บทความวิจัย

ระบบการจัดการหุ่นยนต์หลายตัวสำหรับการขนถ่ายวัสดุในโรงงานอุตสาหกรรม

พงศกร ชาญชัยชูจิต และ พฤทธิกร สมิตไมตรี*
ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์

* ผู้นิพนธ์ประสานงาน โทรศัพท์ 0 7428 7214 อีเมล: pruittikorn.s@psu.ac.th DOI: 10.14416/j.kmutnb.2019.02.003
รับเมื่อ 9 กันยายน 2561 แก้ไขเมื่อ 14 พฤศจิกายน 2561 ตอบรับเมื่อ 18 ธันวาคม 2561 เผยแพร่ออนไลน์ 25 กุมภาพันธ์ 2562
© 2019 King Mongkut's University of Technology North Bangkok. All Rights Reserved.

บทคัดย่อ

ปัจจุบันการขนถ่ายวัสดุในโรงงานอุตสาหกรรมมีการนำหุ่นยนต์ AGV มาใช้กันอย่างแพร่หลาย ซึ่งพบว่าส่วนใหญ่มักนำหุ่นยนต์ AGV หนึ่งตัวมาใช้ในการขนถ่ายวัสดุเพียงหนึ่งชนิด จากสถานีงานต้นทางไปยังสถานีงานปลายทาง และทำซ้ำแบบเดิมตลอดเวลา งานวิจัยนี้จึงนำเสนอการออกแบบระบบจัดการหุ่นยนต์หลายตัวสำหรับขนถ่ายวัสดุในโรงงานอุตสาหกรรม โดยใช้โปรแกรม ROS (Robot Operating System) เพื่อพัฒนาศักยภาพระบบขนถ่ายวัสดุ เพิ่มความยืดหยุ่น ลดเวลาในการทำงานของหุ่นยนต์ ด้วยวิธี Market-based Approach ร่วมกับ Dijkstra's Algorithm สำหรับการขนถ่ายวัสดุในโรงงานอุตสาหกรรม ผลการจำลองการทำงานพบว่า ระบบสามารถควบคุมหุ่นยนต์ AGV ต่างชนิดกันได้ สามารถเลือกสั่งการหุ่นยนต์ขนถ่ายวัสดุได้คุ้มค่าที่สุดจากการประเมินงานของหุ่นยนต์ทั้งหมดในระบบ ณ สภาวะปัจจุบันของการทำงาน ระบบจัดการนี้สามารถลดเวลาการทำงานได้ครึ่งหนึ่งเมื่อเทียบกับวิธีแบบ FIFO นอกจากนี้ยังพบว่าการเลือกจำนวนหุ่นยนต์ให้กับระบบการจัดการหุ่นยนต์หลายตัวส่งผลต่อความเร็วในการขนถ่ายวัสดุอีกด้วย ผลการจำลองพื้นที่ทำงานขนาด 65 ตารางเมตร แสดงให้เห็นว่าหุ่นยนต์ 3 ตัว สามารถขนถ่ายวัสดุ 100 ชุดคำสั่งงานได้เร็วที่สุด งานวิจัยนี้แสดงให้เห็นว่าระบบสามารถจัดการงานสำหรับการขนถ่ายวัสดุโดยเลือกหุ่นยนต์ทำงานให้มีรอบเวลาทำงาน (Cycle Time) ได้เร็วที่สุด สามารถควบคุมหุ่นยนต์ที่มีลักษณะแตกต่างกันทำให้เกิดความยืดหยุ่นในการขนถ่ายวัสดุ ซึ่งสามารถนำไปใช้ขนถ่ายวัสดุในโรงงานอุตสาหกรรมที่ใช้หุ่นยนต์หลายตัวในการขนถ่ายวัสดุหลายชนิดจากสถานีเริ่มต้นไปยังสถานีปลายทางได้มากกว่าหนึ่งสถานี

คำสำคัญ: ระบบหุ่นยนต์หลายตัว วิธีการเปรียบเทียบราคาตลาด ขั้นตอนวิธีของไดก์สตรา การขนถ่ายวัสดุ

การอ้างอิงบทความ: พงศกร ชาญชัยชูจิต และ พฤทธิกร สมิตไมตรี, "ระบบการจัดการหุ่นยนต์หลายตัวสำหรับการขนถ่ายวัสดุในโรงงานอุตสาหกรรม," *วารสารวิชาการพระจอมเกล้าพระนครเหนือ*, ปีที่ 29, ฉบับที่ 3, หน้า 454–464, ก.ค.–ก.ย. 2562.

Research Article

Management of a Multi-robots System for Industrial Material Handling

Pongsakorn Chanchaichujit and Pruittikorn Smithmaitrie*
Department of Mechanical Engineering, Faculty of Engineering, Prince of Songkla University, Songkhla, Thailand

* Corresponding Author, Tel. 0 7428 7214, E–mail: pruittikorn.s@psu.ac.th
DOI: 10.14416/j.kmutnb.2019.02.003
Received 9 September 2018; Revised 14 November 2018; Accepted 18 December 2018; Published online: 25 February 2019
© 2019 King Mongkut's University of Technology North Bangkok. All Rights Reserved.

Abstract

Recently, material handling of manufacturing factories tends to widely adopt AGV robots. However, AGV robots mostly handle one-by-one material from a start station to a final station, repeatedly. This research introduces a multi-robots material handling system that can handle multiple robots and stations at the same time. The control program is based on ROS (Robot Operating System). The objective of this research is to improve the effectiveness of material handling system by adding more flexibility and reducing the robot work time. The management system uses the market-based approach combined with Dijkstra's algorithm for material handling in a manufacture factory. The simulation results found that the system is able to control various types of AGV robots. This means that the system has ability to order and execute the best robot among all working robots in the system for an incoming task at the current situation. This reduces the total working time by a half comparing with FIFO method. In the case study of the 65-m^2 working area, the optimum number of the robots is 3 to execute 100 tasks. The research results shown that the system can arrange materials handling by choosing the suitable robots for overall fastest cycle time. This system increases flexibility in handling materials in various working spaces. The knowledge of this research can be applied on many factories, which use multiple robots for various tasks to hand materials from start stations to final stations.

Keywords: Multi-robots System, Market-based Approach, Dijkstra's Algorithm, Material Handling

Please cite this article as: P. Chanchaichujit and P. Smithmaitrie, "Management of a multi-robots system for industrial material handling," *The Journal of KMUTNB*, vol. 29, no. 3, pp. 454–464, Jul.–Sep. 2019 (in Thai).

1. บทนำ

หุ่นยนต์ AGV (Automated Guided Vehicles) ถูกนำมาใช้ช่วยทำงานแทนคน ทั้งแบบอัตโนมัติ แบบเดินตามแถบแม่เหล็ก หรือแบบใช้สัญญาณไร้สายในการระบุตำแหน่งของหุ่นยนต์ [1] ลักษณะการนำหุ่นยนต์มาทำงานร่วมในกระบวนการผลิตมักจะให้หุ่นยนต์ทำงานตามลำดับขั้นตอน FIFO (First In First Out) กล่าวคือ งานที่มาก่อนทำก่อน และงานที่มาหลังทำทีหลัง จนกระทั่งถึงงานลำดับสุดท้าย ซึ่งไม่ได้คำนึงถึงการเพิ่มประสิทธิภาพการขนถ่ายวัสดุ หากในระบบเพิ่มจำนวนหุ่นยนต์มาใช้ในการขนถ่ายวัสดุมากขึ้น จะทำให้ประสิทธิภาพที่ได้ลดลง เนื่องจากขาดการวิเคราะห์เวลาที่ใช้ในการทำงานแต่ละงาน ซึ่งหากวิเคราะห์การจัดการงานของหุ่นยนต์จะพบว่า ยังสามารถเพิ่มการจัดการงานที่เหมาะสมเพื่อช่วยลดเวลาการทำงานของการขนย้ายวัสดุในภาพรวมได้อีก [2]

งานวิจัยเกี่ยวกับระบบการจัดการหุ่นยนต์หลายตัวถูกนำมาใช้เพื่อช่วยจัดการระบบให้มีประสิทธิภาพเพิ่มขึ้นในหลายๆ ด้าน เช่น การควบคุมกลยุทธ์สำหรับหุ่นยนต์หลายตัวส่งผลต่อการช่วยลดเวลาในการทำงาน [3] ปัญหาที่มักพบสำหรับวิธีการจัดเก็บและขนส่งวัตถุดิบทั่วไปอย่าง FIFO คือ ข้อจำกัดเรื่องการใช้เวลารวมมากเกินไปสำหรับการจัดการขนส่งหลายชุดคำสั่ง เพราะไม่ได้เปรียบเทียบและเลือกใช้หุ่นยนต์ที่เหมาะสมกับงานที่สุด ยกตัวอย่างเช่น หุ่นยนต์ที่ว่างงานอยู่ใกล้กับสถานีงานคำสั่งที่สองมากที่สุด แต่ต้องไปรับงานคำสั่งแรกตามวิธี FIFO ทำให้เสียเวลาในการเดินทาง เป็นต้น งานวิจัยนี้จึงมุ่งเน้นการออกแบบระบบการจัดการหุ่นยนต์หลายตัวสำหรับขนถ่ายวัสดุด้วยวิธี Market-based Approach โดยวิธีการนี้จะแปลงระบบเส้นทางให้อยู่ในรูปแบบกริด จุดปลายของเส้นทางเปรียบเสมือนสินค้าที่มีการแลกเปลี่ยนอยู่ในตลาด และระยะทางระหว่างจุดเริ่มต้นไปยังจุดปลายทางเปรียบเสมือนต้นทุนที่ใช้ กำไรจะขึ้นอยู่กับข้อมูลต่างๆ ที่ใช้ประกอบไปจนถึงจุดปลายทาง งานวิจัยนี้ใช้เวลาในการทำงานของหุ่นยนต์เป็นตัวชี้วัดเพื่อเลือกงานให้เหมาะสมกับตำแหน่งปัจจุบันของหุ่นยนต์ โดยหุ่นยนต์ทุกตัวจะมีเส้นทางการเดินที่สั้นที่สุดซึ่งในกรณีนี้วิธี Market-based Approach ก็คือการเลือกหุ่นยนต์ที่ใช้เวลาน้อยที่สุดไปทำงาน

วิธีหาเส้นทางแบบกริดที่สั้นที่สุดนั้นมีอยู่หลายอัลกอริทึม งานวิจัยนี้ได้เลือกใช้ขั้นตอนวิธีของไดก์สตรา (Dijkstra's Algorithm) เพราะเข้ากับรูปแบบของระบบขนส่งวัสดุด้วยหุ่นยนต์หลายตัว รวมถึงเวลาที่ใช้ในการประมวลหาระยะทางที่สั้นที่สุดก็อยู่ในระดับที่ไม่นานจนเกินไป และเมื่อมีจำนวนโหนดในการประมวลผลมากขึ้น ระยะเวลาในการคำนวณก็ไม่ได้เพิ่มขึ้นมากนัก เมื่อทำการเปรียบเทียบกับอัลกอริทึมเบลแมน-ฟอร์ด [4] ผลลัพธ์จากการหาระยะทางที่สั้นที่สุดเมื่อเทียบกับ Genetic Algorithm (GA) มีค่าเท่ากัน [5] สำหรับวิธีของไดก์สตรานั้นเป็นวิธีการหาเส้นทางที่สั้นที่สุด โดยมีงานวิจัยที่นำวิธีการเลือกหุ่นยนต์ที่ใช้เวลาน้อยที่สุดไปทำงาน ร่วมกับอัลกอริทึมการหาเส้นทางที่สั้นที่สุด เช่น การจัดสรรงานหุ่นยนต์หลายตัวสำหรับค้นหาและช่วยเหลือ [6] โดยใช้ค่าน้ำหนักระหว่างโหนด เมื่อมีการนำค่าน้ำหนักระหว่างโหนดแทนด้วยระยะทางระหว่างโหนด จึงสามารถนำมาใช้ในการหาระยะทางที่สั้นที่สุดของหุ่นยนต์ได้ [7] แต่ระยะทางที่ใช้ในการเดินทางที่สั้นที่สุดนั้นไม่เพียงพอต่อการนำไปตัดสินใจเลือกหุ่นยนต์ งานวิจัยนี้จึงนำเรื่องของเวลาขนถ่ายวัสดุมาช่วยในการตัดสินใจ เพราะทำให้เงื่อนไขการตัดสินใจนั้นขึ้นอยู่กับตัวแปรสองตัวคือ ความเร็วในการทำงาน และระยะทางของหุ่นยนต์

ในการพัฒนาระบบจัดการหุ่นยนต์หลายตัวพบว่ามีความยุ่งยากในเรื่องของการสื่อสาร การจัดการทรัพยากรของระบบจัดการหุ่นยนต์ การสื่อสารผ่านระบบเครือข่าย รวมถึงหากต้องการควบคุมหุ่นยนต์ที่แต่ละตัวมีลักษณะแตกต่างกันยิ่งเพิ่มความซับซ้อนในระบบควบคุมมากยิ่งขึ้น แต่ในปัจจุบันมีการพัฒนาซอฟต์แวร์ที่ใช้ในการสื่อสารกันแบบ Machine-to-machine นั่นคือ ROS (Robot Operating System) ข้อดีของโปรแกรม ROS คือ มีความยืดหยุ่นในการควบคุมหุ่นยนต์ต่างชนิดหลายตัวพร้อมกันได้ รวมถึงเป็นโอเพนซอร์สที่สามารถใช้งานได้ฟรี และถูกขยายผลไปใช้งานในอุตสาหกรรมมากขึ้นในปัจจุบัน [8]

ขั้นตอนวิธีของไดก์สตรากำหนดให้ G คือ กราฟที่มีค่าถ่วงน้ำหนัก โดยกำหนดให้ทุกทางมีค่าถ่วงน้ำหนักเป็นบวก G ประกอบด้วยจุดต่างๆ คือ $v_0, v_1,..., v_n$ และมีค่าถ่วงน้ำหนักเท่ากับ $w(u_i, v_j)$ เมื่อ u_i คือ พิกัดปัจจุบัน และ V_j คือ พิกัดถัดไป โดยให้ a เป็นจุดเริ่มต้น มี z เป็นจุดปลายทาง กำหนดให้ S คือ เซตของจุดซึ่งเก็บค่าจุดที่เคยเดินผ่านมาแล้ว [9]

ขั้นตอนของ Dijkstra's Algorithm

For $i = 1$ to n

$L(v_1) = \infty$

$L(a) = 0$

$S = \varnothing$

While $z \notin S$

Begin

U = จุดที่ไม่ได้อยู่ใน S และมีระยะทางน้อยที่สุดจาก $L(u)$

$S = S \cup \{u\}$

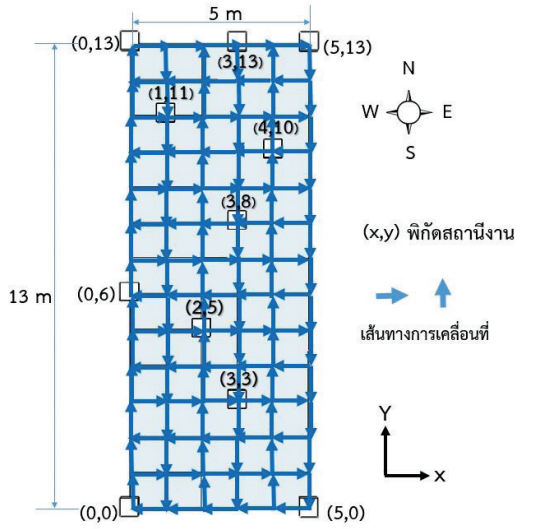
For ทุกจุดของ v ที่ไม่ได้อยู่ใน S

If $L(u) + W(u,v) < L(v)$ *Then*

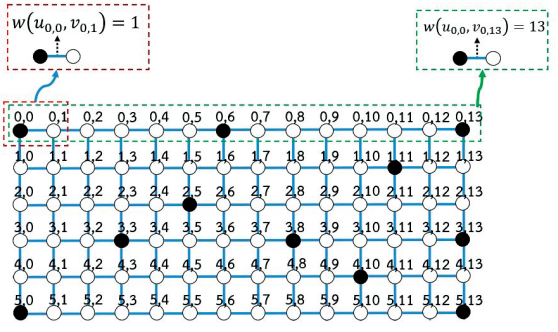
$L(v) = L(u) + W(u,v)$

Return $L(z)$

END



รูปที่ 1 แผนที่จำลองในโรงงานอุตสาหกรรมสำหรับทดสอบการจัดการงาน



รูปที่ 2 แผนภูมิที่สร้างจากขั้นตอนวิธีของไดก์สตรา

2. วิธีการวิจัย

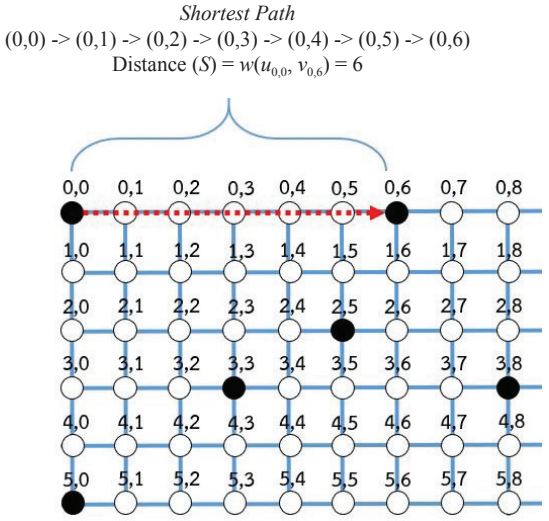
สำหรับขั้นตอนวิธีของไดก์สตรานั้นถูกนำมาใช้ในการหาผลรวมของเส้นทางที่สั้นที่สุดของหุ่นยนต์จากสถานีเริ่มต้นไปยังสถานีปลายทาง วิธีการหาเส้นทางที่สั้นที่สุดหาจากแผนที่แบบตารางแสดงดังรูปที่ 1

งานวิจัยนี้จะใช้แผนที่ดังรูปที่ 1 ในการทดสอบระบบจัดการงานให้กับหุ่นยนต์ ซึ่งขนาดของแผนที่มีขนาดกว้าง 5 เมตร ยาว 13 เมตร ประกอบด้วยสถานีงานทั้งหมด 11 สถานี สำหรับค่าถ่วงน้ำหนักที่ใช้แทนในการขั้นตอนวิธีของไดก์สตรา แทนด้วยค่าระยะห่างของจุดแต่ละจุดเท่ากับ 1 เมตร และแทนจุดแต่ละจุดบนเส้นทางด้วยพิกัด (x, y)

เมื่อใช้ขั้นตอนวิธีของไดก์สตรา จะได้แผนผังสำหรับการหาเส้นทางที่สั้นที่สุดดังแสดงในรูปที่ 2 ทดสอบหาระยะทางที่สั้นที่สุดจากสถานีพิกัด (0, 0) ไปยังพิกัด (0, 6) เมื่อจุดดำแทนสถานีงาน และจุดขาวแทนตำแหน่งเส้นทางเดิน ผลจากทดสอบแสดงได้ดังรูปที่ 3 จะได้ระยะทางในการเดินทางจากสถานีพิกัด (0, 0) ไปสถานีพิกัด (0, 6) เป็นระยะทาง 6 เมตร โดยผ่านเส้นทางพิกัด (0, 1), (0, 2), (0, 3), (0, 4), (0, 5) ตามลำดับ

3. ระบบการจัดการงาน

ระบบการจัดการงานสำหรับหุ่นยนต์หลายตัวแบ่งออกเป็นสองประเภท คือ ระบบศูนย์กลางหุ่นยนต์หลายตัว



รูปที่ 3 เส้นทางที่สั้นที่สุดจากสถานีพิกัด (0, 0) ไปยังสถานีพิกัด (0, 6)

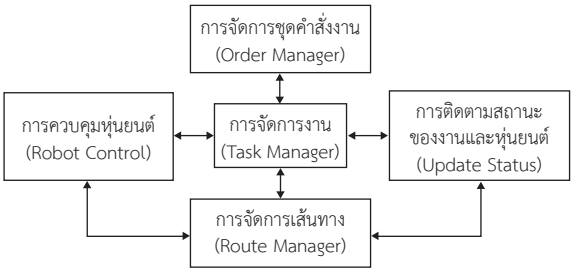
(Centralized Multi-robot System) และระบบไม่มีศูนย์กลางหุ่นยนต์หลายตัว (Decenrtalized Multi-robot System) งานวิจัยนี้เลือกใช้แบบระบบศูนย์กลางหุ่นยนต์หลายตัว เพราะสามารถควบคุมและจัดการได้ง่ายเหมาะสำหรับสภาวะแวดล้อมเป็นปกติไม่เปลี่ยนแปลง จำนวนหุ่นยนต์คงที่ สำหรับระบบการจัดการงานของงานวิจัยนี้แบ่งการจัดการงานออกเป็น 5 ส่วน คือ การจัดการชุดคำสั่งงาน การติดตามสถานะของงานและหุ่นยนต์ การจัดการงาน การควบคุมหุ่นยนต์ และการจัดการเส้นทาง ดังรูปที่ 4

3.1 การจัดการชุดคำสั่งงาน (Order Manager)

เป็นส่วนที่ให้ผู้ใช้ป้อนชุดคำสั่งงานเข้าสู่ระบบ เพื่อให้ระบบทำหน้าที่จัดการงาน

3.2 การติดตามสถานะของงานและหุ่นยนต์ (Update Status)

เป็นส่วนที่จัดการในเรื่องของการอัปเดทสถานะของหุ่นยนต์และชุดคำสั่งงานที่หุ่นยนต์แต่ละตัวกำลังทำงานอยู่ เพื่อให้ส่วนของการจัดการงานสามารถตรวจสอบสถานะของงานและสถานะของหุ่นยนต์ ณ เวลาปัจจุบันได้



รูปที่ 4 ภาพรวมของระบบจัดการงาน

3.3 การจัดการงาน (Task Manager)

เป็นส่วนหลักที่ใช้ในการคำนวณเวลาสำหรับชุดคำสั่งงานและตัดสินใจเลือกหุ่นยนต์ที่ทำงานได้รวดเร็วที่สุด สำหรับการประเมินค่าเวลาการทำงานของหุ่นยนต์จะเริ่มคำนวณหลังจากที่ได้เส้นทางที่สั้นที่สุดในการเคลื่อนที่ของหุ่นยนต์แต่ละงานจากขั้นตอนของไดก์สตรา จากนั้นนำมาคำนวณหาเวลาการทำงานของหุ่นยนต์ในแต่ละงานได้จากสมการที่ (1)

$$T_{m,n} = \left(M \frac{S_n}{V_n} \right) + \left(N \frac{\theta_n}{\omega_n} \right) \quad (1)$$

เมื่อ T คือ เวลาในการทำงานของหุ่นยนต์
S คือ ผลรวมของระยะทางที่สั้นที่สุดของหุ่นยนต์
V คือ ความเร็วเชิงเส้นของหุ่นยนต์
θ คือ ผลรวมของมุมการเคลื่อนที่ของหุ่นยนต์
ω คือ ความเร็วเชิงมุมของหุ่นยนต์
n คือ หุ่นยนต์ตัวที่ n
m คือ งานที่ m
M คือ จำนวนครั้งของการเคลื่อนที่แบบเชิงเส้น
N คือ จำนวนครั้งของการเคลื่อนที่แบบเชิงมุม

สมการที่ (1) นำความเร็วของหุ่นยนต์ทั้งความเร็วเชิงเส้นและเชิงมุมที่ใช้ในการเคลื่อนที่มาแปลงให้อยู่ในรูปของเวลาเพื่อนำมาเลือกหุ่นยนต์ที่ใช้เวลาทำงานน้อยที่สุด

อย่างไรก็ตาม แผนที่ที่ใช้ในการทดลองนี้มีระยะห่างระหว่างจุดพิกัดเป็นระยะคงที่ ดังนั้นหากนำไปใช้ในการคำนวณเวลาสำหรับแผนผังที่มีระยะห่างของจุดพิกัดไม่คงที่จะต้องมีการปรับสมการให้ถูกต้องตามระยะห่างของจุดพิกัดด้วย

3.4 วิธีการเปรียบเทียบราคาตลาด (Market-based Approach)

วิธีการนี้เป็นหนึ่งในการประเมินมูลค่าของสินค้าโดยอาศัยการเปรียบเทียบข้อมูลต่างๆ เช่น ประเภทของสินค้า ขนาด ตำแหน่งที่ตั้ง ราคา ที่มีลักษณะคล้ายคลึงกันในด้านต่างๆ แล้วทำการวิเคราะห์เพื่อสรุปผลเป็นมูลค่าของสินค้านั้น การเปรียบเทียบราคาตลาดมีวิธีย่อยหลายวิธี เช่น วิธีการเปรียบเทียบโดยตรง (Direct Comparison) วิธีปรับเปลี่ยนราคาซื้อขาย (Grid Adjustment) วิธีคะแนนคุณภาพถ่วงน้ำหนัก (Weighted Quality Score; WQS) และวิธีสมการถดถอย (Regression Analysis) [10] มีหลายงานวิจัยนำวิธีการเหล่านี้มาใช้ในการจัดการหุ่นยนต์หลายตัว โดยการนำค่าข้อมูลตำแหน่ง ระยะห่างของหุ่นยนต์ และเวลาการทำงานของหุ่นยนต์แต่ละตัวมาเปรียบเทียบกันด้วยการ Bidding ระหว่างหุ่นยนต์ แล้วใช้วิธีเลือกค่าดัชนีกำหนดแบบน้อยที่สุดหรือมากที่สุดอย่างใดอย่างหนึ่ง เพื่อเลือกหุ่นยนต์ที่เหมาะสมกับงาน [11]–[13]

เวลาทำงานของหุ่นยนต์ที่หาได้จากสมการที่ (1) ถูกนำมาใช้ในการ Bidding เพื่อเลือกงานที่เหมาะสมกับหุ่นยนต์ด้วยระยะทางที่สั้นที่สุดและการทำงานเร็วที่สุด โดยพิจารณาตัวแปร 3 ตัว คือ ระยะทางจากตำแหน่งปัจจุบันของหุ่นยนต์ ความเร็วเชิงเส้น และความเร็วเชิงมุมของหุ่นยนต์ มาประเมินเป็นเวลาที่ใช้ในการทำงานของหุ่นยนต์ตามสมการที่ (1) จะได้เซตเวลาของหุ่นยนต์แต่ละตัวในระบบที่ใช้ในแต่ละงานตามสมการที่ (2) ซึ่งเป็นการแสดงเวลาที่หุ่นยนต์แต่ละตัวต้องใช้ทำงานในแต่ละงานจากตำแหน่งปัจจุบันที่หุ่นยนต์อยู่

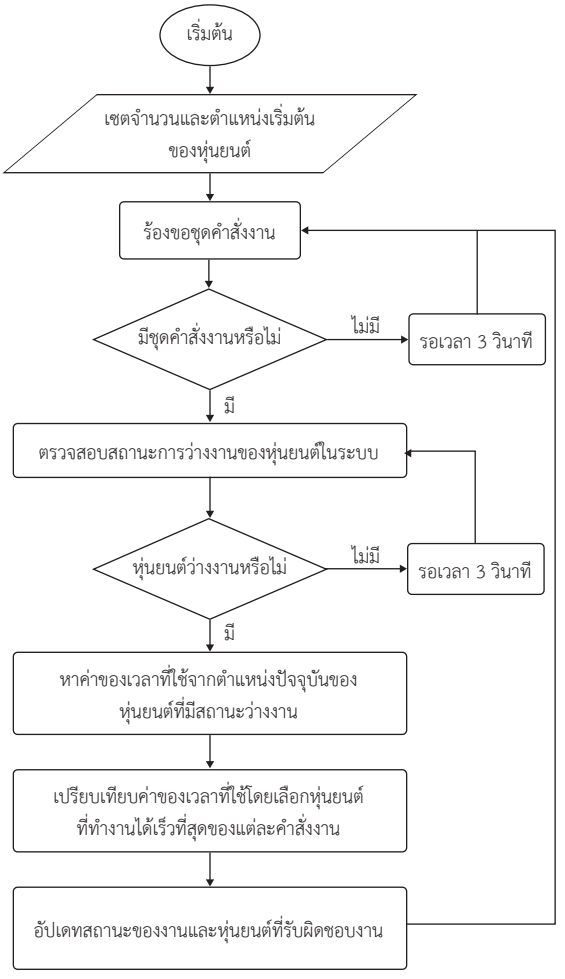
$$J_m = \{R_n : T_m\} \tag{2}$$

เมื่อ J คือ เซตของงาน $\{J_1, J_2, \ldots, J_m\}$

R คือ หุ่นยนต์ในระบบ

T คือ เวลาที่ถูกประเมินสำหรับงาน

จากนั้นนำเซตเวลาการทำงานของหุ่นยนต์ดังสมการที่ (2) มาเปรียบเทียบหาเวลาที่น้อยที่สุด สำหรับเลือกงานให้กับหุ่นยนต์ด้วยการนำค่าเวลาในการทำงานของหุ่นยนต์แต่ละตัวมา Bidding สำหรับชุดงานนั้นๆ ดังสมการที่ (3)



รูปที่ 5 ขั้นตอนการทำงานของการจัดการงาน

$$Task_i = \min(J_m) \tag{3}$$

เมื่อ i คือ ลำดับการมอบหมายงาน

เมื่อสามารถทำการ Bidding งานให้กับหุ่นยนต์ได้แล้ว ระบบการจัดการจะทำการมอบหมายงานนั้นๆ ให้กับหุ่นยนต์ตัวที่เหมาะสม และประเมินเวลางานใหม่ที่เข้ามาให้กับหุ่นยนต์ที่ว่างงานอยู่กระทำซ้ำเช่นนี้จนมอบหมายงานที่มีอยู่ทั้งหมดให้หุ่นยนต์ทำงานจนเสร็จสิ้น ขั้นตอนการทำงานของการจัดการงานแสดงดังรูปที่ 5

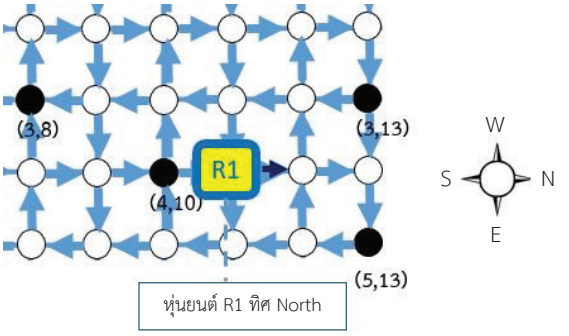
3.5 การควบคุมหุ่นยนต์ (Robot Control)

เมื่อระบบควบคุมและจัดการงานมีหุ่นยนต์ที่แตกต่างกันทำงานร่วมกัน จะส่งผลถึงการควบคุมการเคลื่อนที่ของหุ่นยนต์ที่มีความแตกต่างกัน เช่น ระบบของล้อหุ่นยนต์เป็นแบบ Differential Drive [14] หรือแบบ Mecanum Drive [15] ทำให้ระบบสั่งงานการเคลื่อนที่ของหุ่นยนต์มีตัวแปรที่ต่างกันและมีความซับซ้อนมากขึ้น เพื่อให้ง่ายต่อการสั่งการของระบบการจัดการ จึงอาศัยการส่งคำสั่งรูปแบบการเคลื่อนที่แทนการส่งตำแหน่งปลายทางให้กับหุ่นยนต์ โดยส่วนการควบคุมการเคลื่อนที่จะทำหน้าที่บังคับการเคลื่อนที่ของหุ่นยนต์ตัวนั้นแทน เพื่อให้ตัวระบบการจัดการงานกับการควบคุมการเคลื่อนที่ของหุ่นยนต์ถูกแยกออกจากกันแต่ยังสามารถทำงานร่วมกันได้ และสร้างการควบคุมหุ่นยนต์แต่ละตัวให้มีฟังก์ชันการเคลื่อนที่ตามระบบขับเคลื่อนล้อของตนเอง โดยมีลักษณะฟังก์ชันการเคลื่อนที่ของหุ่นยนต์ทั้งหมด 3 รูปแบบ คือ Forward (เดินหน้า) Left (หุ่นยนต์เลี้ยวซ้ายเป็นมุม 90 องศา) Right (หุ่นยนต์เลี้ยวขวาเป็นมุม 90 องศา) และ Turnaround (หุ่นยนต์หมุนตามเข็มนาฬิกาเป็นมุม 180 องศา) ระบบการจัดการหุ่นยนต์จะใช้ข้อมูลตำแหน่งปัจจุบันแทนด้วย (x_i, y_i) ตำแหน่งถัดไปแทนด้วย (x_j, y_j) และทิศทางของหุ่นยนต์แทนด้วย North, East, West, South จากนั้นหาความสัมพันธ์ของตำแหน่งถัดไปและตำแหน่งปัจจุบัน โดยการหาค่า x และ y จากสมการที่ (4) และ (5) ตามลำดับ

$$x = x_j - x_i \quad (4)$$

$$y = y_j - y_i \quad (5)$$

แล้วนำค่าที่ได้ไปเทียบกับตารางที่ 1 เพื่อแปลงเส้นทางการเคลื่อนที่ที่สั้นที่สุดจากขั้นตอนของไดก์สตรา เป็นรูปแบบการเคลื่อนที่ของหุ่นยนต์ ตัวอย่างเช่น จากรูปที่ 6 หุ่นยนต์ R1 ตำแหน่งปัจจุบัน คือ (4,11) ทิศ North มีพิกัดเป้าหมายตำแหน่งถัดไป คือ (4,12) หาค่า x และ y จากสมการที่ (4) และ (5) จะได้ว่า x = 0, y = 1 เมื่อนำค่าไปแทนในตารางที่ 1 จะได้ชุดคำสั่งการเคลื่อนที่ คือ Forward เป็นต้น



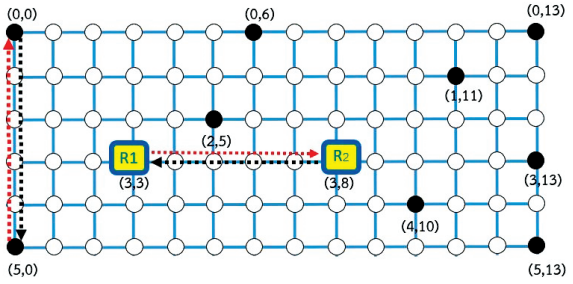
รูปที่ 6 การระบุทิศและการเคลื่อนที่ของหุ่นยนต์เทียบกับแผนที่

ตารางที่ 1 การแปลงเส้นทางเป็นรูปแบบการเคลื่อนที่

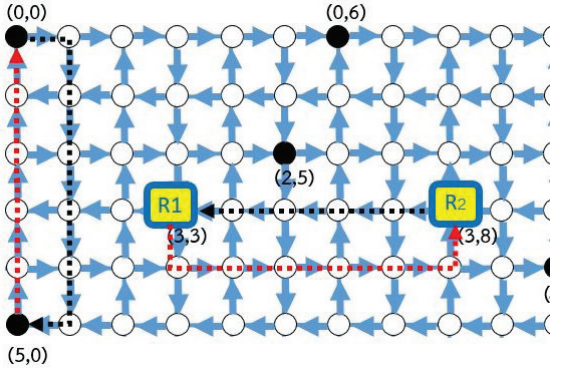
ทิศทางของหุ่นยนต์ ณ ตำแหน่งปัจจุบัน	ความสัมพันธ์ของตำแหน่งถัดไปกับตำแหน่งปัจจุบันของหุ่นยนต์		คำสั่งการเคลื่อนที่
	x	y	
North	0	1	Forward
	0	–1	Turnaround and Forward
	–1	0	Left and Forward
	1	0	Right and Forward
East	0	1	Left and Forward
	0	–1	Right and Forward
	–1	0	Turnaround and Forward
	1	0	Forward
West	0	1	Right and Forward
	0	–1	Left and Forward
	–1	0	Forward
	1	0	Turnaround and Forward
South	0	1	Turnaround and Forward
	0	–1	Forward
	–1	0	Right and Forward
	1	0	Left and Forward

3.6 การจัดการเส้นทางของหุ่นยนต์ (Route Manager)

เมื่อหุ่นยนต์ได้รับงานที่มอบหมายแล้ว อีกเหตุการณ์หนึ่งที่จะเกิดขึ้น คือ ระบบจัดการงานหาเส้นทางที่สั้นที่สุดได้ผลเป็นเส้นทางที่ทับซ้อนกันของหุ่นยนต์สองตัว ทำให้เกิดปัญหาการจราจรขณะทำงาน ตัวอย่างเช่นในรูปที่ 7 เมื่อหุ่นยนต์ตัวที่หนึ่งต้องการนำสินค้าจากสถานีพิกัด (3, 3) ไปยังสถานี



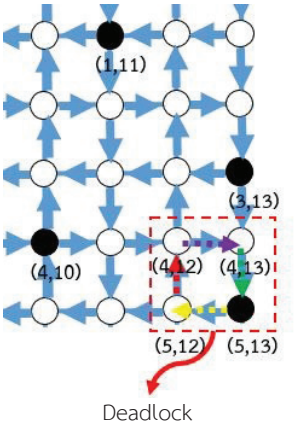
รูปที่ 7 ปัญหาจราจรเมื่อหุ่นยนต์สองตัวทำงานบนเส้นทางที่ทับกัน



รูปที่ 8 การแก้ปัญหาการจราจรด้วยการกำหนดเส้นทางเดินทิศทางเดียว

พิกัด (3, 8) และหุ่นยนต์ตัวที่สองต้องการนำสินค้าจากพิกัด (3, 8) ไปส่งยังพิกัด (3, 3) ผลคำนวณเส้นทางที่สั้นที่สุดของหุ่นยนต์ทั้งสองตัวพบว่าเกิดเส้นทางทับซ้อนกันบนเส้นทางเดียวกัน จึงทำให้เกิดปัญหาการจราจรขึ้น

งานวิจัยนี้ใช้วิธีการแก้ปัญหาด้วยการสร้างเส้นทางเดินแบบทางเดียว (Unidirectional Path) [16] โดยทุกจุดบนแผนที่แบบตารางจะต้องมีทางเข้าและออกอย่างน้อย 1 ทาง และมีจำนวนทางเดินในแนวกว้างและยาวเป็นจำนวนคู่เสมอ หากเกิดกรณีหุ่นยนต์ต้องใช้จุดถัดไปร่วมกัน หุ่นยนต์ที่ได้รับคำสั่งเข้าถึงการจองจุดถัดไปก่อน จะมีสิทธิ์จองพื้นที่นั้นเพื่อใช้เส้นทาง หุ่นยนต์อีกตัวจะต้องหยุดรอจนกว่าจุดถัดไปนั้นจะว่าง แล้วจึงทำการจองเพื่อใช้เส้นทางนั้นในลำดับถัดไป ด้วยเงื่อนไขนี้เส้นทางใหม่ที่ได้จากการกำหนดเส้นทางแบบเดินทางเดียวจะได้ผลการคำนวณเส้นทางดังรูปที่ 8



รูปที่ 9 เหตุการณ์ Deadlock ของหุ่นยนต์ 4 ตัวที่ใช้เส้นทางพร้อมกัน

นอกจากนี้ระหว่างทำการทดลองได้พบกรณีของ Deadlock เมื่อหุ่นยนต์สี่ตัวจองใช้เส้นทางพร้อมกัน ดังรูปที่ 9 โดยหุ่นยนต์ 4 ตัว มีการใช้จุดจราจรร่วมกัน คือ หุ่นยนต์ตัวที่หนึ่งต้องการเคลื่อนตัวจาก (4, 12) ไปยังจุด (4, 13) หุ่นยนต์ตัวที่สองจากจุด (4, 13) ไปยังจุด (5, 13) หุ่นยนต์ตัวที่สามจาก (5, 13) ไปยัง (5, 12) และหุ่นยนต์ตัวที่สี่จากจุด (5, 12) ไปยัง (4, 12) กรณีนี้ Manca *et al*. [17] ได้แสดงวิธีการแก้ปัญหาไว้แล้ว กล่าวคือ ระบบจัดการเส้นทางจะทำการตรวจสอบการใช้พิกัดร่วมกันก่อนจะเกิดเหตุการณ์ดังกล่าวโดยหากมีหุ่นยนต์ใช้พิกัดที่ใช้ร่วมกันมากกว่า 2 จุด เช่น หุ่นยนต์ตัวที่หนึ่งที่ต้องการเคลื่อนตัวจากจุด (4, 12) ไปยังจุด (4, 13) ตรวจสอบพบว่าหุ่นยนต์ตัวที่สองต้องการใช้จุด (4,13) และหุ่นยนต์ตัวที่สี่ต้องการใช้จุด (4, 12) ระบบจัดการเส้นทางจะรอจนกว่าหุ่นยนต์ตัวที่หนึ่งเคลื่อนที่ไปยังจุด (4, 13) จากนั้นจึงจะอนุญาตให้หุ่นยนต์ตัวที่สี่หรือสองสามารถทำงานต่อไปได้

4. ผลการทดลอง

เริ่มต้นจากการทดลองเปรียบเทียบจำนวนหุ่นยนต์ที่เหมาะสมในระบบโดยการคำนวณเวลาที่ใช้ทำงานของระบบด้วยจำนวนหุ่นยนต์หนึ่งตัวถึงสี่ตัวทำงานแบบ FIFO เปรียบเทียบกับวิธี Market-based Approach โดยเพิ่ม

จำนวนชุดคำสั่งงานครั้งละ 10 งาน จนถึง 100 งาน โดยงานที่ใช้ในการจำลองอ้างอิงจากสถานีงานทั้งหมด 11 สถานี (จุด) ยกตัวอย่างการสร้างชุดคำสั่งงานดังตารางที่ 2 โดยมีสถานีเริ่มต้นและสิ้นสุดที่ได้มาจากการสุ่ม กำหนดให้หุ่นยนต์มีความเร็วเชิงเส้น 1 เมตร/วินาที ความเร็วเชิงมุม 1.57 เรเดียน/วินาที สำหรับตำแหน่งเริ่มต้นนั้นจะกำหนดตำแหน่งหุ่นยนต์ให้กระจายทั่วแผนที่ โดยให้ตำแหน่งเริ่มต้นของหุ่นยนต์ตัวที่หนึ่งอยู่ที่พิกัด (3, 13) หุ่นยนต์ตัวที่สองอยู่ที่พิกัด (0, 0) หุ่นยนต์ตัวที่สามอยู่พิกัด (4, 7) หุ่นยนต์ตัวที่สี่อยู่พิกัด (5, 3) ทำงานบนแผนที่จำลองโรงงานอุตสาหกรรมตามรูปที่ 1

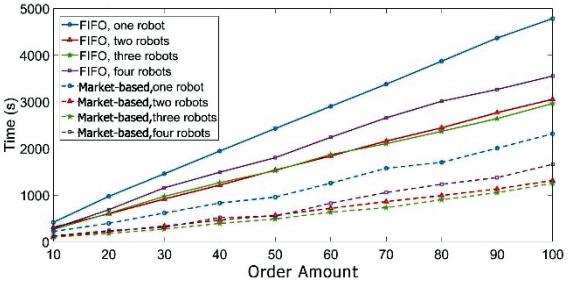
ตารางที่ 2 ตัวอย่างการสร้างชุดคำสั่งงาน

คำสั่งงาน	สถานีเริ่มต้น	สถานีปลายทาง
01	0,0	0,6
02	0,6	0,13
03	0,13	1,11
04	1,11	2,5
05	2,5	3,3
06	3,3	3,8
07	3,8	3,13
08	3,13	4,10
09	4,10	5,0
10	5,0	5,13

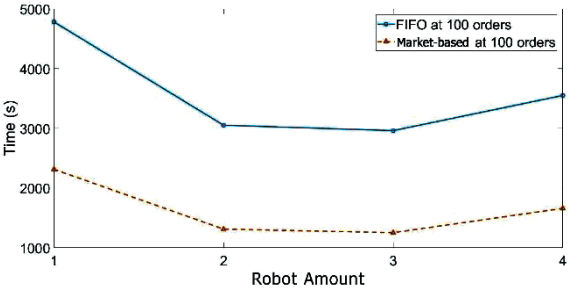
ผลการจำลองการทำงานของโปรแกรมระบบควบคุมด้วย ROS แสดงให้เห็นความสัมพันธ์ของจำนวนงานและเวลาที่ใช้ในการจัดการงานทั้งหมดของทั้งสองวิธีเปรียบเทียบกันดังรูปที่ 10

จากผลการทดลองพบว่าวิธีการเลือกหุ่นยนต์ที่ใช้เวลาน้อยที่สุดไปทำงาน ใช้เวลาในการจัดการงานทั้งหมดเสร็จสิ้นได้เร็วกว่าวิธีแบบ FIFO ถึง 1 เท่าตัว เมื่อเปรียบเทียบด้วยกรณีที่มีหุ่นยนต์จำนวนทำงานเท่ากันแต่วิธีจัดการงานต่างกัน และเวลาที่ใช้ในการทำภารกิจจนเสร็จสิ้น จะเพิ่มขึ้นเป็นเชิงเส้นเมื่อจำนวนชุดคำสั่งเพิ่มขึ้น

นอกจากนี้เมื่อพิจารณาที่จำนวนชุดคำสั่งเท่ากัน รูปที่ 11 แสดงให้เห็นว่าเมื่อหุ่นยนต์ในระบบมีจำนวนเพิ่มขึ้น เวลาที่



รูปที่ 10 เวลาที่ใช้ในการทำงานของระบบหุ่นยนต์หลายตัวแบบ FIFO เปรียบเทียบกับวิธี Market-based Approach



รูปที่ 11 เวลาการทำงานของระบบจัดการหุ่นยนต์ที่จำนวนชุดคำสั่งเท่ากัน เมื่อจำนวนหุ่นยนต์ในระบบเพิ่มขึ้น

ใช้ในการจัดการงานทั้งหมดจนเสร็จสิ้นจะลดลงจนถึงจุดหนึ่งเท่านั้น เพราะหากมีจำนวนหุ่นยนต์เยอะจนเกินไปจะทำให้เกิดการจราจรแออัด หุ่นยนต์ต้องเสียเวลารอเส้นทางว่าง เป็นผลให้ทั้งระบบใช้เวลามากกว่าที่ควรจะเป็น จากการทดลองการจัดการงานที่ 100 ชุดคำสั่งงาน โดยเพิ่มจำนวนหุ่นยนต์ในระบบจาก 1 ถึง 4 ตัว พบว่าสำหรับพื้นที่การทำงานที่กว้าง 5 เมตร ยาว 13 เมตร สำหรับการจัดการชุดคำสั่ง 100 งาน การเลือกใช้หุ่นยนต์ทำงาน 3 ตัว ด้วยวิธีการเลือกหุ่นยนต์ที่ใช้เวลาน้อยที่สุดไปทำงานสามารถจัดการงานได้เร็วที่สุดด้วยเวลา 1,252 วินาที ใกล้เคียงกับหุ่นยนต์จำนวน 2 ตัว

5. สรุป

จากผลการจำลองการจัดการงานด้วยวิธีแบบ FIFO เปรียบเทียบกับวิธี Market-based Approach หรือการ

เลือกหุ่นยนต์ที่ใช้เวลาน้อยที่สุดไปทำงาน พบว่า วิธีการเลือกหุ่นยนต์ที่ใช้เวลาน้อยที่สุดไปทำงาน ทำงานสำเร็จได้เร็วกว่า รวมถึงจำนวนหุ่นยนต์ที่เพิ่มขึ้น สามารถช่วยลดเวลาในการทำงานได้ ซึ่งจากการทดลองสำหรับจำนวนงานทั้งหมด 100 งาน ด้วยวิธีการเลือกหุ่นยนต์ที่ใช้เวลาน้อยที่สุดไปทำงาน ระบบสามารถลดเวลาการทำงานลงครึ่งหนึ่งเมื่อเทียบกับระบบ FIFO นอกจากนี้ยังพบว่าการนำหุ่นยนต์หลายตัวมาใช้ในการทำงาน จำนวนหุ่นยนต์ที่มากขึ้นไม่ได้ส่งผลให้เวลาในการทำงานลดลงเสมอไป เมื่อมีหุ่นยนต์มากถึงจำนวนหนึ่งก็จะส่งผลให้ระบบการทำงานช้าลงได้ด้วย ยกตัวอย่างเช่น เวลาที่ใช้ในการคำนวณเส้นทางมากขึ้น ปัญหาการจัดการจราจรที่พบบ่อยขึ้น ดังนั้นควรให้ความสำคัญกับจำนวนหุ่นยนต์ที่เลือกใช้ในการขนถ่ายวัสดุในอุตสาหกรรมด้วยเช่นกัน

สำหรับวิธีการเลือกหุ่นยนต์ที่ใช้เวลาน้อยที่สุดไปทำงาน เป็นอีกหนึ่งวิธีที่น่าสนใจสำหรับนำมาใช้จัดการชุดคำสั่งงานสำหรับระบบหุ่นยนต์หลายตัวหลายสถานี เพื่อการขนถ่ายวัสดุในโรงงานอุตสาหกรรม เพราะช่วยลดเวลาในการทำงานถึงครึ่งหนึ่งเมื่อเปรียบเทียบกับวิธีแบบ FIFO อีกทั้งวิธีการเลือกหุ่นยนต์ที่ใช้เวลาน้อยที่สุดไปทำงาน ยังนำข้อมูลของหุ่นยนต์แต่ละแบบมาเปรียบเทียบและใช้ตัดสินในการเลือกงานที่เหมาะสมที่สุดให้กับหุ่นยนต์แต่ละประเภทในระบบได้ โดยคำนวณจากคุณสมบัติของตัวหุ่นยนต์แต่ละตัวและตำแหน่งของหุ่นยนต์มาเปรียบเทียบกัน เพื่อส่งมอบงานให้กับหุ่นยนต์ที่เหมาะสมที่สุดที่ว่างงานอยู่ ณ เวลานั้น ซึ่งแก้ไขข้อบกพร่องของการสั่งงานแบบ FIFO ที่ไม่ได้นำเวลาการทำงานของหุ่นยนต์แต่ละตัวมาเปรียบเทียบเพื่อเลือกใช้หุ่นยนต์ให้เหมาะสมกับงาน

งานวิจัยนี้สามารถนำไปใช้กับการขนถ่ายวัสดุในโรงงานอุตสาหกรรมที่มีหุ่นยนต์หลายตัวในการขนถ่ายวัสดุหลายชนิด จากสถานีเริ่มต้นไปยังสถานีปลายทางได้มากกว่าหนึ่งสถานี

เอกสารอ้างอิง

[1] C. J. Malmborg, “A model for the design of zone control automated guided vehicle systems,” *International Journal of Production Research*, vol. 28, no. 10, pp. 1741–1758, 1990.

[2] F. Tang and L. E. Parker, “A complete methodology for generating multi-robot task solutions using ASyMTRe-D and market-based task allocation,” in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 3351–3358.

[3] P. K. Das, H. S. Behera, and B. K. Panigrahi, “Intelligent-based multi-robot path planning inspired by improved classical Q-learning and improved particle swarm optimization with perturbed velocity,” *International Journal Engineering Science and Technology*, vol. 19, no. 1, pp. 651–669, 2016.

[4] K. Thippeswamy, J. Hanumanthappa, and D. H. Manjaiah, “A study on contrast and comparison between Bellman-Ford algorithm and Dijkstra’s algorithm,” presented at the National Conference on Wireless Networks-09 (NCOWN-2010), 2010.

[5] Y. Sharma, S. C. Saini, and M. Bhandhari, “Comparison of Dijkstra’s shortest path algorithm with genetic algorithm for static and dynamic routing network,” *International Journal of Electronics and Computer Science Engineering*, vol. 1, no. 2, pp. 416–425, 2012.

[6] A. Hussein, M. Adel, M. Bakr, O. M. Shehata, and A. Khamis, “Multi-robot task allocation for search and rescue missions,” *Journal of Physics: Conference Series*, vol. 570, pp. 1–10, 2014.

[7] S. A. Fadzli, S. I. Abdulkadir, M. Makhtar, and A. A. Jamal, “Robotic indoor path planning using Dijkstra’s algorithm with multi-layer dictionaries,” in *Proceedings 2015 2nd*

International Conference on Information Science and Security (ICISS), 2015, pp. 1–4.

[8] M. Quigley, K. Conley, B. Gerkey, and J. Faust, "ROS: An open-source robot operating system," presented at the ICRA workshop on open source software, 2009.

[9] M. Ben-Ari and F. Mondada, "Mapping-based navigation," in *Proceedings Elements of Robotics*, 2018, pp. 165–178.

[10] M. Karlsson, F. Ygge, and A. Andersson, "Market-based approaches to optimization," *Computational Intelligence*, vol. 23, no. 1, pp. 92–109, 2007.

[11] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz, "Market-based multirobot coordination: A survey and analysis," in *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1257–1270, 2006.

[12] M. B. Dias and A. Stentz, "Opportunistic optimization for market-based multirobot control," presented at the IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, 2002.

[13] R. Zlot and A. Stentz, "Market-based multirobot coordination for complex tasks," *International Journal of Robotics Research*, vol. 25, no. 1, pp. 73–101, 2006.

[14] C. Myint and N. N. Win, "Position and velocity control for two-wheel differential drive mobile robot," *International Journal of Science Engineering and Technology Research (IJSETR)*, vol. 5, no. 9, pp. 2849–2855, 2016.

[15] H. Taheri, B. Qiao, and N. Ghaeminezhad, "Kinematic model of a four mecanum wheeled mobile robot," *International Journal of Computer Applications*, vol. 113, no. 3, pp. 6–9, 2015.

[16] J. Rubaszewski, A. Yalaoui, L. Amodeo, and S. Fuchs, "Extensions of the unidirectional flow path design problem solved by efficient metaheuristics," *IFAC Proceedings Volumes*, vol. 46, no. 9, pp. 784–789, 2013.

[17] S. Manca, A. Fagiolini, and L. Pallottino, "Decentralized coordination system for multiple AGVs in a structured environment," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 6005–6010, 2011.