

ระบบจำลองหุ่นยนต์เสมือนโดยใช้การจำลองฮาร์ดแวร์ภายในลูป Virtual Robot Simulation System Using Hardware in the Loop Simulation

สมหวัง อริสริยวงศ์

ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ องครักษ์

อ.องครักษ์ จ.นครนายก 26120

somwang@swu.ac.th, โทรศัพท์ 037-322609, โทรสาร 037-322609

บทคัดย่อ

ระบบจำลองฮาร์ดแวร์ภายในลูป (Hardware-in-the-loop simulation) ได้รับความนิยมเพิ่มขึ้นอย่างสูงในการใช้เป็นเครื่องมือสำหรับออกแบบและทดสอบระบบควบคุมในงานวิศวกรรม เพื่อลดต้นทุนและเวลาที่ใช้ในการออกแบบ งานวิจัยนี้นำเสนอระบบจำลองหุ่นยนต์เสมือนโดยการประยุกต์ใช้ระบบจำลองฮาร์ดแวร์ภายในลูปเพื่อออกแบบและทดสอบหุ่นยนต์แบบสกรารา ซึ่งระบบดังกล่าวประกอบไปด้วยส่วนที่เป็นฮาร์ดแวร์ทำหน้าที่รับส่งสัญญาณระหว่างตัวควบคุมกับระบบจำลองหุ่นยนต์เสมือนในรูปของสัญญาณไฟฟ้า และ ส่วนที่เป็นซอฟต์แวร์ทำหน้าที่กำหนดค่าตัวแปรต่างๆของหุ่นยนต์รวมถึงแสดงการเคลื่อนที่ของหุ่นยนต์ในรูปแบบสามมิติ จากผลการทดลองควบคุมหุ่นยนต์เสมือนโดยใช้แบบบังคับด้วยตัวเองและแบบบังคับด้วยตัวควบคุมแบบ พี แอล ซี พบว่าระบบจำลองหุ่นยนต์เสมือนที่พัฒนาขึ้นสามารถทำงานได้อย่างถูกต้องตามที่กำหนดไว้

คำหลัก: หุ่นยนต์, ระบบจำลองฮาร์ดแวร์ภายในลูป, ระบบจำลอง, ตัวควบคุม

Abstract

Hardware-in-the-loop simulation is an increasingly popular engineering tool for its effectiveness in testing the software and hardware of the control system, decrease development time and costs. This paper proposes a virtual robot simulation system for applying hardware-in-the-loop simulation techniques to the design and testing of SCARA robot manipulators. This system consists of two parts, the hardware part which interfaces with controller in electrical signals and the software part which use to setting virtual robot's parameters and display the motion of virtual robot in 3D. The results obtained from control the virtual robot simulation system by manual and PLC controller show that the system can work correctly and effectively.

Keywords: Robot, Hardware-in-the-loop simulation, SCARA robot, Control, Simulation

1. บทนำ

ในปัจจุบันมีการใช้หุ่นยนต์ในอุตสาหกรรมกันอย่างแพร่หลาย [1] เนื่องจากหุ่นยนต์ สามารถปรับเปลี่ยนรูปแบบการทำงานได้อย่างหลากหลายตามโปรแกรมที่ป้อนเข้าไป ทำให้การใช้งานหุ่นยนต์มีความยืดหยุ่นสูง การโปรแกรมให้หุ่นยนต์ทำงานได้ตามที่ต้องการนั้นในปัจจุบันมีอยู่สองวิธี [2] คือ Offline Programming และ Online Programming โดยที่ Offline Programming จะทำการโปรแกรมและทดสอบหุ่นยนต์ผ่านโปรแกรมจำลองการทำงานของหุ่นยนต์โดยไม่ใช้หุ่นยนต์จริง ส่วน Online Programming นั้นจะทำการโปรแกรมและทดสอบหุ่นยนต์โดยผ่านทางตัวหุ่นยนต์จริง จะเห็นได้ว่าการโปรแกรมแบบ Offline Programming จะประหยัดเวลาและค่าใช้จ่ายได้มากกว่าการโปรแกรมแบบ Online Programming เนื่องจากสามารถทดสอบโปรแกรมที่เขียนขึ้นได้หลายครั้งและสามารถป้องกันความเสียหายที่อาจเกิดขึ้นได้ในกรณีที่เกิดความผิดพลาดในการเขียนโปรแกรม แต่การโปรแกรมแบบ Online Programming ยังไม่ได้ตอบโจทย์เกี่ยวกับการโปรแกรมหุ่นยนต์ในเชิงลึกเนื่องจากในปัจจุบันโปรแกรมจำลองการทำงานของหุ่นยนต์ที่พัฒนาขึ้นไม่สามารถจำลองสัญญาณของตัวแปรต่างๆของหุ่นยนต์ในรูปของสัญญาณทางไฟฟ้าได้เช่นเดียวกับที่สามารถวัดได้จากหุ่นยนต์จริง ทำให้ไม่สามารถออกแบบโปรแกรมหรือออกแบบตัวควบคุมของหุ่นยนต์ในเชิงลึกได้

ระบบจำลองฮาร์ดแวร์ภายในรูปได้ถูกนำมาใช้ในการออกแบบ และ ทดสอบ ระบบควบคุมในอุตสาหกรรมเพื่อลดเวลา ลดความผิดพลาด และ ลดต้นทุนการผลิต ดังเช่นงานวิจัยของ R. Isermann, J. Schaffnit and S. Sinsel [3] ได้นำระบบจำลองฮาร์ดแวร์ภายในรูปมาใช้ในการออกแบบระบบควบคุมเครื่องยนต์ ซึ่งงานวิจัยนี้ได้ใช้การจำลองชิ้นส่วนบางชิ้นภายในเครื่องยนต์ด้วยซอฟต์แวร์แบบเวลาจริง ทำให้ประหยัดค่าใช้จ่ายในการสร้างฮาร์ดแวร์และผลการทดลองที่ได้ยังมีค่าใกล้เคียงกับฮาร์ดแวร์จริง งานวิจัย

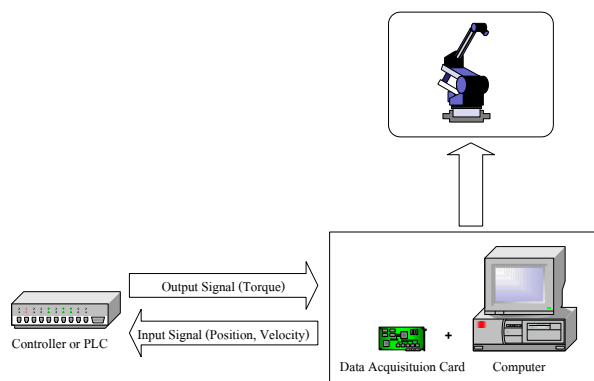
ของ M. Linjama, T. Virvalo, J. Gustafsson, J. Lintula, V. Aaltonen และ M. Kivikoski [4] ที่ออกแบบระบบควบคุมกระบอบสูบไฮดรอลิกโดยใช้เทคนิคจำลองฮาร์ดแวร์ภายในรูป โดยใช้ซอฟต์แวร์ทำการสร้างสัญญาณเซนเซอร์ที่ใช้วัดตำแหน่งของกระบอบสูบแทนเซนเซอร์ที่มีอยู่จริงผ่านทางการ์ดประมวลผลข้อมูลเพื่อดูพฤติกรรมของตัวควบคุมว่าเป็นไปตามที่ต้องการหรือไม่ งานวิจัยของ Kenji Hagiwara, Satoshi Terayama, Youhei Takeda, Ko Yoda และ Shoichi Suzuki [5] ได้นำระบบจำลองฮาร์ดแวร์ภายในรูป ไปใช้ออกแบบระบบส่งกำลังเกียร์อัตโนมัติโดยได้จำลองระบบส่งกำลังของเกียร์อัตโนมัติทั้งหมดด้วยซอฟต์แวร์แบบเวลาจริงเพื่อดูผลการตอบสนองของตำแหน่งเกียร์ ณ ความเร็วรถต่างๆ ในงานวิจัยนี้ได้แสดงให้เห็นว่าระบบจำลองฮาร์ดแวร์ภายในรูปสามารถช่วยลดเวลาในการออกแบบ และ ทดสอบ ระบบส่งกำลังเกียร์อัตโนมัติลงได้อย่างมาก เนื่องจากไม่ต้องไปทดสอบกับรถยนต์จริง ๆ ในส่วนทางด้านการศึกษานั้นก็ได้เริ่มมีการนำระบบจำลองฮาร์ดแวร์ภายในรูปมาช่วยใช้ในการเรียนการสอนเช่นกัน เช่น งานวิจัยของ Panayiotis S. Shiakolas และ Damrongrit Piyabongkarn [6] ได้ใช้ระบบจำลองฮาร์ดแวร์ภายในรูปมาทำการจำลองระบบการควบคุมตำแหน่งวัตถุด้วยสนามแม่เหล็ก และ จำลองตัวควบคุมชนิดดิจิตอล จากงานวิจัยนี้แสดงให้เห็นว่าระบบจำลองฮาร์ดแวร์ภายในรูปสามารถช่วยให้การเรียนการสอนในเรื่องวิศวกรรมการควบคุมเป็นไปได้อย่างมีประสิทธิภาพมากขึ้น เนื่องจากสามารถปรับเปลี่ยนค่าตัวแปรของระบบได้อย่างรวดเร็ว อีกทั้งยังราคาถูกกว่าอุปกรณ์ทดลองที่เป็นฮาร์ดแวร์เป็นอย่างมาก

จากที่กล่าวมาข้างต้นถ้าเรานำระบบจำลองฮาร์ดแวร์ภายในรูปมาใช้ร่วมกับโปรแกรมจำลองการทำงานของหุ่นยนต์โดยจะทำการสร้างแบบจำลองของหุ่นยนต์ในคอมพิวเตอร์ให้สามารถเคลื่อนไหวได้ตามสัญญาณที่มาจากตัวควบคุมตามที่ได้อัปเดตโปรแกรมไว้และสามารถสร้างสัญญาณไฟฟ้าซึ่งเป็นตัวแทนของตัวแปรที่สำคัญภายในหุ่นยนต์ เช่น ตำแหน่ง และ

ความเร็วของข้อต่อต่างๆ เพื่อให้สามารถตรวจวัดได้ เช่นเดียวกับสัญญาณที่มาจากหุ่นยนต์จริง ทำให้การออกแบบโปรแกรมหรือออกแบบตัวควบคุมของหุ่นยนต์ในเชิงลึกเป็นไปได้โดยมีประสิทธิภาพ

2. ระบบจำลองหุ่นยนต์เสมือนโดยใช้การจำลองฮาร์ดแวร์ภายในรูป

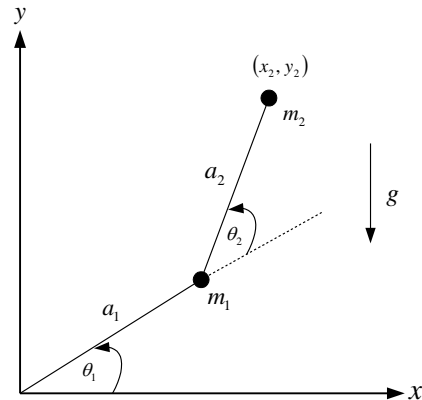
ระบบจำลองหุ่นยนต์เสมือนโดยใช้การจำลองฮาร์ดแวร์ภายในรูป เป็นการพัฒนาโปรแกรมที่ใช้จำลองการทำงานของหุ่นยนต์ ให้สามารถสร้างสัญญาณไฟฟ้าซึ่งเป็นตัวแทนของตัวแปรที่สำคัญภายในหุ่นยนต์เพื่อให้สามารถตรวจวัดได้ โดยใช้เทคนิคการสร้างภาพกราฟิกในคอมพิวเตอร์ร่วมกับการจำลองฮาร์ดแวร์ภายในรูป ซึ่งการจำลองการจำลองฮาร์ดแวร์ภายในรูปหมายถึงการจำลองชิ้นส่วนที่เป็นฮาร์ดแวร์บางส่วนหรือทั้งหมดของหุ่นยนต์ด้วยซอฟต์แวร์ แต่ส่วนที่เป็นสัญญาณอินพุต และ สัญญาณเอาต์พุตของหุ่นยนต์จะเป็นสัญญาณทางไฟฟ้าจริงที่สามารถตรวจวัดได้ โดยจะทำการสร้างแบบจำลองของหุ่นยนต์ในคอมพิวเตอร์ให้สามารถเคลื่อนไหวได้ตามสัญญาณที่มาจากตัวควบคุม และสามารถสร้างสัญญาณไฟฟ้าซึ่งเป็นตัวแทนของตัวแปรที่สำคัญภายในหุ่นยนต์ เช่น ตำแหน่ง และ ความเร็ว ของข้อต่อต่างๆ โดยสัญญาณไฟฟ้าจะถูกสร้างขึ้นผ่านทางการ์ดดาต้าแอคควิซิชัน (Data Acquisition Card) ดังแสดงได้ดังรูปที่ 1



รูปที่ 1 โครงสร้างระบบจำลองหุ่นยนต์เสมือน

2.1 หุ่นยนต์สการา (SCARA Robot)

หุ่นยนต์ที่นำมาใช้เป็นตัวอย่างในการสร้างระบบจำลองหุ่นยนต์เสมือนโดยใช้การจำลองฮาร์ดแวร์ภายในรูปนั้นจะใช้หุ่นยนต์แบบสการา (SCARA: Selective Compliance Assembly Robot Arm) ที่มีการทำงานแบบ 2 แกนในแนวระนาบ มีจุดหมุนของข้อต่อจำนวน 2 จุด ดังรูปที่ 2



รูปที่ 2 แบบจำลองหุ่นยนต์สการา 2 แกน

โดยแบบจำลองทางคณิตศาสตร์ของหุ่นยนต์สการา สามารถเขียนได้ดังสมการที่ 1

$$\begin{bmatrix} (m_1 + m_2)a_1^2 + m_2a_2^2 + 2m_2a_1a_2 \cos \theta_2 & m_2a_2^2 + m_2a_1a_2 \cos \theta_2 \\ m_2a_2^2 + m_2a_1a_2 \cos \theta_2 & m_2a_2^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} -m_2a_1a_2(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2) \sin \theta_2 \\ m_2a_1a_2\dot{\theta}_1^2 \sin \theta_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad [1]$$

3. การออกแบบระบบจำลองหุ่นยนต์เสมือน

ในหัวข้อนี้จะกล่าวถึงการออกแบบระบบจำลองหุ่นยนต์เสมือนโดยใช้การจำลองฮาร์ดแวร์ภายในรูป โดยการออกแบบจะแบ่งออกเป็นสองส่วนคือ การออกแบบทางด้านฮาร์ดแวร์ และ การออกแบบทางด้านซอฟต์แวร์ ซึ่งมีรายละเอียดดังต่อไปนี้

3.1 การออกแบบทางด้านฮาร์ดแวร์

ในส่วนของการออกแบบทางด้านฮาร์ดแวร์ของระบบจำลองหุ่นยนต์เสมือนโดยใช้การจำลองฮาร์ดแวร์ภายในรูปนั้นจะมีส่วนประกอบหลักดังรูปที่ 3



รูปที่ 3 ส่วนประกอบหลักทางด้านฮาร์ดแวร์

จากรูปที่ 3 จะพบว่าส่วนประกอบหลักจะมีอยู่ด้วยกัน 3 ส่วน ดังต่อไปนี้

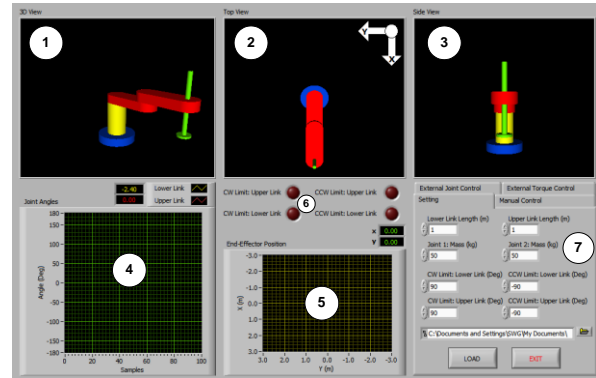
1 คอมพิวเตอร์ (Computer) ทำหน้าที่ประมวลผลการเคลื่อนที่ของหุ่นยนต์เสมือนที่สร้างขึ้นในโปรแกรม จากข้อมูลที่ได้รับมาจากการดักค่าแอดคิวชิชัน และส่งข้อมูลตัวแปรที่สำคัญของหุ่นยนต์เสมือนกลับไปที่การดักค่าแอดคิวชิชัน เพื่อแปลงเป็นสัญญาณไฟฟ้าส่งต่อไปให้ตัวควบคุมภายนอกต่อไป โดยการติดต่อระหว่างการ์ดดักค่าแอดคิวชิชันกับคอมพิวเตอร์กระทำผ่านทางพอร์ต ยู เอส บี (USB)

2 การ์ดดักค่าแอดคิวชิชัน (Data Acquisition Card: DAQ Card) ทำหน้าที่รับส่งสัญญาณทั้งแบบดิจิทัลและอนาลอกกับตัวควบคุมภายนอกเพื่อส่งต่อไปยังคอมพิวเตอร์ โดยสัญญาณที่เป็นดิจิทัลที่มีระดับสัญญาณสูงกว่าสัญญาณ TTL จะรับส่งผ่านทางบอร์ดเชื่อมต่อสัญญาณ (Interface Signals Board) ส่วนสัญญาณที่เป็นอนาลอกสามารถเชื่อมต่อโดยตรงเข้าที่การ์ดดักค่าแอดคิวชิชัน

3 บอร์ดเชื่อมต่อสัญญาณ (Interface Signals Board) ทำหน้าที่แปลงสัญญาณดิจิทัลจากอุปกรณ์ควบคุมภายนอกที่มีระดับสัญญาณต่างจากระดับสัญญาณ TTL ให้ไปเป็นระดับสัญญาณ TTL เพื่อให้สามารถรับส่งสัญญาณกับการดักค่าแอดคิวชิชันได้

3.2 การออกแบบทางด้านซอฟต์แวร์

ในส่วนของการออกแบบทางด้านซอฟต์แวร์ของระบบจำลองหุ่นยนต์เสมือนโดยใช้การจำลองฮาร์ดแวร์ภายในลูปนั้น ได้ใช้ซอฟต์แวร์ LabVIEW ในการเขียนโปรแกรม โดยออกแบบโปรแกรมให้ใช้งานได้ง่ายและมองเห็นค่าตัวแปรต่างๆที่สำคัญได้แบบเวลาจริง โดยโปรแกรมที่พัฒนาขึ้นนี้สามารถรองรับไฟล์ของโมเดลหุ่นยนต์ที่สร้างจากโปรแกรมเขียนแบบ CAD ที่มีนามสกุลเป็น ASE, STL และ VRML ได้ ส่วนประกอบของโปรแกรมสามารถแสดงได้ดังรูปที่ 4



รูปที่ 4 ส่วนประกอบของโปรแกรมจำลองหุ่นยนต์

เสมือน

จากรูปที่ 4 ส่วนประกอบของโปรแกรมจำลองหุ่นยนต์เสมือนจะมีอยู่ด้วยกัน 7 ส่วนดังต่อไปนี้

1. หน้าต่าง 3D View ใช้สำหรับแสดงผลภาพสามมิติของแบบจำลองหุ่นยนต์เสมือน
2. หน้าต่าง Top View ใช้สำหรับแสดงผลภาพด้านบนของแบบจำลองหุ่นยนต์เสมือน
3. หน้าต่าง Side View ใช้สำหรับแสดงผลภาพด้านข้างของแบบจำลองหุ่นยนต์เสมือน
4. กราฟ Joint Angles ใช้สำหรับแสดงค่ามุมของข้อต่อทั้งสองข้อต่อของแบบจำลองหุ่นยนต์เสมือน
5. กราฟ End-Effector Position ใช้สำหรับแสดงตำแหน่งในแนวระนาบของจุดปลายแขนหุ่นยนต์เสมือน

6. หลอดไฟแสดงสถานะลิมิตการเคลื่อนที่ของแขนที่ 1 (Lower Link) และ แขนที่ 2 (Upper Link) ของแบบจำลองหุ่นยนต์เสมือน โดยหลอดไฟจะติดเมื่อแขนของหุ่นยนต์เสมือนเคลื่อนที่เกินจากที่ตั้งค่าไว้

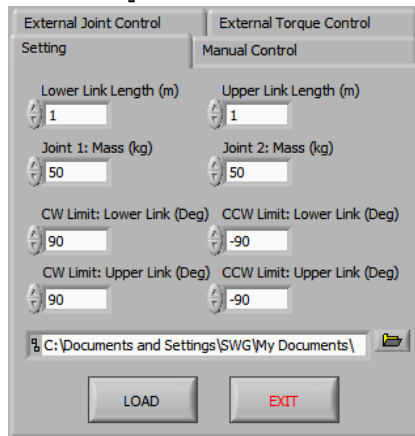
7. หน้าต่างแท็บเมนู เป็นหน้าต่างที่ใช้ในการตั้งค่าตัวแปรต่างๆของหุ่นยนต์เสมือนและตั้งค่าการทำงานของโปรแกรม

3.3 การตั้งค่าการทำงานของโปรแกรมจำลองหุ่นยนต์เสมือน

หลังจากเปิดโปรแกรมจำลองหุ่นยนต์เสมือนขึ้นมาแล้วสิ่งแรกที่ใช้ต้องทำคือการตั้งค่าการทำงานของ

ของโปรแกรมเสียก่อน ซึ่งมีขั้นตอนและรายละเอียดดังต่อไปนี้

1. ไปที่หน้าต่างแท็บเมนู คลิกที่แท็บ Setting ซึ่งจะมีรายละเอียดดังรูปที่ 5



รูปที่ 5 แท็บเมนู Setting

จากรูปที่ 5 สิ่งที่ต้องป้อนให้กับโปรแกรมมีดังต่อไปนี้

1.1 Lower Link Length (m) คือ ความยาวของแขนที่ 1 หรือ แขนด้านล่างของหุ่นยนต์เสมือน

1.2 Upper Link Length (m) คือ ความยาวของแขนที่ 2 หรือ แขนด้านบนของหุ่นยนต์เสมือน

1.3 Joint 1: Mass (kg) คือ น้ำหนักรวมของข้อต่อที่ 1 ของหุ่นยนต์เสมือน

1.4 Joint 2: Mass (kg) คือ น้ำหนักรวมของข้อต่อที่ 2 ของหุ่นยนต์เสมือน

1.5 CW Limit: Lower Link (Deg) คือ ค่าลิมิตของมุมการเคลื่อนที่ของแขนที่ 1 หรือ แขนด้านล่างของหุ่นยนต์เสมือน ในทิศทางตามเข็มนาฬิกา

1.6 CCW Limit: Lower Link (Deg) คือ ค่าลิมิตของมุมการเคลื่อนที่ของแขนที่ 1 หรือ แขนด้านล่างของหุ่นยนต์เสมือน ในทิศทางทวนเข็มนาฬิกา

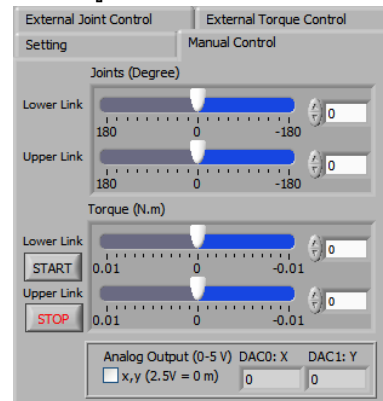
1.7 CW Limit: Upper Link (Deg) คือ ค่าลิมิตของมุมการเคลื่อนที่ของแขนที่ 2 หรือ แขนด้านบนของหุ่นยนต์เสมือน ในทิศทางตามเข็มนาฬิกา

1.8 CCW Limit: Upper Link (Deg) คือ ค่าลิมิตของมุมการเคลื่อนที่ของแขนที่ 2 หรือ แขนด้านบนของหุ่นยนต์เสมือน ในทิศทางทวนเข็มนาฬิกา

ส่วนปุ่ม LOAD มีไว้สำหรับการโหลดไฟล์แบบจำลองหุ่นยนต์เสมือนเข้าสู่โปรแกรม

3.4 การใช้งานแท็บเมนู Manual Control

แท็บเมนู Manual Control จะใช้สำหรับการควบคุมการเคลื่อนที่ของหุ่นยนต์จากภายในโปรแกรม โดยผู้ใช้งานเป็นคนป้อนค่าด้วยตัวเอง ซึ่งจะแบ่งออกเป็นสองโหมดย่อย คือ Joints Control และ Torque Control รายละเอียดของแท็บเมนู Manual Control แสดงในรูปที่ 6



รูปที่ 6 แท็บเมนู Manual Control

สิ่งที่ผู้ใช้งานต้องป้อนในเมนูนี้คือ

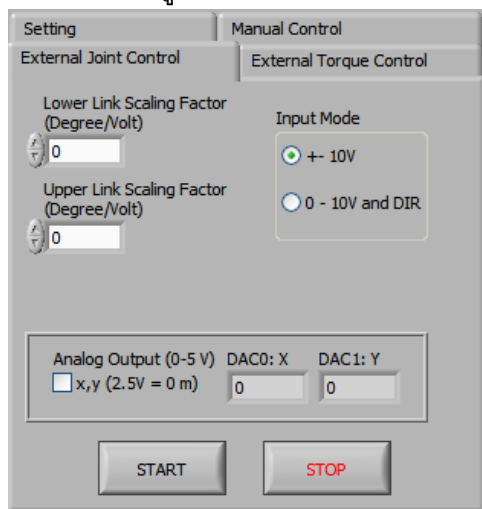
1. ในกรณีของ Joints Control ต้องป้อนค่าของมุมที่ข้อต่อที่ 1 และ ข้อต่อที่ 2 ของหุ่นยนต์เสมือน เพื่อสังเกตการณ์เคลื่อนที่ทั้งหมดของหุ่นยนต์ หลังจากป้อนค่าแล้วจะพบว่าหุ่นยนต์เสมือนจะมีการเคลื่อนที่และค่าของตัวแปรต่างๆในกราฟก็จะมีการพล็อตให้เห็นทันที

2. ในกรณีของ Torque Control ต้องทำการป้อนค่าแรงบิดที่กระทำต่อข้อต่อทั้งสองของหุ่นยนต์จากนั้นกดปุ่ม START หุ่นยนต์เสมือนก็จะเคลื่อนที่ตามแรงบิดที่ใส่เข้าไป และ ถ้าต้องการหยุดการเคลื่อนที่ให้กดปุ่ม STOP ส่วน Check Box: Analog Output (0-5 V) นั้นจะใช้สำหรับการส่งค่าสัญญาณอนาล็อกของตำแหน่งปลายแขนหุ่นยนต์ (x, y) ออกไปยัง Analog Output ของการ์ดดาต้าแอดคิวิตีชันเพื่อนำไปเป็นข้อมูลให้ตัวควบคุมภายนอกต่อไป

3.5 การใช้งานแท็บเมนู External Joint Control

แท็บเมนู External Joint Control จะใช้สำหรับจำลองการเคลื่อนที่ของหุ่นยนต์เสมือนจากตัวควบคุมภายนอกแบบการควบคุมมุมของข้อต่อ โดยที่ตัวควบคุมภายนอกจะต้องส่งสัญญาณอนาล็อกผ่านมา

ทางการ์ดดาต้าแอดคิวิตีชัน เพื่อเป็นข้อมูลให้โปรแกรมทราบว่าการเคลื่อนที่เป็นมุมเท่าใด รายละเอียดของแท็บเมนู External Joint Control สามารถแสดงได้ดังรูปที่ 7



รูปที่ 7 แท็บเมนู External Joint Control
สิ่งที่จะต้องป้อนให้โปรแกรมมีดังต่อไปนี้

1. Lower Link Scaling Factor (Degree/Volt) เป็นค่าคงที่แสดงถึงอัตราส่วนระหว่างมุมที่ข้อต่อที่ 1 หรือ ข้อต่อของแขนหุ่นยนต์เสมือนด้านล่างเคลื่อนที่ต่อ ค่าสัญญาณอนาลอกที่การ์ดดาต้าแอดคิวิตีชันอ่านได้จากตัวควบคุมภายนอก

2. Upper Link Scaling Factor (Degree/Volt) เป็นค่าคงที่แสดงถึงอัตราส่วนระหว่างมุมที่ข้อต่อที่ 2 หรือ ข้อต่อของแขนหุ่นยนต์เสมือนด้านบนเคลื่อนที่ต่อ ค่าสัญญาณอนาลอกที่การ์ดดาต้าแอดคิวิตีชันอ่านได้จากตัวควบคุมภายนอก

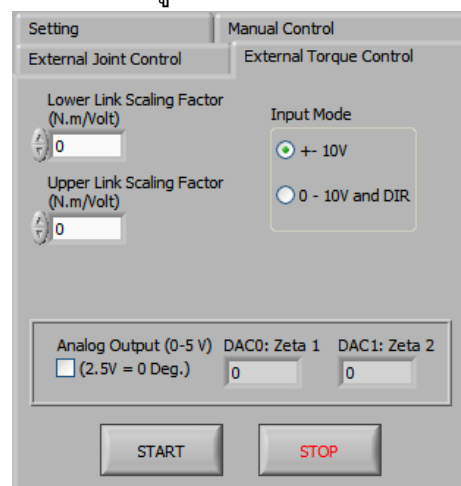
3. Input Mode ซึ่งหมายถึงสัญญาณอนาลอกที่ส่งเข้ามามีลักษณะอย่างไร ถ้าเลือกเป็น +- 10V หมายถึงสัญญาณที่ส่งเข้ามามีลักษณะเป็นบวกและลบ โดยเครื่องหมายจะบอกถึงทิศทางที่ต้องการให้เคลื่อนที่ ส่วนขนาดจะเป็นตัวบอกถึงจำนวนองศาที่ต้องการให้หมุน แต่ถ้าเลือกเป็น 0 - 10V and DIR จะหมายถึงสัญญาณที่เข้ามาจะมีสองส่วนคือ ขนาดของจำนวนองศาที่ต้องการให้หมุนจะเป็น 0 - 10V ส่วนทิศทางการหมุนให้ดูจากสัญญาณดิจิตอลของช่อง DIR ซึ่งต่อกับพอร์ต P0.0 และ P0.1 ของการ์ดดาต้าแอดคิวิตีชัน ส่วน Check Box: Analog Output (0-5 V)

นั้นจะใช้สำหรับการส่งค่าสัญญาณอนาลอกของตำแหน่งปลายแขนหุ่นยนต์ (x, y) ออกไปยัง Analog Output ของการ์ดดาต้าแอดคิวิตีชันเพื่อนำไปเป็นข้อมูลให้ตัวควบคุมภายนอกต่อไป

4. ถ้าต้องการให้หุ่นยนต์เสมือนเริ่มทำงานให้กดปุ่ม START ถ้าต้องการให้หยุดกดปุ่ม STOP

3.6 การใช้งานแท็บเมนู External Torque Control

แท็บเมนู External Torque Control จะใช้สำหรับจำลองการเคลื่อนที่ของหุ่นยนต์เสมือนจากตัวควบคุมภายนอกแบบการควบคุมแรงบิดที่กระทำต่อข้อต่อ โดยที่ตัวควบคุมภายนอกจะต้องส่งสัญญาณอนาลอกผ่านมาทางการ์ดดาต้าแอดคิวิตีชัน เพื่อเป็นข้อมูลให้โปรแกรมทราบว่าการเคลื่อนที่เป็นมุมเท่าใด รายละเอียดของแท็บเมนู External Torque Control สามารถแสดงได้ดังรูปที่ 8



รูปที่ 8 แท็บเมนู External Torque Control
สิ่งที่จะต้องป้อนให้โปรแกรมมีดังต่อไปนี้

1. Lower Link Scaling Factor (N.m/Volt) เป็นค่าคงที่แสดงถึงอัตราส่วนระหว่างแรงบิดที่ข้อต่อที่ 1 หรือ ข้อต่อของแขนหุ่นยนต์เสมือนด้านล่างเคลื่อนที่ต่อ ค่าสัญญาณอนาลอกที่การ์ดดาต้าแอดคิวิตีชันอ่านได้จากตัวควบคุมภายนอก

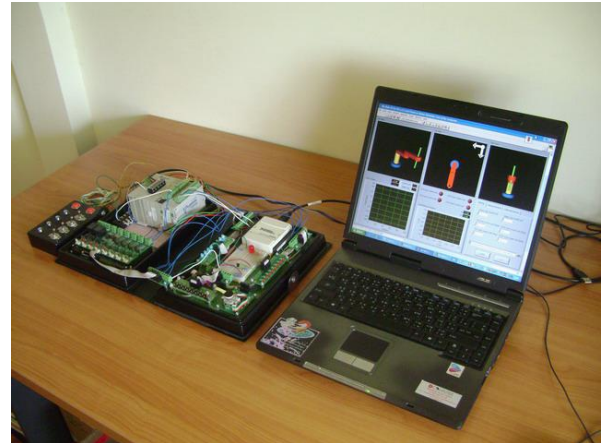
2. Upper Link Scaling Factor (N.m/Volt) เป็นค่าคงที่แสดงถึงอัตราส่วนระหว่างแรงบิดที่ข้อต่อที่ 2 หรือ ข้อต่อของแขนหุ่นยนต์เสมือนด้านบนเคลื่อนที่ต่อ ค่าสัญญาณอนาลอกที่การ์ดดาต้าแอดคิวิตีชันอ่านได้จากตัวควบคุมภายนอก

3. Input Mode ซึ่งหมายถึงสัญญาณอนาลอกที่ส่งเข้ามามีลักษณะอย่างไร ถ้าเลือกเป็น $\pm 10V$ หมายถึงสัญญาณที่ส่งเข้ามามีลักษณะเป็นบวกและลบ โดยเครื่องหมายจะบอกถึงทิศทางของแรงบิด ส่วนขนาดจะเป็นตัวบอกถึงแรงบิดที่กระทำ แต่ถ้าเลือกเป็น $0 - 10V$ and DIR จะหมายถึงสัญญาณที่เข้ามาจะมีสองส่วนคือ ขนาดแรงบิดที่กระทำ จะเป็น $0 - 10V$ ส่วนทิศทางของแรงบิดให้ดูจากสัญญาณดิจิตอลของช่อง DIR ซึ่งต่อกับพอร์ต P0.0 และ P0.1 ของการ์ดดาต้าแอดคิวิตีชัน ส่วน Check Box: Analog Output ($0-5 V$) นั้นจะใช้สำหรับการส่งค่าสัญญาณอนาลอกแทนตำแหน่งมุมของข้อต่อทั้งสองออกไปยัง Analog Output ของการ์ดดาต้าแอดคิวิตีชันเพื่อนำไปเป็นข้อมูลให้ตัวควบคุมภายนอกต่อไป

4. ถ้าต้องการให้หุ่นยนต์เสมือนเริ่มทำงานให้กดปุ่ม START ถ้าต้องการให้หยุดกดปุ่ม STOP

4. การทดลองและผลการทดลอง

การทดลองและผลการทดลองการใช้งานระบบจำลองหุ่นยนต์เสมือนโดยใช้การจำลองฮาร์ดแวร์ภายในรูป จะทดสอบทั้งในส่วนของ Manual Control และ External Control โดยในส่วนของ Manual Control จะทดลองป้อนค่าของมุมและแรงบิดของแต่ละข้อต่อของหุ่นยนต์จากภายในโปรแกรมและสังเกตการเคลื่อนที่ของหุ่นยนต์รวมถึงค่า Limit ต่างๆที่ตั้งไว้ว่าสามารถทำงานได้อย่างถูกต้องหรือไม่ และ ในส่วน External Control จะเป็นการนำตัวควบคุมแบบพีแอลซี (PLC) ซึ่งเป็นตัวควบคุมที่นิยมใช้ในอุตสาหกรรม มาทำการต่อร่วมกับระบบจำลองหุ่นยนต์เสมือนโดยใช้การจำลองฮาร์ดแวร์ภายในรูป เพื่อทดลองควบคุมหุ่นยนต์สกรูที่ได้จำลองขึ้นมา โดยสังเกตการเคลื่อนที่ของหุ่นยนต์รวมถึงค่า Limit ต่างๆที่ตั้งไว้ว่าสามารถทำงานได้อย่างถูกต้องหรือไม่ อุปกรณ์การทดลองทั้งหมดสามารถแสดงได้ดังรูปที่ 9



รูปที่ 9 อุปกรณ์ทดลองทั้งหมดเมื่อนำมาต่อรวมกัน

4.1 การทดลองแบบ Manual Control

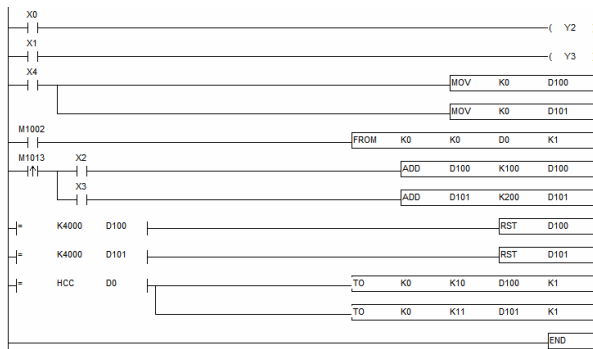
เป็นการทดลองป้อนค่าของมุมและแรงบิดของแต่ละข้อต่อของหุ่นยนต์จากภายในโปรแกรมและสังเกตการเคลื่อนที่ของหุ่นยนต์รวมถึงค่า Limit ต่างๆที่ตั้งไว้ว่าสามารถทำงานได้อย่างถูกต้องหรือไม่

4.2 ผลการทดลองแบบ Manual Control

ในส่วนของการเปลี่ยนค่ามุม (Joints (Degree)) ของ Lower Link และ Upper Link ไปเป็นค่าต่างๆ พบว่าการเคลื่อนที่ของหุ่นยนต์เป็นไปได้อย่างถูกต้อง และค่า Limit ต่างๆที่ตั้งก็สามารถทำงานได้อย่างถูกต้อง หลอดไฟเตือน Over Limit ของทั้งสองแกนก็สามารถทำงานได้อย่างถูกต้อง การค่อยๆเปลี่ยนค่าแรงบิดของ Lower Link และ Upper Link จะพบว่าหุ่นยนต์เคลื่อนที่ได้อย่างถูกต้อง และ หลอดไฟเตือน Over Limit ของทั้งสองแกนก็สามารถทำงานได้อย่างถูกต้อง

4.3 การทดลองแบบ External Control

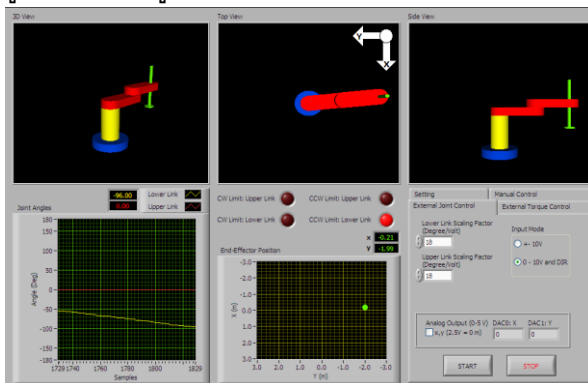
เป็นการนำตัวควบคุมแบบ พีแอลซี (PLC) ซึ่งเป็นตัวควบคุมที่นิยมใช้ในอุตสาหกรรม มาทำการต่อร่วมกับระบบจำลองหุ่นยนต์เสมือนโดยใช้การจำลองฮาร์ดแวร์ภายในรูป เพื่อทดลองควบคุมหุ่นยนต์สกรูที่ได้จำลองขึ้นมา โดยสังเกตการเคลื่อนที่ของหุ่นยนต์รวมถึงค่า Limit ต่างๆที่ตั้งไว้ว่าสามารถทำงานได้อย่างถูกต้องหรือไม่ โดยได้โปรแกรมตัวควบคุมแบบพีแอลซี ด้วยภาษาแลตเตอร์ไดอะแกรม (Ladder Diagram) ตัวอย่างของแลตเตอร์ไดอะแกรมสามารถแสดงได้ดังรูปที่ 10



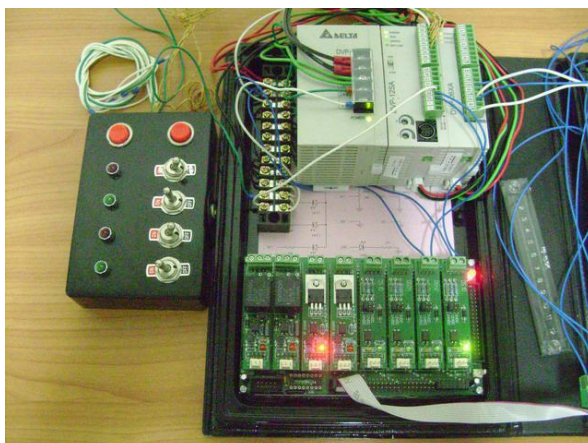
รูปที่ 10 แลตเตอร์ไคอะแกรมของ พี แอล ซี

4.4 ผลการทดลองแบบ External Control

จากผลการทดลองทั้งในส่วนของ External Joint Control และ External Torque Control พบว่าการเคลื่อนที่ของหุ่นยนต์เป็นไปได้อย่างถูกต้องและค่า Limit ต่างๆที่ตั้งก็สามารถทำงานได้อย่างถูกต้องเช่นกัน ตัวอย่างของผลการทดลองสามารถแสดงได้ดังรูปที่ 11 และ รูปที่ 12



รูปที่ 11 การเคลื่อนที่ของหุ่นยนต์เมื่อ X0 เป็น ON และ X2 เป็น ON ในส่วนของ External Joint Control



รูปที่ 12 สถานะของบอร์ดเชื่อมต่อสัญญาณระหว่าง ดาต้าแอดคิควิชชันกับตัวควบคุม พี แอล ซี

5. สรุปผลการทดลอง

จากผลการทดลองจะพบว่าการทำงานของระบบจำลองหุ่นยนต์เสมือนโดยใช้การจำลองฮาร์ดแวร์ภายในรูป ในส่วนของ Manual Control และ External Control ทั้งแบบ Joint Control และ Torque Control พบว่าระบบสามารถทำงานได้อย่างถูกต้อง โดยสามารถแสดงภาพการเคลื่อนที่ของหุ่นยนต์เสมือนแบบสามมิติได้อย่างราบรื่น ค่าลิมิตต่างๆสามารถแสดงผลได้อย่างถูกต้องทั้งในส่วนซอฟต์แวร์และฮาร์ดแวร์ และสามารถรับส่งสัญญาณกับตัวควบคุมภายนอกแบบ พี แอล ซี ได้อย่างถูกต้อง แต่จะพบว่าในส่วนของ External Control จะมีสัญญาณรบกวนเข้ามาในระบบทำให้การเคลื่อนที่ของหุ่นยนต์ไม่ราบรื่นในบางครั้ง ทั้งนี้เนื่องมาจากว่าในการทดลองไม่ได้ใช้สายสัญญาณแบบป้องกันสัญญาณรบกวนทั้งหมด ทำให้อาจมีสัญญาณรบกวนเข้ามาได้ อีกทั้งแหล่งจ่ายไฟของระบบไฟฟ้ากำลังกับระบบไฟฟ้าควบคุมเป็นแหล่งเดียวกันทำให้สัญญาณอาจรบกวนกันได้ แนวทางการแก้ไขคือ ใช้สายสัญญาณแบบป้องกันสัญญาณรบกวนทั้งหมด และ ควรแยกแหล่งจ่ายไฟของระบบไฟฟ้ากำลังออกจากระบบไฟฟ้าควบคุม

6. กิตติกรรมประกาศ

งานวิจัยนี้ได้รับทุนอุดหนุนการวิจัยจากเงินงบประมาณแผ่นดิน ประจำปีงบประมาณ พ.ศ. 2551 มหาวิทยาลัยศรีนครินทรวิโรฒ

7. เอกสารอ้างอิง

- [1] F. L. Lewis, C. T. Abdallah and D. M. Dawson, "Control of Robot Manipulators", Macmillan Publishing Co., 1993.
- [2] William A. Wolovich, "robotics:Basic Analysis and Design", CBS College Publishing, 1986.
- [3] R. Isermann, J. Schaffnit and S. Sinsel, "Hardware-in-the-loop simulation for design and testing of engine-control systems", Control Engineering Practice, Pergamon, vol.7, pp.643-653, 1999.

- [4] M. Linjama, T. Virvalo, J. Gustafsson, J. Lintula, V. Aaltonen and M. Kivikoski, "Hardware-in-the-loop environment for servo controller design, tuning and testing", *Microprocessors and Microsystems*, Elsevier, vol.24, pp.13-21, 2000.
- [5] Kenji Hagiwara, Satoshi Terayama, Youhei Takeda, Ko Yoda and Shoichi Suzuki, "Development of automatic transmission control system using hardware-in-the-loop simulation system", *JSAE Review*, Elsevier, vol.23, pp.55-59, 2002.
- [6] Panayiotis S. Shiakolas and Damrongrit Piyabongkarn, "Development of a Real-Time Digital Control System With a Hardware-in-the-Loop magnetic Levitation Device for Reinforcement of Controls Education", *IEEE Trans. on Education*, vol.46, no.46, pp.79-87, 2003.