บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในการควบคุมระบบแขนหุ่นยนต์ (Robot Manipulator) แบบใดๆ จำเป็นต้องมีโปรแกรมสำหรับการจำลองระบบพลวัต (Dynamic model) ของแขนหุ่นยนต์เพื่อใช้ในการทำงาน 2 ประการ ประการแรกใช้ในการทดสอบระบบควบคุมกับแบบจำลองของแขนหุ่นยนต์ ก่อนที่จะนำวิธีควบคุมนั้นไปใช้กับแขนจริง ส่วนประการที่สอง ใช้สำหรับการควบคุมการเคลื่อนที่ของแขนจริง เนื่องด้วยการควบคุมขั้นสูงบางวิธี ต้องอาศัยแบบจำลองพลวัตของแขนในการคำนวณหาสัญญาณควบคุม

นอกจากนั้นคลังโปรแกรม ยังช่วยให้ผู้ที่ต้องการพัฒนาวิธีการควบคุมแขนหุ่นยนต์ ทำงานสะดวกขึ้นกล่าวคือ ทำให้ผู้ใช้เขียนโปรแกรมเพิ่มเติมเฉพาะในส่วนของตัวควบคุม โดยไม่ต้องสนใจกับส่วนของแบบจำลองพลวัตและ ช่วยให้ผู้เริ่มทำการศึกษาด้านนี้สามารถเรียนรู้และเข้าใจได้เร็วขึ้น โดยโปรแกรมดังกล่าวควรมีคุณสมบัติ ดังนี้

1. มีความยืดหยุ่นให้ผู้ใช้สามารถเลือกลักษณะโครงสร้างของแขนได้ตามความต้องการ
2. สามารถคำนวณหาแบบจำลองได้เร็วเพียงพอที่จะนำไปใช้การควบคุมแขนหุ่นยนต์จริง
3. สามารถติดต่อกับผู้ใช้ผ่านทางภาพกราฟิก (Graphic User Interface หรือย่อว่า GUI) เพื่อความสะดวกในการใช้งาน
4. ไม่ต้องอาศัยซอฟต์แวร์ทางการค้าอื่นๆ ในการทำงาน เนื่องจากโปรแกรมหลายตัวที่พัฒนาก่อนนี้ต้องอาศัยซอฟต์แวร์ทางการค้า เช่น Matlab หรือ Mathematica

คลังโปรแกรมที่ได้จะนำไปใช้ในการจำลองระบบแขนหุ่นยนต์เพื่อควบคุมการเคลื่อนที่ โดยวิธีโครงสร้างแปรผันได้ (Variable Structure Control หรือย่อว่า VSC)

การหาแบบจำลองพลวัตแขนหุ่นยนต์

วิธีในการหาแบบจำลองพลวัตของแขนหุ่นยนต์สามารถหาได้ 2 รูปแบบ [18, 20, 22] คือ

1. รูปแบบลากรองจ์ (Lagrange Formulation) พัฒนาขึ้นโดย J. Uicker (1965), M. Kahn (1969) โดยแบบจำลองหาจากสมการลากรองจ์ ซึ่งจะได้มาจากการหาพลังงานจลน์และพลังงานศักย์รวมของแขน ข้อดีของวิธีนี้คือ มองแขนหุ่นยนต์ในภาพรวม สามารถแสดงให้เห็นถึงชนิดของแรงต่างๆที่มากระทำแต่ละข้อต่อ เช่น แรงจากความเฉื่อยของแขน, แรงโคริออลิส (Coriolis), แรงโน้มถ่วง ฯลฯ แต่เนื่องจากต้องอาศัยการคำนวณที่ซับซ้อนมาก กล่าวคือ มีต้นทุนในการคำนวณ (computation cost) เป็น $O(n^4)$ โดยที่ n เป็นจำนวนของข้อต่อ ทำให้ใช้เวลาในการคำนวณมากและไม่เหมาะสำหรับการจำลองระบบ

2. รูปแบบนิวตันออยเลอร์ (Newton-Euler Formulation) พัฒนาโดย D. Orin et al. (1979), W. Armstrong (1979), Luh et al. (1980) วิธีนี้มองแขนหุ่นยนต์เป็นท่อนๆ ซึ่งมีแรงกระทำระหว่างกัน ทำให้ไม่สามารถแยกแยะให้เห็นถึงชนิดของแรงต่างๆ ได้เหมือนวิธีแรก แต่มีข้อดีคือ ต้นทุนในการคำนวณ (computation cost) เป็น $O(n)$ เหมาะสำหรับการจำลองระบบและควบคุมแขนแบบเวลาจริง (real time)

คลังโปรแกรมในการหาแบบจำลองพลวัต

คลังโปรแกรมคอมพิวเตอร์สำหรับใช้จำลองระบบแขนหุ่นยนต์ มีความสามารถหลัก คือ การหาความสัมพันธ์ของแขนในส่วนของจลนศาสตร์ (kinematics) และการหาสมการพลวัตของแขนหุ่นยนต์แบบใดๆ โดยอาศัยวิธีการคำนวณตามรูปแบบของนิวตันออยเลอร์เนื่องจากสามารถหาแบบจำลองได้เร็วกว่ารูปแบบของลากรองจ์ ในส่วนของคลังโปรแกรมนี้ได้มีผู้พัฒนาขึ้นมาหลายรูปแบบ [4, 5, 7, 10, 13, 14] แต่ละแบบก็มีข้อดีข้อเสียแตกต่างกัน ซึ่งโดยมากขาดในส่วนติดต่อกับผู้ใช้แบบรูปกราฟิก การแสดงแขนกลเป็นรูปภาพ 3 มิติ และการควบคุมแขนกลวิธีต่างๆ ตัวอย่างของโปรแกรม ได้แก่

1. Robotica [14] มีข้อดี คือสามารถคำนวณหาแบบจำลองพลวัตได้ทั้งในรูปแบบปิด (closed form) และแบบเชิงเลข (numerical form) ซึ่งโดยทั่วไปจะหาเฉพาะแบบเชิงเลขเท่านั้น แต่ข้อเสียคือต้องอาศัยซอฟต์แวร์ของ Mathematica

2. Robotics Toolbox [5] เป็นชุดคำสั่งที่ใช้กับโปรแกรม Matlab แต่เนื่องจากลักษณะของ Matlab ที่ไม่สามารถใช้ทรัพยากรในการคำนวณได้อย่างมีประสิทธิภาพ ทำให้ใช้เวลาในการคำนวณนาน ไม่เหมาะกับการนำไปใช้ควบคุมแบบเวลาจริง (real time) [7]

3. ROBOOP [7] โปรแกรมเขียนโดยใช้ภาษา C++ ซึ่งมีลักษณะเป็นภาษาเชิงวัตถุ (object oriented language) สามารถใช้กับการทำงานแบบเวลาจริงได้ และไม่ต้องอาศัยซอฟต์แวร์อื่น แต่ยังขาดในส่วนของการแสดงผลแบบรูปกราฟิก

4. Xanimate [11] โปรแกรมเขียนโดยใช้ภาษา C และใช้ Athena Widget สำหรับสร้างส่วนติดต่อกับผู้ใช้แบบกราฟิก แต่ตัวโปรแกรมนี้ค่อนข้างล้าสมัยและยังขาดการคำนวณในส่วนของระบบพลวัตแขนกล

5. QMotor Robotic Toolkit [10] เป็นโปรแกรมที่ทำงานบน Matlab สามารถใช้ควบคุมแขนกล โดยผ่านทางแบบจำลองบน Simulink ทำให้สามารถใช้งานได้ง่ายในการควบคุมแขนจริง แต่ก็เป็นโปรแกรมที่ต้องเสียค่าใช้จ่าย และมีส่วนกราฟิกเฉพาะแขนแบบ Puma

การควบคุมการเคลื่อนที่ของแขนหุ่นยนต์

แขนกลเป็นระบบที่มีความไม่เป็นเชิงเส้นสูงทำให้ยากต่อการออกแบบระบบควบคุม ทำให้ในโรงงานอุตสาหกรรมทั่วๆไป นิยมใช้ตัวควบคุมที่ออกแบบโดยอาศัยทฤษฎีเชิงเส้น เช่น PID ซึ่งเป็นการพิจารณาให้ข้อต่อแต่ละข้อต่อเป็นอิสระต่อกันและให้ผลของความไม่เป็นเชิงเส้นต่างๆ ให้เป็น

สัญญาณรบกวน การควบคุมนี้ให้ผลดีเฉพาะในกรณีที่แขนไม่ได้เคลื่อนที่เร็วนัก แต่เมื่อต้องการให้แขนเคลื่อนที่เร็วขึ้น ความคลาดเคลื่อนจากแนวทางที่ต้องการจะมีค่ามากขึ้นตาม [18] โดยเฉพาะในกรณีที่มีการเปลี่ยนแปลงของโหลด [17] ดังนั้นเพื่อให้ได้การควบคุมได้ผลดีจึงต้องออกแบบการควบคุมโดยพิจารณาส่วนของความไม่เป็นเชิงเส้นด้วย

ในกรณีที่รู้แบบจำลองและค่าพารามิเตอร์ของแขนกลอย่างถูกต้อง เราสามารถหักล้างความไม่เป็นเชิงเส้นทั้งหมด ทำให้ได้ระบบใหม่ที่เป็นเชิงเส้น ซึ่งกล่าวละเอียดในส่วนของการควบคุมแบบพลวัตผกผันแต่ในความเป็นจริงจะต้องมีความไม่แน่นอนเกิดขึ้นทั้งในแบบจำลอง และค่าพารามิเตอร์ การจัดการความไม่แน่นอนของค่าพารามิเตอร์นี้ทำได้ 2 วิธีหลักๆ คือ การควบคุมแบบปรับตัวเอง และ ควบคุมแบบคงทน

1. วิธีควบคุมแบบปรับตัวเอง (adaptive control) [15] จะพยายามเรียนรู้ถึงค่าพารามิเตอร์ของระบบและปรับค่าพารามิเตอร์ในตัวควบคุมเพื่อให้ความคลาดเคลื่อนลดลง จึงทำให้ผลการควบคุมที่ได้มีประสิทธิภาพดีและจะมีช่วงของพารามิเตอร์ในการทำงานกว้าง แต่มีข้อเสียคือ ต้องมีภาระในการคำนวณค่าพารามิเตอร์ตลอดเวลา และเนื่องจาก พารามิเตอร์ของแขนกลมีการเปลี่ยนแปลงอย่างรวดเร็วตลอดเวลา ทำให้การควบคุมแบบปรับตัวเองไม่เหมาะสมที่จะใช้ ในการควบคุมแบบเวลาจริง [17] รวมถึงความไม่คงทนต่อความผิดพลาดของแบบจำลองที่เกิดขึ้น หรือเมื่อมีสัญญาณรบกวน

2. วิธีควบคุมแบบคงทน (robust control) [3] จะออกแบบตัวควบคุมที่สามารถจัดการค่าของพารามิเตอร์ในช่วงหนึ่งได้โดยไม่ต้องมีการปรับตัว ทำให้ประสิทธิภาพในการควบคุมจะด้อยกว่าการควบคุมแบบปรับตัวเอง แต่สามารถคำนวณหาสัญญาณควบคุมได้เร็วกว่า และมีความคงทนต่อความไม่แน่นอนของแบบจำลองและสัญญาณรบกวน

ในส่วนของการควบคุมแบบ VSC ซึ่งเป็นการควบคุมแบบคงทนวิธีหนึ่ง ถูกเสนอขึ้นครั้งแรกโดย Utkin (1977) และได้ถูกนำมาประยุกต์ใช้กับระบบต่างๆมากมาย [6, 25] สำหรับระบบแขนหุ่นยนต์นั้นเริ่มครั้งแรกจากงานของ Young (1978) และมีผู้ทำการพัฒนาต่อมาจำนวนมาก [3]

หลักการของ VSC คืออาศัยการสวิตชิ่งเพื่อเอาชนะ ความไม่แน่นอนของพารามิเตอร์ ทำให้เป็นข้อดีของ VSC เนื่องจากเป็นการควบคุมแบบง่ายๆ ไม่ต้องมีการคำนวณที่ยุ่งยากทำให้สามารถทำงานแบบเวลาจริงได้และมีความคงทนต่อความไม่แน่นอนและต่อสิ่งรบกวน แต่ก็มีข้อเสีย 2 ประการ ประการแรก การเกิดปรากฏการณ์ที่สวิตช์ถูกผลักกลับไปมาด้วยความถี่สูง หรือที่เรียกว่า แชทเทอริ่ง (chattering) ซึ่งมีผู้พยายามแก้โดยใช้วิธีต่างๆ เช่น ออกแบบชั้นเขตแดน (boundary layer) เพื่อให้สัญญาณควบคุมที่ได้มีความต่อเนื่องในบริเวณพื้นผิว Slotine (1985) แต่ก็จะมีความคลาดเคลื่อนเกิดขึ้น ซึ่งขนาดความคลาดเคลื่อนขึ้นอยู่กับขนาดความกว้างของชั้นเขตแดนปัญหาอีกประการหนึ่ง คือ ต้องรู้อัตราขยายของตัวสวิตชิ่งที่ได้จากขอบเขตความไม่แน่นอนของพารามิเตอร์ ถ้าอัตราขยายมีค่าสูงจะทำให้ขนาดการเกิดแชทเทอริ่งสูงมากขึ้น แต่ถ้าน้อยไปจะทำให้แนวการเคลื่อนที่คลาดเคลื่อนจากแนวทางที่กำหนด ส่วนวิธีการแก้ไขได้มีผู้เสนอให้พารามิเตอร์ของขอบเขตความไม่แน่นอนในตัวควบคุม สามารถปรับค่าได้ซึ่งค่าจะลู่เข้าสู่ค่าขอบเขตที่แท้จริงได้ [21] ต่อมามีการพัฒนาวิธีการปรับตัวเพิ่มขึ้น [26] เพื่อให้ระบบควบคุมในส่วนของการ

ปรับตัวทนต่อสัญญาณรบกวนมากขึ้น

1.2 จุดประสงค์ในการทำวิทยานิพนธ์

สร้างคลังโปรแกรมสำหรับจำลองระบบแขนหุ่นยนต์ และส่วนติดต่อกับผู้ใช้ทางภาพกราฟิก สำหรับใช้เป็น เครื่องมือในการศึกษาความรู้ด้านแขนหุ่นยนต์และทฤษฎีการควบคุมการเคลื่อนที่แบบต่างๆ

1.3 ขอบเขตของวิทยานิพนธ์

งานวิทยานิพนธ์ประกอบด้วย 2 ส่วนหลัก ได้แก่

1. เขียนคลังโปรแกรมภาษา C สำหรับการคำนวณในส่วนของจลนศาสตร์และสมการพลวัตของแขนหุ่นยนต์ โดยคุณลักษณะของคลังโปรแกรมนี้ ประกอบด้วย
   - คลังโปรแกรมนี้ทำงานบนระบบปฏิบัติการลินุกซ์ (Linux) แต่สามารถนำรหัสต้นฉบับ (source code) มาทำการแปลโปรแกรม (compile) เพื่อให้ทำงานบนระบบปฏิบัติการอื่นๆ ได้
   - คลังโปรแกรมสามารถคำนวณหาจลนศาสตร์ (Kinematics) และส่วนของพลวัต (Dynamics) สำหรับแขนกลแบบท่อนแขนเรียงลำดับ (serial-link manipulators) ซึ่งผู้ใช้สามารถเลือกจำนวนของข้อต่อ (joint), ชนิดของข้อต่อ, ความยาวหรือค่าพารามิเตอร์อื่นๆ ของแขนได้เอง โดยรูปแบบของพารามิเตอร์ที่ใช้ในการอธิบายถึงโครงสร้างของแขน เป็นรูปแบบที่เป็นมาตรฐานเพื่อสามารถนำไปเปรียบเทียบกับคลังโปรแกรมอื่น [5, 7] โดยง่าย
   - สามารถคำนวณเร็วเพียงพอ สำหรับการทำงานแบบเวลาจริง บนเครื่องคอมพิวเตอร์ที่มี CPU ความเร็วมากกว่า Pentium 133 MHz ขึ้นไป และมีหน่วยความจำที่มากกว่า 24 MB
   - ผู้ใช้มีหน้าที่เพียงแค่เขียนโปรแกรมหลัก (main program) เพื่อเรียกใช้คลังโปรแกรมนี้ในการแปลโปรแกรมและเชื่อมโยง (link) เป็นไฟล์กระทำการ (executable file) และผู้ใช้สามารถเขียนฟังก์ชันอื่นเพิ่มเติมได้ตามความต้องการ
   - ตัวคลังโปรแกรมภาษา C ไม่ต้องอาศัยซอฟต์แวร์ทางการค้าอื่นๆ ในการทำงาน
2. ทำส่วนติดต่อกับผู้ใช้ทางภาพกราฟิก โดยมีคุณลักษณะ ดังนี้
   - สร้างขึ้นจากโปรแกรม GTK (Gimp Toolkits) เนื่องจากโปรแกรมนี้สามารถหาได้ในอินเทอร์เน็ต และนำมาใช้ได้อย่างถูกลิขสิทธิ์โดยไม่ต้องเสียค่าใช้จ่าย
   - สามารถแสดงภาพเคลื่อนไหว (animation) การเคลื่อนที่ของแขนกลในรูปแบบของภาพ 3 มิติ
   - สามารถแสดงกราฟของสัญญาณต่างๆ เช่น แรงบิด, ตำแหน่ง, ความเร็ว
   - สามารถเลือกรูปกราฟเพื่อเก็บเป็นไฟล์ข้อมูล สำหรับนำมาแสดงภายหลังโดยใช้โปรแกรมอื่น เช่น Matlab, gnuplot (โปรแกรมวาดกราฟบนลินุกซ์)

- มีฟังก์ชันสำหรับใช้ในการควบคุมแขนหุ่นยนต์โดยวิธีควบคุมแบบพลวัตผกผัน (Inverse Dynamic Control), VSC และตัวควบคุม PID ซึ่งผู้ใช้สามารถปรับเปลี่ยนค่าพารามิเตอร์ในตัวควบคุมเหล่านี้ได้ โดยไม่ต้องเขียนฟังก์ชันอื่นเพิ่มเติม
- เปรียบเทียบความถูกต้องและความเร็วของการคำนวณ ในส่วนของจลนศาสตร์, พลวัต และ ผลการจำลองการเคลื่อนที่ของแขนกล ที่ได้จากคลังโปรแกรมนี้ กับผลจาก Robotics Toolbox [5] โดยให้ค่าความผิดพลาดอยู่ในขอบเขตที่ยอมรับได้

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษาความรู้พื้นฐานของแขนหุ่นยนต์ในส่วนของจลนศาสตร์และสมการพลวัต
2. เขียนคลังโปรแกรมภาษา C สำหรับคำนวณหาจลนศาสตร์และสมการพลวัตสำหรับแขน หุ่นยนต์แบบใดๆ
3. ศึกษาการใช้โปรแกรม GTK เพื่อทำส่วนของการติดต่อกับผู้ใช้ทางภาพกราฟิก
4. ทำการจำลองระบบแขนหุ่นยนต์และควบคุมการเคลื่อนที่โดยวิธีโครงสร้างแปรผันได้
5. ทำเอกสารวิธีการใช้คลังโปรแกรม

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้คลังโปรแกรมสำหรับใช้ในการจำลองระบบและควบคุมแขนหุ่นยนต์ ซึ่งจะเป็นประโยชน์ต่อผู้ต้องการพัฒนาวิธีการควบคุมแบบต่างๆ
2. ส่วนของการติดต่อกับผู้ใช้ทางภาพกราฟิก เพื่อช่วยให้ผู้เริ่มต้นศึกษาวิชาด้านนี้สามารถเข้าใจการทำงานได้เร็วและชัดเจนขึ้น

1.6 โครงสร้างของวิทยานิพนธ์

วิทยานิพนธ์นี้ประกอบด้วยเนื้อหาทั้งหมด 5 บท โดยแต่ละบทมีเนื้อหาดังนี้

บทที่ 1 เป็นการกล่าวถึงที่มาและความสำคัญของปัญหา จากนั้นเป็นส่วนของจุดประสงค์ในการทำวิทยานิพนธ์ ขอบเขตของวิทยานิพนธ์ ขั้นตอนในการดำเนินงานและประโยชน์ที่คาดว่าจะได้รับ

บทที่ 2 กล่าวถึงความรู้พื้นฐานของแขนหุ่นยนต์ได้แก่ จลนศาสตร์และการหาสมการพลวัตสำหรับใช้ในการสร้างคลังโปรแกรม และทฤษฎีการควบคุมการเคลื่อนที่แบบพลวัตผกผัน, VSC และ PID

บทที่ 3 กล่าวถึงโครงสร้างของคลังโปรแกรมที่สร้างขึ้น รายละเอียดของตัวแปรแขนกลที่นิยามขึ้นฟังก์ชันต่างๆ ภายในคลังโปรแกรม และตัวอย่างในการใช้งาน รวมถึงการเปรียบเทียบผลการคำนวณกับ Robotic Toolbox

บทที่ 4 กล่าวถึงความสามารถของ ส่วนติดต่อกับผู้ใช้ทางรูปกราฟิก และตัวอย่างการใช้งานสำหรับจำลองระบบแขนกล โดยใช้การควบคุมทั้งสามวิธี

บทที่ 5 เป็นบทสรุปของงานที่ได้ทำมา ข้อดีและข้อเสีย รวมทั้งข้อเสนอแนะในการปรับปรุงคลังโปรแกรมนี้

ภาคผนวก ก และ ข เป็นส่วนของวิธีการใช้งาน การติดตั้ง คลังโปรแกรมนี้ และส่วนติดต่อกับผู้ใช้ทางรูปกราฟิก