

การพัฒนาแบบจำลองระบบจัดเก็บและเรียกคืนสินค้าอัตโนมัติด้วยตัวควบคุมพีไอดี Development Model of Automated Storage and Retrieval System with PID Controller

เวียงชัย ภาระมัตย์¹ จันทิมา ทูมมะลา¹ และนรรรัตน์ วัฒนมงคล¹

¹ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยบูรพา

169 ถนนลงหาดบางแสน ตำบลแสนสุข อำเภอเมือง จังหวัดชลบุรี 20131 E-mail: norrarat@eng.buu.ac.th

บทคัดย่อ

ระบบการจัดการคลังเก็บสินค้าของโรงงานในปัจจุบันได้เริ่มพัฒนาการนำระบบอัตโนมัติมาใช้อย่างต่อเนื่องเพื่อลดแรงงานคน และเพิ่มประสิทธิภาพการขนส่งทางโลจิสติกส์ งานวิจัยฉบับนี้นำเสนอการพัฒนาแบบจำลองระบบจัดเก็บและเรียกคืนสินค้าอัตโนมัติด้วยตัวควบคุมพีไอดี (PID Controller) เพื่อควบคุมการทำงานของมอเตอร์ให้สามารถเคลื่อนไปยังตำแหน่งที่ต้องการได้แม่นยำขึ้นและใช้เวลาในการทำงานลดลง การควบคุมการทำงานของโรบอทจะใช้มอเตอร์ 3 ตัว ซึ่งแต่ละตัวจะเคลื่อนที่ในแต่ละตำแหน่งของแกน 3 มิติ ชุดมอเตอร์จะควบคุมด้วยคอนโทรลเลอร์ชนิด NI myRIO-1900 และใช้โปรแกรมแลปวิว (LabVIEW) ทำหน้าที่สั่งงานมอเตอร์ของโรบอทให้ทำงาน ผู้ใช้จะทำหน้าที่สั่งงานผ่านทางหน้าจอคอมพิวเตอร์ด้วยโปรแกรมอัตโนมัติซึ่งแสดงผลแบบเวลาจริง อย่างไรก็ตาม หากระบบไฟฟ้าเกิดขัดข้องจนโรบอทต้องหยุดการทำงานชั่วคราว โรบอทสามารถจดจำตำแหน่งล่าสุดและโรบอทสามารถกลับมาเริ่มทำงานต่อจากตำแหน่งปัจจุบันได้ทันที จากผลการทดสอบเพื่อเปรียบเทียบกับระบบที่ไม่ใช้ตัวควบคุมพีไอดีพบว่า เปอร์เซนต์โอเวอร์ชูตของระบบลดลงประมาณ 15% และโรบอทใช้เวลาในการเคลื่อนที่ไปยังตำแหน่งที่ต้องการลดลงประมาณ 0.2 วินาที

คำสำคัญ: โรบอท, ระบบควบคุมแบบอัตโนมัติ, โปรแกรมแลปวิว, ระบบควบคุมแบบพีไอดี

Abstract

The inventory management system of company in the present has been continuously developed to reduce worker costs and increase logistics efficiency. This research presents the development model of automated storage and retrieval system (ASRS) with PID Controller to control the motor moving to the desired position more precisely and reduce the operation time. Robotic control uses three motors, each of which moves in each of the three axis positions. The motor is controlled by the controller namely, NI myRIO-1900 and using program LabVIEW

to control the movement of motor. User will perform the task automatically with the program, which is displayed in the real time. If the power supply failures until the robot stopped working, the robots can remember the latest position and can resume from the current position immediately. From the results were compared with the non-PID control systems, the maximum percent overshoot is reduced about 15% and the operation time (setting time) of robots is reduced about 0.2 seconds.

Keywords: Robot, autonomous control system, LabVIEW, PID Controller

1. บทนำ

จากการศึกษาพบว่าในประเทศไทยมีบริษัทประเภทโรงงานอุตสาหกรรมที่มีการขนส่งทางโลจิสติกส์มีแนวโน้มเพิ่มขึ้น [1] โรงงานเหล่านี้จะประกอบธุรกิจเกี่ยวกับขนส่งสินค้าให้กับลูกค้าเพื่อส่งสินค้าไปยังสถานที่ต่างๆ ทั้งภายในประเทศและต่างประเทศ โดยทั่วไปจึงต้องมีคลังเก็บสินค้าเป็นของตนเองเพื่อจัดเก็บสินค้าที่รอการส่งออก สินค้าแต่ละชนิดก็มีความแตกต่างกันหลากหลาย รวมถึงลูกค้าที่มาใช้บริการก็อาจมีจำนวนมาก ซึ่งระบบการจัดการคลังสินค้าหลายแห่งในปัจจุบันยังใช้แรงงานคนเป็นจำนวนมาก ถึงแม้ว่าจะมีการรอก (Folk life) ก็อาจมีจำนวนไม่เพียงพอต่อการใช้งานหรือบางแห่งชั้นวางของมีความสูงมากเกินกว่าที่รอกจะทำงานได้ หากโรงงานมีคลังสินค้าขนาดใหญ่จะทำให้สิ้นเปลืองค่าใช้จ่ายในการจ้างพนักงานมากขึ้น ด้วยเหตุนี้ในปัจจุบันหลายๆ โรงงานจึงได้มีแนวคิดและเริ่มมีการใช้งานระบบจัดเก็บสินค้าโดยใช้พนักงานเพียงไม่กี่คนเท่านั้น และมีบางแห่งอาจจะไม่จำเป็นต้องใช้คนงานในการจัดระบบคลังสินค้า โดยมีเพียงเครื่องจักรทำงานเท่านั้น [2] ทำให้ผู้ปฏิบัติงานไม่ต้องเข้าไปยุ่งเกี่ยวกับอันตราย เช่น อุบัติเหตุจากการควบคุมรอก อุบัติเหตุจากการตกจากที่สูง และอุบัติเหตุจากการร่วงหล่นของสินค้า อย่างไรก็ตาม เทคโนโลยีการจัดการคลังสินค้าแบบอัตโนมัติมักจะนำเข้ามาจากต่างประเทศซึ่งมีค่าใช้จ่ายสูง ถึงแม้ว่าในระยะยาวจะคุ้มค่าต่อการลงทุน

จากปัญหาที่ได้กล่าวมาข้างต้นจึงได้มีการพัฒนาแบบจำลองระบบจับเก็บและเรียกคืนสินค้าอัตโนมัติด้วยตัวควบคุมพีไอดีเพื่อเป็นต้นแบบในการนำไปใช้ในทางปฏิบัติได้จริง โดยใช้ไมโครคอนโทรลเลอร์ Nation Instrument รุ่น myRIO-1900 ในการควบคุมมอเตอร์ที่ใช้ในการขับเคลื่อนโรบอทให้ไปตามตำแหน่งที่ระบุให้มีการหยิบหรือวางสินค้าซึ่งสามารถเคลื่อนที่ได้ทั้งในแนวราบและแนวดิ่ง ในการพัฒนาจะใช้โปรแกรม LabVIEW ในการควบคุมการทำงานต่าง ๆ ของระบบ โดยมีหน้าจอควบคุมการทำงานและแสดงผลการทำงานของตัวโรบอทแบบเวลาจริงผ่านหน้าจอคอมพิวเตอร์

2. ระบบที่นำเสนอ

2.1 วัสดุอุปกรณ์

อุปกรณ์ที่ใช้สร้างแบบจำลองระบบจับเก็บและเรียกคืนสินค้าอัตโนมัติจะประกอบด้วย 2 ส่วน คือ ส่วนที่เป็นชั้นวางสินค้า (Shelf) และส่วนที่เป็นโรบอทสำหรับการจับเก็บและเรียกคืนสินค้า วัสดุที่นำมาใช้สร้างแบบจำลองชั้นวางสินค้าจะใช้ฟิวเจอร์บอร์ดที่หาได้ตามร้านค้าทั่วไป ชั้นวางสินค้ามีขนาดคือ 60×13×50 ซม. มีความสูง 3 ชั้น แต่ละชั้นแบ่งเป็น 5 ช่อง รวมทั้งหมด 15 ช่อง โดยช่องวางสินค้าแต่ละช่องจะมีขนาด 8.5×6×8.5 ลบ.ซม. ระยะการเคลื่อนที่ในแต่ละแกนของโรบอทที่เป็นแบบ 3 มิติคือ 90, 25 และ 50 ซม. ตามแนวแกน X, Y และ Z ตามลำดับ โดยใช้มอเตอร์ที่ยึดติดกับสายพานเคลื่อนที่จำนวน 3 ตัว ชนิด DC motor ควบคุมด้วยสัญญาณพัลส์ (PWM) ส่วนแขนกลที่ใช้สำหรับหยิบหรือวางสินค้า (Gripper) จะออกแบบเป็นลักษณะมือหนีบเพื่อหยิบสินค้าออกมาและขาส์ไคด์เพื่อวางสินค้าเข้าไปยังช่องวางสินค้าให้พอดีโดยไม่ทำให้สินค้าตกลงหล่นเสียหายโดยสร้างจากเครื่องพิมพ์ 3 มิติเพื่อให้ได้รูปแบบและขนาดตามที่ต้องการ และใช้ท่อพีวีซีขนาด ¼ นิ้ว สำหรับทำเป็นรางเลื่อนตัวโรบอทในแนวแกน X

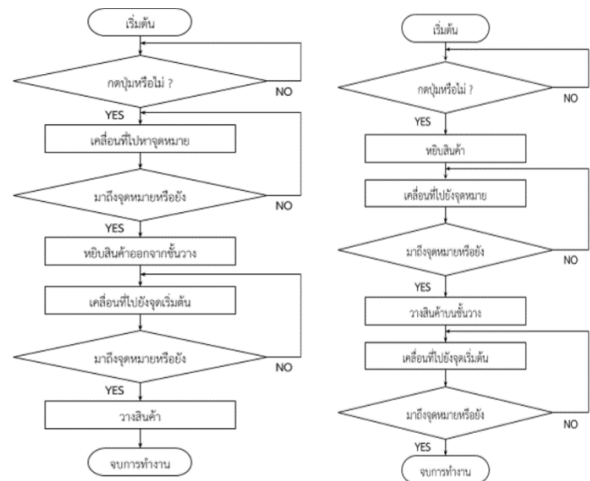


รูปที่ 1 แบบจำลองระบบที่นำเสนอและ NI myRIO 1900

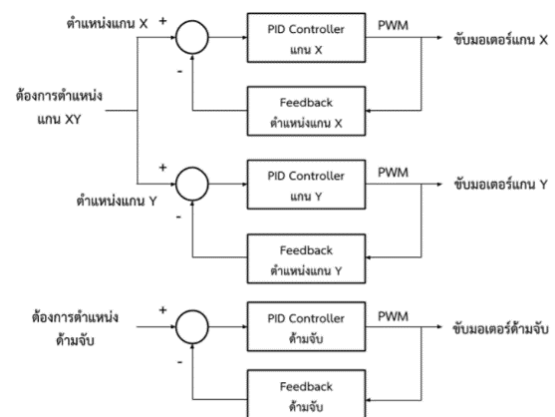
2.2 ขั้นตอนการดำเนินการ

การสร้างแบบจำลองการจัดเก็บคลังสินค้าแบบอัตโนมัติจะเริ่มจากการออกแบบการทำงานของระบบก่อนซึ่งจะมีอยู่ 2 รูปแบบคือการนำสินค้าเข้าไปเก็บบนชั้นวางสินค้า และการนำเอาสินค้าออกมาจาก

ชั้นวางสินค้า นอกจากนี้การควบคุมมอเตอร์จะใช้วิธีการควบคุมด้วย Proportional-Integral-Derivative (PID) แบบลูปปิด [3] และใช้ Encoder Sensor [4] สำหรับบ่อนค่าความผิดพลาดกลับไปยังไมโครคอนโทรลเลอร์เพื่อให้เกิดความแม่นยำในการเคลื่อนที่ไปยังตำแหน่งเป้าหมายที่ต้องการ ซึ่งพิจารณาได้จากรูปที่ 2 และ 3 ตามลำดับ



รูปที่ 2 (ซ้าย) การนำสินค้าเข้าไปเก็บบนชั้นวางสินค้า และ (ขวา) การนำเอาสินค้าออกมาจากชั้นวางสินค้า



รูปที่ 3 ระบบควบคุม PID แบบลูปปิดของแกน X, Y และ Z

ในการออกแบบหน้าต่างควบคุมจะใช้โปรแกรม LabVIEW ในการเขียนโดยจะเขียนอยู่ 2 ส่วนคือ หน้าต่าง GUI บนคอมพิวเตอร์ กับส่วนที่ใช้ในความคุมฮาร์ดแวร์โดยจะโปรแกรมไว้ใน NI myRIO 1900 ในการกำหนดค่าตำแหน่งต่างๆ จะใช้โปรแกรมอีกตัวที่เขียนขึ้นมาเพื่ออ่านค่าจาก Encoder sensor อย่างเดียว เพราะค่าที่อ่านได้คือค่าอ้างอิงตำแหน่งต่างๆ ของชั้นวางสินค้า โดยทดลองปรับมอเตอร์ไปตำแหน่งต่างๆ ของชั้นวางสินค้า จากนั้นก็เอาค่าที่อ่านได้มาเป็นตำแหน่งอ้างอิง

บทความวิจัย

การประชุมวิชาการ งานวิจัยและพัฒนาเชิงประยุกต์ ครั้งที่ 10

10th ECTI-CARD 2018, Phitsanulok Thailand

หากมีชั้นวางที่ขนาดใหญ่ขึ้นหรือเล็กลงต้องทำการปรับตำแหน่งอ้างอิงของชั้นวางใหม่ทั้งหมดเพื่อให้โรบอทราบขอบเขตในการเคลื่อนที่

การแก้ปัญหาคำนวณผิดพลาดในการระบุตำแหน่งของโรบอทจะใช้ตัวควบคุมพีไอดีเพื่อต้องการลดค่าเวลาในการเคลื่อนที่ (setting time) โอเวอร์ชูตสูงสุด (Maximum overshoot) และค่าความผิดพลาดเมื่อเข้าสู่สถานะคงตัว (Steady-state error) ซึ่งค่าพารามิเตอร์ที่ได้กล่าวมานี้จะเป็นพารามิเตอร์ในการปรับค่าอัตราขยายต่างๆ ของตัวควบคุมพีไอดีเพื่อให้สามารถควบคุมตำแหน่งการเคลื่อนที่ได้อย่างแม่นยำและรวดเร็วยิ่งขึ้น โดยฟังก์ชันถ่ายโอน (Transfer Function, $G(s)$) ของตัวควบคุมพีไอดีแสดงได้ดังสมการ (1)

$$G(s) = K_p + K_I \frac{1}{s} + K_D s \quad (1)$$

โดยที่ K_p ทำหน้าที่ควบคุมค่าความผิดพลาด ณ เวลาปัจจุบัน

K_I ทำหน้าที่ควบคุมการสะสมของค่าความผิดพลาด

K_D ทำหน้าที่ควบคุมอัตราการเปลี่ยนแปลงของค่าความผิดพลาด และ s คือตัวดำเนินการในโดเมน s

ความสัมพันธ์ของตัวควบคุมพีไอดีกับ setting time, maximum overshoot และ steady-state error พิจารณาได้จากตารางที่ 1

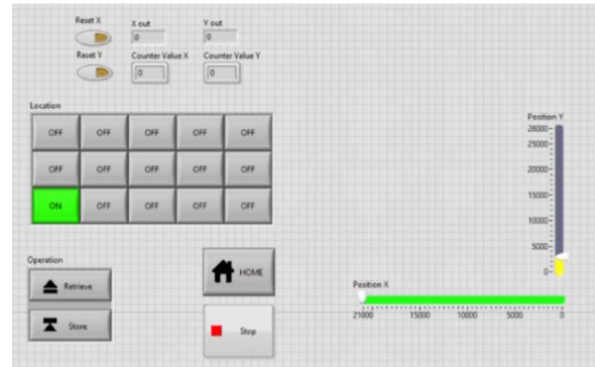
ตารางที่ 1 ความสัมพันธ์ของค่าอัตราขยาย K_p , K_I และ K_D

อัตราขยาย (เพิ่มขึ้น)	Maximum overshoot	Setting time	Steady-state error
K_p	เพิ่มขึ้น	เปลี่ยนน้อยมาก	ลดลง
K_I	เพิ่มขึ้น	เพิ่มขึ้น	ลดลง (จนไม่มี)
K_D	ลดลง	ลดลง	เปลี่ยนน้อยมาก

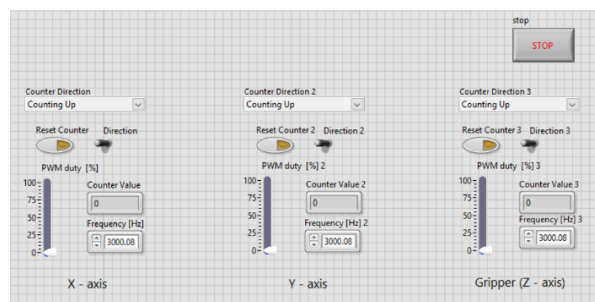
สำหรับส่วนของหน้าต่างควบคุมเปิดใช้งานได้จากโปรแกรม LabVIEW ซึ่งประกอบด้วยตำแหน่งของชั้นวางสินค้าทั้งหมด ดังแสดงในรูปที่ 4 มีปุ่มสำหรับเรียกใช้งานได้จำนวน 4 ปุ่ม คือปุ่มสำหรับนำสินค้าเข้าไปเก็บบนชั้นวาง (Store) ปุ่มสำหรับนำสินค้าออกจากชั้นวาง (Retrieval) ปุ่มสำหรับการรีเซ็ตค่าโรบอทให้เป็นตำแหน่งเริ่มต้น (HOME) และปุ่มสำหรับหยุดการทำงานของโปรแกรมทั้งหมด (Stop) นอกจากนี้ยังแสดงตำแหน่งปัจจุบันของโรบอทจากแถบเลื่อนด้านขวามือ ด้านซ้ายมือจะแสดงถึงระยะในแกน แกน X และ Y ดังรูปที่ 4

ในการออกแบบหน้าต่างควบคุมจะใช้โปรแกรม LabVIEW ในการเขียนโดยจะเขียนอยู่ 2 ส่วนคือ หน้าต่าง GUI บนคอมพิวเตอร์ กับส่วนที่ใช้ในควบคุมฮาร์ดแวร์โดยจะโปรแกรมไว้ใน NI myRIO 1900 ในการกำหนดค่าตำแหน่งต่าง ๆ จะใช้โปรแกรมอีกตัวเพื่ออ่านค่าจาก

Encoder sensor อย่างเดียว จากนั้นก็เอาค่าที่อ่านได้มาเป็นตำแหน่งอ้างอิงในการที่จะให้โรบอทรู้ว่าจะเคลื่อนที่ไปตำแหน่งไหนบนชั้นวางสินค้าซึ่งแสดงดังรูปที่ 5



รูปที่ 4 หน้าต่างควบคุมการทำงานของระบบ



รูปที่ 5 หน้าต่างสำหรับการกำหนดค่าตำแหน่งต่าง ๆ บนชั้นวางสินค้า

3. การทดสอบและผลการทดสอบ

หลังจากที่สร้างแบบจำลองการจัดเก็บคลังสินค้าอัตโนมัติเสร็จแล้ว จะได้ทำการทดสอบ โปรแกรมที่ใช้ในการควบคุมโรบอท โดยขั้นตอนแรกคือการกำหนดตำแหน่งของชั้นวางสินค้าโดยใช้โปรแกรมสำหรับอ่านค่า Encoder Sensor เมื่อได้ตำแหน่งมาครบแล้วก็เอาค่าตำแหน่งต่าง ๆ ไปใส่ไว้ในโปรแกรมหลักเพื่อให้โรบอทรู้ว่าจะต้องเคลื่อนที่ไประยะเท่าใด นอกจากนี้ในการแสดงผลการทำงานของโรบอทแบบเวลาจริงผ่านหน้าจอคอมพิวเตอร์จะเห็นได้ว่าเมื่อมีการกดปุ่มสั่งให้โรบอททำงาน โรบอทจะสามารถเริ่มทำงานได้ตามคำสั่งโดยจะเคลื่อนที่ไปตามช่องวางสินค้าในตำแหน่งต่าง ๆ ของแบบจำลองทั้งในแกน X และ Y และยังสามารถแสดงค่าตำแหน่งปัจจุบันของโรบอทผ่านทางหน้าจอคอมพิวเตอร์ได้ ทั้งนี้ระยะการเคลื่อนที่ในแนวแกนต่าง ๆ จะเกิดความผิดพลาดขึ้นเสมอ เนื่องจากการออกแบบทางกลของมอเตอร์ที่ยังไม่สมบูรณ์ รวมถึงเวลาในการเคลื่อนที่ไปยังแต่ละตำแหน่งก็ช้าด้วย

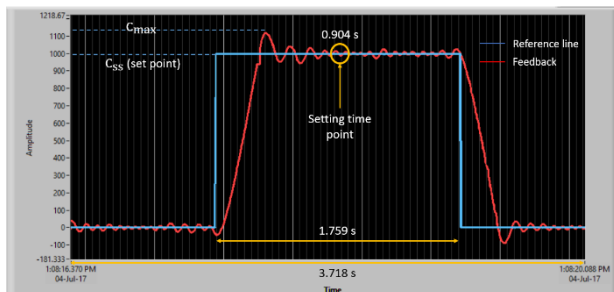
ในการทดสอบจากโปรแกรม LabVIEW จะได้กราฟการทำงานของมอเตอร์ที่ใช้ควบคุมโรบอทโดยพิจารณาจากรูปที่ 6 แสดงให้

เห็นว่าค่าป้อนกลับจากมอเตอร์จะมีค่าที่ผิดพลาดไปจากค่าอ้างอิง (set point) และแกว่งไปมาอยู่บริเวณค่าอ้างอิงในช่วงเข้าสู่สถานะคงตัว หลังจากการปรับค่าอัตราขยายพีไอดีแล้ว ค่าป้อนกลับมอเตอร์จะมีการเข้าสู่ค่าอ้างอิงและไม่เกิดการแกว่งตัวของค่าป้อนกลับในช่วงเข้าสู่สถานะคงตัวในที่สุด เราสามารถหาค่า Maximum percent overshoot (MP) [5] ได้จากสมการ (2)

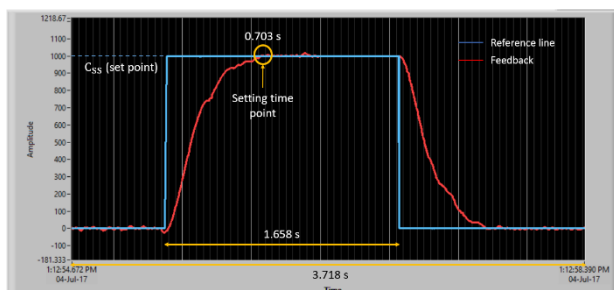
$$MP = \frac{c(t_p) - c(\infty)}{c(\infty)} \times 100\% \quad (2)$$

เมื่อ $c(t_p)$ คือค่าแอมพลิจูดสูงสุด และ $c(\infty)$ คือค่าแอมพลิจูดอ้างอิง

จากรูปที่ 6 ค่าแอมพลิจูดอ้างอิงเท่ากับ 1,000 ที่วัดได้จาก Encoder sensor มีหน่วยเป็นพัลส์ ค่าโอเวอร์ชูตสูงสุดหรือค่าแอมพลิจูดสูงสุดคือ 1,150 พัลส์ เมื่อนำค่าทั้งสองนี้ไปแทนในสมการ (2) จะสามารถหาค่า MP ได้เท่ากับ 15% จากนั้นได้ทำการปรับค่าอัตราขยายพีไอดีดังตารางที่ 2 เพื่อต้องการลดค่า MPO ให้ได้มากที่สุด ซึ่งสภาวะของระบบที่ดีที่สุดคือเมื่อ $K_p = 0.5, K_i = 0$ และ $K_d = 0.003$ จากรูปที่ 7 จะพบว่าค่าโอเวอร์ชูตสูงสุดเท่ากับ 1,002 พัลส์ จะส่งผลให้ค่า MPO ลดลงเหลือเพียง 0.2% เวลาเข้าสู่ setting time ได้เร็วขึ้นจากเดิมคือ 0.904 วินาที เป็น 0.703 วินาที และ steady-stage error มีค่าเท่ากับ 0.2%



รูปที่ 6 ก่อนการปรับค่าอัตราขยายพีไอดี



รูปที่ 7 หลังการปรับค่าอัตราขยายพีไอดี

ตารางที่ 2 ผลการทดลองปรับค่าอัตราขยายพีไอดี

ค่าอัตราขยาย (K_p, K_i, K_d)	Maximum overshoot (%)	Setting time (s)	Steady stage error (%)
$K_p = 0.5,$ $K_i = 0,$ $K_d = 0.003$	0.2	0.703	0.2
$K_p = 0.5,$ $K_i = 0.001,$ $K_d = 0.003$	5	1.02	0.05
$K_p = 0.5,$ $K_i = 0.01,$ $K_d = 0.01$	10	1.312	0.01

4. สรุป

งานวิจัยฉบับนี้นำเสนอการพัฒนาแบบจำลองระบบจัดเก็บและเรียกคืนสินค้าอัตโนมัติด้วยตัวควบคุมพีไอดี (PID Controller) เพื่อควบคุมการทำงานของมอเตอร์ให้สามารถเคลื่อนไปยังตำแหน่งที่ต้องการได้แม่นยำขึ้นและใช้เวลาในการทำงานลดลงโดยใช้มอเตอร์ 3 ตัว ซึ่งแต่ละตัวจะเคลื่อนที่ในแต่ละตำแหน่งของแกน 3 มิติ ชุดมอเตอร์จะควบคุมด้วยคอนโทรลเลอร์ NI myRIO-1900 และใช้โปรแกรมแลปวิว (LabVIEW) ทำหน้าที่สั่งงานมอเตอร์ของโรบอทให้ทำงาน จากผลการทดสอบพบว่า เปอร์เซนต์โอเวอร์ชูตของระบบลดลงประมาณ 15% และโรบอทใช้เวลาในการเคลื่อนที่ไปยังตำแหน่งที่ต้องการลดลงประมาณ 0.2 วินาที เมื่อเปรียบเทียบกับระบบที่ไม่ใช้ตัวควบคุมพีไอดี

5. กิตติกรรมประกาศ

คณะผู้วิจัยขอขอบคุณ ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยบูรพา ที่สนับสนุนงบประมาณสำหรับการเข้าร่วมการประชุมทางวิชาการระดับชาติ ประจำปีงบประมาณ 2561

เอกสารอ้างอิง

- [1] ธุรกิจโลจิสติกส์ได้รับดิจิทัลบูม, สืบค้นวันที่ 12 ก.พ. 2561, https://www.prachachat.net/news_detail.php?newsid=1497430736
- [2] SCGL Warehouse & Transportation, สืบค้นวันที่ 12 ก.พ. 2561, <https://www.scglogistics.co.th/network/index/3>
- [3] A. Kiam Heong, G. Chong, and L. Yun, "PID control system analysis, design, and technology," IEEE Transactions on Control Systems Technology, vol. 13, pp. 559-576, 2005.
- [4] Rotary encoder, https://en.wikipedia.org/wiki/Rotary_encoder
- [5] Katsuhiko Ogata, Modern Control Engineering, 3rd ed., Prentice Hall, 1997, pp. 150-152.