การพัฒนาโปรแกรมจำลองแขนกลด้วยไดเรคเอกซ์

Development of a DirectX-based Robot Arm Simulator

สยาม เจริญเสียง วิทยา ธรรมาวุฒิกุล
สถาบันวิทยาการหุ่นยนต์ภาคสนาม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
91 ถ.ประชาอุทิศ แขวงบางมด เขตทุ่งครุ กรุงเทพฯ 10140
โทร 0-2470-9339 โทรสาร 0-2470-9691 E-mail: siam@fibo.kmutt.ac.th, witya@fibo.kmutt.ac.th

Siam Charoenseang Wittaya Thumawutikun
Institute of Field Robotics, King Mongkut's University of Technology Thonburi
91 Prachauthit Rd, Bangmod, Thung Kharu, Bangkok 10140, Thailand
Tel: 0-2470-9339 Fax: 0-2470-9691 E-mail: siam@fibo.kmutt.ac.th, witya@fibo.kmutt.ac.th

บทคัดย่อ

บทความนี้นำเสนอการพัฒนาโปรแกรมจำลองแขนกลที่แสดงผลการทำงานในรูปแบบของภาพกราฟฟิกคอมพิวเตอร์ 3 มิติ ซึ่งได้รับการพัฒนาบนระบบปฏิบัติการวินโดวส์โดยใช้ชุดคำสั่งของไมโครซอฟไดเรคเอกซ์ เพื่อเพิ่มความสามารถในการประมวลผลภาพกราฟฟิก 3 มิติ ได้อย่างรวดเร็ว รวมทั้งช่วยให้ผู้ใช้สามารถปรับเปลี่ยนตำแหน่งและทิศทางการมองภาพทำให้สามารถพิจารณาแบบจำลองได้อย่างละเอียดและง่ายขึ้น นอกจากนี้โปรแกรมจำลองยังมีโครงสร้างข้อมูลที่ยืดหยุ่นทำให้ผู้ใช้งานสามารถทำการออกแบบ สร้าง หรือแก้ไขแบบจำลองแขนกลได้อย่างหลากหลายรูปแบบ ผู้ใช้ยังสามารถทดสอบการเคลื่อนที่ข้อต่อแต่ละข้อของแขนกลได้ โดยไม่จำเป็นต้องทำการเชื่อมต่อกับแขนกลจริง โปรแกรมยังมีความสามารถในการคำนวณฟอร์เวิร์ดไคเนเมติกส์และอินเวิร์ดไคเนเมติกส์ของแบบจำลองที่สร้างขึ้นได้อย่างอัตโนมัติ ซึ่งอำนวยความสะดวกในการหาความสัมพันธ์ระหว่างตำแหน่งและมุมการวางตัวของปลายแขนกลในพิกัด 3 มิติ และตำแหน่งของข้อต่อที่สอดคล้องกัน รวมถึงความสามารถในการแสดงผลใน 3 มิติของรูปทรงรีของความสามารถในการเคลื่อนที่ (Manipulability Ellipsoid) ของปลายแขนกลที่ตำแหน่งการเคลื่อนที่ต่างๆ

Abstract

This paper presents development of a 3-D robot arm simulator. It is developed on the Microsoft Windows operating system. It utilizes the Microsoft DirectX engine to generate 3-D graphics. In the 3-D world, user can design, build, and edit robot models in various and flexible configurations such as defining difference position and orientation of each component. In addition, user can test each joint's movement of robot model without connecting to the real robot. Finally, the simulator automatically generates and calculates robot forward kinematics, inverse kinematics, manipulability measurement, and manipulability ellipsoid for each movement.

1. บทนำ

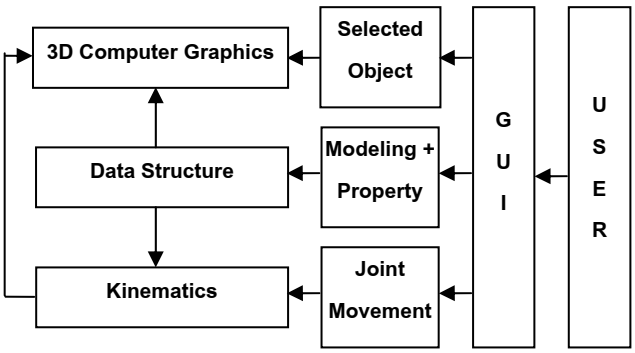
ปัจจุบันหุ่นยนต์ถูกนำมาใช้งานกันอย่างกว้างขวาง ทั้งในอุตสาหกรรมขนาดเล็ก และขนาดใหญ่ เนื่องจากมีความรวดเร็ว ความเที่ยงตรง ความสามารถทำงานอย่างต่อเนื่อง สามารถในการทำงานในสภาวะแวดล้อมต่างๆ หรือสภาวะแวดล้อมที่เป็นอันตรายต่อมนุษย์ แต่อย่างไรก็ตามในการปรับเปลี่ยนการทำงานจำเป็นจะต้องมีการเปลี่ยนแปลงโปรแกรมการทำงานของหุ่นยนต์ ซึ่งหมายถึงการหยุดกระบวนการผลิตลง ดังนั้นวิธีการโปรแกรมแบบออฟไลน์ (Off-line Programming) จึงเป็นที่นิยมใช้กัน โดยสามารถทำการวิเคราะห์และจำลองการทำงานหุ่นยนต์จากภายนอกกระบวนการผลิต ซึ่งต้องอาศัยโปรแกรมจำลองพฤติกรรมของหุ่นยนต์

ถึงแม้โปรแกรมการจำลองหุ่นยนต์จะถูกพัฒนา และมีอยู่หลากหลายในปัจจุบัน ทั้งโปรแกรมเชิงการค้าอย่าง WORKSPACE [1] และ EASY-ROB [2] แต่โปรแกรมเหล่านี้เน้นไปที่หุ่นยนต์อุตสาหกรรมที่นิยมใช้แพร่หลาย ซึ่งจะขาดความสามารถในการแก้ไขดัดแปลงแบบจำลองได้ด้วยโปรแกรมเอง หรือสามารถแก้ไขได้แต่ต้องการโมดูลหรือโปรแกรมเสริมอื่นๆ โปรแกรมการคำนวณทางวิศวกรรมอย่าง MATLAB [3] และ Mathematica [4] เองก็ได้พัฒนา toolbox และ package สำหรับการวิเคราะห์หุ่นยนต์ แต่การจำลองหุ่นยนต์ในโปรแกรมเหล่านี้จะขาดการแสดงผลในรูปกราฟฟิก 3 มิติ อีกทั้งผู้ใช้ต้องมีความรู้และเข้าใจในการเขียนภาษาโปรแกรมนั้นๆ โปรแกรม

จำลองหุ่นยนต์ยังถูกพัฒนาให้สามารถใช้งานแพร่หลายโดยเพิ่มความสามารถในการทำงานบนหลายแฟลตฟอร์ม ผ่านทางอินเตอร์เน็ตด้วยการพัฒนาโปรแกรมด้วยภาษา JAVA [5] หรือ VRML [6] แต่โปรแกรมเหล่านี้จะไม่สามารถดัดแปลงแบบจำลองได้เลย แนวความคิดของโปรแกรมจำลองแขนกลที่สามารถสร้างแบบจำลองได้ยืดหยุ่นและหลากหลาย การใช้งานโปรแกรมที่มีการแสดงผลแบบกราฟฟิก 3 มิติ ช่วยให้พิจารณาการเคลื่อนที่ของแขนกลใน 3 มิติ จึงเป็นที่มาของการพัฒนาโปรแกรมจำลองแขนกลนี้

2. โครงสร้างโปรแกรมจำลองแขนกล

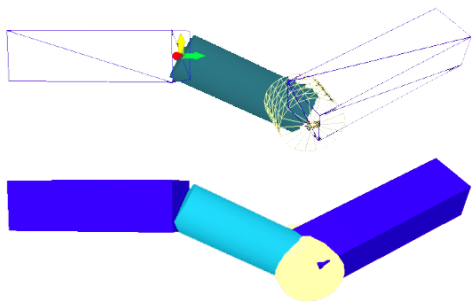
โปรแกรมจำลองแขนกลที่ได้พัฒนาขึ้นนี้จะเน้นไปที่การจำลองการเคลื่อนไหวสำหรับแขนกลแบบอนุกรม โดยที่ผู้ใช้งานสามารถทำการปรับเปลี่ยนแบบจำลองของแขนกลได้ตามต้องการ สามารถกำหนดข้อต่อต่างๆ ให้เป็นข้อหมุน หรือข้อเลื่อน รวมถึงการวางตัวของแต่ละชิ้นส่วน ทำให้สามารถสร้างแบบจำลองแขนกลได้หลากหลาย และยืดหยุ่น การนำเสนอและแสดงผลของโปรแกรมในรูปแบบภาพคอมพิวเตอร์กราฟฟิก 3 มิติ เป็นการเพิ่มประสิทธิภาพในการแสดงผลสร้างความสมจริง รวมทั้งโปรแกรมที่ใช้ในการออกแบบหุ่นยนต์มีส่วนติดต่อกับผู้ใช้แบบกราฟฟิก (Graphic User Interface : GUI) บนระบบปฏิบัติการวินโดวส์ ทำให้ผู้ใช้สามารถใช้งานโปรแกรมได้โดยง่าย สำหรับโครงสร้างโดยรวมของโปรแกรมจำลองจะเป็นไปตามรูปที่ 1



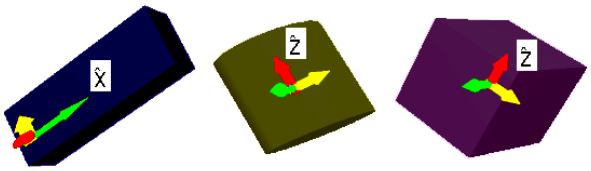
รูปที่ 1 โครงสร้างโปรแกรมโดยรวม

2.1 แบบจำลองแขนกล

การสร้างแบบจำลองแขนกลของโปรแกรมทั่วไปจะอาศัยการกำหนดค่าตัวแปรตามหลักการของ Denavit – Hartenberg [7] แต่ในโปรแกรมนี้เปิดโอกาสให้ผู้ใช้สามารถสร้างแบบจำลองโดยการสร้างก้านต่อและข้อหมุน มาต่อเชื่อมกันแบบอนุกรมและสามารถกำหนดคุณสมบัติ ความยาว ขนาด และการวางตัวของแต่ละชิ้นส่วน โดยที่การวางตัวของแต่ละชิ้นส่วนจะวางตัวเทียบกับระบบแกนของวัตถุที่ต่อเชื่อมอยู่ก่อนหน้า ซึ่งการต่อเชื่อมนี้จะอนุญาตให้เกิดขึ้นที่ปลายของการต่อเท่านั้นดังแสดงในรูปที่ 2 สำหรับแต่ละชิ้นส่วนจะมีระบบแกนของตนเอง โดยกำหนดให้ก้านต่อจะต้องวางตัวตามความยาวในแกน X และแกนของข้อหมุน-ข้อเลื่อน จะอยู่ในแกน Z ดังแสดงในรูปที่ 3

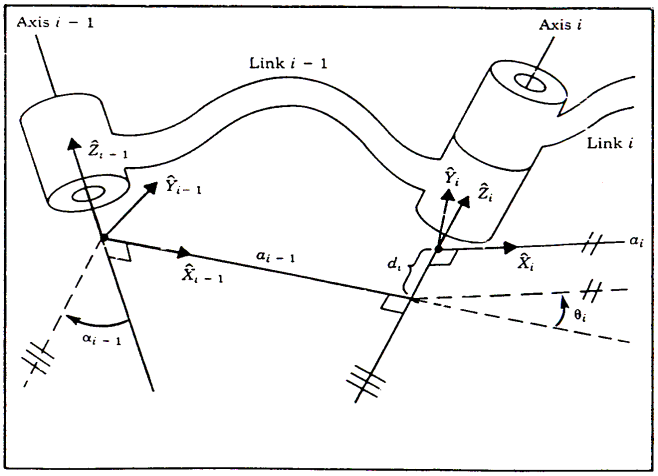


รูปที่ 2 การต่อเชื่อมของแต่ละชิ้นส่วน

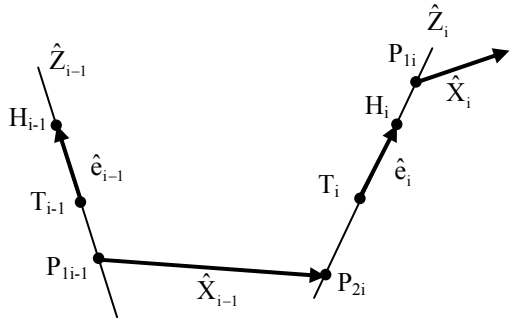


รูปที่ 3 ระบบแกนของก้านต่อ ข้อหมุน และข้อเลื่อนตามลำดับ

เมื่อทำการสร้างการต่อเชื่อมของส่วนประกอบต่างๆเสร็จสิ้นแล้ว โปรแกรมจะทำการคำนวณหาค่าตัวแปรตามหลักการของ Denavit – Hartenberg ดังรูปที่ 4 ต่อไป เพื่อใช้ในการคำนวณฟอร์เวิร์ดไคเนเมติกส์ ซึ่งการคำนวณหาตัวแปรเหล่านี้อาศัยตำแหน่งและแกนหมุนของข้อต่อแต่ละข้อต่อ โดยจะทำการหาตำแหน่งจุดกำเนิดของระบบแกนหมุน (Origin Point) เทียบกับระบบแกนหลัก (Global Frame) ซึ่งกำหนดด้วย T_i และจุดที่อยู่บนแกนหมุนและห่างจากจุดกำเนิดของระบบแกน 1 หน่วย กำหนดด้วย H_i ดังแสดงในรูปที่ 5 ด้วยตำแหน่งของ H_i และ T_i ทำให้เราสามารถทราบการวางตัวของแกนหมุนในระบบแกนหลัก



รูปที่ 4 ตัวแปรตามหลักการของ Denavit – Hartenberg [7]



รูปที่ 5 การใช้ T_i และ H_i ในการหาการวางตัวของแกนหมุน และการหาตัวแปรตามหลักการของ Denavit – Hartenberg

ในการหาค่าตัวแปรตามหลักการของ Denavit – Hartenberg พิจารณาจากรูปที่ 4 และ 5 อาศัยพื้นฐานความรู้ทางด้านเวกเตอร์ จะสามารถหาค่าตัวแปรได้จากสมการ (1)-(4)

$$d_i = \left|P_{1i} - P_{2i}\right| \tag{1}$$

$$a_{i-1} = \left|\hat{X}_{i-1}\right| = \left|P_{2i} - P_{1i-1}\right| \tag{2}$$

$$\alpha_{i-1} = a\cos\left(\left|\hat{e}_i \cdot \hat{e}_{i-1}\right| / \left(\left|\hat{e}_i\right| \cdot \left|\hat{e}_{i-1}\right|\right)\right) \tag{3}$$

$$\theta_i = a\cos\left(\left|\hat{X}_i \cdot \hat{X}_{i-1}\right| / \left(\left|\hat{X}_i\right| \cdot \left|\hat{X}_{i-1}\right|\right)\right) \tag{4}$$

โดยที่ตำแหน่ง P_{1i-1} และ P_{2i} คือตำแหน่งที่ใกล้ที่สุดของแกนหมุน Z_{i-1} และ Z_i ซึ่งระยะ a_{i-1} ก็คือระยะสั้นที่สุด และเป็นระยะตั้งฉากระหว่างแกนหมุน Z_{i-1} และ Z_i ซึ่งจะสามารถหาจุดทั้ง 2 ได้ตามสมการที่(5)-(10)

$$\hat{e}_{i-1} = H_{i-1} - T_{i-1} \text{ และ } \hat{e}_i = H_i - T_i \tag{5}$$

$$P_{1i-1} = H_{i-1} + s_{i-1}\left(T_{i-1} - H_{i-1}\right) = H_{i-1} + s_{i-1}\hat{e}_{i-1} \tag{6}$$

$$P_{2i} = H_i + u_i\left(T_i - H_i\right) = H_i + u_i\hat{e}_i \tag{7}$$

$$\begin{aligned} s_{i-1} = {} & \left(\hat{e}_{i-1} \cdot \hat{e}_i\right)\left[\left(H_{i-1} - H_i\right) \cdot \hat{e}_i\right] / \det \\ & - \left(\hat{e}_i \cdot \hat{e}_i\right)\left[\left(H_{i-1} - H_i\right) \cdot \hat{e}_{i-1}\right] / \det \end{aligned} \tag{8}$$

$$\begin{aligned} u_i = {} & \left(\hat{e}_{i-1} \cdot \hat{e}_{i-1}\right)\left[\left(H_{i-1} - H_i\right) \cdot \hat{e}_i\right] / \det \\ & - \left(\hat{e}_{i-1} \cdot \hat{e}_i\right)\left[\left(H_{i-1} - H_i\right) \cdot \hat{e}_{i-1}\right] / \det \end{aligned} \tag{9}$$

$$\det = \left(\left(\hat{e}_{i-1} \cdot \hat{e}_i\right) \cdot \left(\hat{e}_{i-1} \cdot \hat{e}_i\right)\right) - \left(\left(\hat{e}_{i-1} \cdot \hat{e}_{i-1}\right) \cdot \left(\hat{e}_i \cdot \hat{e}_i\right)\right) \tag{10}$$

ในแต่ละแกนหมุน Z_i จะเสมือนมีตัวแปรเพิ่มเติม u_i และ s_i เป็นตัวแปรช่วยในการหาค่าตัวแปรอื่นๆ โดยค่า u_i จะได้จากการพิจารณาระหว่างแกนหมุน Z_{i-1} และ Z_i ส่วน s_i จะได้จากการพิจารณาระหว่างแกนหมุน Z_i และ Z_{i+1}

ในการพิจารณาเครื่องหมายของ d_i, α_{i-1} และ θ_i นั้นจะพิจารณาดังต่อไปนี้

$d_i = (-1) \cdot d_i$ เมื่อ $s_i > u_i$

$\alpha_{i-1} = (-1) \cdot \alpha_{i-1}$ เมื่อ $\hat{e}_i \times \hat{e}_{i-1}$ มีทิศตรงข้ามกับ $\hat{X}_{i-1}$

$\theta_i = (-1) \cdot \theta_i$ เมื่อ $\hat{X}_{i-1} \times \hat{X}_i$ มีทิศตรงข้ามกับ $\hat{e}_i$

กรณีที่ $\hat{e}_i$ ขนานกับ $\hat{e}_{i-1}$ จะพิจารณาสมการข้างต้นให้ $s_{i-1} = 1$ จะสามารถหาค่า u_i ได้จากสมการที่ (11)

$$u_i = -\left(T_{i-1} - H_i\right) \cdot \hat{e}_{i-1} / \left(\hat{e}_i \cdot \hat{e}_{i-1}\right) \tag{11}$$

2.2 อินเวิร์ดไคเนเมติกส์ของแบบจำลอง

เนื่องจากโปรแกรมมีความหลากหลาย และยืดหยุ่นในการสร้างแบบจำลอง ดังนั้นการแก้ปัญหาอินเวิร์ดไคเนเมติกส์เพื่อที่จะสามารถรองรับแบบจำลองต่างๆการใช้ระเบียบวิธีเชิงตัวเลข (Numerical Method) จึงเป็นวิธีที่เหมาะสม ซึ่งเทคนิคที่ใช้ในการคำนวณที่นำมาใช้ภายในโปรแกรมได้แก่

2.2.1 Iterative Newton Raphson Method

จากการหาฟอร์เวิร์ดไคเนเมติกส์ และ ความสัมพันธ์ของจาโคเบียนเมตริก ปัญหาอินเวิร์ดไคเนเมติกส์สามารถแก้ได้โดยวิธีการทำซ้ำด้วยการอาศัยเมตริกผกผัน (Inverse Matrix) ของเมตริกจาโคเบียน (Jacobian Matrix) หรือ pseudo inverse ในกรณีที่เมตริกจาโคเบียนไม่สามารถหาเมตริกผกผันได้

2.2.2 Singularity Robust Inverse Method (SR-Inverse)

ในบริเวณจุด singular จะมีบางทิศทางที่หุ่นยนต์จะไม่สามารถเคลื่อนไปได้ หรือถูกลดความสามารถในการเคลื่อนที่ลง การใช้ inverse หรือ pseudo inverse ในการแก้ปัญหาอินเวิร์ดไคเนเมติกส์ หากค่า $\delta\theta$ นั้นอยู่ในทิศทางที่ถูกลดความสามารถในการเคลื่อนที่ลง ความพยายามที่จะหาคำตอบที่ถูกต้องเที่ยงตรงส่งผลให้คำตอบที่ได้มีความเป็นไปได้ หรือน่าเชื่อถือลดลง ดังนั้นการยอมให้เกิดความผิดพลาดขึ้นบ้างในบริเวณที่เป็น singular คำตอบก็จะส่งผลให้คำตอบที่ได้มีความเป็นไปได้มากขึ้น [8]

2.2.3 Optimum Step Size Control for Newton-Raphson Method

การแก้ปัญหาอินเวิร์ดไคเนเมติกส์โดยวิธีการทำซ้ำตามวิธี Newton-Raphson อาจจะไม่ลู่เข้าสู่คำตอบ ซึ่งส่งผลให้ไม่สามารถหาคำตอบของปัญหาได้ แนวความคิดในการใช้ step size ที่เหมาะที่สุด ที่ทำให้คำตอบในแต่ละรอบการทำซ้ำพยายามใกล้คำตอบหรือลู่เข้ามากที่สุดจะช่วยแก้ปัญหาได้เป็นอย่างดี [9]

2.3 รูปทรงรีของความสามารถในการเคลื่อนที่ของแบบแขนกล (Manipulability Ellipsoid)

เมื่อนำเมตริกจาโคเบียนมาเขียนในรูป Singular Value Decomposition (SVD) สามารถแสดงได้ดังสมการที่ (12) [10]

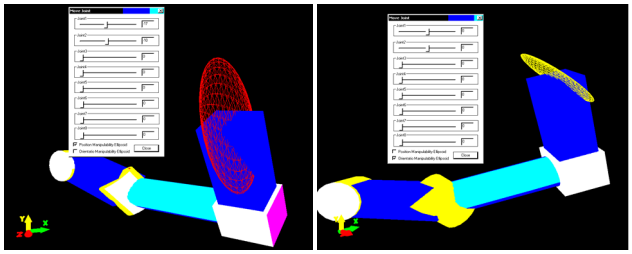
$$\begin{gathered} J = U \sum V^T, \sum \triangleq \operatorname{diag}(\sigma_1, \cdots, \sigma_p) \in R^{m \times n} \\ p = \min\{m, n\} \qquad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k > 0 \\ \sigma_{k+1} = \cdots = \sigma_p = 0 \end{gathered} \tag{12}$$

จะสามารถพิจารณารูปทรงรีของความสามารถในการเคลื่อนที่ของแขนกล ซึ่งจะมีแกนของทรงรี (principal axes) ในทิศทางเวกเตอร์ $u_1, u_2, \ldots u_m$ จากเมตริก $U = [u_1 | u_2 | \ldots | u_m]$ โดยที่ค่า Singular Value (σ_i) จะแสดงค่ารัศมีของทรงรีในทิศทาง u_i ที่สัมพันธ์กัน

เนื่องจากเมตริกจาโคเบียนสามารถเขียนแยกเป็นเมตริกย่อยในรูปของส่วนที่เป็นตำแหน่ง J_P (position) และการวางตัว J_o (orientation) ดังสมการที่ (13)

$$J = \begin{bmatrix} J_P \\ J_o \end{bmatrix} \tag{13}$$

เมื่อพิจารณาสมการที่ (12) ร่วมกับสมการที่ (13) จะสามารถแยกรูปทรงรีของการเคลื่อนที่ออกเป็นสองรูป คือรูปทรงรีของการเคลื่อนที่ของตำแหน่ง และรูปทรงรีของการเคลื่อนที่ของการวางตัว ดังแสดงตัวอย่างรูปทรงรีที่ได้ในโปรแกรมจำลองได้ดังรูปที่ 6 โดยที่รูปทรงรีดังกล่าวจะแสดงถึงความสามารถในการเคลื่อนที่ได้ดีมากน้อยเพียงใดในทิศทางต่าง ๆโดยเปรียบเทียบได้จากรัศมีของทรงรีในทิศทางนั้น ซึ่งจะช่วยในการวิเคราะห์ และพิจารณาแบบจำลองแขนกลถึงความเหมาะสมในการทำงาน และความคล่องตัวในการเคลื่อนที่ในพื้นที่ทำงาน

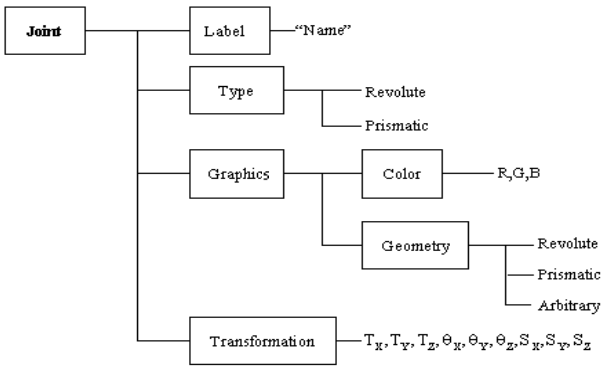


a. b.

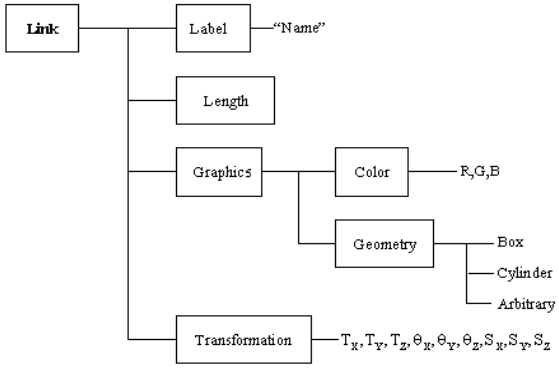
รูปที่ 6 ตัวอย่างรูปทรงรีของการเคลื่อนที่ของตำแหน่ง (a.) และรูปทรงรีของการเคลื่อนที่ของการวางตัว (b.) ที่ได้ในโปรแกรมจำลอง

2.4 โครงสร้างข้อมูลของระบบ

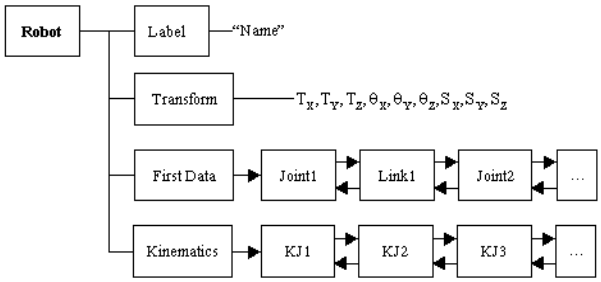
โครงสร้างข้อมูลของแขนกล (Robot Data Structure) เป็นส่วนที่เก็บข้อมูล รูปแบบ โครงสร้าง และสถานะของหุ่นยนต์ โดยโครงสร้างข้อมูลของแขนกลยังแยกออกเป็นตามลักษณะของวัตถุเป็น ข้อต่อ ก้านต่อ แขนกล และไคเนเมติกส์



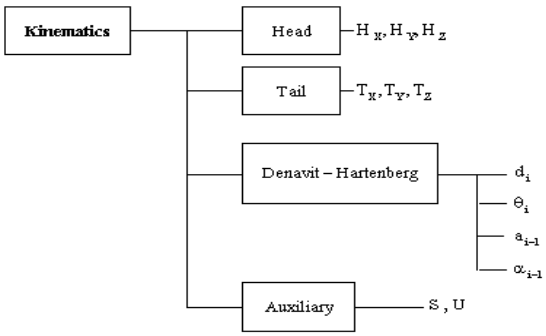
รูปที่ 7 โครงสร้างข้อมูลข้อต่อ



รูปที่ 8 โครงสร้างข้อมูลก้านต่อ



รูปที่ 9 โครงสร้างข้อมูลแขนกล



รูปที่ 10 โครงสร้างข้อมูลไคเนเมติกส์

แบบจำลองแขนกลที่ผู้ใช้กำหนดจะเกิดจากการต่อกันแบบอนุกรมของก้านต่อ และข้อหมุน ดังนั้นโครงสร้างข้อมูลของแขนกลจึงเกิดจากการต่ออนุกรมของโครงสร้างข้อมูลข้อหมุนดังรูปที่ 7 และโครงสร้างข้อมูลก้านต่อดังรูปที่ 8 ดังนั้นรูปแบบโครงสร้างข้อมูลที่เหมาะสมกับโครงสร้างข้อมูลของแขนกล คือ ลักษณะการต่อของข้อมูลแบบอนุกรม โดยโครงสร้างสามารถยืดหยุ่นได้การเพิ่ม ลด หรือแทรกข้อมูลในโครงสร้างทำได้โดยง่าย รวดเร็ว ซึ่งรูปแบบที่เหมาะสมก็คือโครงสร้างแบบดับเบิ้ลลิงค์ลีส แต่ลิงค์ลีสนี้จะเกิดจากการต่ออนุกรมของข้อมูลต่างชนิดกัน คือข้อมูลที่เป็นโครงสร้างแบบข้อหมุน และข้อมูลที่เป็นโครงสร้างแบบก้านต่อ

โครงสร้างข้อมูลของแขนกลนั้นดังแสดงในรูปที่ 9 จะทำการเก็บตำแหน่งหรือพอยเตอร์ (pointer) ที่ชี้ไปยังข้อมูลตัวแรกของลิงค์ลีสเท่านั้น เนื่องจากคุณสมบัติของลิงค์ลีสที่ข้อมูลแต่ละตัวสามารถชี้ไปยัง

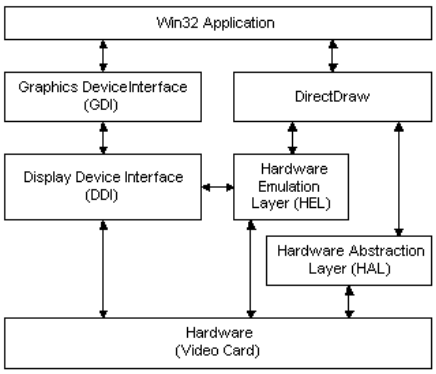
ข้อมูลตัวต่อๆไปเนื่องจากแต่ละข้อมูลมีการเก็บตำแหน่งของข้อมูลก่อนหน้า และต่อหลังจากตัวข้อมูลไว้ด้วย โครงสร้างข้อมูลแขนกลยังเก็บข้อมูลเกี่ยวกับตำแหน่งของแขนกล ชื่อ และตำแหน่งข้อมูลเริ่มต้นของข้อมูลไคเนเมติกส์

โครงสร้างข้อมูลไคเนเมติกส์ดังแสดงในรูปที่ 10 จะเกิดขึ้นเมื่อมีการสร้างการต่อเชื่อมของชิ้นส่วนข้อหมุนและก้านต่อเป็นแขนกลเกิดขึ้น โดยที่ข้อมูลไคเนเมติกส์นี้จะมีจำนวนเท่ากับจำนวนข้อต่อในแขนกลนั้น แต่ละข้อมูลนั้นแทนข้อมูลที่สัมพันธ์กับแต่ละข้อต่อนั้น และเช่นเดียวกันการต่อเชื่อมของแต่ละข้อมูลนี้จะเป็นรูปแบบดับเบิ้ลลิ้งค์ลีสโดยที่โครงสร้างนี้จะเก็บข้อมูลของตัวแปรที่ใช้ตามหลักการของ Denavit – Hartenberg รวมถึงตำแหน่งของจุดที่แทนการวางตัวของแกนหมุน และข้อมูลตัวแปรที่จำเป็นในการคำนวณในสมการที่ (1)–(11)

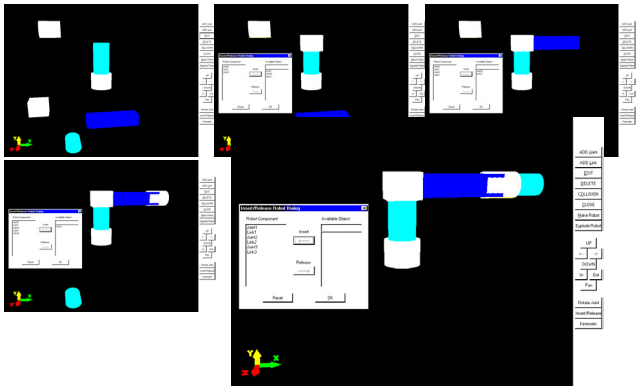
2.5 คอมพิวเตอร์กราฟฟิก 3 มิติ

ปัจจุบันความสามารถทางด้านกราฟฟิกของเครื่องคอมพิวเตอร์ส่วนบุคคลมีประสิทธิภาพที่โดดเด่นขึ้น อุปกรณ์แสดงผล (Graphic Card) ในท้องตลาดทั่วไปที่มีราคาไม่สูงก็สามารถรองรับการทำงานและสร้างภาพ 3 มิติได้ ทำให้การพัฒนาและใช้งานระบบจำลองการทำงานต่างๆ สามารถทำได้อย่างกว้างขวาง รวมทั้งการที่โปรแกรมจำลองมีการทำงานและแสดงผลในรูปแบบ 3 มิติ ร่วมกับส่วนติดต่อกับผู้ใช้แบบกราฟฟิก (GUI) บนวินโดวส์ ส่งผลให้ผู้ใช้สามารถใช้งานได้อย่างสะดวกและเข้าใจการทำงานของโปรแกรมได้โดยง่าย

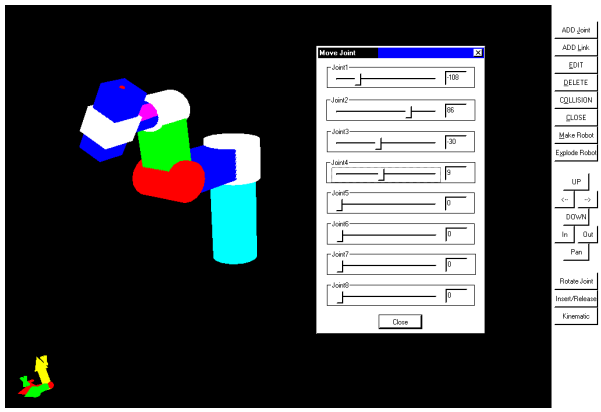
สำหรับการสร้างภาพกราฟฟิกในเครื่องคอมพิวเตอร์ส่วนบุคคลนั้น จะอาศัยเครื่องมือในการสร้างภาพได้ 2 แบบใหญ่ๆ คือ DirectX Engine และ OpenGL Engine ความแตกต่างระหว่างเครื่องมือทั้งสอง คือ DirectX สามารถเข้าถึงฮาร์ดแวร์ และเร่งการสร้างภาพ 3 มิติ รวมถึงเข้าไปใช้ประสิทธิภาพความสามารถของฮาร์ดแวร์ได้อย่างเต็มที่ในกรณีที่กราฟฟิกฮาร์ดแวร์นั้นรองรับการทำงานของ DirectX โดยอาศัย hardware abstraction layer (HAL) ในกรณีที่ฮาร์ดแวร์ไม่รองรับการทำงานของ DirectX ก็จะทำการจำลองการทำงานโดย hardware emulation layer (HEL) ซึ่งจะให้ผลการทำงานที่ช้ากว่า HAL Device ดังแสดงในรูปที่ 11 ตัวอย่างขั้นตอนการสร้างแบบจำลองแขนกล และตัวอย่างแบบจำลองที่สร้างขึ้นได้อย่างหลากหลายตามความต้องการ แสดงตามลำดับในรูปที่ 12 และ 13



รูปที่ 11 สถาปัตยกรรมของ DirectX [11]



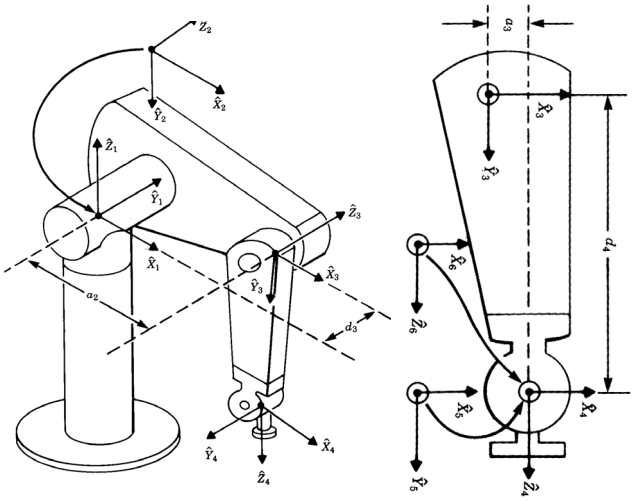
รูปที่ 12 ตัวอย่างขั้นตอนการสร้างแบบจำลองแขนกล



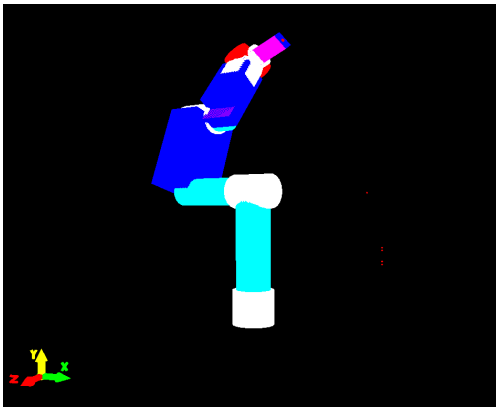
รูปที่ 13 ตัวอย่างแบบจำลองที่สร้างขึ้น

3. การทดลอง

ในการทดลองโมเดลของหุ่นยนต์ PUMA 560 ดังแสดงในรูปที่ 14 เป็นตัวเปรียบเทียบในการทดลอง ซึ่งกำหนดค่าพารามิเตอร์ดังนี้คือ $a_2 = 30$ $a_3 = 5$ $d_3 = 15$ และ $d_4 = 25$ โดยการทดลองสร้างแบบจำลอง 3 มิติ แล้วจึงจำลองหาตำแหน่งของปลายแขนกลโดยการคำนวณฟอร์เวิร์ดไคเนเมติกส์ และการคำนวณหาตำแหน่งของข้อต่อต่างๆย้อนกลับจากตำแหน่งปลายแขนกล หรือการหาอินเวิร์ดไคเนเมติกส์ ผลการทดลองดังแสดงในตารางที่ 1 และ 2 ตามลำดับ



รูปที่ 14 แบบจำลองหุ่นยนต์ PUMA 560 [7]



รูปที่ 15 แบบจำลองหุ่นยนต์ PUMA 560 ที่ได้จากการสร้างในโปรแกรมจำลอง

ตารางที่ 1 การทดลองหาฟอร์เวิร์ดไคเนเมติกส์

θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	**Error Position (unit)**	**Error Orientation**
36°	-36°	-36°	36°	-36°	36°	0.05654	0.07783°
54°	-54°	-54°	54°	-54°	54°	0.06681	0.09467°
90°	-90°	-90°	90°	-90°	90°	0.04097	0.20811°
108°	-108°	-108°	108°	-108°	108°	0.03241	0.22732°
126°	-126°	-126°	126°	-126°	126°	0.05305	0.41601°
162°	-162°	-162°	162°	-162°	162°	0.0697	0.20613°
180°	-180°	-180°	180°	-180°	180°	0.05977	0.16218°

ตารางที่ 2 การทดลองหาอินเวิร์ดไคเนเมติกส์

	Desired Position	**SR-inverse(SRI)**	**Optimum Step Size(Ops)**	**SRI+Ops**
X (unit)	-0.922	-0.8383	-0.9225	-0.9218
Y (unit)	-31.483	-31.338	-31.483	-31.4829
Z (unit)	-48.067	-48.163	-48.066	-48.0663
|Error| (unit)	-	0.1934	0.0011	0.00039
Iteration	-	21	10	11
θ_1	-120°	-120°	-120°	-120°
θ_2	60°	60.367°	59.688°	59.7915°
θ_3	-78.69°	-79.054°	-78.011°	-78.236°
θ_4	0°	-85.685°	0.9353°	-4.18159°
θ_5	0°	164.16°	40.372°	41.9283°
θ_6	0°	63.929°	-124.13°	-118.294°

4. สรุป

โปรแกรมจำลองแขนกลที่ได้นำเสนอนี้ ได้แสดงให้เห็นถึงความยืดหยุ่นในการสร้างแบบจำลองได้อย่างหลากหลาย การใช้ภาพกราฟฟิก 3 มิติเข้ามาช่วยในการนำเสนอและแสดงผล รวมทั้งความสามารถคำนวณฟอร์เวิร์ดไคเนเมติกส์ และอินเวิร์ดไคเนเมติกส์ของแบบจำลองได้อย่างถูกต้องโดยอัตโนมัติ ซึ่งเป็นผลจากการใช้โครงสร้างข้อมูล และแบบจำลองแขนกลที่มีประสิทธิภาพได้อย่างเหมาะสม ปัจจุบันโปรแกรมมีความสามารถในการคำนวณเฉพาะในทางไคเนเมติกส์เท่านั้น ดังนั้นการพัฒนาในส่วนของการจำลองทางพลศาสตร์ (Dynamic Simulation) จึงเป็นส่วนสำคัญในการทำให้โปรแกรมจำลองแขนกลนี้สมบูรณ์มากยิ่งขึ้น

เอกสารอ้างอิง

[1] Owens, J., “WORKSPACE-a microcomputer-based industrial robot simulator and off-line programming system”, Next Steps for Industrial Robotics, IEE Colloquium on, 1994, pp. 4/1 -4/4.

[2] EASY-ROB, http://www.easyrob.de.

[3] P.I. Corke, “A computer tool for simulation and analysis: the Robotics Toolbox for MATLAB”, Proceedings of the 1995 National Conference of the Australian Robot Association, Melbourne, Australia, pp. 319-330, July 1995.

[4] Nethery, J.F. and Spong, M.W., “Robotica: a Mathematica package for robot analysis”, IEEE Robotics & Automation Magazine , Volume: 1 Issue: 1,March 1994 , pp. 13 –20.

[5] Speck, A. and Klaeren, H., “RoboSiM: Java 3D robot visualization”, In Proceedings of IECON'99, The 25th Annual Conference of the IEEE Industrial Electronics Society, Volume: 2, 1999, pp. 821-826.

[6] Rohrmeier, M., “Web based robot simulation using VRML”, Proceedings of the 2000 Winter Simulation Conference, Volume: 2 , 2000 , pp. 1525 –1528.

[7] J.J. Craig, “Introduction to Robotics: Mechanics and Control”, Reading, MA: Addison-Wesley, 1989.

[8] Y. Nakamura and H. HanaFusa, ”Inverse Kinematic Solutions With Singularity Robustness for Robot Manipulator Control”, ASME Journal of Dynamic Systems, Measurement, and Control, Vol. 108, pp. 163-171,1986.

[9] D.E. Whitney, “Optimum Step Size Control for Newton-Raphson Solution of Non-Linear Vector Equations”, IEEE Trans. Autom. Control, pp. 572-574, 1969.

[10] Tsuneo Yoshikawa, “Foundations of Robotics Analysis and Control”, The MIT Press, 1990.

[11] Microsoft Corporation, “DirectX 8.0 Documentation (Visual C++)”.