บทที่ 1

บทนำ

ความสำคัญและที่มาของโครงการวิทยานิพนธ์

ในปัจจุบันมีงานอุตสาหกรรมหลายๆ ประเภทได้มีการนำเอาหุ่นยนต์อุตสาหกรรมมาใช้งานกันอย่างกว้างขวาง เช่น งานประเภทจับและวาง (pick and place) , งานในด้านการประกอบ (assembly) , งานพ่นสี (spray painting) หรืองานเชื่อมประเภทต่างๆ (spot and arc welding) ซึ่งหุ่นยนต์ในปัจจุบันได้มีการพัฒนาจากผู้ผลิตจนมีขีดความสามารถที่มนุษย์ยอมรับได้ กล่าวคือ มีความเที่ยงตรงมากขึ้น มีความสามารถทำงานได้อย่างต่อเนื่อง และทำงานเป็นระบบกับเครื่องมือประเภทอื่นๆ เช่น เครื่อง CNC (Computer Numerical Control) หรือระบบลำเลียงชิ้นงาน (material handling) เป็นต้น ซึ่งจากประโยชน์ดังกล่าวทำให้ได้มีการพัฒนาและนำหุ่นยนต์มาใช้ในงานอุตสาหกรรมมากขึ้นตามลำดับ

เมื่อพิจารณาขั้นตอนการใช้งาน เราจะพบว่าสิ่งที่สำคัญที่สุดที่ใช้ในการควบคุมหุ่นยนต์ให้ทำงานในลักษณะต่างๆ ก็คือ วิธีการโปรแกรม (Programming Technique) ซึ่งมีหลายวิธี เริ่มตั้งแต่การสอนและสั่งให้หุ่นยนต์ทำตามที่สอน (Teach and Repeat) ซึ่งเป็นวิธีการพื้นฐานที่ใช้งานได้สะดวก แต่ก็มีข้อจำกัด เช่นจะต้องมีหุ่นยนต์พร้อมอุปกรณ์อื่นๆที่ใช้งานร่วมกันก่อนที่จะเริ่มทำการสอนหุ่นยนต์ , การไม่สามารถทำงานในลักษณะที่ซับซ้อน และเมื่อมีการเปลี่ยนแปลงลักษณะงาน ก็ต้องมาสอนใหม่ และเก็บไว้ วิธีต่อมาคือการใช้ภาษาควบคุม (Programming Language) ซึ่งต้องใช้ความรู้ด้านการเขียนโปรแกรมแต่สามารถสั่งให้หุ่นยนต์ทำงานที่มีความซับซ้อนได้ตามระดับของภาษาที่ใช้ แต่ถึงกระนั้นในงานซับซ้อนบางประเภทจำเป็นต้องมีการจำลองการทำงานของหุ่นยนต์เสียก่อนเพื่อป้องกันความผิดพลาดในการออกแบบและวางระบบการทำงาน ทำให้ในปัจจุบันได้มีการนำระบบ Computer-Aided-Design (CAD) ซึ่งเป็นการนำคอมพิวเตอร์มาช่วยออกแบบในอุตสาหกรรมการผลิตมาประยุกต์ใช้เข้าด้วยกันกับวิธีการโปรแกรมโดยใช้ภาษา ซึ่งทำให้สามารถศึกษาและพิจารณาการทำงาน การวางแผนการทำงาน การเปลี่ยนแปลงลักษณะงานของหุ่นยนต์ รวมถึงการควบคุมหุ่นยนต์ให้ทำงานตามลักษณะงานนั้นๆ ได้อย่างถูกต้องตามต้องการ

ในโครงการวิทยานิพนธ์ฉบับนี้จะเป็นการศึกษาการจำลองการทำงานของหุ่นยนต์ในระบบการผลิตอัตโนมัติ MPS โดยใช้ระบบ CAD และทำการเขียนโพสต์โปรเซสเซอร์เพื่อแปลงข้อมูล

ที่ได้จากการจำลองการทำงานของแขนกลบนคอมพิวเตอร์ไปเป็นภาษาที่ใช้ในการควบคุมโดย ในท้ายที่สุดจะนำภาษาควบคุมที่ได้ดังกล่าวไปใช้ควบคุมแขนกลจริงให้ทำงานภายในระบบผลิตอัตโนมัติตามต้องการ

วัตถุประสงค์ของโครงการวิทยานิพนธ์

1. เพื่อศึกษาถึงวิธีการโปรแกรมหุ่นยนต์แบบ OFF-LINE Programming โดยวิเคราะห์และวางแผนลักษณะการทำงานของหุ่นยนต์ในเซลการทำงาน และทำการจำลองการทำงาน (simulation) ของหุ่นยนต์และอุปกรณ์อื่นด้วยคอมพิวเตอร์

2. เพื่อศึกษาและเขียนโพสต์โปรเซสเซอร์เพื่อใช้ในการแปลงข้อมูลที่ได้จากการจำลองการทำงานหุ่นยนต์ Mitsubishi บนโปรแกรม CAD (CATIA) ไปเป็นภาษาที่ใช้ในการควบคุมแขนกลจริงให้ทำงานตามที่ต้องการ

3. เพื่อศึกษาถึงวิธีการส่งข้อมูลด้วยการสื่อสารแบบอนุกรม , ระบบโปรฟิบัส และระบบผลิตอัตโนมัติแบบ MPS ซึ่งลักษณะการทำงานของแต่ละหน่วยงานจะถูกควบคุมด้วย PLC และ PLC ของแต่ละหน่วยงานจะเชื่อมต่อกันผ่านเครือข่าย (network) โดยใช้พิธีการ (protocol) แบบ Profibus

4. เพื่อเป็นพื้นฐานในการพัฒนาเทคโนโลยีในอุตสาหกรรมผ่านการควบคุมโดยไมโครคอมพิวเตอร์ และการสื่อสารข้อมูลผ่านระบบเครือข่าย

ขั้นตอนและขอบเขตการดำเนินโครงการวิทยานิพนธ์

1. ศึกษาทฤษฎีเกี่ยวกับหุ่นยนต์และลักษณะการโปรแกรมหุ่นยนต์วิธีต่างๆ

2. ศึกษาวิธีการใช้งานซอฟท์แวร์ Robot Module ใน CATIA version4

3. ทำการกำหนดลักษณะงานของแขนกล ซึ่งเป็นงานประเภทเคลื่อนย้ายชิ้นงานในระบบจำลองการผลิตโดยมีขั้นตอนดังต่อไปนี้

3.1.ทำการหยิบชิ้นงาน ซึ่งมีลักษณะเป็นทรงกระบอกตัน แบน จากสายพานลำเลียงไปยังโต๊ะกำหนดตำแหน่ง

3.2.ทำการเคลื่อนย้ายชิ้นงานจากโต๊ะกำหนดตำแหน่งเข้าในที่จัดเก็บตามลักษณะข้อมูลการตรวจสอบ ซึ่งเป็นลักษณะการใส่ชิ้นงาน ลงในแมกกาซีนจัดเก็บโดยการสวมลงไป หรือ การวางชิ้นงานบนสถานีแยกชิ้นงาน

และทำการจำลองและวิเคราะห์ลักษณะการทำงานข้างต้นบนคอมพิวเตอร์ ก่อนที่จะนำข้อมูลไปใช้ในการควบคุมแขนกลในระบบงานจริงต่อไป

4. ศึกษาการควบคุมและภาษาที่ใช้ในตัวควบคุม (controller) ของแขนกล Mitsubishi

5. เขียนโพสต์โปรเซสเซอร์ (post-processor) เพื่อใช้ในการแปลงไฟล์ข้อมูลเอาต์พุต (output file) ที่ได้จากการจำลองลักษณะงานดังที่ได้กล่าวในข้อ 3 ไปเป็นภาษาที่ใช้ในการควบคุมหุ่นยนต์

6. ศึกษาการใช้งานซอฟท์แวร์สื่อสาร (communication software) เพื่อติดต่อสื่อสารกับตัวควบคุมโดยผ่านพอร์ตอนุกรม (RS-232 Interface)

7. ศึกษาการส่งข้อมูลผ่าน Profibus (Process Field Bus) ภายในระบบ MPS

8. ทดลองควบคุมแขนกลจริงตามลักษณะงานที่ได้กำหนดไว้

9. สรุปผลโครงการวิทยานิพนธ์

ประโยชน์ที่คาดว่าจะได้รับจากโครงการวิทยานิพนธ์

1. ได้หลักการทำงานของระบบ MPS ซึ่งเป็นระบบ FMS (Flexible Manufacturing System) ประเภทหนึ่งซึ่งประกอบไปด้วยสถานีการทำงานต่างๆ หุ่นยนต์ ระบบลำเลียงชิ้นงาน และระบบควบคุมด้วยคอมพิวเตอร์โดยมีการสื่อสารข้อมูลผ่านโปรฟิบัส ซึ่งจะเป็นประโยชน์ในการควบคุมระบบการผลิตอัตโนมัติในอุตสาหกรรม

2. ได้โปรแกรมโพสต์โปรเซสเซอร์สำหรับหุ่นยนต์ Misubishi ที่สามารถนำไปใช้เพื่อศึกษาและประยุกต์ใช้ควบคุมหุ่นยนต์ในงานอุตสาหกรรม รวมไปถึงการพัฒนาโปรแกรมเพิ่มเติมเพื่อใช้กับหุ่นยนต์หรืออุปกรณ์ประเภทอื่นๆที่ควบคุมด้วย PLC ได้ต่อไป

3. เป็นพื้นฐานในการพัฒนาการควบคุมหุ่นยนต์แบบ Off-line Programming โดยใช้ระบบ CAD มาช่วยจำลองและวิเคราะห์สภาพการทำงานของหุ่นยนต์ก่อนทำงานจริง ซึ่งจะสามารถนำไปประยุกต์ใช้เพื่อให้เกิดประโยชน์ในอุตสาหกรรมประเภทต่างๆ เช่น การเชื่อม การประกอบ และ การพ่นสี เป็นต้น

4. เป็นพื้นฐานในการศึกษาเกี่ยวกับระบบการติดต่อสื่อสารข้อมูลภายในระบบผลิตอัตโนมัติในอุตสาหกรรม

ประเภทของการโปรแกรมหุ่นยนต์ (Method of Robot Programming)

สามารถแบ่งได้เป็น 2 ประเภทพื้นฐานคือ

1) ON-LINE PROGRAMMING METHOD

เป็นการโปรแกรมโดยตรงจากการควบคุมดำเนินการของมนุษย์ซึ่งหุ่นยนต์จะทำงานตามคำสั่งทีละคำสั่งตามการควบคุมของผู้ควบคุม (programmer) โดยข้อมูลของตำแหน่งและความเร็วที่ใช้ในการเคลื่อนที่จะถูกอ่านค่าโดยตรงจากอุปกรณ์ป้อนกลับ (feedback devices) บนแขนของหุ่นยนต์ ซึ่งวิธีการโปรแกรมแบบนี้จะเป็นวิธีที่ง่ายในการ ใช้งานแต่จะมีปัญหาเมื่อนำไปใช้กับงานที่มีความซับซ้อน

1.1) Teach-and-Repeat Method

เป็นการควบคุมหุ่นยนต์โดยการใช้แป้นบังคับ (pendant) ควบคุมการเคลื่อนที่ของแต่ละข้อต่อ (joint) ของหุ่นยนต์ ไปในตำแหน่งที่ต้องการ โดยใช้ความเร็วซึ่งสามารถควบคุมได้จากแป้นบังคับ นอกจากนี้อาจมีฟังก์ชันที่เกี่ยวกับการบีบหรือปล่อยของมือจับ (gripper) ประกอบด้วย หลังจากควบคุมการเคลื่อนที่แล้วลักษณะการเคลื่อนที่ดังกล่าวจะถูกบันทึกไว้ในหน่วยความจำ ซึ่งจะสามารถสั่งให้หุ่นยนต์ทำงานซ้ำได้ในภายหลัง ดังแสดงในรูปที่ 1.1

1.2) Lead-Through Method

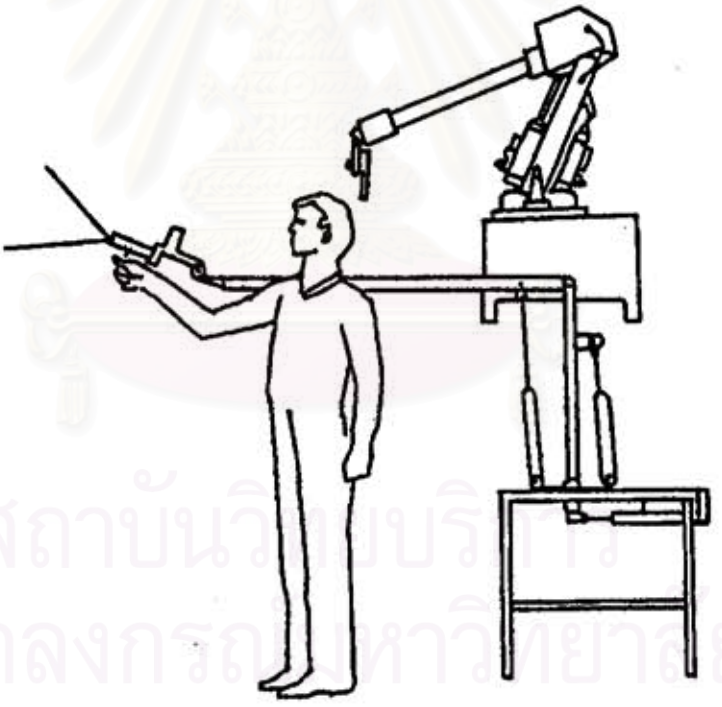
เป็นวิธีการโปรแกรม ที่เหมาะสมกับการควบคุมหุ่นยนต์บนเส้นทางการเคลื่อนที่ที่ต่อเนื่อง (continuous path) และการทำงานในสภาวะแวดล้อมที่เป็นอันตรายซึ่งไม่เหมาะสมในการโปรแกรมด้วยแป้นบังคับ โดยผู้ควบคุมจะเป็นผู้กำหนดการเคลื่อนที่จากอุปกรณ์จำลองการเคลื่อนที่ที่เรียกว่า master โดยหุ่นยนต์หรือ slave จะเคลื่อนที่ในลักษณะเดียวกันก่อนที่ลักษณะการเคลื่อนที่ดังกล่าวจะถูกเก็บบันทึกและถูกนำมาใช้เพื่อทำให้หุ่นยนต์เคลื่อนที่โดยอัตโนมัติในภายหลัง ดังแสดงในรูปที่ 1.2

2) OFF-LINE PROGRAMMING METHOD

เป็นวิธีการโปรแกรม ที่เตรียมและสร้างโปรแกรมที่เครื่องคอมพิวเตอร์ (computer terminal) หนึ่งๆ และถ่ายโอนโปรแกรมดังกล่าวเข้าสู่ตัวควบคุมของหุ่นยนต์ในภายหลัง โดยจะสามารถเตรียมและทดสอบโปรแกรมใหม่ได้ ในขณะที่หุ่นยนต์ยังทำงานหนึ่งๆอยู่ วิธีดังกล่าวจะมีความคล้ายคลึงกับขั้นตอนการทำ off-line ที่ใช้ใน NC part programming ซึ่งจะสามารถลดเวลาที่ใช้ในการเปลี่ยนโปรแกรม อันจะส่งผลให้หุ่นยนต์ทำงานได้อย่างต่อเนื่อง

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รูปที่ 1.1 แสดงวิธีการโปรแกรมโดยใช้ Teach pendant



รูปที่ 1.2 แสดงการโปรแกรมโดยใช้วิธี Lead through

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ข้อดีของการโปรแกรมหุ่นยนต์แบบ off-line programming

1) หุ่นยนต์สามารถทำงานได้อย่างต่อเนื่องขณะเมื่อทำการโปรแกรม ซึ่งจะไม่ทำให้สูญเสียเวลาในการผลิต

2) สามารถนำไปประยุกต์ใช้กับระบบ Computer-Aided-Design (CAD) โดยจะสามารถศึกษาการจำลองการทำงานของหุ่นยนต์ ก่อนที่จะสร้างเป็นโปรแกรมไปควบคุมหุ่นยนต์ให้ทำงานตามที่ต้องการต่อไป

3) สามารถที่จะเขียนโปรแกรมเพื่อศึกษาการทำงานของหุ่นยนต์ในสถานการณ์ที่มิได้คาดไว้

4) สามารถศึกษาถึงความเหมาะสมของการทำงานของหุ่นยนต์กับงานประเภทต่างๆ ได้อย่างสะดวกและรวดเร็ว เช่น ศึกษาถึงลักษณะที่ต้องการในการเชื่อม เมื่อเปลี่ยนรูปทรง (model) ของรถยนต์ในสายการผลิต

5) สามารถนำไปประยุกต์ใช้กับอุปกรณ์ประเภทอื่นๆ เพื่อเพิ่มความสามารถในการทำงาน เช่น ระบบตรวจจับสัญญาณ (sensoring system)

6) สามารถนำไปใช้ทดลองควบคุมการทำงานเพื่อศึกษาถึงความเป็นไปได้ของโครงงานที่ต้องการ ก่อนจะตัดสินใจนำหุ่นยนต์ดังกล่าวมาใช้งาน

2.1) Robot Programming Language

เป็นวิธีการโปรแกรมทางอ้อม (Indirectly) โดยการใช้ภาษาควบคุม แบบตัวอักษร (textual programming language) และใช้เครื่องคอมพิวเตอร์เป็นตัวควบคุม และสื่อสารกับหุ่นยนต์ เพื่อให้ทำงานตามลักษณะงานที่ต้องการ ตัวอย่างของภาษาที่ใช้ควบคุมหุ่นยนต์ในทางการค้า เช่น VAL II (by Unimation Westinghouse) , AML (IBM Corp) , RAIL (Automatix Inc.) , HELP (General Electric) , Autopass และอื่นๆ

ข้อได้เปรียบของการโปรแกรม โดยใช้ภาษาควบคุม เมื่อเทียบกับวิธี on-line programming คือ

- สามารถใช้ความสามารถของอุปกรณ์ตรวจจับสัณญาณ รวมถึงการใช้สัญญานอินพุต / เอาต์พุต ที่เป็นสัญญาน analog หรือ digital ได้
- ปรับปรุงความสามารถของเอาต์พุตสำหรับการควบคุมอุปกรณ์ภายนอก
- สามารถใช้รูปแบบการควบคุมแบบมีเงื่อนไข (Program Logic Control) เช่น คำสั่ง IF, GOTO เป็นต้น
- การคำนวณ และการประมวลข้อมูลมีความคล้ายคลึงกับ ภาษาคอมพิวเตอร์ทั่วไปทำให้การศึกษาใช้งานทำได้ง่ายขึ้น
- สามารถติดต่อสื่อสารระหว่าง คอมพิวเตอร์อื่นๆ ได้

ในส่วนของการควบคุมนั้น เมื่อมีการโปรแกรมเรียบร้อยแล้ว คำสั่งต่างๆ จะถูกแปลงให้เป็นภาษาเครื่อง (machine language) ซึ่งอยู่ในรูปเลขฐานสองโดยใช้ตัวแปลง compiler หรือ interpreter เพื่อนำไปใช้ในการควบคุมหุ่นยนต์ต่อไป ดังแสดงในรูปที่ 1.3

การโปรแกรมโดยใช้ภาษาควบคุม สามารถแบ่งออกเป็น 2 รูปแบบคือ

2.1.1) Explicit Language (Robot Oriented programing)

เป็นภาษาในการโปรแกรมที่จะต้องกำหนดข้อมูลต่างๆ คือ ตำแหน่งของวัตถุและหุ่นยนต์ รูปแบบการเคลื่อนที่ของปลายจับ และ ข้อมูลจากการตรวจจับสัญญาณ รวมถึงจะต้องคำนึงถึงลำดับการทำงาน (flow of control) และ ส่วนสนับสนุนการโปรแกรมต่างๆ เช่น editor หรือ debugger เป็นต้น

ลักษณะของการโปรแกรมจะแบ่งออกเป็น 2 ส่วน คือ รูปแบบการควบคุมโดยใช้ภาษาโปรแกรม ซึ่งใช้ควบคุมการเคลื่อนที่ต่างๆ และรูปแบบการควบคุมแบบ on-line ซึ่งเป็นวิธีที่สะดวกที่สุดในการกำหนดตำแหน่ง และ ทิศทาง (orientation) ของจุดปลายในขณะ และ/หรือหยุดการเคลื่อนที่

(<u>หมายเหตุ</u> ในส่วนของการกำหนดตำแหน่ง และ ทิศทางนั้น อาจจะใช้วิธีการกำหนดตำแหน่งของวัสดุต่างๆ โดยการใช้แกนอ้างอิงในระบบพิกัดฉาก (coordinate frame) ซึ่งจะแสดงในรูปของเมทริกซ์เปลี่ยนแปลง (transformation matrix) ขนาด 4x4 ของเวคเตอร์ระยะทาง และเมตริกซ์ของทิศทางเทียบกับแกนอ้างอิง)

<u>ตัวอย่างของประโยคในภาษาควบคุม</u> (ภาษา AML)

- MOVE P1 เป็นคำสั่งให้เคลื่อนที่จากตำแหน่งปัจจุบันไปยัง ตำแหน่ง และทิศทาง ซึ่งกำหนดด้วยตัวแปร P1 ซึ่งตัวแปรดังกล่าวจะถูกกำหนดโดยใช้วิธีการควบคุมแบบ on-line ไปยังตำแหน่งดังกล่าว

- DEFINE PATH 123=PATH (P1, P2, P3) เป็นคำสั่งกำหนดทางเดินที่ประกอบด้วยจุด P1, P2, P3

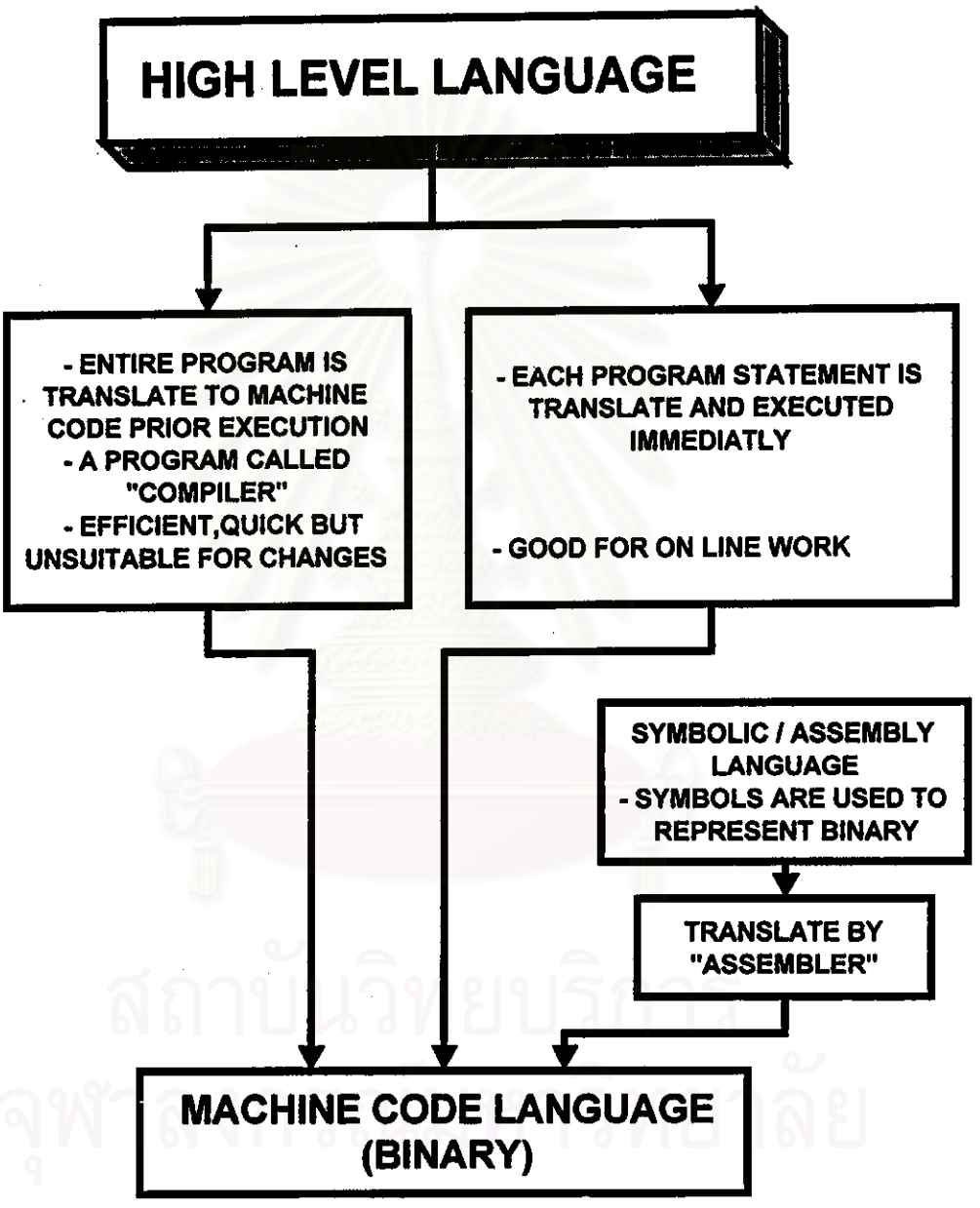
- SPEED 75 เป็นคำสั่งที่ใช้ในการกำหนดความเร็ว โดยมีความเร็วเป็น 75 % ของความเร็วเริ่มต้น

- WAIT 20, ON เป็นคำสั่งให้โปรแกรมหยุดที่ประโยคนี้จนกระทั่งมีสัญญาณอินพุตที่มีเงื่อนไข "ON" เข้ามาที่พอร์ต 20 ของ controller

- SIGNAL 10,ON เป็นคำสั่งที่ใช้สื่อสารกับอุปกรณ์ภายนอกโดยจะมีสัญญาณเอาต์พุตออกที่พอร์ต 10 ของ controller

- GO TO 150 เป็นคำสั่งที่แยกไปทำยังประโยคที่ 150

- IF (logical expression) GO TO 150 เป็นคำสั่งที่มีเงื่อนไข



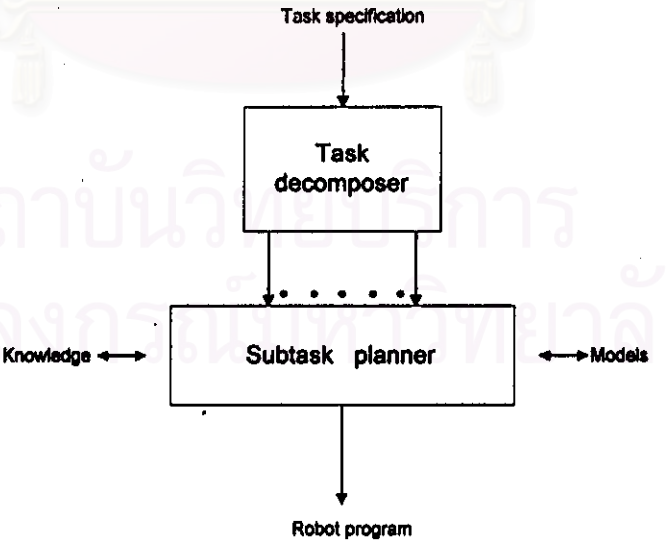
รูปที่ 1.3 แสดงขั้นตอนการโปรแกรมโดยใช้ HIGH LEVEL LANGUAGE

2.1.2) Implicit Language or Task-level programming ("World Model" language)

เป็นภาษาที่ใช้โปรแกรมโดยการกำหนดรายละเอียดของงาน ที่ประกอบเป็นงานที่ต้องการ ตามลำดับ แทนที่จะกำหนดการเคลื่อนที่ของหุ่นยนต์ ดังเช่น Explicit Langauge ทำให้การโปรแกรมโดยวิธีนี้ใช้งานได้ง่ายกว่า แต่จะต้องมีข้อมูลที่เกี่ยวกับสภาพแวดล้อมการทำงานของหุ่นยนต์ และลักษณะทางกายภาพของหุ่นยนต์ เก็บไว้ในหน่วยความจำมากขึ้น รวมถึงอาจจะต้องมีการนำระบบตรวจจับสัญญาณ มาประยุกต์ใช้งานด้วย ข้อมูลต่างๆ ที่ใช้ในการโปรแกรมจะแบ่งออกเป็น 3 ส่วนคือ ลักษณะงาน (task specification) , ข้อมูลกลาง (world models) และการแปลงให้เป็นภาษาของหุ่นยนต์ (robot program synthesis)

พื้นฐานของการโปรแกรมประเภทนี้นั้น จะมีความคล้ายคลึงกับการโปรแกรมโดยใช้ปัญญาประดิษฐ์ (artificial intelligence) กล่าวคือผู้ออกแบบจะต้องมีข้อมูล อินพุต / เอาต์พุตในการออกแบบลักษณะงาน และเมื่อออกแบบเสร็จตัวสร้างโปรแกรม (program generator) จะสร้างโปรแกรมซึ่งตรงกับข้อมูลดังกล่าวเพื่อนำมาใช้งานต่อไป

รูปที่ 1.4 แสดงลักษณะที่เป็นไปได้ในการโปรแกรม โดยลักษณะงานจะถูกแยกออกเป็นลำดับของลักษณะงานย่อย โดย ตัวแยกลักษณะงาน (task decomposer) และข้อมูลต่างๆ เช่น ตำแหน่งเริ่มต้น และสุดท้าย (Initial / final state) , ตำแหน่งการหยิบ (grasping position) , ค่าตัวดำเนินการ (operand) จะถูกขยายออกมา ก่อนที่จะส่งค่าดังกล่าวไปยังตัววางแผนโปรแกรมย่อย (subtask planner) เพื่อสร้างโปรแกรมของหุ่นยนต์ที่ต้องการต่อไป



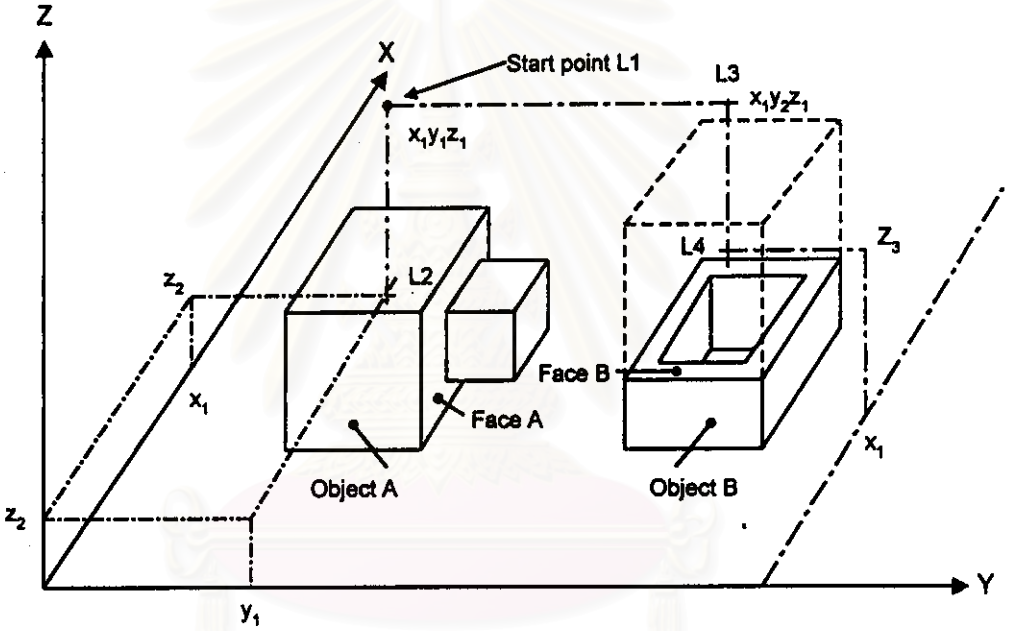
รูปที่ 1.4 แสดงตัวอย่างลักษณะที่เป็นไปได้ในการโปรแกรมแบบ Implicit Language

หมายเหตุ World Modelling เป็นการอธิบายถึงคุณสมบัติทางรูปทรง และกายภาพของวัตถุรวมถึงหุ่นยนต์ และแสดงถึง สภาวะของการประกอบวัตถุต่างๆ ในขอบเขตการทำงาน

1) คุณสมบัติทางรูปทรงและกายภาพของวัตถุสามารถแสดง โดย กลุ่มของมุมและจุด กลุ่มของพื้นผิว การสร้างเป็นรูปทรงกระบอก และการแบ่งเป็นเซลย่อย

2) สภาวะของวัตถุ จะแสดงถึงรูปแบบของวัตถุ เช่น attachment ซึ่งจะบอกว่าวัตถุแข็ง (rigid) หรือไม่ หรือค่าบังคับ (constraints) ซึ่งจะบอกว่าการเคลื่อนที่ของวัตถุเป็นเชิงเส้น (translation) หรือการหมุน (rotation)

รูปที่ 1.5 แสดงลักษณะความแตกต่างระหว่างการโปรแกรมโดยใช้ Explicit และ Implicit Language ส่วนรูปที่ 1.6 จะแสดงถึงระดับของวิธีการโปรแกรมหุ่นยนต์ด้วยวิธีต่างๆ



รูปที่ 1.5 แสดงความแตกต่างระหว่าง Explicit Language และ Implicit Language ในการยกวัตถุ A , หมุน และเสียบลงในวัตถุ B

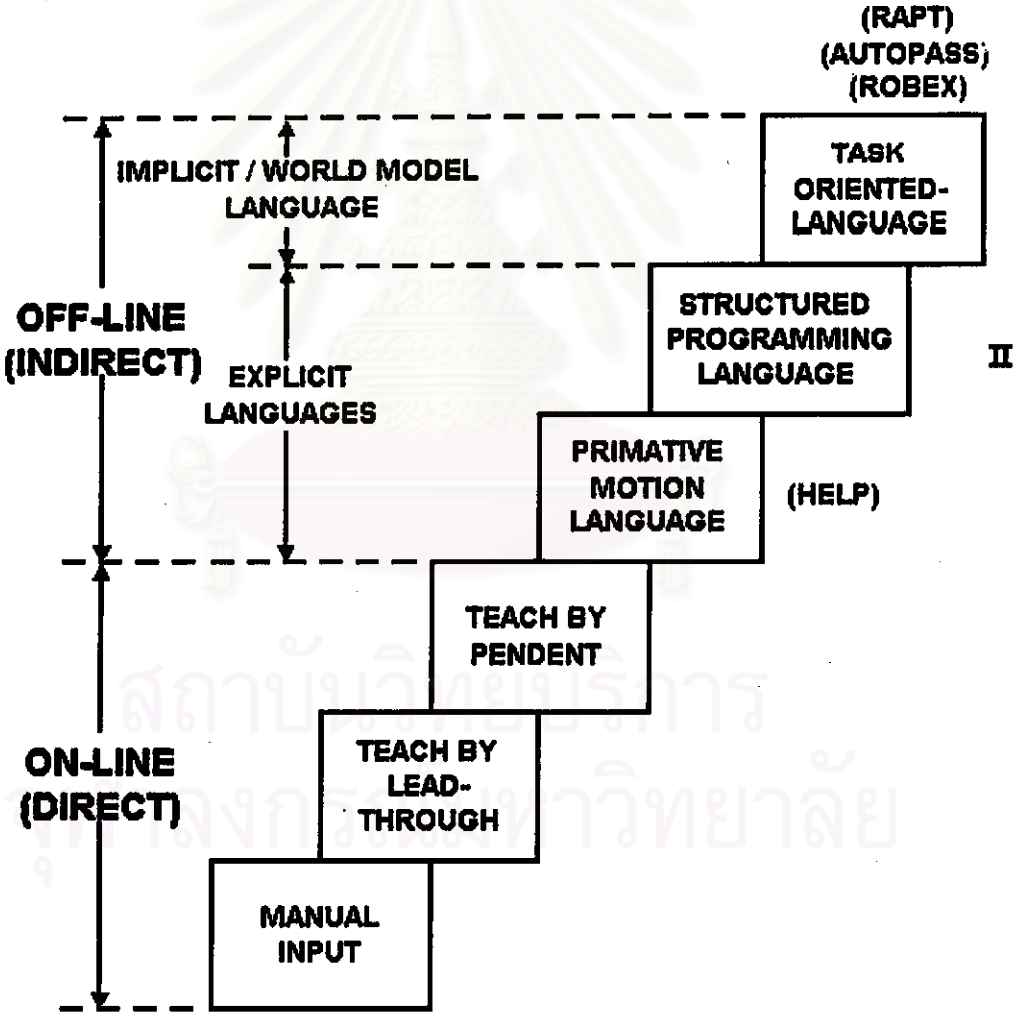
Explicit Language	Implicit Language
Definition of locations	World Model is defined
$L1= X_1Y_1Z_1\alpha_1\beta_1\gamma_1$	This includes description of object and
$L2= X_2Y_2Z_2\alpha_2\beta_2\gamma_2$	environment geometry,gripping points,etc.
$L3= X_3Y_3Z_3\alpha_3\beta_3\gamma_3$	
$L4= X_4Y_4Z_4\alpha_4\beta_4\gamma_4$	

Program

OPEN GRIPPER
FROM L1
GOTO L2
CLOSE GRIPPER
GOTO L1
GOTO L3
GOTO L4
OPEN GRIPPER

Program

FROM START POINT
MOVE OBJECT 1 ONTO OBJECT 2
WITH FACE A AGAINST FACE B



รูปที่ 1.6 แสดงระดับของวิธีการโปรแกรมหุ่นยนต์ประเภทต่าง ๆ

2.2) Simulation and Off-line programming (Off-line programming with a CAD System)

ในปัจจุบันได้มีการพัฒนาระบบ CAD (Computer-Aided-Design) ซึ่งเป็นการนำเอาคอมพิวเตอร์มาช่วยในการออกแบบผลิตภัณฑ์มาประยุกต์ใช้กับการควบคุมหุ่นยนต์ภายในอุตสาหกรรมต่างๆ กล่าวคือจะสามารถจำลองและวิเคราะห์การทำงานในลักษณะต่างๆ ของหุ่นยนต์บนคอมพิวเตอร์ ก่อนที่จะนำข้อมูลที่ได้ไปใช้ในการออกแบบ และแปลงไปเป็นภาษาที่ใช้ควบคุมหุ่นยนต์ต่อไป ดังแสดงในรูปที่ 1.7

ตัวอย่างของโปรแกรมการจำลองที่ใช้ระบบ CAD ในการควบคุมหุ่นยนต์แบบ off-line programming ก็เช่น PLACE (McDonnell Douglas Manufacturing Industry Systems Company) , Robographix (Computervision Corporation), Robot-SIM (General Electric Calma Company) และ ROBO-CAM (Silma, Inc) เป็นต้น

ประโยชน์ที่ได้จากการทำการจำลองการทำงานของหุ่นยนต์ (ROBOT SIMULATION)

1) นำมาใช้เพื่อตรวจสอบการชนและหลีกเลี่ยงการชน (collision detection and avoidance) ระหว่างหุ่นยนต์หลายๆตัว และวัตถุอื่นในพื้นที่การทำงาน (workcell)

2) นำมาใช้เพื่อดูค่าปัจจัย (factor) ต่างๆที่เกี่ยวข้องกับการเคลื่อนที่ของหุ่นยนต์ เช่น ความเร่ง ความหน่วง น้ำหนักของแขนหุ่นยนต์ รวมถึงน้ำหนักสูงสุดซึ่งหุ่นยนต์จะสามารถรับและทำงานได้

3) นำมาใช้เพื่อศึกษา และปรับเปลี่ยนรูปแบบการทำงานของหุ่นยนต์ เพื่อให้เกิดประสิทธิภาพสูงสุดในงานแต่ละประเภท เช่น ศึกษาตำแหน่งที่ไปได้ไกลที่สุดของแขนของหุ่นยนต์ เป็นต้น

4) นำมาใช้เพื่อพิจารณา การเลือกซื้อหุ่นยนต์ที่เหมาะสมกับงานประเภทนั้นๆ โดยพิจารณาถึงปัจจัยที่สำคัญต่างๆ เช่น ขอบเขตการทำงาน (work envelope) , ช่วงเวลาที่ใช้ในการเคลื่อนที่ในแต่ละรอบ (estimated cycle time) หรือ ขอบเขตการเคลื่อนที่จำกัดของแต่ละส่วนของหุ่นยนต์ เป็นต้น

5) นำมาใช้เพื่อประโยชน์ในการศึกษา และอบรมซึ่งจะทำให้สามารถทดลองและควบคุมหุ่นยนต์ได้โดยปราศจากอันตรายที่อาจจะเกิดขึ้นในตามความเป็นจริง

6) สามารถใช้ประโยชน์จากฐานข้อมูลในระบบ CAD/CAM ของโรงงาน เช่น ถ้ามีการเก็บข้อมูลของตัวถังรถยนต์ ภายในคอมพิวเตอร์ที่ใช้ในระบบ CAD/CAM ผู้ออกแบบสามารถกำหนดจุดที่จะใช้ในการทำ spot-welding ได้โดยส่งข้อมูลผ่านระบบเครือข่าย (network)

รูปที่ 1.7 แสดงลักษณะการทำ Simulation ของ Robot Workcell ในการทำงานเชื่อม

ลักษณะสำคัญของโปรแกรม Robot Module ในซอฟท์แวร์ CATIA version4

CATIA Robotic Module เป็นซอฟท์แวร์ที่ใช้ประโยชน์ในการออกแบบและการโปรแกรมแบบ off-line programming ของลักษณะงานของหุ่นยนต์ (robotics workcell) โดยสามารถใช้ประโยนช์จากฐานข้อมูลและการออกแบบในระบบ 3 มิติของซอฟท์แวร์ CATIA ได้

ขั้นตอนของการทำงานจะเริ่มจากการกำหนด และเลือกส่วนประกอบต่างๆ ของขอบเขตของงาน คือ หุ่นยนต์ ตัวควบคุม และการกำหนดงาน (task) หนึ่งๆ หรือหลายๆ งาน ซึ่งมีจุดประสงค์ในการศึกษาถึงการกระทำระหว่างหุ่นยนต์และสภาพแวดล้อมภายนอก หลังจากนั้นลักษณะงานงานดังกล่าวจะถูกจำลองการทำงาน และวิเคราะห์เพื่อทำให้เกิดประโยชน์สูงสุด (optimization) ในงานนั้นๆ

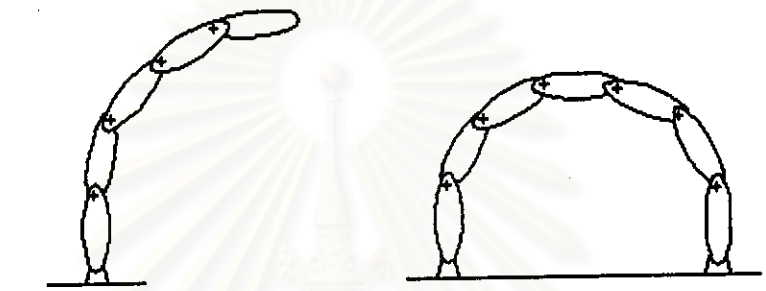
ผลที่ได้เมื่อทำการจำลองการทำงานแล้วนั้น งานดังกล่าวจะถูกแปลงออกมาในรูปไฟล์ข้อมูล (output file) ด้วยยูทิลิตี้ของซอฟท์แวร์ CATIA ซึ่งจะนำไปใช้งานได้โดยการเขียนโพสต์โปรเซสเซอร์ (post-processor) เพื่อแปลงไฟล์ดังกล่าวให้อยู่ในรูปแบบของภาษาของตัวควบคุมที่ควบคุมหุ่นยนต์นั้นๆ

ขั้นตอนสุดท้ายคือการส่งไฟล์ที่ได้ดังกล่าวผ่านพอร์ต RS-232 ซึ่งเป็นพอร์ตอนุกรมบนเครื่องคอมพิวเตอร์ไปควบคุมหุ่นยนต์เพื่อให้ทำงานตามลักษณะงานที่กำหนดไว้ดังกล่าว

ข้อจำกัดของโปรแกรม Robot Module

ถึงแม้ว่าซอฟท์แวร์ Robot Module จะสามารถนำมาใช้งานได้ดังกล่าว แต่ก็มีข้อจำกัดในการใช้งานที่ต้องทราบไว้ ดังต่อไปนี้

1) ในการพิจารณาหุ่นยนต์ที่จะนำมาใช้กับ Robot Module นั้น หุ่นยนต์ดังกล่าวจะต้องมีโครงสร้างแบบโซ่เปิด (open chain) ซึ่งจะสามารถหาค่า Trasformation Matrix ซึ่งเป็นเมทริกซ์ ที่ใช้บอกตำแหน่ง และ orientation เทียบกับแกนอ้างอิงได้ดังแสดงในรูปที่ 1.8



รูปที่ 1.8 โครงสร้างของหุ่นยนต์ (structure of a robot) ที่ยอมรับ

2) ลักษณะการต่อเชื่อมระหว่างหุ่นยนต์และส่วนควบคุม (Link between Robots and Controllers)

กฎข้อที่ 1 : หุ่นยนต์ตัวหนึ่งๆ จะไม่สามารถต่อเชื่อมกับตัวควบคุมมากกว่าหนึ่งตัวได้

กฎข้อที่ 2 : หุ่นยนต์จะต้องต่อเชื่อมกับตัวควบคุม หุ่นยนต์จะไม่สามารถเคลื่อนที่ได้ถ้าไม่ต่อเชื่อมกับตัวควบคุม

กฎข้อที่ 3 : หุ่นยนต์หลายตัวสามารถที่จะต่อเชื่อมกับตัวควบคุมตัวหนึ่งๆ ได้

กฎข้อที่ 4 : งาน (Task) หนึ่งๆ จะสามารถกำหนดตัวควบคุมได้เพียงหนึ่งตัวควบคุม

กฎข้อที่ 5 : งาน (Task) หลายงานจะไม่สามารถทำได้พร้อมกัน ถ้างานดังกล่าวอ้างอิงถึงตัวควบคุมตัวเดียวกัน

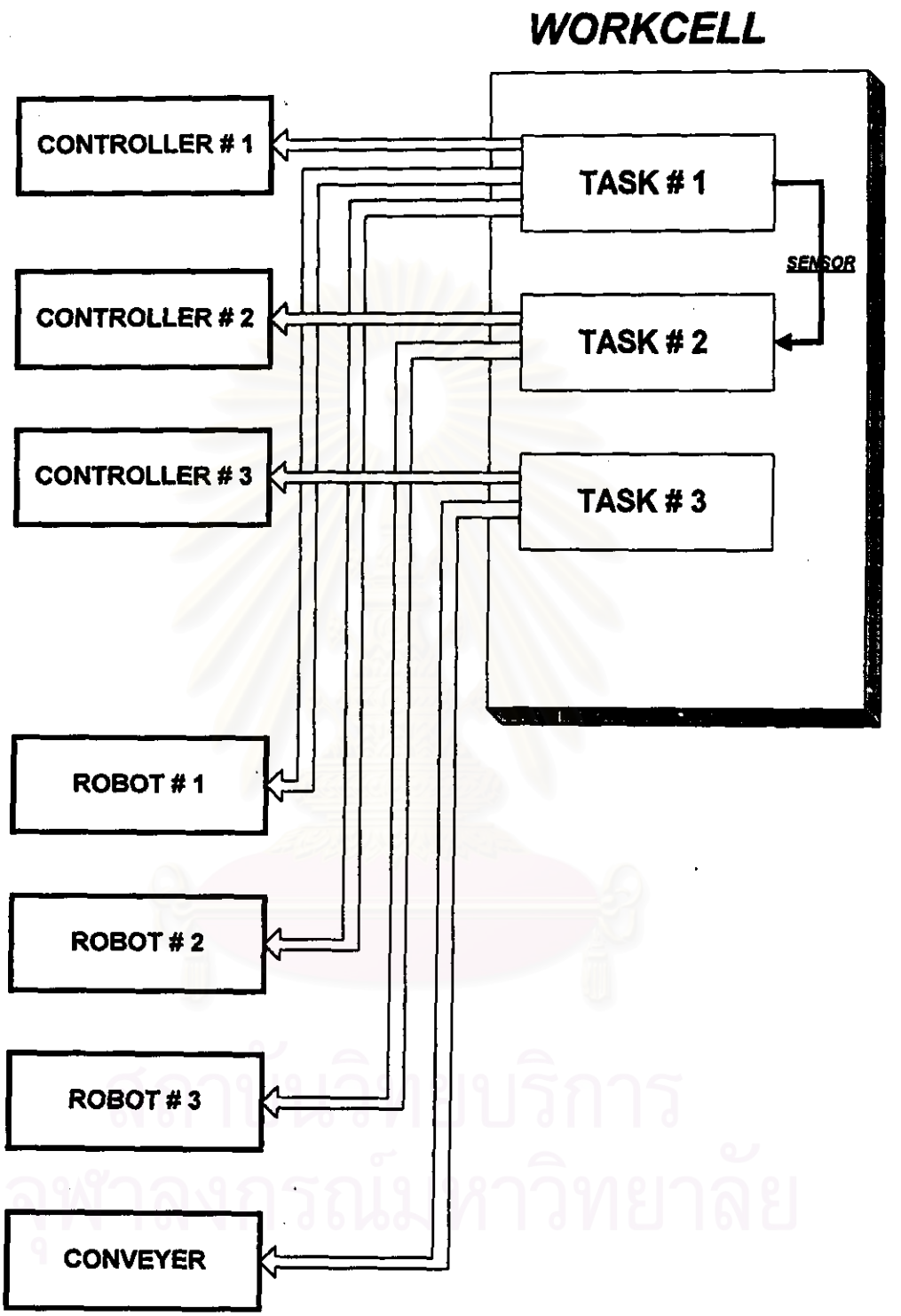
รูปที่ 1.9 แสดงถึงลักษณะการเชื่อมต่อระหว่างหุ่นยนต์ และ ส่วนควบคุมภายในซอฟท์แวร์ โดยแสดงให้เห็นถึงในลักษณะการทำงานหนึ่งๆนั้น จะประกอบไปด้วยลักษณะงานย่อยๆที่ทำงานสัมพันธ์กันโดยใช้การส่งข้อมูลของอุปกรณ์ตรวจจับสัญญาณ ในลักษณะงานหนึ่งๆนั้นอาจจะประกอบไปด้วยการทำงานของหุ่นยนต์หนึ่งหรือมากกว่าหนึ่งตัว แต่ลักษณะงานแต่ละส่วนดังกล่าว จะต้องถูกควบคุมด้วยตัวควบคุมเพียงตัวเดียวเท่านั้น

รูปที่ 1.10 แสดงขั้นตอนการทำ off-line programming ซึ่งขั้นตอนการทำงานจะเริ่มจากการจำลองการทำงานของหุ่นยนต์บนซอฟท์แวร์ CATIA ROBOT MODULE โดยจะทำการวิเคราะห์

และวางแผนการทำงานตามลักษณะงานที่ได้กำหนด จากนั้นจึงทำการแปลงให้เป็นไฟล์เอาต์พุต และใช้โพสต์โปรเซสเซอร์ที่พัฒนาขึ้นแปลงไฟล์เอาต์พุตดังกล่าวไปเป็นคำสั่งที่ใช้ในการควบคุมหุ่นยนต์ ก่อนที่จะส่งไปควบคุมหุ่นยนต์จริงโดยผ่านการสื่อสารแบบอนุกรมต่อไป

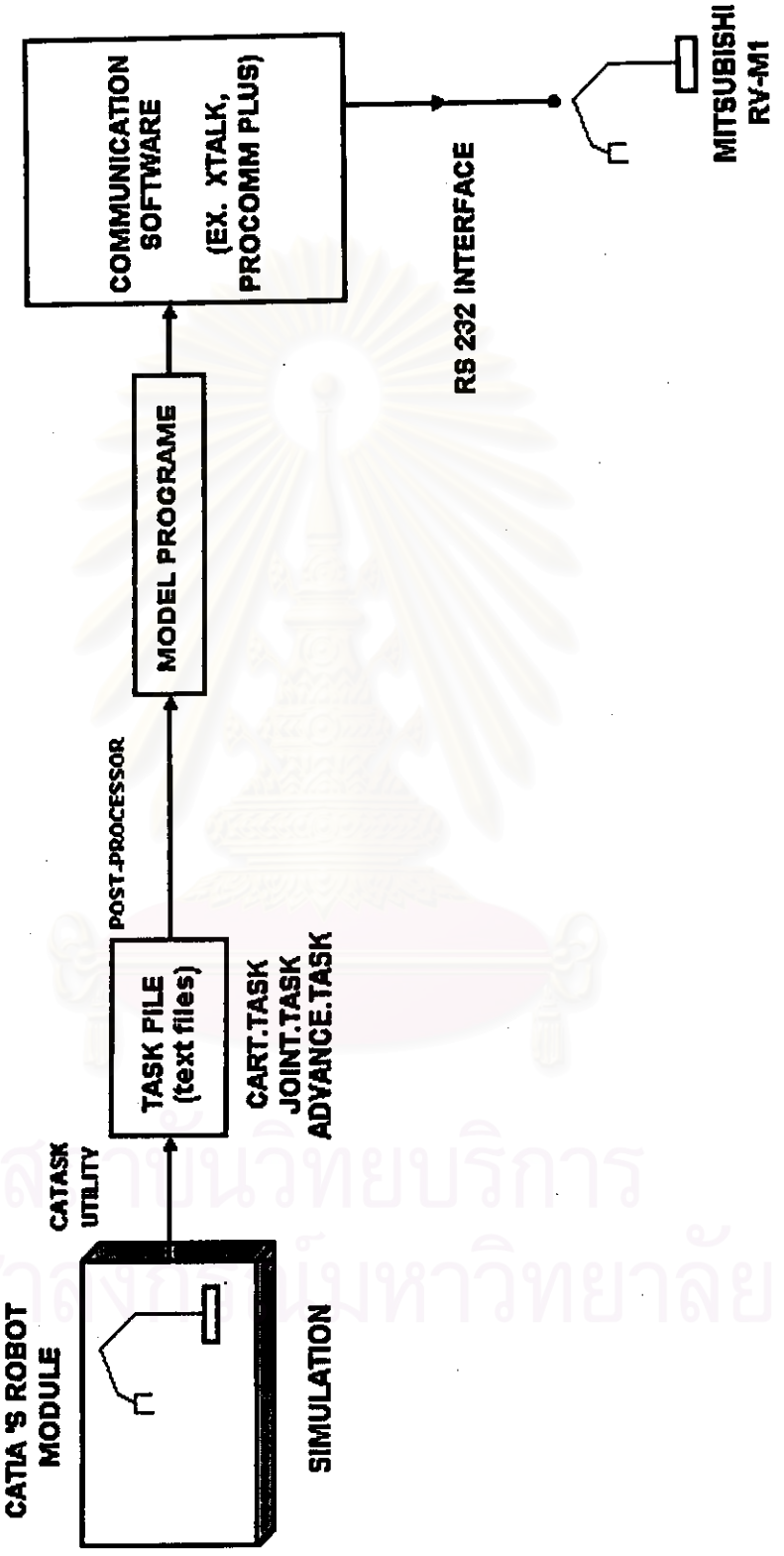
รูปที่ 1.11 แสดงถึงขั้นตอนการส่งข้อมูลผ่าน RS-232 และ โปรฟิบัสโดยใช้มาตรฐานการสื่อสารแบบ Ethernet ซึ่งแสดงให้เห็นถึงการส่งข้อมูลในลักษณะต่างๆ โดยในส่วนของการสื่อสารผ่าน Ethernet นั้นจะใช้สำหรับส่งข้อมูลเอาต์พุตจากการจำลองการทำงานไปยังคอมพิวเตอร์ที่ใช้ในการส่งโปรแกรม ส่วนการสื่อสารผ่าน RS-232 นั้นจะใช้ในการส่งภาษาควบคุมที่ได้จากการแปลงแล้วไปยังตัวควบคุมของหุ่นยนต์เพื่อทำงานจริง และในส่วนที่เป็นการสื่อสารภายในระบบผลิตอัตโนมัติที่ใช้ในการวิจัยนี้จะใช้การสื่อสารข้อมูลผ่านระบบ Profibus ซึ่งจะได้กล่าวรายละเอียดต่อไป

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

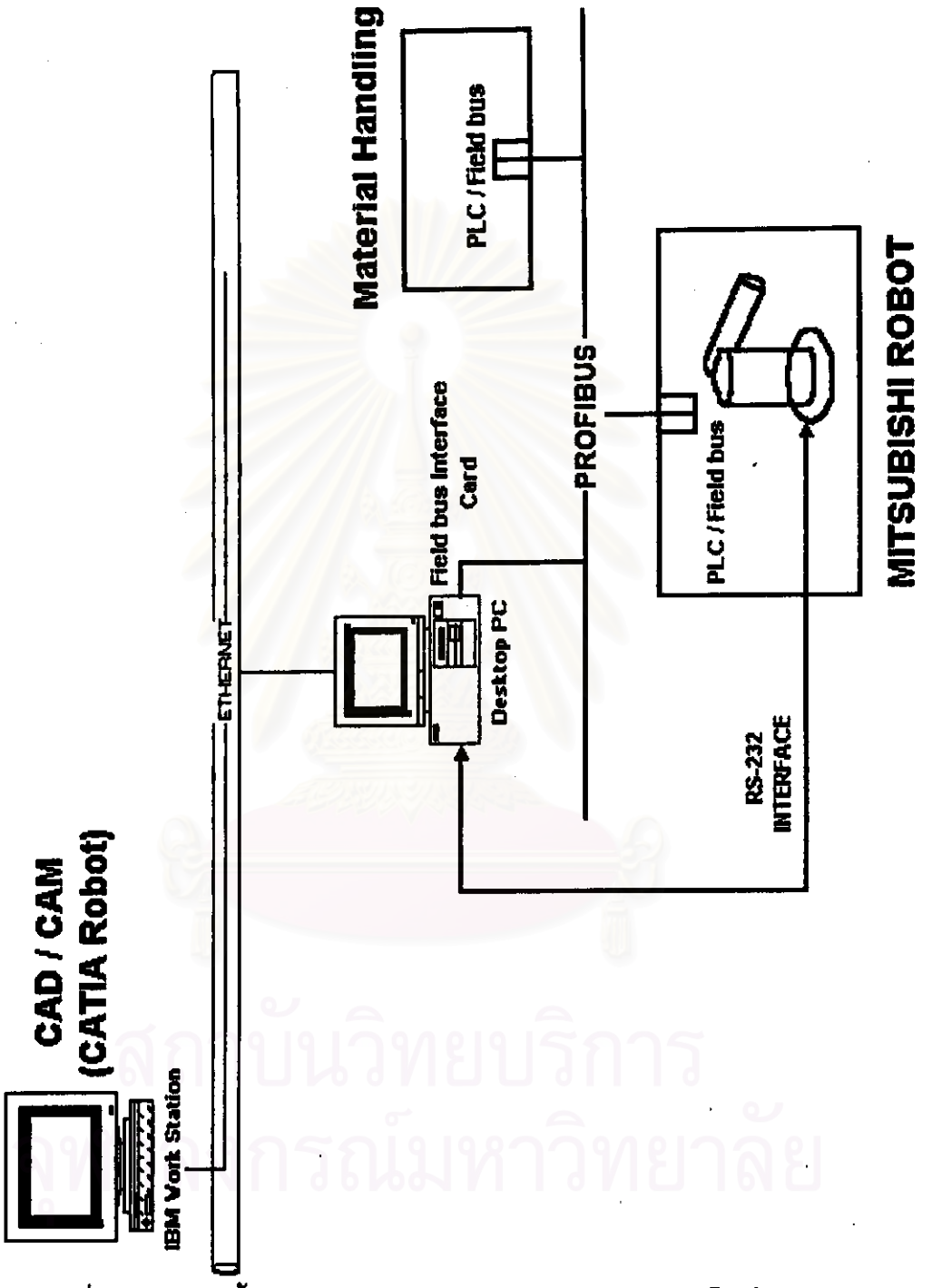


รูปที่ 1.9 แสดงลักษณะการต่อเชื่อมระหว่างหุ่นยนต์และส่วนควบคุม

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 1.10 แสดงขั้นตอนการทำ off-line programming



รูปที่ 1.11 แสดงขั้นตอนการส่งข้อมูลผ่าน RS-232 , Profibus โดยใช้ Ethernet