บทที่ 5
การควบคุม

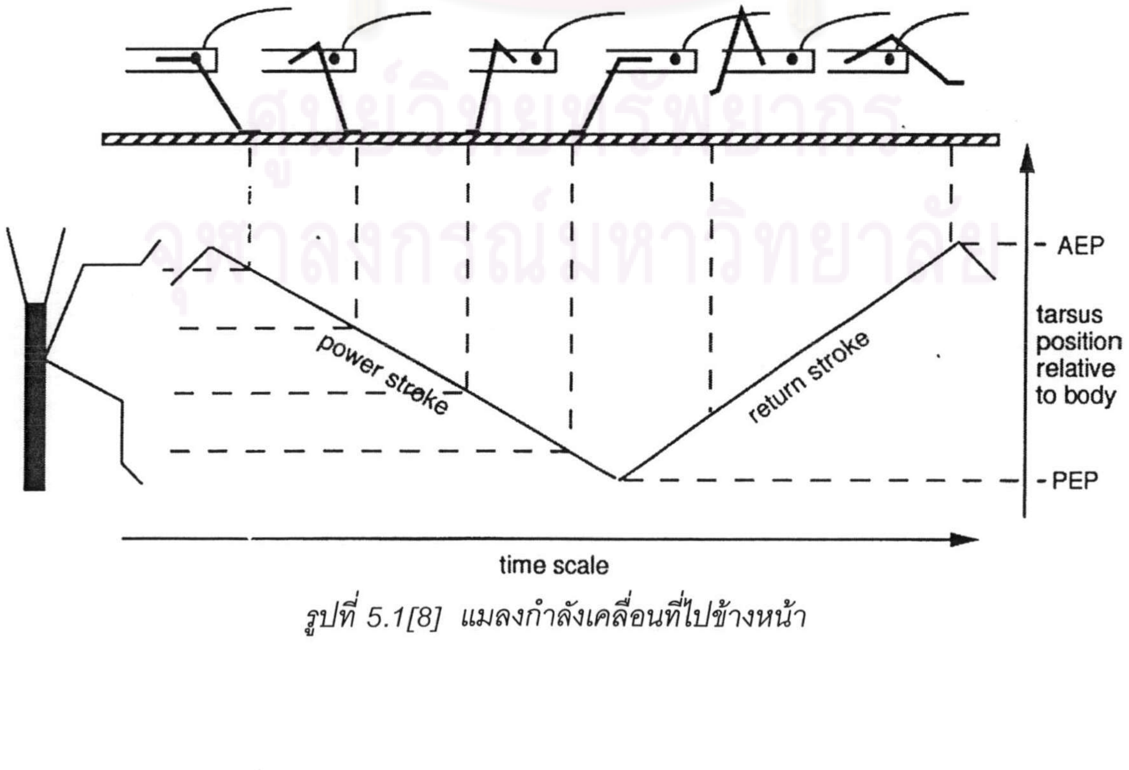
ในหัวข้อที่ 3.4 ได้กล่าวถึงรูปแบบการควบคุมแบบขนาน โดยตัวประมวลผลหลักจะทำหน้าที่ ประมวลผลระดับบนคือ กำหนดรูปแบบการเดิน (Gait) และตัวประมวลผลของแต่ละขาจะทำหน้าที่ ประมวลผลระดับล่าง คือ การควบคุมปลายขาของหุ่นยนต์ให้เคลื่อนที่ไปยังตำแหน่งที่กำหนด

ในบทนี้จะกล่าวถึง 3 ส่วนใหญ่ ๆ คือ

1. รูปแบบการเดินของหุ่นยนต์ที่จะศึกษา
2. โปรแกรมควบคุมท่าเดิน
3. โปรแกรมช่วยเหลือที่ใช้ในการแก้ไขรูปแบบการเดินของหุ่นยนต์

5.1 รูปแบบการเดินของหุ่นยนต์ที่จะศึกษา

ในกรณีของหุ่นยนต์หกขา มีผู้สนใจที่จะนำหลักการเดินของแมลงมาสร้างระบบควบคุมเพื่อให้หุ่นยนต์มีลักษณะการเดินที่คล้ายแมลง เพราะแมลงเป็นสัตว์ชนิดหนึ่งที่มีการเดินที่เสถียรภาพ



รูปที่ 5.1[8] แมลงกำลังเคลื่อนที่ไปข้างหน้า

ศูนย์วิทยทรัพยากร จุฬาลงกรณ์มหาวิทยาลัย

จากรูปที่ 5.1 แสดงถึงการเดินของแมลงโดยในการเดิน 1 ก้าว สามารถแบ่งได้เป็น 2 ช่วง คือ

Power Stroke คือ ช่วงที่เท้าของแมลงสัมผัสกับพื้นรับน้ำหนักตัวแมลง และเป็นช่วงที่แมลงออกแรงพยายามที่จะผลักตัวไปข้างหน้า

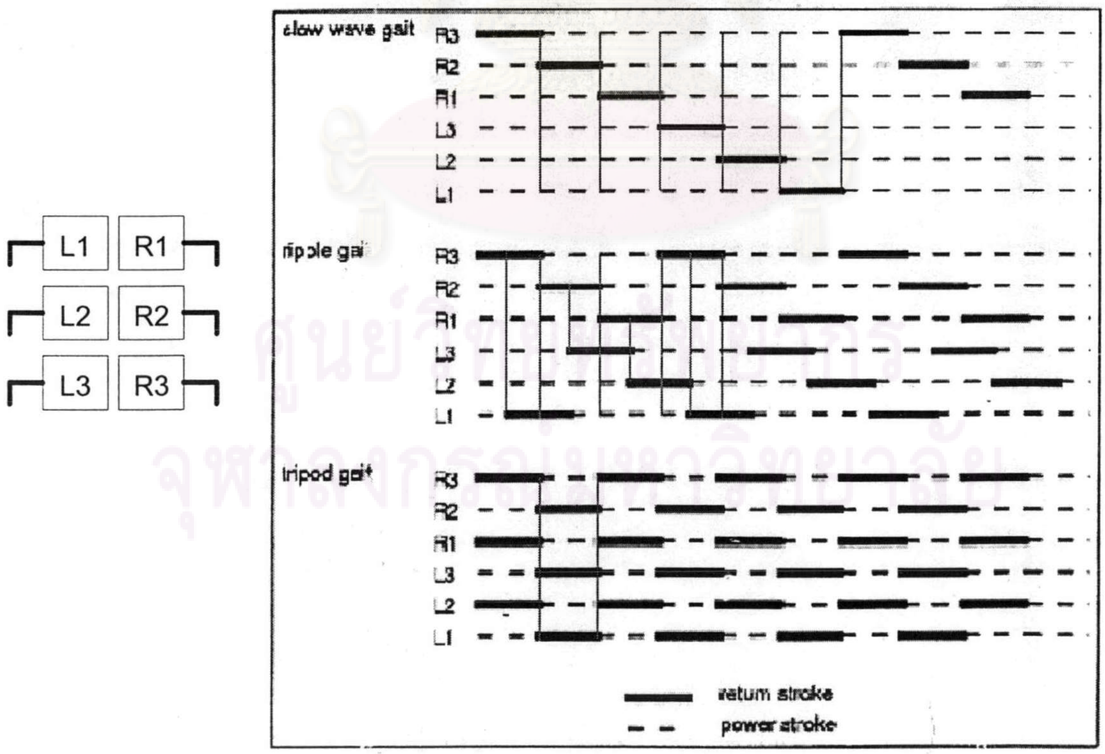
Return Stroke คือ ช่วงแมลงผลักตัวเองไปแล้ว และเท้าของแมลงยกขึ้นจากพื้น และพยายามเคลื่อนที่ไปสู่ตำแหน่งของการเปลี่ยนเป็นช่วง Power Stroke

โดย

ตำแหน่งที่เท้าของแมลงกำลังจะเปลี่ยนจาก Return Stroke ไปเป็น Power Stroke เรียกว่า **AEP (Anterior Extreme Position)**

ตำแหน่งที่เท้าของแมลงที่กำลังจะเปลี่ยนจาก Power Stroke ไปเป็น Return Stroke เรียกว่า **PEP (Posterior Extreme Position)**

วิลสัน (Wilson) [8] เป็นนักกีฏวิทยาชาวอเมริกัน ในปี 1966 ได้เสนอแบบจำลองซึ่งอธิบายท่าเดิน (Gaits) ที่สังเกตได้ทั่วไปทั้งหมดของแมลงสาบ (Cockroaches) ดังรูปที่ 5.2 แบบจำลองนี้เป็นการศึกษาลักษณะท่าเดินของแมลงสาบในขณะที่มันเดินบนพื้นราบเรียบ แต่ถึงอย่างไรก็ตามแบบจำลองนี้ไม่ได้อธิบายถึงข้อมูลจากการทดลองทั้งหมดจริงๆ มันเป็นลักษณะส่วนใหญ่ที่สังเกตได้



รูปที่ 5.2[8] โมเดลการเดินของแมลง ที่สังเกตได้โดยทั่วไป (Wilson 1966)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

จากรูปที่ 5.2 สามารถอธิบายแบบจำลองการเดินทั้งสามท่าดังนี้

ท่าเดินแบบลูกคลื่นต่ำ (Slow wave gait) เป็นท่าเดินของแมลงที่มีขาหนึ่งขาเป็นอิสระและอีกห้าขาที่เหลือวางอยู่บนพื้น

ในการเริ่มเดินจะก้าวขา R3 ขึ้นก่อน โดยขาที่เหลือวางอยู่บนพื้น

ในขณะที่ขา R3 สัมผัสพื้น ขา R2 ก็จะก้าวขึ้นทันที โดยขาที่เหลือวางอยู่บนพื้น

ในขณะที่ขา R2 สัมผัสพื้น ขา R1 จะก้าวขึ้นทันที โดยขาที่เหลือวางอยู่บนพื้น

ในขณะที่ขา R1 สัมผัสพื้น ขา L3 จะก้าวขึ้นทันที โดยขาที่เหลือวางอยู่บนพื้น

ในขณะที่ขา L3 สัมผัสพื้น ขา L2 จะก้าวขึ้นทันที โดยขาที่เหลือวางอยู่บนพื้น

ในขณะที่ขา L2 สัมผัสพื้น ขา L1 จะก้าวขึ้นทันที โดยขาที่เหลือวางอยู่บนพื้น

เมื่อครบวงจรของการเคลื่อนที่ ก็จะกลับมาเริ่มต้นที่เดิมอีกครั้ง

ท่าเดินแบบกระเพื่อม (Ripple gait) เป็นท่าเดินของแมลงที่มีขา 2 ขาเป็นอิสระ และอีก 4 ขาที่เหลือสัมผัสอยู่บนพื้น

ในการเริ่มเดินจะก้าวขา R3 ขึ้นก่อน โดยขาที่เหลือวางอยู่บนพื้น

ในขณะที่ขา R3 ยกขึ้นมาครึ่งก้าว ขา L1 ก็ยกขึ้นทันที โดยขาที่เหลือวางอยู่บนพื้น

ในขณะที่ขา R3 วางถึงพื้น ขา R2 ก็ยกขึ้นทันที ในขณะที่ขา L1 ก้าวไปได้ครึ่งก้าว

ในขณะที่ขา L1 วางถึงพื้น ขา L3 ก็ยกขึ้นทันที ในขณะที่ขา R2 ก้าวไปได้ครึ่งก้าว

ในขณะที่ขา R2 วางถึงพื้น ขา R1 ก็ยกขึ้นทันที ในขณะที่ขา L3 ก้าวไปได้ครึ่งก้าว

ในขณะที่ขา L3 วางถึงพื้น ขา L1 ก็ยกขึ้นทันที ในขณะที่ขา R1 ก้าวไปได้ครึ่งก้าว

ในขณะที่ขา R1 วางถึงพื้น ขา R3 ก็กลับมาเริ่มต้นใหม่อีกครั้ง

ท่าเดินแบบไทรพอด (Tripod gait) เป็นท่าเดินของแมลงที่มีขา 3 ขาเป็นอิสระ และอีก 3 ขาที่เหลือวางอยู่บนพื้น

ในการเริ่มเดินจะก้าวขา R3,R1 และ L2 ขึ้นพร้อมกัน โดยขาที่เหลือวางอยู่บนพื้น

ในขณะที่ขา R3, R1 และ L2 สัมผัสพื้นขา R2, L3 และ L1 ก็ก้าวขึ้นมาทันที จากนั้นเมื่อครบวงจรของการเคลื่อนที่ ก็จะกลับมาเริ่มต้นที่เดิมอีกครั้ง

5.2 โปรแกรมควบคุมท่าเดิน

สามารถแบ่งได้เป็น 3 ส่วนใหญ่ๆ ตามรูปแบบการเดินที่จะศึกษาคือ

- โปรแกรมท่าเดินแบบลูกคลื่นต่ำ (Slow wave gait)
- โปรแกรมท่าเดินแบบกระเพื่อม (Ripple gait)
- โปรแกรมท่าเดินแบบไทรพอด (Tripod gait)

เพื่อความสะดวกต่อการเข้าใจจะขออธิบาย โปรแกรมท่าเดินแบบไทรพอด (Tripod gait) ท่าเดินแบบลูกคลื่นต่ำ (Slow wave gait)และท่าเดินแบบกระเพื่อม (Ripple gait) ตามลำดับ

5.2.1 โปรแกรมท่าเดินแบบไทรพอด (Tripod gait)

แบ่งเป็น 2 ส่วนคือ โปรแกรมของตัวประมวลผลของแต่ละขาของหุ่นยนต์ และโปรแกรมของตัวประมวลผลกลาง

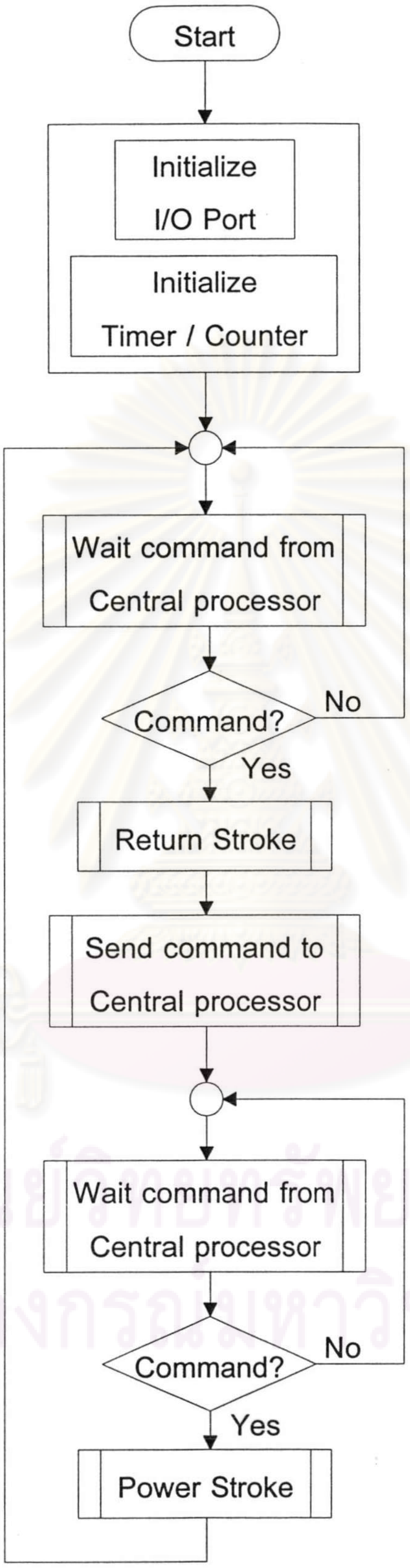
5.2.1.2 โปรแกรมของตัวประมวลผลของแต่ละขาของท่าเดินแบบไทรพอด (Tripod gait)

ตัวประมวลผลที่ใช้เป็นตัวประมวลผลตระกูล 8051 หมายเลข 89C1051 มีพอร์ต Input/Output ขนาด 8 บิต จำนวน 2 พอร์ท (P1 และ P3) วงจรที่ออกแบบไว้จะใช้พอร์ท 1.6 ในการรับคำสั่งจากตัวประมวลผลกลาง พอร์ท 3.7 ในการส่งคำสั่งกลับไปยังตัวประมวลผลกลาง และ พอร์ท 1.3 ในการรับค่าจากอุปกรณ์ตรวจรู้ที่ปลายขาหุ่นยนต์

เมื่อเริ่มโปรแกรมจะต้องกำหนดค่าเริ่มต้นก่อน การที่จะใช้พอร์ตใดเป็น Input หรือ Output ถูกกำหนดโดยโปรแกรมส่วนค่าเริ่มต้น (Initialize) โปรแกรมส่วนนี้ยังกำหนดให้ตัวประมวลผลสามารถใช้ วงจรนับ/จับเวลาภายในได้ ซึ่งถูกนำไปใช้ในการสร้างพัลซ์ให้กับอาร์ซีเซอร์โวมอเตอร์

หลังจากได้กำหนดแล้วว่า พอร์ตใดเป็น input หรือ output จะมีโปรแกรมส่วนติดต่อ Hardware มารองรับการรับและส่งค่าไปมาระหว่างตัวประมวลผลหลักและตัวประมวลผลของแต่ละขา โดยโปรแกรมส่วนดังกล่าวนี้ถูกเขียนเป็นโปรแกรมย่อย (Subroutine) ซึ่งจะถูกเรียกใช้โดยวงวนการทำงานหลัก (Main loop) ต่อไป

ในส่วนวงวนการทำงานหลัก จะมีวงวนการทำงานย่อยไว้รับคำสั่งที่ตัวประมวลผลหลักส่งมา 2 วงวนการทำงาน วงวนการทำงานย่อยแรกมีไว้สำหรับรับคำสั่งจากตัวประมวลผลหลักเพื่อเข้าสู่ช่วง Return stroke และ วงวนการทำงานย่อยที่สองมีไว้สำหรับรับคำสั่งจากตัวประมวลผลหลักเพื่อเข้าสู่ช่วง Power stroke ส่วนการส่งคำสั่งกลับไปยังตัวประมวลผลหลักสำหรับโปรแกรมท่าเดินแบบไทรพอด (Tripod gait) จะส่งกลับไปเมื่อขาของหุ่นยนต์เคลื่อนที่ผ่านช่วง Return stroke ไปแล้ว ดังรูปที่ 5.3



รูปที่ 5.3 แผนผังการไหลแสดงการทำงานของตัวประมวลผลของแต่ละขาสำหรับโปรแกรมท่าเดินแบบไทรพอด (Tripod gait)

5.2.1.2 โปรแกรมของตัวประมวลผลกลางของท่าเดินแบบไทรพอด (Tripod gait)

ตัวประมวลผลที่ใช้เป็นตัวประมวลผลตระกูล 8051 หมายเลข 89C51 มีพอร์ต Input/Output ขนาด 8 บิต จำนวน 4 พอร์ท (P0 – P4) วงจรที่ออกแบบไว้จะใช้พอร์ท 1.0 ส่งคำสั่งไปยังตัวประมวลผลของขา R1 ใช้พอร์ท 1.1 ในการรับคำสั่งจากตัวประมวลผลของขา R1

ใช้พอร์ท 1.2 ส่งคำสั่งไปยังตัวประมวลผลของขา L1 ใช้พอร์ท 1.3 ในการรับคำสั่งจากตัวประมวลผลของขา L1

ใช้พอร์ท 1.4 ส่งคำสั่งไปยังตัวประมวลผลของขา L2 ใช้พอร์ท 1.5 ในการรับคำสั่งจากตัวประมวลผลของขา L2

ใช้พอร์ท 1.6 ส่งคำสั่งไปยังตัวประมวลผลของขา R2 ใช้พอร์ท 1.7 ในการรับคำสั่งจากตัวประมวลผลของขา R2

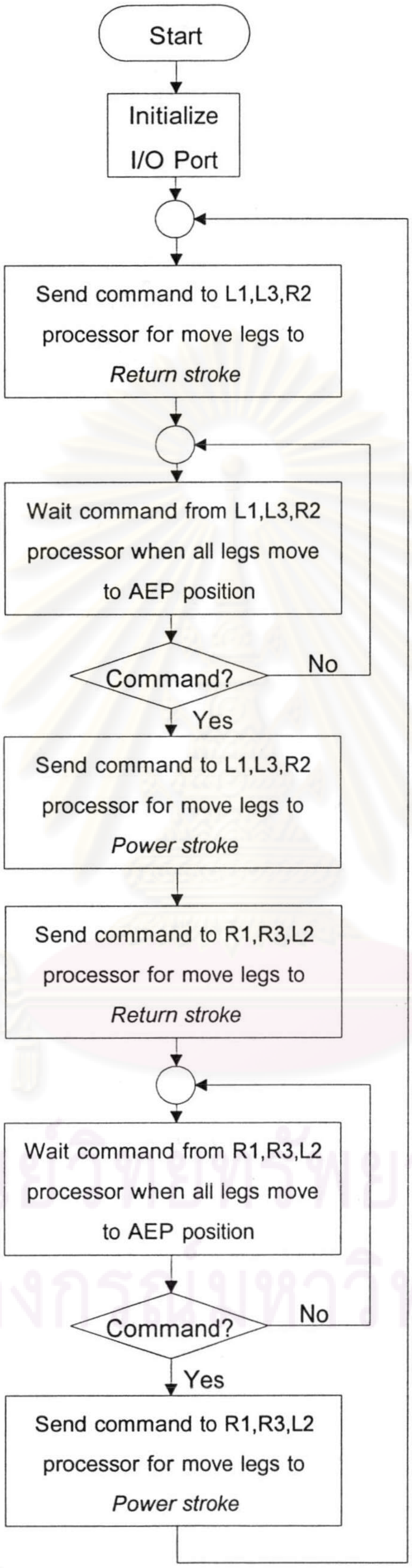
ใช้พอร์ท 3.0 ส่งคำสั่งไปยังตัวประมวลผลของขา L3 ใช้พอร์ท 3.1 ในการรับคำสั่งจากตัวประมวลผลของขา L3

ใช้พอร์ท 3.2 ส่งคำสั่งไปยังตัวประมวลผลของขา R3 ใช้พอร์ท 3.3 ในการรับคำสั่งจากตัวประมวลผลของขา R3

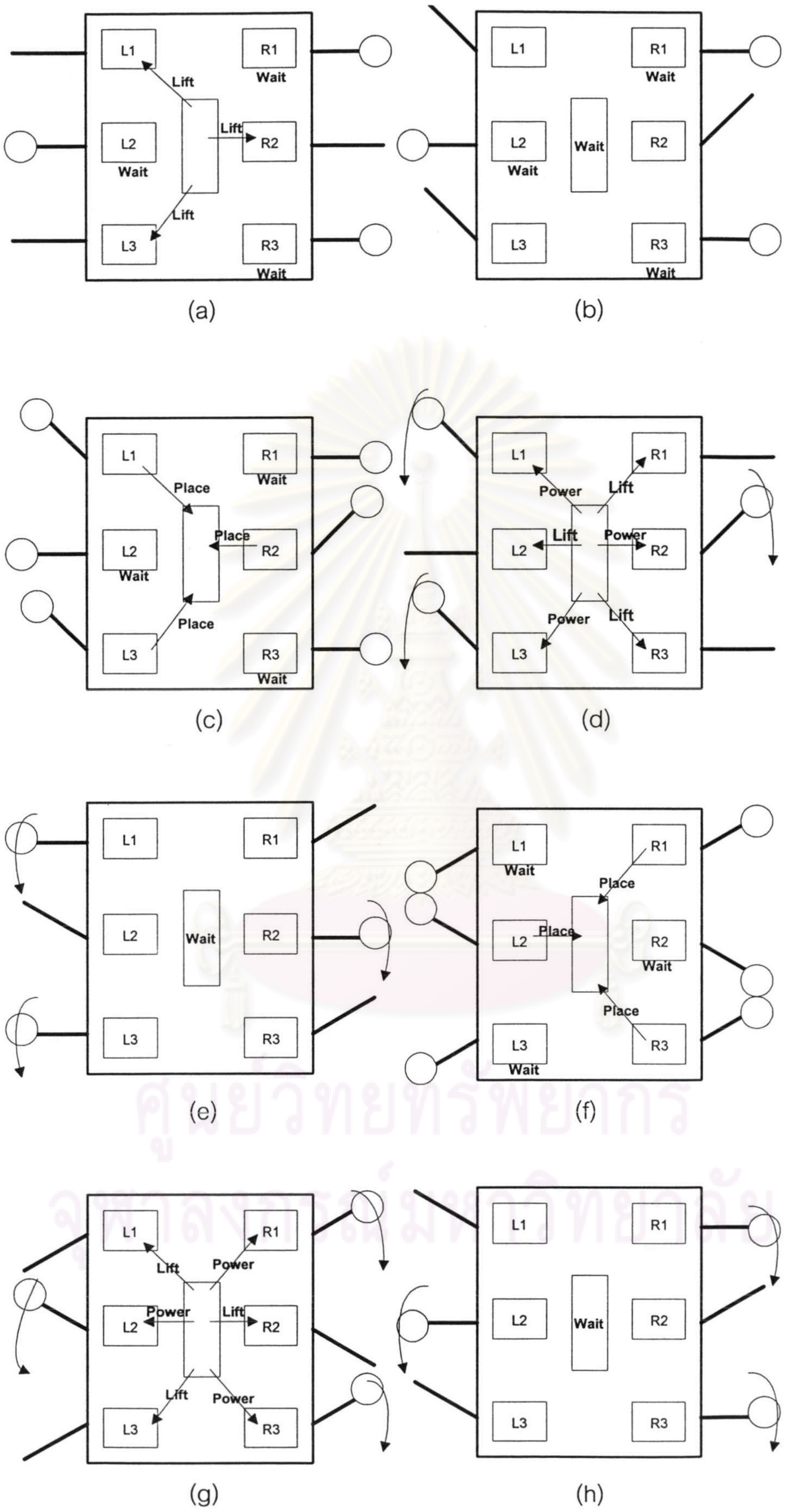
เมื่อเริ่มโปรแกรมจะต้องกำหนดค่าเริ่มต้นก่อน การที่จะใช้พอร์ตใดเป็น Input หรือ Output ถูกกำหนดโดยโปรแกรมส่วนค่าเริ่มต้น (Initialize)

หลังจากได้กำหนดแล้วว่า พอร์ตใดเป็น input หรือ output ในส่วนวงวนการทำงานหลักจะมีวงวนการทำงานย่อยไว้รับคำสั่งจากตัวประมวลผลของแต่ละขา ดังรูปที่ 5.4 เพื่อง่ายต่อการเข้าใจจะของอธิบายระบบทั้งหมด ดังรูปที่ 5.5

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

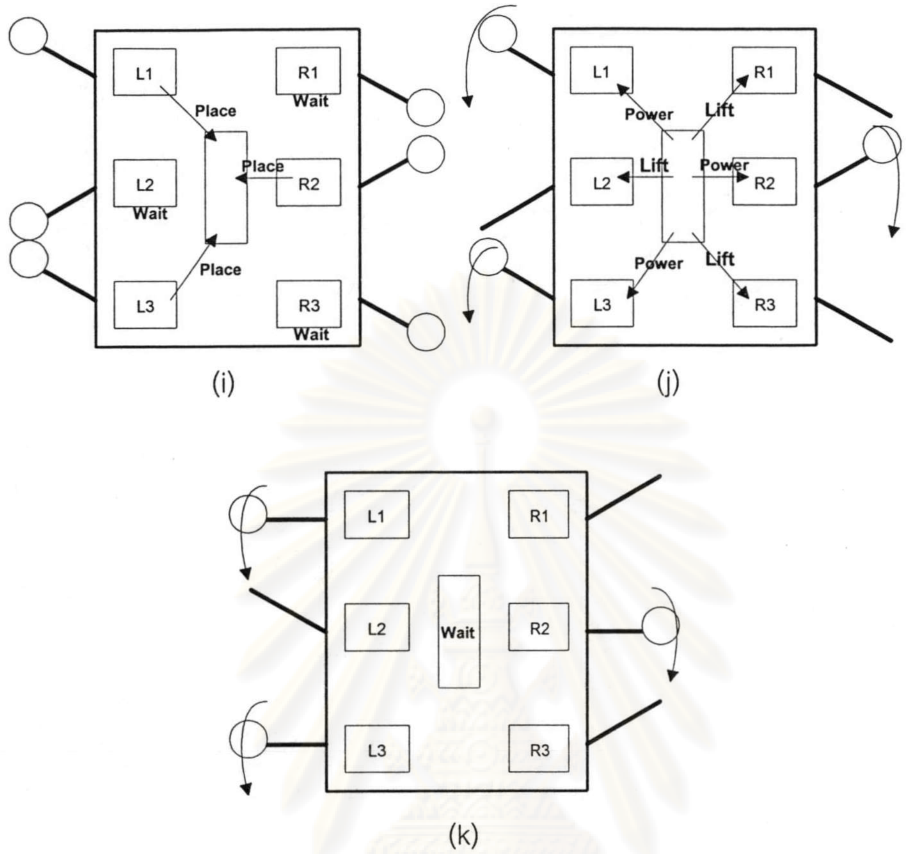


รูปที่ 5.4 แผนผังการไหลแสดงการทำงานของตัวประมวลผลกลางสำหรับโปรแกรมท่าเดินแบบไทรพอด (Tripod gait)



รูปที่ 5.5 การสื่อสารระหว่างตัวประมวลผลสำหรับโปรแกรมท่าเดินแบบไทรพอด (Tripod gait)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 5.5 (ต่อ) การสื่อสารระหว่างตัวประมวลผลสำหรับโปรแกรมท่าเดินแบบไทรพอด (Tripod gait)

จากรูปที่ 5.5

รูปที่ปลายขาของหุ่นยนต์มีปุ่มวงกลมหมายถึง ปลายขาของหุ่นยนต์สัมผัสพื้น

รูปที่ปลายขาของหุ่นยนต์ไม่มีปุ่มวงกลมหมายถึง ปลายขาของหุ่นยนต์ไม่ได้สัมผัสพื้น

(a) เมื่อเริ่มเคลื่อนที่ ตัวประมวลผลหลักจะออกคำสั่งไปยังตัวประมวลผลของขา L1,L3 และ R2 เพื่อเข้าสู่ช่วง Return stroke ในขณะที่ ตัวประมวลผลของขา R1,R3 และ L2 จะหยุดรอคำสั่งจากตัวประมวลผลหลัก

(b) เมื่อตัวประมวลผลหลักออกคำสั่งไปแล้ว จะหยุดรอคำสั่งกลับจากตัวประมวลผลของขา L1,L3 และ R2 ในขณะที่ ตัวประมวลผลของขา R1,R3 และ L2 จะหยุดรอคำสั่งจากตัวประมวลผลหลัก

(c) เมื่อขา L1,L3 และ R2 เคลื่อนที่ถึงตำแหน่ง AEP (เริ่มสัมผัสพื้น) ตัวประมวลผลของขา L1,L3 และ R2 จะส่งคำสั่งกลับมายังตัวประมวลผลหลัก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

(d) เมื่อตัวประมวลหลักได้รับคำสั่งจากตัวประมวลผลของขา L1,L3 และ R2 ทั้งสามขา ตัวประมวลผลหลักจะส่งคำสั่งไปยังตัวประมวลผลของขา R1,R3 และ L2 เพื่อเข้าสู่ช่วง Return stroke และ ส่งคำสั่งไปยังตัวประมวลผลของขา L1,L3 และ R2 เพื่อเข้าสู่ช่วง Power stroke

(e) ตัวประมวลผลหลักจะหยุดรอคำสั่งกลับจากตัวประมวลผลของขา R1,R3 และ L2

(f) เมื่อขา R1,R3 และ L2 เคลื่อนที่ถึงตำแหน่ง AEP (เริ่มสัมผัสพื้น) ตัวประมวลผลของขา R1,R3 และ L2 จะส่งคำสั่งกลับมายังตัวประมวลผลหลัก และ ในขณะที่ขา L1,L3 และ R2 จะหยุดเพื่อรอรับคำสั่งจากตัวประมวลผลหลัก

(g) เมื่อตัวประมวลหลักได้รับคำสั่งจากตัวประมวลผลของขา R1,R3 และ L2 ทั้งสามขา ตัวประมวลผลหลักจะส่งคำสั่งไปยังตัวประมวลผลของขา L1,L3 และ R2 เพื่อเข้าสู่ช่วง Return stroke และ ส่งคำสั่งไปยังตัวประมวลผลของขา R1,R3 และ L2 เพื่อเข้าสู่ช่วง Power stroke

(h) ตัวประมวลผลหลักจะหยุดรอคำสั่งกลับจากตัวประมวลผลของขา L1,L3 และ R2

(i) เมื่อขา L1,L3 และ R2 เคลื่อนที่ถึงตำแหน่ง AEP (เริ่มสัมผัสพื้น) ตัวประมวลผลของขา L1,L3 และ R2 จะส่งคำสั่งกลับมายังตัวประมวลผลหลัก และ ในขณะที่ขา R1,R3 และ L2 จะหยุดเพื่อรอรับคำสั่งจากตัวประมวลผลหลัก

(j) เมื่อตัวประมวลหลักได้รับคำสั่งจากตัวประมวลผลของขา L1,L3 และ R2 ทั้งสามขา ตัวประมวลผลหลักจะส่งคำสั่งไปยังตัวประมวลผลของขา R1,R3 และ L2 เพื่อเข้าสู่ช่วง Return stroke และ ส่งคำสั่งไปยังตัวประมวลผลของขา L1,L3 และ R2 เพื่อเข้าสู่ช่วง Power stroke

(k) ตัวประมวลผลหลักจะหยุดรอคำสั่งกลับจากตัวประมวลผลของขา R1,R3 และ L2 วนกลับไปทำเหมือนในรูป (e)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

5.2.2 โปรแกรมท่าเดินแบบลูกคลื่นต่ำ (Slow wave gait)

แบ่งเป็น 2 ส่วนคือ โปรแกรมของตัวประมวลผลของแต่ละขาของหุ่นยนต์ และโปรแกรมของตัวประมวลผลกลาง

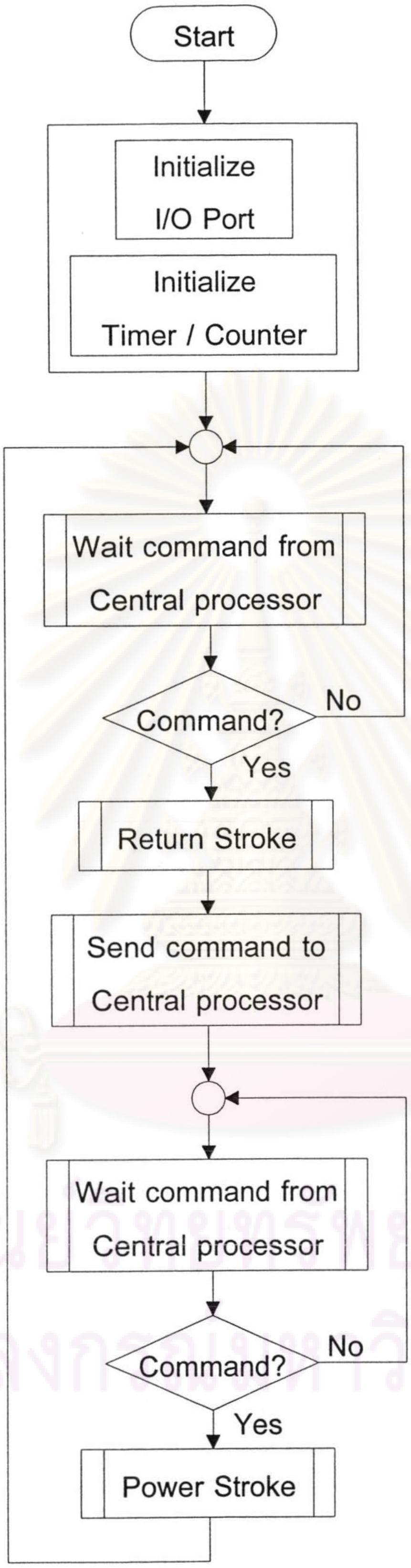
5.2.2.2 โปรแกรมของตัวประมวลผลของแต่ละขาของท่าเดินแบบลูกคลื่นต่ำ (Slow wave gait)

เมื่อเริ่มโปรแกรมจะต้องกำหนดค่าเริ่มต้นก่อน การที่จะใช้พอร์ตใดเป็น Input หรือ Output ถูกกำหนดโดยโปรแกรมส่วนค่าเริ่มต้น (Initialize) โปรแกรมส่วนนี้ยังกำหนดให้ตัวประมวลผลสามารถใช้ วงจรนับ/จับเวลาภายในได้ ซึ่งถูกนำไปใช้ในการสร้างพัลซ์ให้กับอาร์ซีเซอร์โวมอเตอร์

หลังจากได้กำหนดแล้วว่า พอร์ตใดเป็น input หรือ output จะมีโปรแกรมส่วนติดต่อ Hardware มารองรับการรับและส่งค่าไปมาระหว่างตัวประมวลผลหลักและตัวประมวลผลของแต่ละขา โดยโปรแกรมส่วนดังกล่าวนี้ถูกเขียนเป็นโปรแกรมย่อย (Subroutine) ซึ่งจะถูกเรียกใช้โดยวงวนการทำงานหลัก (Main loop) ต่อไป

ในส่วนวงวนการทำงานหลัก จะมีวงวนการทำงานย่อยไว้รับคำสั่งที่ตัวประมวลผลหลักส่งมา 2 วงวนการทำงาน วงวนการทำงานย่อยแรกมีไว้สำหรับรับคำสั่งจากตัวประมวลผลหลักเพื่อเข้าสู่ช่วง Return stroke และ วงวนการทำงานย่อยที่สองมีไว้สำหรับรับคำสั่งจากตัวประมวลผลหลักเพื่อเข้าสู่ช่วง Power stroke ส่วนการส่งคำสั่งกลับไปยังตัวประมวลผลหลักสำหรับโปรแกรมท่าเดินแบบลูกคลื่นต่ำ (Slow wave gait) จะส่งกลับไปเมื่อขาของหุ่นยนต์เคลื่อนที่ผ่านช่วง Return stroke ไปแล้วเช่นเดียวกับท่าเดินแบบไทรพอด (Tripod gait) ดังรูปที่ 5.6

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

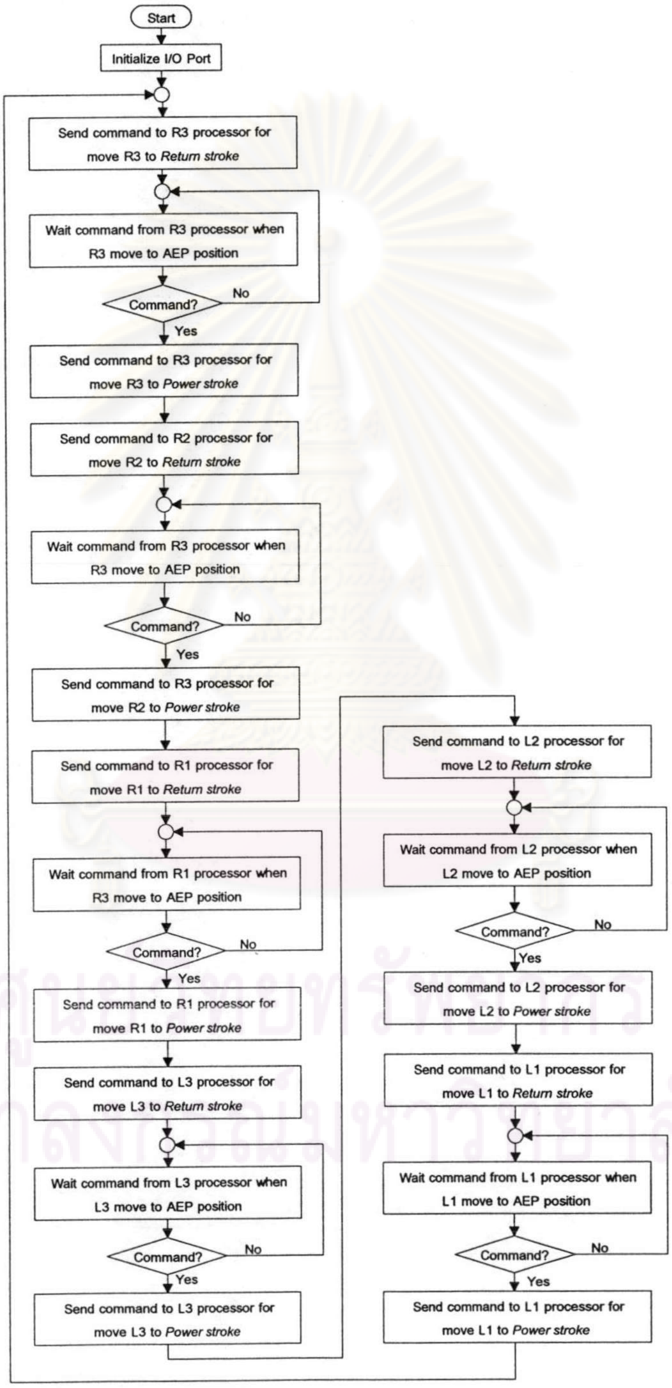


ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

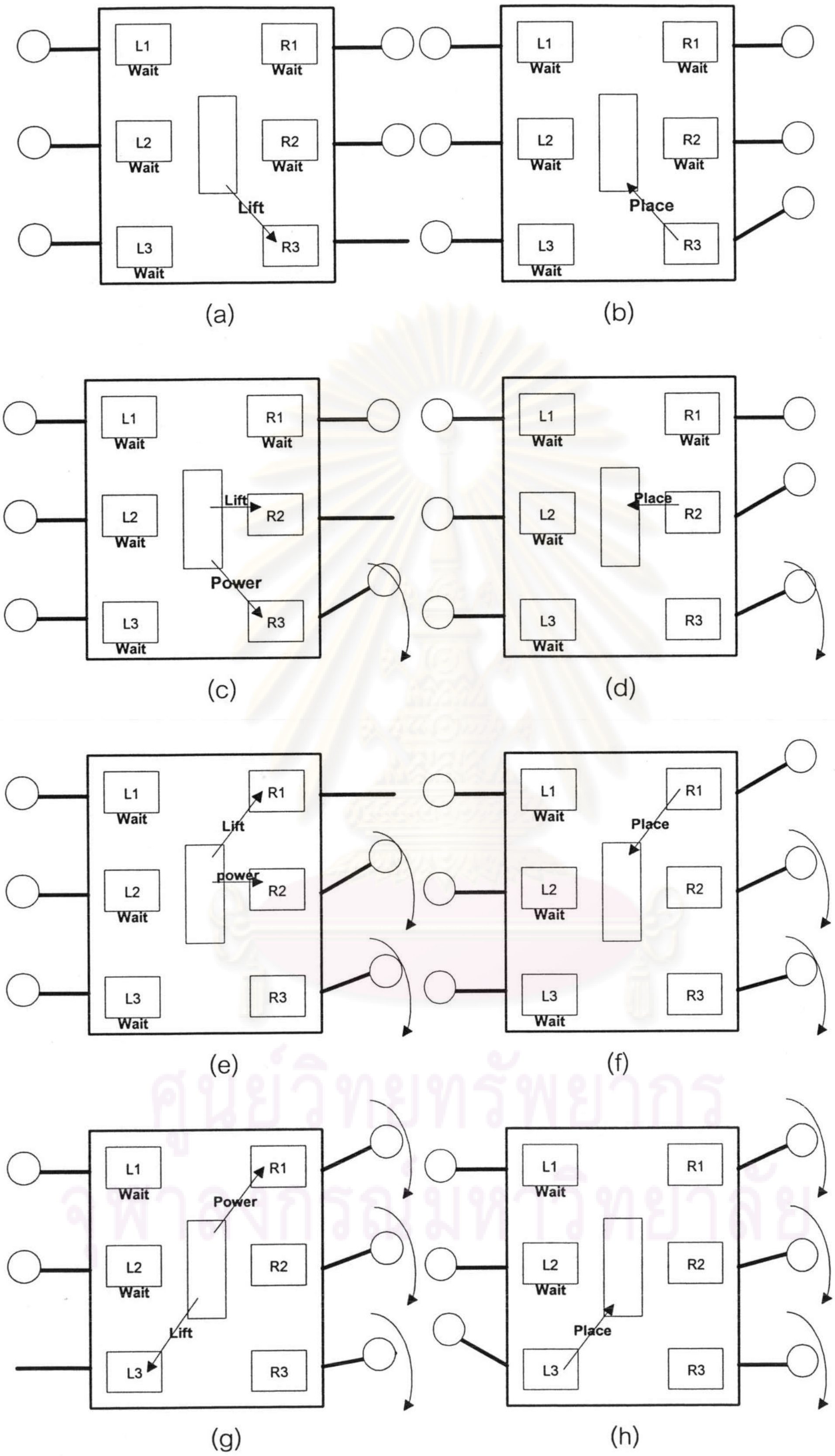
รูปที่ 5.6 แผนผังการไหลแสดงการทำงานของตัวประมวลผลของแต่ละขาสำหรับโปรแกรมท่าเดินแบบลูกคลื่นต่ำ (Slow wave gait)

5.2.2.2 โปรแกรมของตัวประมวลผลกลางของท่าเดินแบบลูกคลื่นต่ำ (Slow wave gait)

เมื่อเริ่มโปรแกรมจะต้องกำหนดค่าเริ่มต้นก่อน การที่จะใช้พอร์ตใดเป็น Input หรือ Output ถูกกำหนดโดยโปรแกรมส่วนค่าเริ่มต้น (Initialize) หลังจากได้กำหนดแล้วว่า พอร์ตใดเป็น input หรือ output ในส่วนวงวนการทำงานหลัก จะมีวงวนการทำงานย่อยไว้รับคำสั่งจากตัวประมวลผลของแต่ละขา ดังรูปที่ 5.7 เพื่อง่ายต่อการเข้าใจจะของอธิบายระบบทั้งหมด ดังรูปที่ 5.8

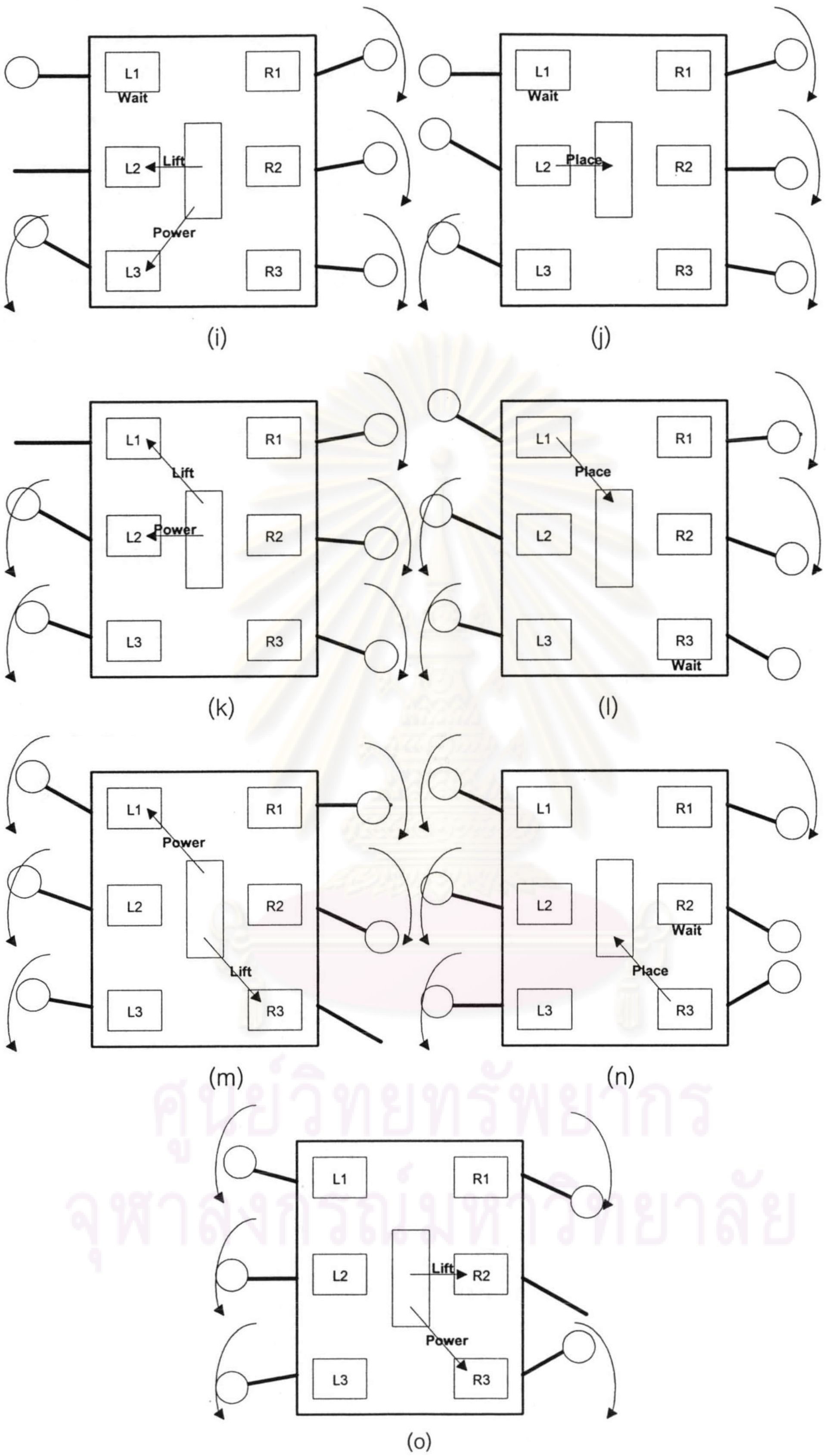


รูปที่ 5.7 แผนผังการไหลแสดงการทำงานของตัวประมวลผลกลางสำหรับโปรแกรมท่าเดินแบบลูกคลื่นต่ำ (Slow wave gait)



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

รูปที่ 5.8 การสื่อสารระหว่างตัวประมวลผลสำหรับโปรแกรมท่าเดินแบบลูกคลื่นต่ำ (Slow wave gait)



รูปที่ 5.8 (ต่อ) การสื่อสารระหว่างตัวประมวลผลสำหรับโปรแกรมท่าเดินแบบลูกคลื่นต่ำ (Slow wave gait)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

จากรูปที่ 5.8

รูปที่ปลายขาของหุ่นยนต์มีปุ่มวงกลมหมายถึง ปลายขาของหุ่นยนต์สัมผัสพื้น

รูปที่ปลายขาของหุ่นยนต์ไม่มีปุ่มวงกลมหมายถึง ปลายขาของหุ่นยนต์ไม่ได้สัมผัสพื้น

(a) เมื่อเริ่มเคลื่อนที่ ตัวประมวลผลหลักจะออกคำสั่งไปยังตัวประมวลผลของขา R3 เพื่อเข้าสู่ช่วง Return stroke หลังจากนั้นตัวประมวลผลหลัก จะหยุดรอคำสั่งกลับจากตัวประมวลผลของขา R3 ในขณะที่ตัวประมวลผลของขา L1,L2,L3,R1 และ R2 จะหยุดรอคำสั่งจากตัวประมวลผลหลัก

(b) เมื่อขา R3 เคลื่อนที่ถึงตำแหน่ง AEP (เริ่มสัมผัสพื้น) ตัวประมวลผลของขา R3 จะส่งคำสั่งกลับมายังตัวประมวลผลหลัก ในขณะที่ตัวประมวลผลของขา L1,L2,L3,R1 และ R2 จะหยุดรอคำสั่งจากตัวประมวลผลหลัก

(c) เมื่อตัวประมวลผลหลักได้รับคำสั่งจากตัวประมวลผลของขา R3 ตัวประมวลผลหลักจะออกคำสั่งไปยังตัวประมวลผลของขา R2 เพื่อเข้าสู่ช่วง Return stroke และ ออกคำสั่งไปยังตัวประมวลผลของขา R3 เพื่อเข้าสู่ช่วง Power stroke

(d) เมื่อขา R2 เคลื่อนที่ถึงตำแหน่ง AEP (เริ่มสัมผัสพื้น) ตัวประมวลผลของขา R2 จะส่งคำสั่งกลับมายังตัวประมวลผลหลัก

(e) เมื่อตัวประมวลผลหลักได้รับคำสั่งจากตัวประมวลผลของขา R2 ตัวประมวลผลหลักจะออกคำสั่งไปยังตัวประมวลผลของขา R1 เพื่อเข้าสู่ช่วง Return stroke และ ออกคำสั่งไปยังตัวประมวลผลของขา R2 เพื่อเข้าสู่ช่วง Power stroke

(f) เมื่อขา R1 เคลื่อนที่ถึงตำแหน่ง AEP (เริ่มสัมผัสพื้น) ตัวประมวลผลของขา R1 จะส่งคำสั่งกลับมายังตัวประมวลผลหลัก

(g) เมื่อตัวประมวลผลหลักได้รับคำสั่งจากตัวประมวลผลของขา R1 ตัวประมวลผลหลักจะออกคำสั่งไปยังตัวประมวลผลของขา L3 เพื่อเข้าสู่ช่วง Return stroke และ ออกคำสั่งไปยังตัวประมวลผลของขา R1 เพื่อเข้าสู่ช่วง Power stroke

(h) เมื่อขา L3 เคลื่อนที่ถึงตำแหน่ง AEP (เริ่มสัมผัสพื้น) ตัวประมวลผลของขา L3 จะส่งคำสั่งกลับมายังตัวประมวลผลหลัก

(i) เมื่อตัวประมวลผลหลักได้รับคำสั่งจากตัวประมวลผลของขา L3 ตัวประมวลผลหลักจะออกคำสั่งไปยังตัวประมวลผลของขา L2 เพื่อเข้าสู่ช่วง Return stroke และ ออกคำสั่งไปยังตัวประมวลผลของขา L3 เพื่อเข้าสู่ช่วง Power stroke

(j) เมื่อขา L2 เคลื่อนที่ถึงตำแหน่ง AEP (เริ่มสัมผัสพื้น) ตัวประมวลผลของขา L2 จะส่งคำสั่งกลับมายังตัวประมวลผลหลัก

(k) เมื่อตัวประมวลผลหลักได้รับคำสั่งจากตัวประมวลผลของขา L2 ตัวประมวลผลหลักจะออกคำสั่งไปยังตัวประมวลผลของขา L1 เพื่อเข้าสู่ช่วง Return stroke และ ออกคำสั่งไปยังตัวประมวลผลของขา L2 เพื่อเข้าสู่ช่วง Power stroke

(l) เมื่อขา L1 เคลื่อนที่ถึงตำแหน่ง AEP (เริ่มสัมผัสพื้น) ตัวประมวลผลของขา L1 จะส่งคำสั่งกลับมายังตัวประมวลผลหลัก และ ในขณะเดียวกันที่ขา R3 เคลื่อนที่ถึงจุด PEP ตัวประมวลผลของขา R3 จะหยุดรอรับคำสั่งจากตัวประมวลผลหลัก

(m) เมื่อตัวประมวลผลหลักได้รับคำสั่งจากตัวประมวลผลของขา L1 ตัวประมวลผลหลักจะออกคำสั่งไปยังตัวประมวลผลของขา R3 เพื่อเข้าสู่ช่วง Return stroke และ ออกคำสั่งไปยังตัวประมวลผลของขา L1 เพื่อเข้าสู่ช่วง Power stroke

(n) เมื่อขา R3 เคลื่อนที่ถึงตำแหน่ง AEP (เริ่มสัมผัสพื้น) ตัวประมวลผลของขา R3 จะส่งคำสั่งกลับมายังตัวประมวลผลหลัก และ ในขณะเดียวกันที่ขา R2 เคลื่อนที่ถึงจุด PEP ตัวประมวลผลของขา R2 จะหยุดรอรับคำสั่งจากตัวประมวลผลหลัก

(o) เมื่อตัวประมวลผลหลักได้รับคำสั่งจากตัวประมวลผลของขา R3 ตัวประมวลผลหลักจะออกคำสั่งไปยังตัวประมวลผลของขา R2 เพื่อเข้าสู่ช่วง Return stroke และ ออกคำสั่งไปยังตัวประมวลผลของขา R3 เพื่อเข้าสู่ช่วง Power stroke เช่นเดียวกับรูป (c)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

5.2.3 โปรแกรมท่าเดินแบบกระเพื่อม (Ripple gait)

แบ่งเป็น 2 ส่วนคือ โปรแกรมของตัวประมวลผลของแต่ละขาของหุ่นยนต์ และโปรแกรมของตัวประมวลผลกลาง

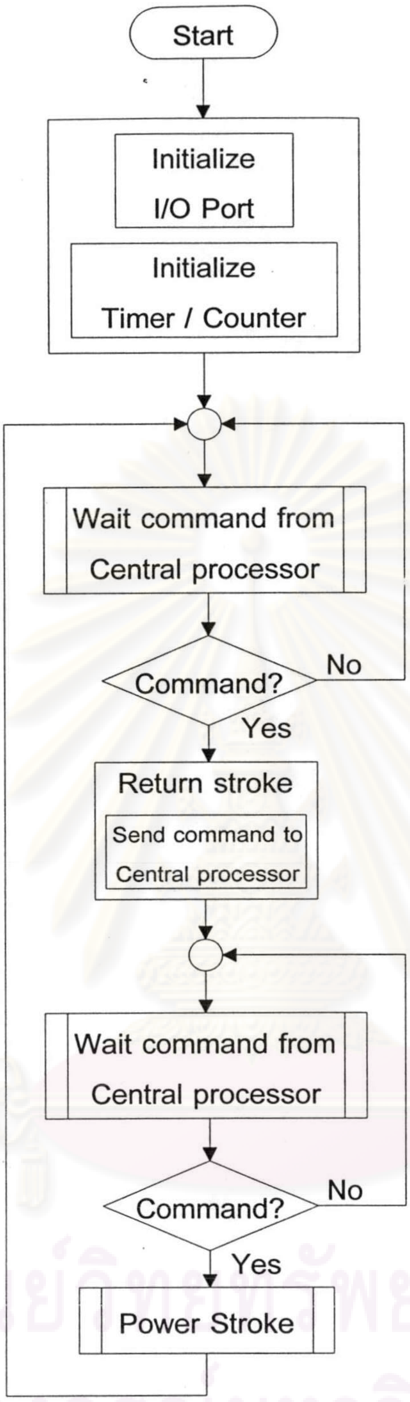
5.2.3.1 โปรแกรมของตัวประมวลผลของแต่ละขาของท่าเดินแบบกระเพื่อม (Ripple gait)

เมื่อเริ่มโปรแกรมจะต้องกำหนดค่าเริ่มต้นก่อน การที่จะใช้พอร์ตใดเป็น Input หรือ Output ถูกกำหนดโดยโปรแกรมส่วนค่าเริ่มต้น (Initialize) โปรแกรมส่วนนี้ยังกำหนดให้ตัวประมวลผลสามารถใช้ วงจรนับ/จับเวลาภายในได้ ซึ่งถูกนำไปใช้ในการสร้างพัลซ์ให้กับอาร์ซีเซอร์โวมอเตอร์

หลังจากได้กำหนดแล้วว่า พอร์ตใดเป็น input หรือ output จะมีโปรแกรมส่วนติดต่อ Hardware มารองรับการรับและส่งค่าไปมาระหว่างตัวประมวลผลหลักและตัวประมวลผลของแต่ละขา โดยโปรแกรมส่วนดังกล่าวนี้ถูกเขียนเป็นโปรแกรมย่อย (Subroutine) ซึ่งจะถูกเรียกใช้โดยวงวนการทำงานหลัก (Main loop) ต่อไป

ในส่วนวงวนการทำงานหลัก จะมีวงวนการทำงานย่อยไว้รับคำสั่งที่ตัวประมวลผลหลักส่งมา 2 วงวนการทำงาน วงวนการทำงานย่อยแรกมีไว้สำหรับรับคำสั่งจากตัวประมวลผลหลักเพื่อเข้าสู่ช่วง Return stroke และ วงวนการทำงานย่อยที่สองมีไว้สำหรับรับคำสั่งจากตัวประมวลผลหลักเพื่อเข้าสู่ช่วง Power stroke ส่วนการส่งคำสั่งกลับไปยังตัวประมวลผลหลักสำหรับโปรแกรมท่าเดินแบบกระเพื่อม (Ripple gait) จะส่งกลับไปเมื่อขาของหุ่นยนต์เคลื่อนที่ถึงครึ่งช่วง Return stroke ดังรูปที่ 5.9

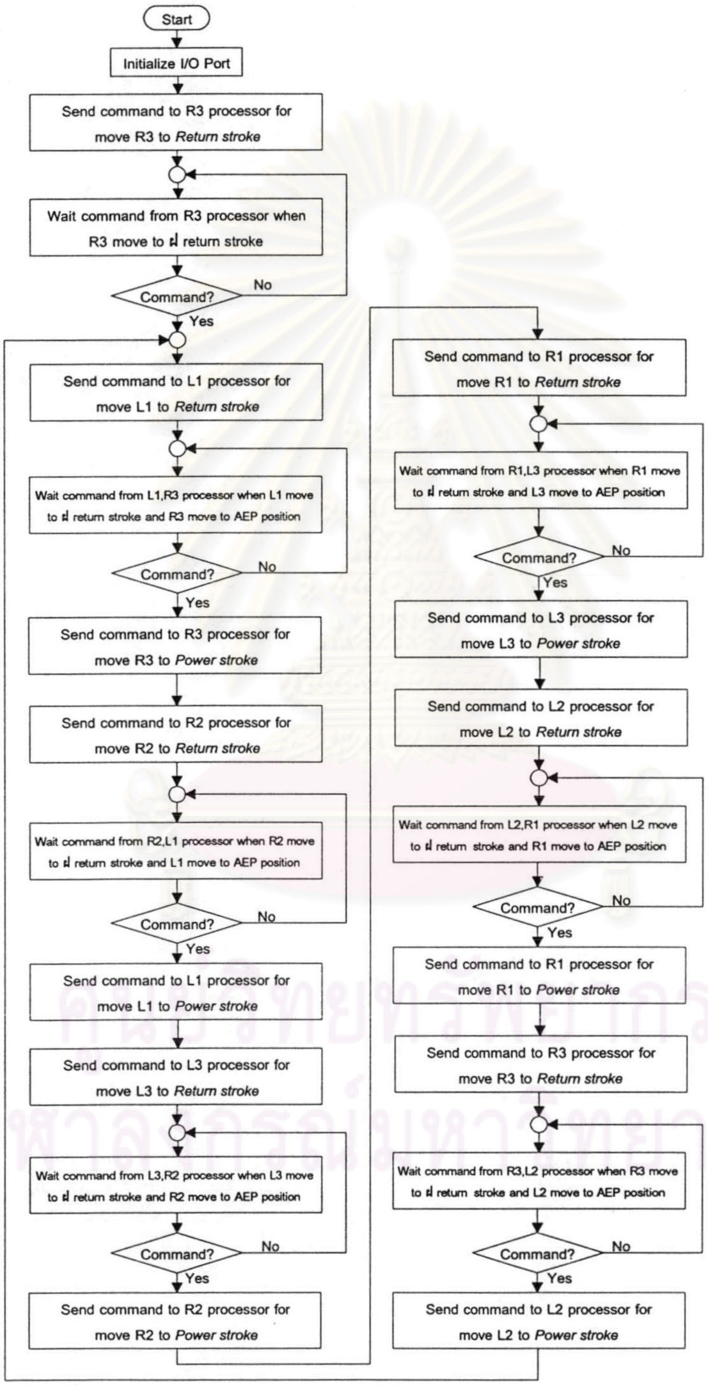
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



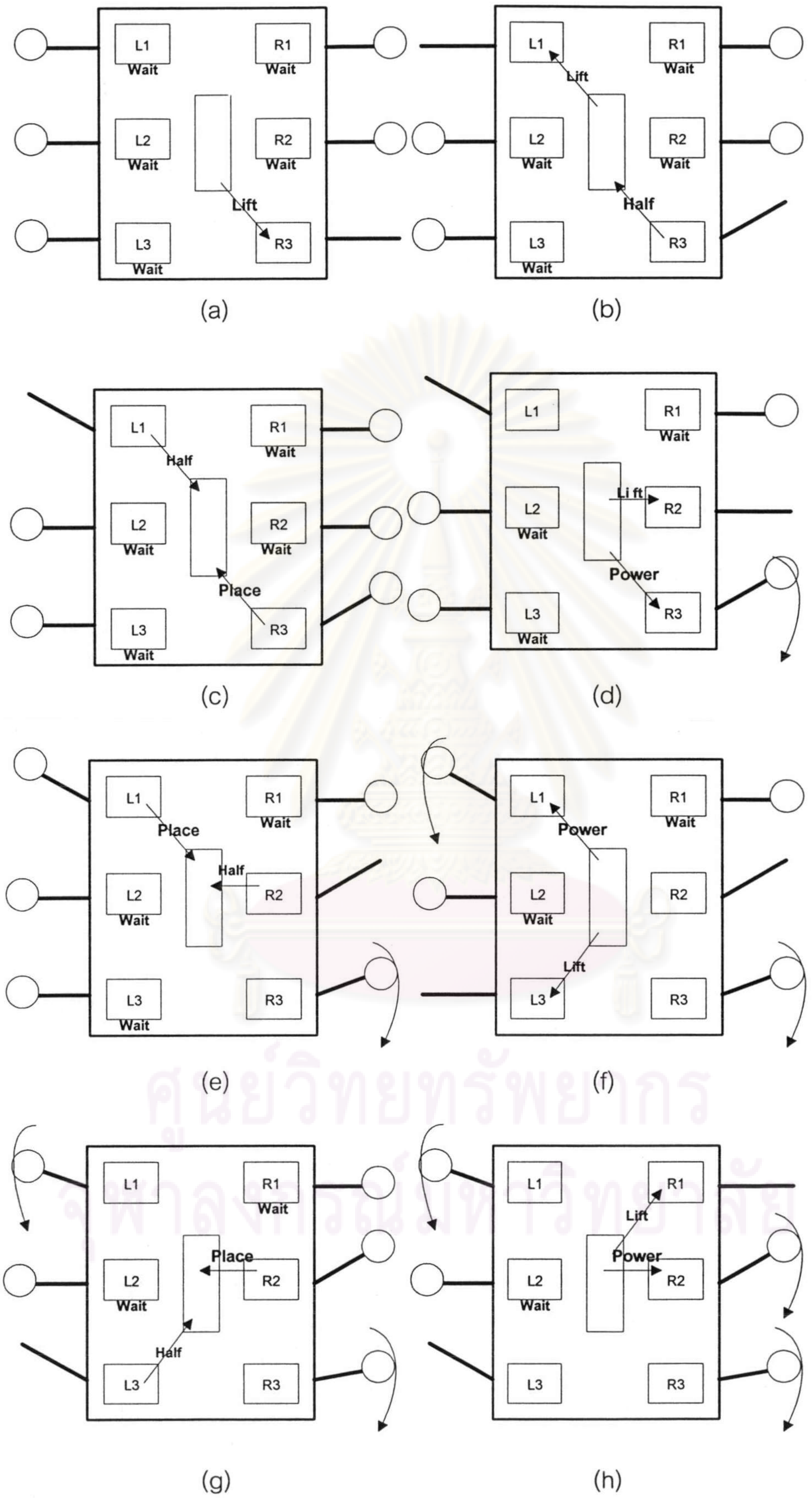
รูปที่ 5.9 แผนผังการไหลแสดงการทำงานของตัวประมวลผลของแต่ละขาสำหรับโปรแกรมท่าเดินแบบกระเพื่อม (Ripple gait)

5.2.3.2 โปรแกรมของตัวประมวลผลกลางของท่าเดินแบบกระเพื่อม (Ripple gait)

เมื่อเริ่มโปรแกรมจะต้องกำหนดค่าเริ่มต้นก่อน การที่จะใช้พอร์ตใดเป็น Input หรือ Output ถูกกำหนดโดยโปรแกรมส่วนค่าเริ่มต้น (Initialize) หลังจากได้กำหนดแล้วว่า พอร์ตใดเป็น input หรือ output ในส่วนวงวนการทำงานหลัก จะมีวงวนการทำงานย่อยไว้รับคำสั่งจากตัวประมวลผลของแต่ละขา ดังรูปที่ 5.10 เพื่อง่ายต่อการเข้าใจจะของอธิบายระบบทั้งหมด ดังรูปที่ 5.11

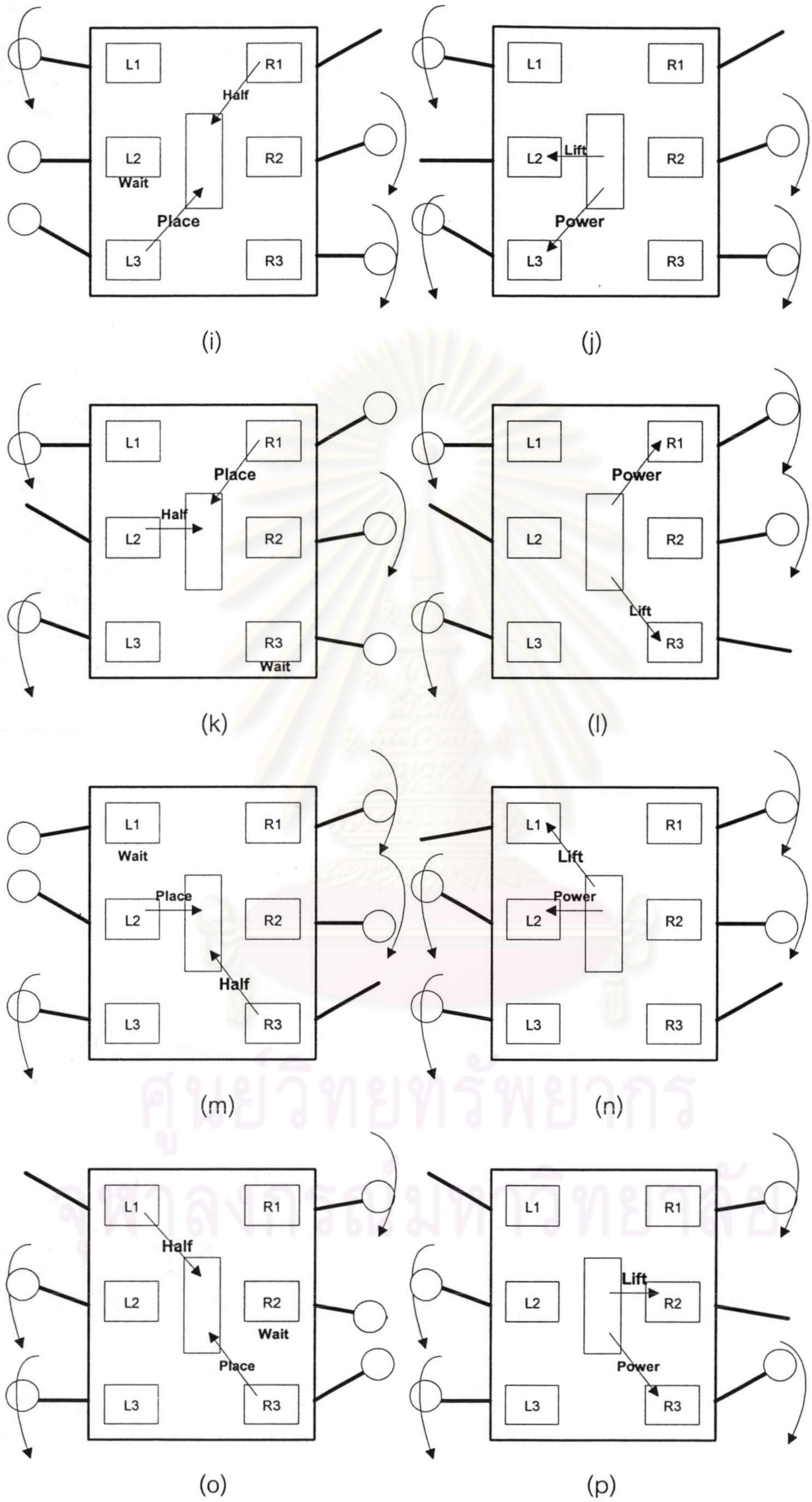


รูปที่ 5.10 แผนผังการไหลแสดงการทำงานของตัวประมวลผลกลางสำหรับโปรแกรมท่าเดินแบบกระเพื่อม (Ripple gait)



รูปที่ 5.11 การสื่อสารระหว่างตัวประมวลผลสำหรับโปรแกรมท่าเดินแบบกระเพื่อม (Ripple gait)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 5.11 (ต่อ) การสื่อสารระหว่างตัวประมวลผลสำหรับโปรแกรมท่าเดินแบบกระเพื่อม

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

จากรูปที่ 5.11

รูปที่ปลายขาของหุ่นยนต์มีปุ่มวงกลมหมายถึง ปลายขาของหุ่นยนต์สัมผัสพื้น

รูปที่ปลายขาของหุ่นยนต์ไม่มีปุ่มวงกลมหมายถึง ปลายขาของหุ่นยนต์ไม่ได้สัมผัสพื้น

(a) เมื่อเริ่มเคลื่อนที่ ตัวประมวลผลหลักจะออกคำสั่งไปยังตัวประมวลผลของขา R3 เพื่อเข้าสู่ช่วง Return stroke หลังจากนั้นตัวประมวลผลหลักจะหยุดรอคำสั่งกลับของตัวประมวลผลของขา R3 ในขณะที่ตัวประมวลผลของขา L1,L2,L3,R1 และ R2 จะหยุดรอคำสั่งจากตัวประมวลผลหลัก

(b) เมื่อขา R3 เคลื่อนที่ถึงประมาณครึ่งช่วงของช่วง Return stroke ตัวประมวลผลของขา R3 จะส่งคำสั่งกลับมายังตัวประมวลผลหลัก เมื่อตัวประมวลผลหลักได้รับคำสั่งกลับจากตัวประมวลผลของขา R3 ตัวประมวลผลหลักกลับไปยังตัวประมวลผลของขา L1 เพื่อเข้าสู่ช่วง Return stroke หลังจากนั้นตัวประมวลผลหลักจะหยุดรอคำสั่งกลับของตัวประมวลผลของขา L1 และ R3 ในขณะที่ตัวประมวลผลของขา L2,L3,R1 และ R2 จะหยุดรอคำสั่งจากตัวประมวลผลหลัก

(c) เมื่อขา L1 เคลื่อนที่ถึงประมาณครึ่งช่วงของช่วง Return stroke และ ขา R3 เคลื่อนที่ถึงตำแหน่ง AEP (เริ่มสัมผัสพื้น) ตัวประมวลผลของขา L1 และ ขา R3 จะส่งคำสั่งกลับมายังตัวประมวลผลหลัก

(d) เมื่อตัวประมวลผลหลักได้รับคำสั่งกลับจากตัวประมวลผลของขา L1 และ R3 ตัวประมวลผลหลักจะส่งคำสั่งกลับไปยังตัวประมวลผลของขา R2 เพื่อเข้าสู่ช่วง Return stroke และ ส่งคำสั่งกลับไปยังตัวประมวลผลของขา R3 เพื่อเข้าสู่ช่วง Power stroke หลังจากนั้นตัวประมวลผลหลักจะหยุดรอคำสั่งกลับของตัวประมวลผลของขา R2 และ L1

(e) เมื่อขา R2 เคลื่อนที่ถึงประมาณครึ่งช่วงของช่วง Return stroke และ ขา L1 เคลื่อนที่ถึงตำแหน่ง AEP (เริ่มสัมผัสพื้น) ตัวประมวลผลของขา R2 และ ขา L1 จะส่งคำสั่งกลับมายังตัวประมวลผลหลัก

(f) เมื่อตัวประมวลผลหลักได้รับคำสั่งกลับจากตัวประมวลผลของขา R2 และ L1 ตัวประมวลผลหลักจะส่งคำสั่งกลับไปยังตัวประมวลผลของขา L3 เพื่อเข้าสู่ช่วง Return stroke และ ส่งคำสั่งกลับไปยังตัวประมวลผลของขา L1 เพื่อเข้าสู่ช่วง Power stroke หลังจากนั้นตัวประมวลผลหลักจะหยุดรอคำสั่งกลับของตัวประมวลผลของขา L3 และ R2

ศูนย์วิทยทรัพยากร จุฬาลงกรณ์มหาวิทยาลัย

(g) เมื่อขา L3 เคลื่อนที่ถึงประมาณครึ่งช่วงของช่วง Return stroke และ ขา R2 เคลื่อนที่ถึงตำแหน่ง AEP (เริ่มสัมผัสพื้น) ตัวประมวลผลของขา L3 และ ขา R2 จะส่งคำสั่งกลับมายังตัวประมวลผลหลัก

(h) เมื่อตัวประมวลผลหลักได้รับคำสั่งกลับจากตัวประมวลผลของขา L3 และ R2 ตัวประมวลผลหลักจะส่งคำสั่งกลับไปยังตัวประมวลผลของขา R1 เพื่อเข้าสู่ช่วง Return stroke และ ส่งคำสั่งกลับไปยังตัวประมวลผลของขา R2 เพื่อเข้าสู่ช่วง Power stroke หลังจากนั้นตัวประมวลผลหลักจะหยุดรอคำสั่งกลับของตัวประมวลผลของขา R1 และ L3

(i) เมื่อขา R1 เคลื่อนที่ถึงประมาณครึ่งช่วงของช่วง Return stroke และ ขา L3 เคลื่อนที่ถึงตำแหน่ง AEP (เริ่มสัมผัสพื้น) ตัวประมวลผลของขา R1 และ ขา R3 จะส่งคำสั่งกลับมายังตัวประมวลผลหลัก

(j) เมื่อตัวประมวลผลหลักได้รับคำสั่งกลับจากตัวประมวลผลของขา R1 และ L3 ตัวประมวลผลหลักจะส่งคำสั่งกลับไปยังตัวประมวลผลของขา L2 เพื่อเข้าสู่ช่วง Return stroke และ ส่งคำสั่งกลับไปยังตัวประมวลผลของขา L3 เพื่อเข้าสู่ช่วง Power stroke หลังจากนั้นตัวประมวลผลหลักจะหยุดรอคำสั่งกลับของตัวประมวลผลของขา L2 และ R1

(k) เมื่อขา L2 เคลื่อนที่ถึงประมาณครึ่งช่วงของช่วง Return stroke และ ขา R1 เคลื่อนที่ถึงตำแหน่ง AEP (เริ่มสัมผัสพื้น) ตัวประมวลผลของขา L2 และ ขา R1 จะส่งคำสั่งกลับมายังตัวประมวลผลหลัก และในขณะเดียวกันขา R3 เคลื่อนที่ถึงตำแหน่ง PEP ตัวประมวลผลของขา R3 จะหยุดรอคำสั่งกลับของตัวประมวลผลหลัก

(l) เมื่อตัวประมวลผลหลักได้รับคำสั่งกลับจากตัวประมวลผลของขา L2 และ R1 ตัวประมวลผลหลักจะส่งคำสั่งกลับไปยังตัวประมวลผลของขา R3 เพื่อเข้าสู่ช่วง Return stroke และ ส่งคำสั่งกลับไปยังตัวประมวลผลของขา R1 เพื่อเข้าสู่ช่วง Power stroke หลังจากนั้นตัวประมวลผลหลักจะหยุดรอคำสั่งกลับของตัวประมวลผลของขา R3 และ L2

(m) เมื่อขา R3 เคลื่อนที่ถึงประมาณครึ่งช่วงของช่วง Return stroke และ ขา L2 เคลื่อนที่ถึงตำแหน่ง AEP (เริ่มสัมผัสพื้น) ตัวประมวลผลของขา R3 และ ขา L2 จะส่งคำสั่งกลับมายังตัวประมวลผลหลัก และในขณะเดียวกันขา L1 เคลื่อนที่ถึงตำแหน่ง PEP ตัวประมวลผลของขา L1 จะหยุดรอคำสั่งกลับของตัวประมวลผลหลัก

(n) เมื่อตัวประมวลผลหลักได้รับคำสั่งกลับจากตัวประมวลผลของขา R3 และ L2 ตัวประมวลผลหลักจะส่งคำสั่งกลับไปยังตัวประมวลผลของขา L1 เพื่อเข้าสู่ช่วง Return stroke และ ส่งคำสั่งกลับไปยังตัวประมวลผลของขา L2 เพื่อเข้าสู่ช่วง Power stroke หลังจากนั้นตัวประมวลผลหลักจะหยุดรอคำสั่งกลับของตัวประมวลผลของขา L1 และ R3

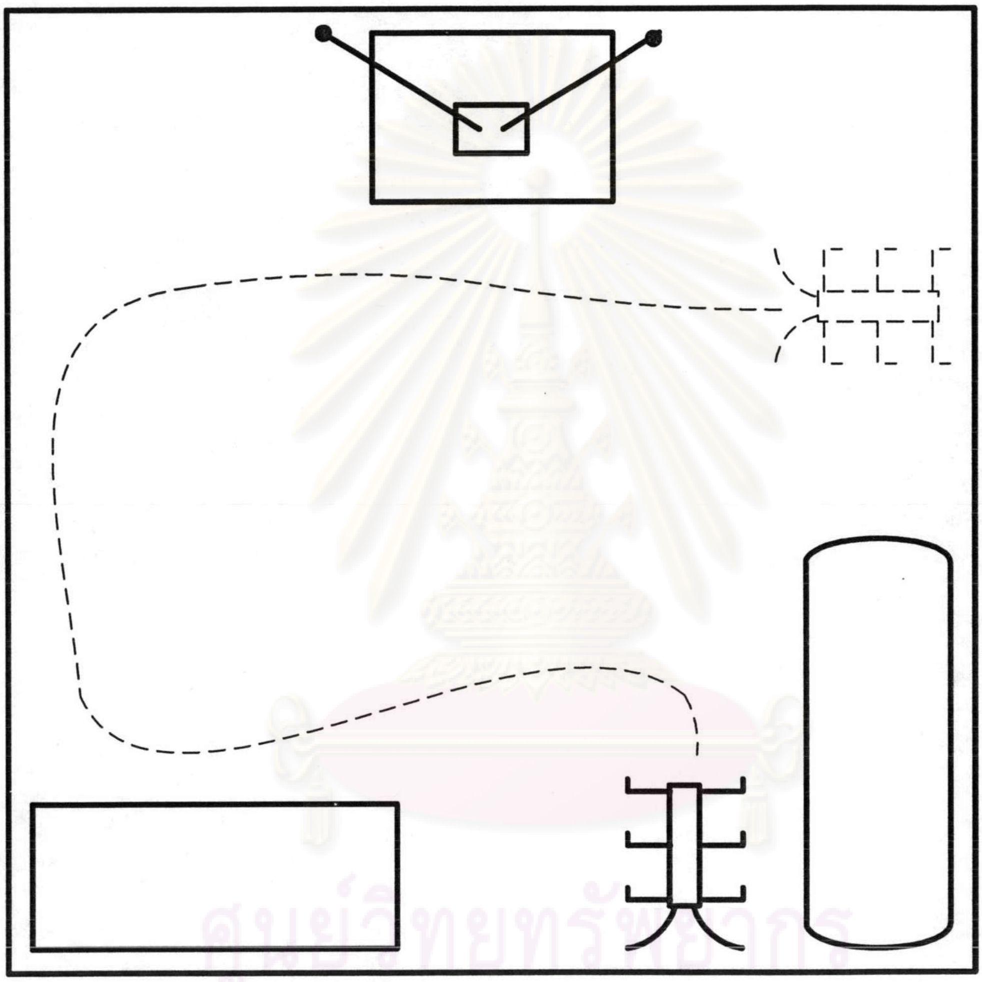
(o) เมื่อขา L1 เคลื่อนที่ถึงประมาณครึ่งช่วงของช่วง Return stroke และ ขา R3 เคลื่อนที่ถึงตำแหน่ง AEP (เริ่มสัมผัสพื้น) ตัวประมวลผลของขา L1 และ ขา R3 จะส่งคำสั่งกลับมายังตัวประมวลผลหลัก และในขณะเดียวกันขา R2 เคลื่อนที่ถึงตำแหน่ง PEP ตัวประมวลผลของขา R2 จะหยุดรอคำสั่งกลับของตัวประมวลผลหลัก

(p) เมื่อตัวประมวลผลหลักได้รับคำสั่งกลับจากตัวประมวลผลของขา L1 และ R3 ตัวประมวลผลหลักจะส่งคำสั่งกลับไปยังตัวประมวลผลของขา R2 เพื่อเข้าสู่ช่วง Return stroke และ ส่งคำสั่งกลับไปยังตัวประมวลผลของขา R3 เพื่อเข้าสู่ช่วง Power stroke หลังจากนั้นตัวประมวลผลหลักจะหยุดรอคำสั่งกลับของตัวประมวลผลของขา R2 และ L1 เช่นเดียวกับในรูป (d)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

5.3 โปรแกรมช่วยเหลือที่ใช้ในการแก้ไขรูปแบบการเดินของหุ่นยนต์

ในการนำไปประยุกษ์ใช้งาน ดังในรูปที่ 5.12 เมื่อหุ่นยนต์เจอสิ่งกีดขว้างหุ่นยนต์จะหลบหลีกสิ่งกีดขวางโดยการเดินเลี้ยวซ้าย เดินเลี้ยวขวา หรือเดินถอยหลังก่อน แล้วจึงเลี้ยวซ้ายหรือขวา ขึ้นอยู่กับโปรแกรมที่เขียนไว้ในตอนแรก ดังนั้นเมื่อต้องการที่จะแก้ไขเพื่อให้หุ่นยนต์หลบหลีกแบบใดเมื่อเจอสิ่งกีดขวาง จะต้องไปแก้ไขที่ตัวโปรแกรมของหุ่นยนต์ ซึ่งเป็นการไม่สะดวก และเป็นเรื่องยากสำหรับการใช้งาน



รูปที่ 5.12 หุ่นยนต์เดินหลบหลีกสิ่งกีดขวางภายในห้อง

ดังนั้นในวิทยานิพนธ์เรื่องนี้จึงสร้างโปรแกรมเพื่อช่วยเหลือการแก้ไขโปรแกรมการเดินของหุ่นยนต์ขึ้นมา เพื่อความสะดวกในการใช้งาน แม้ว่าผู้ใช้จะไม่มีความรู้เรื่องภาษาคอมพิวเตอร์ก็สามารถใช้งานได้

โปรแกรมที่สร้างขึ้นทำงานบนระบบปฏิบัติการ Dos ในการเรียกใช้โปรแกรมสามารถทำได้ด้วยการเรียกไฟล์ Roach.exe

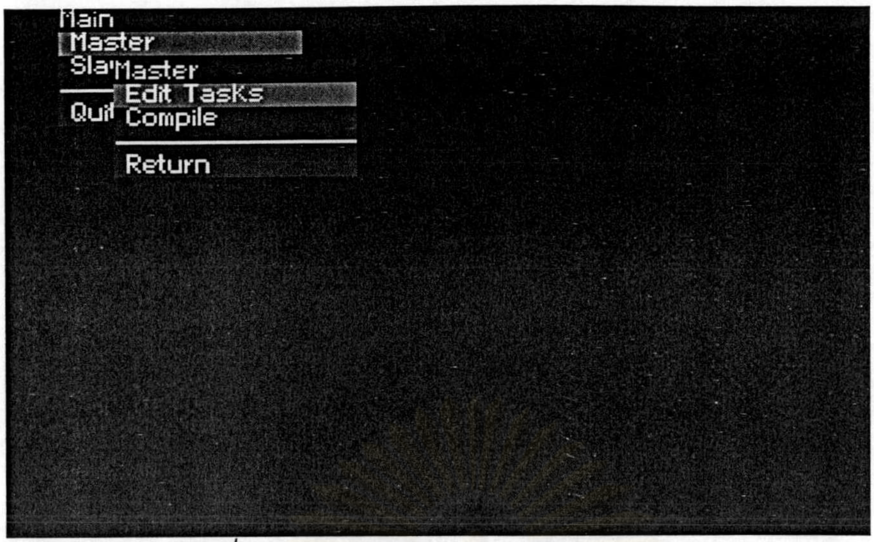
เมื่อเข้าสู่โปรแกรม จะแสดงเมนท์เมนูของโปรแกรม ดังรูปที่ 5.13 โดยมีเมนูย่อยคือ

- Master เลือกเพื่อต้องการโปรแกรมตัวประมวลผลหลัก
- Slave เลือกเพื่อต้องการโปรแกรมตัวประมวลผลของแต่ละขา
- Quit เลือกเพื่อยกเลิกโปรแกรม

รูปที่ 5.13 เมนท์เมนูของโปรแกรม

ในกรณีที่ต้องการโปรแกรมตัวประมวลผลหลัก ให้เลือกที่เมนู Master ของโปรแกรม ดังรูปที่ 5.14 โดยในเมนู Master ของโปรแกรมมีเมนูย่อยคือ

- Edit tasks ใช้กำหนดการกระทำของหุ่นยนต์หลังจากอุปกรณ์ตรวจรู้ของหุ่นยนต์ตัวใดตัวหนึ่งตรวจพบสิ่งกีดขวาง
- Compile ใช้ในการ Compile โปรแกรมของตัวประมวลผลหลักได้ผลลัพท์เป็น trip.hex ซึ่งนำไปเขียนบนตัวประมวลผลหลักต่อไป
- Return ใช้กลับไปสู่เมนูหลัก



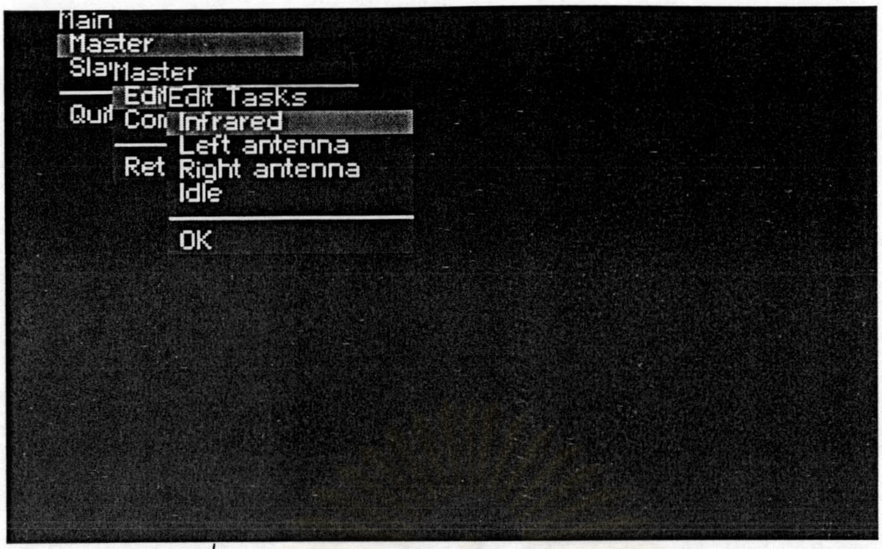
รูปที่ 5.14 ภายในเมนู Master ของโปรแกรม

เมนูย่อย Edit tasks ดังรูปที่ 5.15 ซึ่งอยู่ภายในเมนู Master ใช้ในการกำหนดการกระทำของหุ่นยนต์หลังจากอุปกรณ์ตรวจรู้ของหุ่นยนต์ตัวใดตัวหนึ่งตรวจพบสิ่งกีดขวาง

ภายในเมนู Edit tasks มีเมนูย่อยที่ใช้แทนอุปกรณ์ตรวจรู้แต่ละตัวของหุ่นยนต์คือ

- Infrared ใช้ในการกำหนดการกระทำของหุ่นยนต์เมื่ออินฟาเรดเซ็นเซอร์ซึ่งติดตั้งอยู่ด้านหน้าหุ่นยนต์ตรวจพบสิ่งกีดขวาง
- Left antenna ใช้ในการกำหนดการกระทำของหุ่นยนต์เมื่อลิมิตสวิตช์ซึ่งติดตั้งอยู่ด้านซ้ายของหุ่นยนต์ตรวจพบสิ่งกีดขวาง
- Right antenna ใช้ในการกำหนดการกระทำของหุ่นยนต์เมื่อลิมิตสวิตช์ซึ่งติดตั้งอยู่ด้านขวาของหุ่นยนต์ตรวจพบสิ่งกีดขวาง
- Idle ใช้ในการกำหนดการกระทำของหุ่นยนต์เมื่ออุปกรณ์ตรวจรู้ทุกตัวไม่พบสิ่งกีดขวาง

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 5.15 ภายในเมนู Edit Tasks ของโปรแกรม

ภายในแต่ละเมนูย่อยของเมนู Edit tasks ซึ่งแทนอุปกรณ์ตรวจรู้แต่ละตัวของหุ่นยนต์จะมีคำสั่งที่ใช้กำหนดการกระทำของหุ่นยนต์เมื่ออุปกรณ์ตรวจรู้ตัวนั้นๆตรวจพบสิ่งกีดขวาง ดังรูปที่ 5.16 คือ

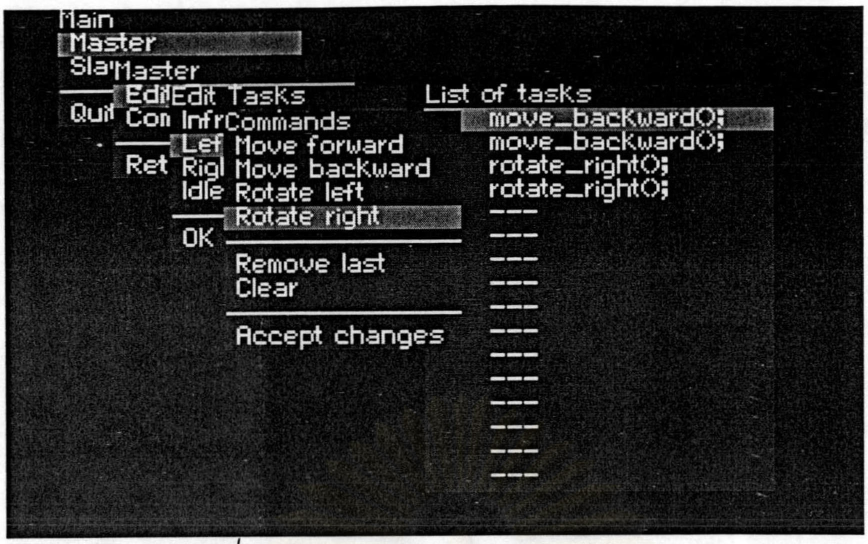
- Move forward เป็นคำสั่งในการเดินหน้า 1 ก้าว
- Move backward เป็นคำสั่งในการเดินถอยหลัง 1 ก้าว
- Rotate left เป็นคำสั่งในการเดินเลี้ยวซ้าย 1 ก้าว
- Rotate right เป็นคำสั่งในการเดินเลี้ยวขวา 1 ก้าว

คำสั่งต่างๆเหล่านี้เวลานำไปใช้จะถูกเรียงลำดับในเมนู List of tasks เพื่อเป็นการกำหนดว่าจะทำคำสั่งใดก่อน ยกตัวอย่างดังรูปที่ 5.16 ในเมนู Edit tasks เลือกไปที่ Left antenna จากรูปหมายความว่า เมื่อลิมิตสวิตช์ทางซ้ายของหุ่นยนต์ตรวจพบสิ่งกีดขวาง หุ่นยนต์จะถอยหลัง 2 ก้าว แล้วจึงเลี้ยวขวาอีก 2 ก้าวตามลำดับ

ส่วนคำสั่งที่เหลือเป็นคำสั่งที่ใช้แก้ไขข้อมูลในเมนู List of tasks คือ

- Remove last ใช้ในการลบคำสั่งหลังสุดในเมนู List of tasks
- Clear ใช้ในการลบคำสั่งทั้งหมดในเมนู List of tasks
- Accept changes ใช้ในการรับคำสั่งในเมนู List of tasks

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

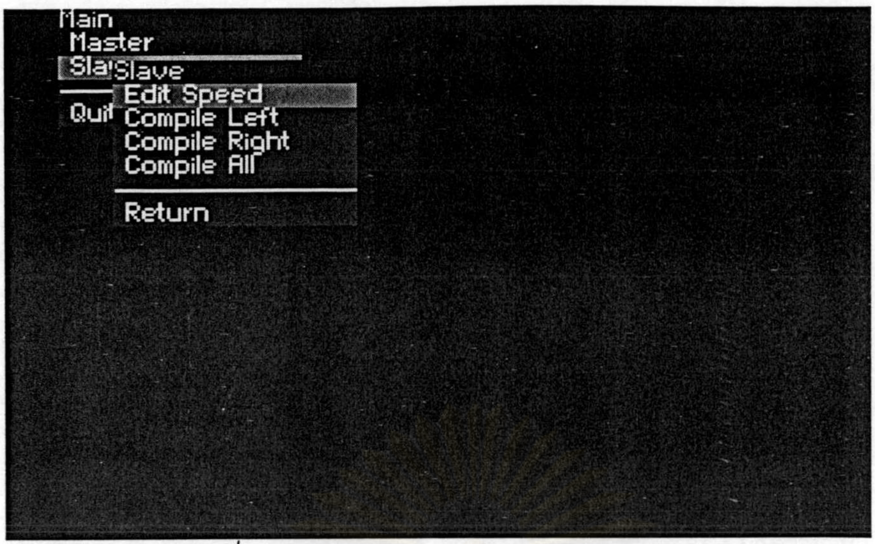


รูปที่ 5.16 เมนู List of tasks ของโปรแกรม

ในกรณีที่ต้องการโปรแกรมตัวประมวลผลของแต่ละขา ให้เลือกที่เมนู Slave ของโปรแกรม ดังรูปที่ 5.17 โดยในเมนู Slave ของโปรแกรมมีเมนูย่อยคือ

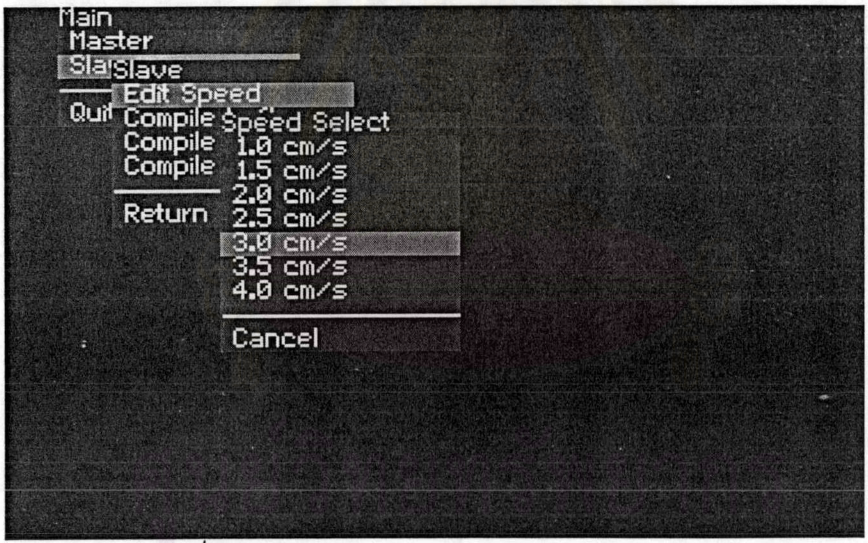
- Edit Speed ใช้ในการกำหนดความเร็วของหุ่นยนต์
- Compile Left ใช้ในการ Compile โปรแกรมของตัวประมวลผลของขาหุ่นยนต์ด้านซ้ายได้ผลลัพท์เป็น lside.hex ซึ่งนำไปเขียนบนตัวประมวลผลของขาหุ่นยนต์ด้านซ้ายต่อไป
- Compile Right ใช้ในการ Compile โปรแกรมของตัวประมวลผลของขาหุ่นยนต์ด้านขวาได้ผลลัพท์เป็น rside.hex ซึ่งนำไปเขียนบนตัวประมวลผลของขาหุ่นยนต์ด้านขวาต่อไป
- Compile All ใช้ในการ Compile โปรแกรมของตัวประมวลผลของขาหุ่นยนต์ด้านซ้ายและขวาพร้อมกันได้ผลลัพท์เป็น lside.hex และ rside.hex ซึ่งนำไปเขียนบนตัวประมวลผลของขาหุ่นยนต์ทั้ง 2 ด้านต่อไป
- Return ใช้กลับไปสู่เมนูหลัก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



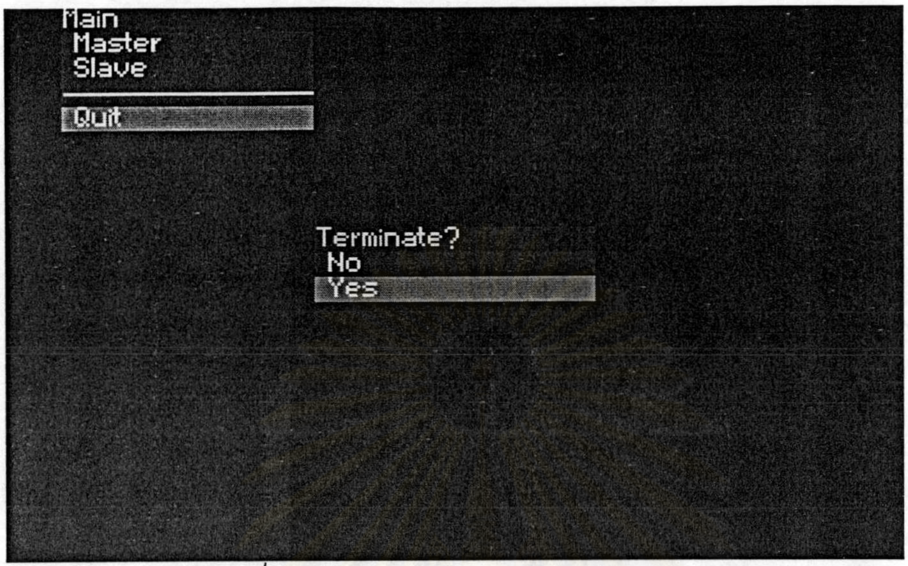
รูปที่ 5.17 ภายในเมนู Slave ของโปรแกรม

เมนูย่อย Edit Speed ดังรูปที่ 5.18 ซึ่งอยู่ภายในเมนู Slave ใช้ในการกำหนดความเร็วของหุ่นยนต์ โดยสมารถปรับความเร็วได้ 7 ระดับ



รูปที่ 5.18 ภายในเมนู Edit Speed ของโปรแกรม

เมนู Quit ใช้เมื่อต้องการออกจากโปรแกรมดังรูปที่ 5.19



รูปที่ 5.19 ภายในเมนู Quit ของโปรแกรม

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย