



การเรียนรู้แบบเสริมกำลังโมเดลเบสสำหรับการควบคุมแขนหุ่นยนต์โดยใช้กล้อง 3 มิติเพื่อการ
โยกย้ายวัตถุ

เมธา เมืองประเสริฐ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมเครื่องกล

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยบูรพา

2566

ลิขสิทธิ์เป็นของมหาวิทยาลัยบูรพา



107404525

BUU-IThesis 64910146-thesis / recv: 20062566 19:32:29 / seq: 122



64910146_107404525

การเรียนรู้แบบเสริมกำลังโมเดลเบสสำหรับการควบคุมแขนหุ่นยนต์โดยใช้กล้อง 3 มิติเพื่อการ
โยกย้ายวัตถุ

เมธา เมืองประเสริฐ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมเครื่องกล
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยบูรพา
2566
ลิขสิทธิ์เป็นของมหาวิทยาลัยบูรพา



107404525

BUU-IThesis 64910146 thesis / recv: 20062566 19:32:29 / seq: 122

MODEL-BASED REINFORCEMENT LEARNING FOR ROBOTIC ARM CONTROL USING
3D CAMERA FOR OBJECT MANIPULATION

META MUEANGPRASERT

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR MASTER DEGREE OF ENGINEERING
IN MECHANICAL ENGINEERING
FACULTY OF ENGINEERING
BURAPHA UNIVERSITY

2023

COPYRIGHT OF BURAPHA UNIVERSITY



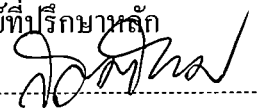
107404525

BUU iThesis 64910146 thesis / recv: 20062566 19:32:29 / seq: 122

คณะกรรมการควบคุมวิทยานิพนธ์และคณะกรรมการสอบวิทยานิพนธ์ได้พิจารณา
วิทยานิพนธ์ของ เมธา เมืองประเสริฐ ฉบับนี้แล้ว เห็นสมควรรับเป็นส่วนหนึ่งของการศึกษาตาม
หลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมเครื่องกล ของมหาวิทยาลัยบูรพาได้

คณะกรรมการควบคุมวิทยานิพนธ์

อาจารย์ที่ปรึกษาหลัก



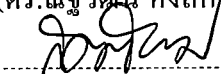
(รองศาสตราจารย์ ดร.กิตติพงษ์ บุญโถ่ง)

คณะกรรมการสอบวิทยานิพนธ์



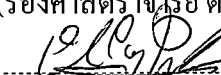
ประธาน

(ดร.ณัฐวัฒน์ หงส์กาญจนกุล)



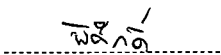
กรรมการ

(รองศาสตราจารย์ ดร.กิตติพงษ์ บุญโถ่ง)



กรรมการ

(รองศาสตราจารย์ ดร.วรเชษฐ์ ภิรมย์ภักดี)



กรรมการ

(ดร.พิศักดิ์ เจิมประยงค์)




คณบดีคณะวิศวกรรมศาสตร์

(ผู้ช่วยศาสตราจารย์ ดร. นยศ คุณกิจ โภศล)

วันที่ A เดือน พฤษภาคม พ.ศ. 2566

บัณฑิตวิทยาลัย มหาวิทยาลัยบูรพา อนุมัติให้รับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของ
การศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมเครื่องกล ของมหาวิทยาลัย
บูรพา



คณบดีบัณฑิตวิทยาลัย

(รองศาสตราจารย์ ดร.วิวัฒน์ แจ่มเอียด)

วันที่ 6 เดือน กรกฎาคม พ.ศ. 2566



64910146: สาขาวิชา: วิศวกรรมเครื่องกล; วศ.ม. (วิศวกรรมเครื่องกล)
 คำสำคัญ: การควบคุมแขนหุ่นยนต์/ การเรียนรู้แบบเสริมกำลัง โมเดลเบส/ กลยุทธ์วิวัฒนาการ
 การปรับตัวของเมทริกซ์ความแปรปรวนร่วม/ การตรวจจับวัตถุด้วยกล้อง 3 มิติ/ การ
 จับและวางวัตถุ

เมธา เมืองประเสริฐ : การเรียนรู้แบบเสริมกำลังโมเดลเบสสำหรับการควบคุมแขนหุ่นยนต์
 โดยใช้กล้อง 3 มิติเพื่อการ โยกย้ายวัตถุ. (MODEL-BASED REINFORCEMENT LEARNING FOR
 ROBOTIC ARM CONTROL USING 3D CAMERA FOR OBJECT MANIPULATION) คณะกรรมการ
 ควบคุมวิทยานิพนธ์: กิตติพงษ์ บุญโล่ง ปี พ.ศ. 2566.

งานวิจัยนี้เป็นการนำเสนอขั้นตอนวิธีการเรียนรู้แบบเสริมกำลังโมเดลเบส (MBRL) สำหรับ
 การควบคุมแขนหุ่นยนต์ร่วมกับการตรวจจับวัตถุด้วยกล้อง 3 มิติในการ โยกย้ายวัตถุ และเพื่อใช้กล้อง 3
 มิติในการตรวจจับพิกัดตำแหน่งของมือจับ วัตถุและเป้าหมายในพิกัดฉาก 3 มิติ เพื่อเป็นข้อมูลสำหรับการ
 ฝึกสอนของวิธี MBRL ซึ่งภายในวิธี MBRL จะนำวิธีการหาค่าเหมาะสมที่สุดกลยุทธ์วิวัฒนาการการ
 ปรับตัวของเมทริกซ์ความแปรปรวนร่วม (CMA-ES) มาทำงานร่วมกับเทคนิคการถดถอยการเรียนรู้ของ
 เครื่องในการสร้างแบบจำลองสิ่งแวดล้อม โดยได้แบ่งการทดสอบออกเป็นสองส่วนคือ การศึกษาเชิง
 ตัวเลขและการทดสอบกับปัญหาจริง ซึ่งในการศึกษาเชิงตัวเลขได้มีการเพิ่มสัญญาณรบกวนแบบเอกรูป
 ของการเคลื่อนที่และนำวิธีจลนศาสตร์แบบผกผัน (IK) มาเปรียบเทียบกับเทคนิคการถดถอยการเรียนรู้
 ของเครื่อง 3 เทคนิคที่ใช้ในวิธี MBRL คือ การถดถอยของกระบวนการเกาส์เซียน (GPR) ซัพพอร์ต
 เวกเตอร์แมชชีนสำหรับการถดถอย (SVR) และโครงข่ายประสาทเทียม (ANN) ผลการทดสอบพบว่า
 เทคนิค GPR จะให้อัตราความสำเร็จสูงที่สุดในทุกปัญหาโดยมีค่ามากที่สุดเท่ากับ 100% เนื่องจากเทคนิค
 GPR เป็นการประมาณความแปรปรวนสัมพัทธ์ที่พิจารณาถึงสัญญาณรบกวน แม้ว่าเทคนิค GPR จะใช้
 เวลาในการฝึกสอนมากที่สุด แต่ก็ถือว่ามีความเหมาะสมกว่าเทคนิคอื่น ซึ่งมีอัตราความสำเร็จเฉลี่ยเพียง
 50% ดังนั้นจึงมีการนำวิธี MBRL(GPR) มาใช้ในการสร้างโครงข่ายนักแสดง (AN) หรือวิธี
 MBRL(GPR)+AN เมื่อนำมาเปรียบเทียบกันแล้วพบว่าวิธี MBRL(GPR) ยังคงให้อัตราความสำเร็จสูงที่สุด
 เท่ากับ 100% และใช้เวลาการฝึกสอนที่น้อยกว่าวิธี MBRL(GPR)+AN ส่วนการทดสอบปัญหาจริง เมื่อนำ
 วิธี MBRL(GPR) มาทำงานร่วมกับการตรวจจับวัตถุจากกล้อง 3 มิติ พบว่าวิธี MBRL(GPR) ยังคงให้อัตรา
 ความสำเร็จสูงถึง 100% ทั้งการควบคุมการเคลื่อนที่ในแนวระนาบและในพิกัดฉากสามมิติ อีกทั้งยัง
 สามารถจับและวางวัตถุได้สำเร็จ 100%



107404525

64910146: MAJOR: MECHANICAL ENGINEERING; M.Eng. (MECHANICAL ENGINEERING)

KEYWORDS: ROBOTIC ARM CONTROL/ MODEL-BASED REINFORCEMENT LEARNING/ COVARIANCE MATRIX ADAPTATION EVOLUTION STRATEGY/ 3D CAMERA OBJECT DETECTION/ PICK-AND-PLACE

META MUEANGPRASERT : MODEL-BASED REINFORCEMENT LEARNING FOR ROBOTIC ARM CONTROL USING 3D CAMERA FOR OBJECT MANIPULATION. ADVISORY COMMITTEE: KITTIPONG BOONLONG, Ph.D. 2023.

This research proposed model-based reinforcement learning (MBRL) algorithm for robotic arm control with 3D camera object detection for object manipulation. 3D camera was used to detect gripper, object, and target position in a 3D coordinates system as training data of the MBRL. Within MBRL, the covariance matrix adaptation evolution strategy (CMA-ES) was combined with machine learning techniques to create an environment model. The test problems are divided into numerical studies and real problems. The numerical studies were investigated with additional uniform noise in movement. The inverse kinematics (IK) was compared with 3 machine learning regression techniques used in the MBRL, Gaussian process regression (GPR), artificial neural network (ANN), and support vector regression (SVR). The results show that the GPR technique has the highest success rate of all numerical studies with a value as high as 100%, because GPR is approximating covariance method that considers noise. Although GPR spent the most training time, GPR was more suitable than other techniques of which the approximately average success rate was only 50%. Therefore, the MBRL(GPR) was used to create Actor network (AN) or MBRL(GPR)+AN. The results of the comparison between MBRL(GPR) and MBRL(GPR)+AN show that MBRL(GPR) still has the highest success rate of 100% and spent training time less than MBRL(GPR)+AN. In real problems, when MBRL(GPR) is combined with 3D camera object detection, The results show that MBRL(GPR) still has the highest success rate of 100% for both planar motion control and three-dimensional space. It can also pick and place object with 100% success.



107404525

กิตติกรรมประกาศ

ขอกราบขอบพระคุณรองศาสตราจารย์ ดร.กิตติพงษ์ บุญโหล่ง อาจารย์ที่ปรึกษาวิทยานิพนธ์ เป็นอย่างสูงที่กรุณาให้โอกาสในการศึกษาต่อในระดับบัณฑิตศึกษา ให้ความช่วยเหลือ ให้คำแนะนำ ให้ความรู้ ตลอดจนช่วยตรวจสอบและแก้ไขจุดบกพร่องต่าง ๆ ด้วยความเอาใจใส่อย่างดีเยี่ยมเสมอมา จนวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี

ขอกราบขอบพระคุณ ดร.ณัฐวัฒน์ หงส์กาญจนกุล ประธานกรรมการสอบวิทยานิพนธ์ รองศาสตราจารย์ ดร.วรเชษฐ ภิรมย์ภักดี ดร.พิศกดิ์ เจิมประยงค์ กรรมการสอบวิทยานิพนธ์ และผู้ช่วยศาสตราจารย์ ดร.จิตติ พัทธวิช กรรมการสอบ โครงร่างวิทยานิพนธ์ ที่กรุณาให้คำแนะนำในการแก้ไขงานวิจัย จนวิทยานิพนธ์ฉบับนี้มีความสมบูรณ์มากยิ่งขึ้น

ขอขอบพระคุณคณะวิศวกรรมศาสตร์ มหาวิทยาลัยบูรพาที่เอื้อเพื่อสถานที่ในการทำงานวิจัย และทุนการศึกษาจากคณะวิศวกรรมศาสตร์ ตามโครงการส่งเสริมและยกระดับคุณภาพบัณฑิตศึกษาเพื่อพัฒนาผู้ช่วยนักวิจัยที่ยั่งยืน คณะวิศวกรรมศาสตร์ มหาวิทยาลัยบูรพา ปีการศึกษา ๒๕๖๕

ขอกราบขอบพระคุณบิดา มารดา และครอบครัวของผู้วิจัยที่คอยให้กำลังใจ และให้การสนับสนุนในทุก ๆ ด้าน และสุดท้ายขอขอบคุณตัวผู้วิจัยเองที่มีความมุ่งมั่น เพียรพยายามและอดทนจนสำเร็จการศึกษาได้ตามเป้าหมายที่วางไว้

เมธา เมืองประเสริฐ



107404525

BUU_1Thesis 64910146 thesis / recv: 20062566 19:32:29 / seq: 122

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ง
บทคัดย่อภาษาอังกฤษ	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ	ช
สารบัญตาราง	ฅ
สารบัญภาพ	ฉุ
บทที่ 1 บทนำ.....	1
ที่มาและความสำคัญของงานวิจัย	1
วัตถุประสงค์ของงานวิจัย	2
ขอบเขตของงานวิจัย.....	2
ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	4
หุ่นยนต์แบบอนุกรม (SERIAL ROBOT)	4
จลนศาสตร์ (KINEMATICS).....	7
การเรียนรู้แบบเสริมกำลัง (REINFORCEMENT LEARNING: RL)	10
ขั้นตอนวิธีการเรียนรู้ของเครื่องสำหรับปัญหาการถดถอย (MACHINE LEARNING ALGORITHMS FOR REGRESSION PROBLEM).....	18
ขั้นตอนวิธีการหาค่าเหมาะสมที่สุด (OPTIMIZATION ALGORITHM).....	31
โครงข่ายประสาทเทียมแบบสังวัตนาการ (CONVOLUTIONAL NEURAL NETWORK: CNN)	34
งานวิจัยที่เกี่ยวข้อง.....	42



107404525

BUU-IThesis 64910146 thesis / recv: 20062566 19:32:29 / seq: 122

บทที่ 3 ระเบียบวิธีวิจัย.....	45
อุปกรณ์ที่ใช้ในการทดสอบ.....	45
การออกแบบชุดการทดลอง.....	46
การทำงานของชุดการทดลอง.....	47
การสร้างโมเดลการตรวจจับวัตถุ.....	48
การเรียนรู้แบบเสริมกำลังโมเดลเบส (MODEL-BASED REINFORCEMENT LEARNING) ..	53
การศึกษาเชิงตัวเลข (NUMERICAL STUDY)	57
การทดสอบปัญหาจริง	63
บทที่ 4 ผลการวิจัย	70
ผลการทดสอบการศึกษาเชิงตัวเลข	71
ผลการทดสอบปัญหาจริง	84
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ	94
สรุปผลการวิจัย	94
ข้อเสนอแนะ	96
บรรณานุกรม.....	97
ภาคผนวก	103
ภาคผนวก ก	103
ประวัติย่อของผู้วิจัย	105



107404525

สารบัญตาราง

	หน้า
ตารางที่ 2-1 ตารางคิว (Baeldung, 2023).....	14
ตารางที่ 2-2 ตารางคิวของปัญหาการให้หุ่นยนต์เคลื่อนที่ไปยังเป้าหมาย.....	16
ตารางที่ 3-1 ความแตกต่างของระยะความลึกในแกน z ที่ได้จากกล้อง 3 มิติและการวัดของมือจับ.....	53
ตารางที่ 4-1 ผลการทดสอบปัญหาการวางด้วยวิธี MBRL.....	76
ตารางที่ 4-2 ผลการทดสอบปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 3-DOF	78
ตารางที่ 4-3 ผลการทดสอบปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4-DOF ในแนวระนาบด้วยวิธี MBRL(GPR)	80
ตารางที่ 4-4 ผลการทดสอบปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4-DOF ในพิกัดฉาก 3 มิติด้วยวิธี MBRL(GPR).....	82



107404525

สารบัญภาพ

หน้า

ภาพที่ 2-1 องค์ประกอบของหุ่นยนต์แบบอนุกรม (Khasale et al., 2019)	4
ภาพที่ 2-2 ลักษณะของข้อต่อแบบหมุนและข้อต่อแบบเชิงเส้น (Jazar, 2022)	5
ภาพที่ 2-3 แขนหุ่นยนต์แบบข้อต่อ (Rootstrap, 2022).....	5
ภาพที่ 2-4 แขนหุ่นยนต์ทรงกลม (Rootstrap, 2022)	6
ภาพที่ 2-5 แขนหุ่นยนต์สี่ขาร (Rootstrap, 2022)	6
ภาพที่ 2-6 แขนหุ่นยนต์ทรงกระบอก (Rootstrap, 2022)	7
ภาพที่ 2-7 แขนหุ่นยนต์คาร์ทีเซียน (Rootstrap, 2022).....	7
ภาพที่ 2-8 แขนหุ่นยนต์สองชั้นต่อ โยง (Mistakes., 2022)	8
ภาพที่ 2-9 องค์ประกอบของการเรียนรู้แบบเสริมกำลัง (Shweta, 2018)	11
ภาพที่ 2-10 ความแตกต่างระหว่าง MFRL และ MBRL (Gireesh, 2020)	11
ภาพที่ 2-11 Markov Decision Process (Sagi, 2020).....	12
ภาพที่ 2-12 รหัสเทียม (Pseudocode) การทำงานของการเรียนรู้แบบคิว (Mateo, 2020).....	13
ภาพที่ 2-13 หลักการของ Epsilon-Greedy (Thomas, 2018)	15
ภาพที่ 2-14 ปัญหาการให้หุ่นยนต์เคลื่อนที่ไปยังเป้าหมาย (Jaiswal, 2017)	16
ภาพที่ 2-15 การเคลื่อนที่ตามนโยบายสำหรับปัญหาการให้หุ่นยนต์เคลื่อนที่ไปยังเป้าหมาย	17
ภาพที่ 2-16 โครงข่ายประสาทเทียมที่ใช้ใน DDPG (Gupta et al., 2021)	18
ภาพที่ 2-17 การทำนายราคาทองคำในอนาคต	18
ภาพที่ 2-18 การแมปข้อมูลในปริภูมิอินพุตไปเป็นปริภูมิลักษณะเด่น (Wilimitis, 2018)	22
ภาพที่ 2-19 ฟังก์ชันการสูญเสียแบบเอปซิลอนอินเซนซิทีฟ (Lins et al., 2010)	23
ภาพที่ 2-20 เซลล์ประสาทในสมองของมนุษย์ (Laubscher & Engelbrecht, 2019).....	26
ภาพที่ 2-21 เซลล์ประสาทเทียม (กอบเกียรติ สระอุบล, 2565).....	26



107404525

BUU-IThesis 64910146 thesis / recv: 20062566 19:32:29 / seq: 122

ภาพที่ 2-22 กราฟฟังก์ชันซิกมอยด์ (Van Rossum & Drake, 2009)27

ภาพที่ 2-23 กราฟฟังก์ชันไฮเปอร์โบลิกแทนเจนต์ (Van Rossum & Drake, 2009)28

ภาพที่ 2-24 กราฟฟังก์ชันเรอู (Van Rossum & Drake, 2009)28

ภาพที่ 2-25 ชั้นการทำงานของเพอร์เซ็ปตรอนชั้นเดียว (กอบเกียรติ สระอุบล, 2565).....29

ภาพที่ 2-26 ชั้นการทำงานของเพอร์เซ็ปตรอนแบบหลายชั้น (Luigi, 2020)30

ภาพที่ 2-27 การแพร่เดินหน้า (Camuñas-Mesa et al., 2019)30

ภาพที่ 2-28 การแพร่ย้อนกลับ (Camuñas-Mesa et al., 2019)31

ภาพที่ 2-29 ฟังก์ชันทรงกลม (Sonja & Derek, 2013).....32

ภาพที่ 2-30 การหาคำตอบของ CMA-ES (Tan et al., 2019).....33

ภาพที่ 2-31 การทำงานของโครงข่ายประสาทเทียมแบบสั่งวัตนาการ (Sumit, 2018)34

ภาพที่ 2-32 กระบวนการทำคอนโวลูชัน (Kumar, 2021)35

ภาพที่ 2-33 การพูลลิ่งจากค่าสูงสุด (Roman, 2020).....36

ภาพที่ 2-34 การพูลลิ่งจากค่าเฉลี่ย (Roman, 2020)36

ภาพที่ 2-35 โครงสร้างแบบเลเน็ต-5 (LeCun et al., 1998).....37

ภาพที่ 2-36 โครงสร้างแบบอเล็กเน็ต (Krizhevsky et al., 2012)38

ภาพที่ 2-37 โครงสร้างแบบกูเกิลเน็ต (Szegedy et al., 2015)38

ภาพที่ 2-38 การทำคอนโวลูชันเชิงลึกและการทำคอนโวลูชันเชิงจุด (Neha & Sridhar, 2021)39

ภาพที่ 2-39 การตรวจจับวัตถุ (Prakhar, 2019).....39

ภาพที่ 2-40 ความแตกต่างระหว่างวิธีการประมวลผลแบบสองครั้งและแบบครั้งเดียว (Phan & Yamamoto, 2020)40

ภาพที่ 2-41 หลักการทำงาน YOLO (Wai et al., 2019).....41

ภาพที่ 2-42 การคำนวณหาค่า IoU (Padilla et al., 2020).....42

ภาพที่ 3-1 กล้อง OAK-D45

ภาพที่ 3-2 แขนหุ่นยนต์ Braccio46



107404525

ภาพที่ 3-3 ชุดการทดลอง47

ภาพที่ 3-4 การทำงานของชุดการทดลอง48

ภาพที่ 3-5 การสร้างกรอบรูปภาพและระบุประเภทของวัตถุที่สนใจ48

ภาพที่ 3-6 การปรับขนาดภาพ49

ภาพที่ 3-7 การเพิ่มปริมาณข้อมูลภาพสอนด้วยการขยายรูปภาพ49

ภาพที่ 3-8 กราฟแสดงความแม่นยำในการฝึกสอนของแต่ละรอบ50

ภาพที่ 3-9 กราฟแสดงความผิดพลาดในการฝึกสอนของแต่ละรอบ51

ภาพที่ 3-10 การตรวจจับตำแหน่งวัตถุ ตำแหน่งเป้าหมาย และตำแหน่งมือจับของแขนหุ่นยนต์51

ภาพที่ 3-11 ระบบพิกัดของกล้อง OAK-D.....52

ภาพที่ 3-12 การวัดความลึกในแกน z ของมือจับด้วยดัลต้าเมตร.....52

ภาพที่ 3-13 แผนผังการทำงาน (Flowchart) ของวิธี MBRL54

ภาพที่ 3-14 การสร้างแบบจำลองสิ่งแวดล้อมด้วยเทคนิคการถอดการเรียนรู้ของเครื่องในวิธี MBRL55

ภาพที่ 3-15 การหาการกระทำที่ดีที่สุดจากแบบจำลองสิ่งแวดล้อม55

ภาพที่ 3-16 แผนผังการทำงาน (Flowchart) ของวิธี MBRL+AN56

ภาพที่ 3-17 การสร้างแบบจำลอง ANN ในวิธี MBRL+AN56

ภาพที่ 3-18 การจำลองปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 2-DOF57

ภาพที่ 3-19 ตำแหน่งเป้าหมายที่เป็นไปได้ของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 2-DOF58

ภาพที่ 3-20 สถานะ การกระทำและรางวัลของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 2-DOF58

ภาพที่ 3-21 การจำลองปัญหาการวางวัตถุ.....59

ภาพที่ 3-22 สถานะ การกระทำและรางวัลของปัญหาการวางวัตถุ59

ภาพที่ 3-23 การจำลองปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 3-DOF60

ภาพที่ 3-24 ตำแหน่งเป้าหมายที่เป็นไปได้ของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 3-DOF60

ภาพที่ 3-25 สถานะ การกระทำและรางวัลของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 3-DOF60



107404525

ภาพที่ 3-26 การจำลองปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4-DOF61

ภาพที่ 3-27 ตำแหน่งเป้าหมายที่เป็นไปได้ของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4-DOF.....62

ภาพที่ 3-28 สถานะ การกระทำและรางวัลของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4-DOF
กรณีเคลื่อนที่ในแนวระนาบ yz62

ภาพที่ 3-29 สถานะ การกระทำและรางวัลของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4-DOF
กรณีเคลื่อนที่ในพิกัดฉาก 3 มิติ.....62

ภาพที่ 3-30 เซอร์โวมอเตอร์ที่ใช้ควบคุมการหมุนในแต่ละข้อต่อของแขนหุ่นยนต์63

ภาพที่ 3-31 รูปทรงที่ใช้ในการตรวจจับตำแหน่งมือจับแขนหุ่นยนต์63

ภาพที่ 3-32 ขอบเขตการหมุนของมุมข้อต่อที่ 2, 3 และ 4 ในระนาบ yz64

ภาพที่ 3-33 การคำนวณหาระยะห่างระหว่างตำแหน่งมือจับกับจุดศูนย์กลางการหมุนมุมข้อต่อที่ 1
.....65

ภาพที่ 3-34 สถานะ การกระทำและรางวัลของการควบคุมแขนหุ่นยนต์ในแนวระนาบ65

ภาพที่ 3-35 การหมุนมุมข้อต่อที่ 1 ไปยังมุมต่าง ๆ เพื่อหาดำแหน่งจุดศูนย์กลางการหมุน66

ภาพที่ 3-36 การหาดำแหน่งจุดศูนย์กลางการหมุนของมุมข้อต่อที่ 166

ภาพที่ 3-37 ขอบเขตการหมุนของมุมข้อต่อที่ 1 ในระนาบ xy.....67

ภาพที่ 3-38 วัตถุที่ใช้ในการหยิบจับ67

ภาพที่ 3-39 เป้าหมายที่ใช้ในการวางวัตถุ.....68

ภาพที่ 3-40 ฐานที่ใช้ในการวางวัตถุและเป้าหมาย.....68

ภาพที่ 3-41 การวิเคราะห์การหมุนของมุมข้อต่อที่ 1.....69

ภาพที่ 3-42 การหมุนของมุมข้อต่อที่ 1 ไปยังแนวเส้นตรงเดียวกับตำแหน่งที่ต้องการจับวัตถุ69

ภาพที่ 4-1 การเปรียบเทียบอัตราความสำเร็จในแต่ละเทคนิคเมื่อสัญญาณรบกวนเพิ่มขึ้นของปัญหา
การไปถึงเป้าหมายของแขนหุ่นยนต์ 2-DOF.....72

ภาพที่ 4-2 ระยะห่างระหว่างปลายแขนหุ่นยนต์และเป้าหมายในแต่ละรอบของปัญหาการไปถึง
เป้าหมายของแขนหุ่นยนต์ 2-DOF จากเทคนิค GPR กรณีสัญญาณรบกวน = 0.0%72

ภาพที่ 4-3 ระยะห่างระหว่างปลายแขนหุ่นยนต์และเป้าหมายในแต่ละรอบของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 2-DOF จากเทคนิค ANN กรณีสัญญาณรบกวน = 0.0%	73
ภาพที่ 4-4 ระยะห่างระหว่างปลายแขนหุ่นยนต์และเป้าหมายในแต่ละรอบของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 2-DOF จากเทคนิค SVR กรณีสัญญาณรบกวน = 0.0%	73
ภาพที่ 4-5 เส้นทางการเคลื่อนที่ของวิธี IK ในปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 2-DOF .	74
ภาพที่ 4-6 ข้อมูลการฝึกสอนในแต่ละเทคนิคเมื่อสัญญาณรบกวนเพิ่มขึ้นของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 2-DOF	75
ภาพที่ 4-7 เวลาการฝึกสอนในแต่ละเทคนิคเมื่อสัญญาณรบกวนเพิ่มขึ้นของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 2-DOF	75
ภาพที่ 4-8 ระยะห่างระหว่างปลายแขนหุ่นยนต์และเป้าหมายในแต่ละรอบของปัญหาการวาง.....	77
ภาพที่ 4-9 การเปรียบเทียบอัตราความสำเร็จเมื่อสัญญาณรบกวนเพิ่มขึ้นของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 3-DOF	78
ภาพที่ 4-10 การเปรียบเทียบอัตราความสำเร็จในแต่ละวิธีเมื่อสัญญาณรบกวนเพิ่มขึ้นของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4-DOF ในแนวระนาบ	79
ภาพที่ 4-11 การเปรียบเทียบเวลาการฝึกสอนในแต่ละวิธีเมื่อสัญญาณรบกวนเพิ่มขึ้นของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4-DOF ในแนวระนาบ	80
ภาพที่ 4-12 การเปรียบเทียบเวลาการทดสอบในแต่ละวิธีเมื่อสัญญาณรบกวนเพิ่มขึ้นของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4-DOF ในแนวระนาบ	81
ภาพที่ 4-13 การเปรียบเทียบอัตราความสำเร็จในแต่ละวิธีเมื่อสัญญาณรบกวนเพิ่มขึ้นของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4-DOF ในแนวระนาบ	82
ภาพที่ 4-14 การเปรียบเทียบเวลาการฝึกสอนในแต่ละวิธีเมื่อสัญญาณรบกวนเพิ่มขึ้นของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4-DOF ในพิกัดฉาก 3 มิติ	83
ภาพที่ 4-15 การเปรียบเทียบเวลาการทดสอบในแต่ละวิธีเมื่อสัญญาณรบกวนเพิ่มขึ้นของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4-DOF ในพิกัดฉาก 3 มิติ	83
ภาพที่ 4-16 อัตราความสำเร็จในแต่ละวิธีของการควบคุมแขนหุ่นยนต์ในแนวระนาบ	84
ภาพที่ 4-17 การเคลื่อนที่ของแขนหุ่นยนต์ในแนวระนาบเพื่อไปยังเป้าหมายของวิธี MBRL(GPR)	85

ภาพที่ 4-18 การเคลื่อนที่ของแขนหุ่นยนต์ในแนวระนาบเพื่อไปยังเป้าหมายของวิธี MBRL(GPR)+AN.....	85
ภาพที่ 4-19 การเคลื่อนที่ของแขนหุ่นยนต์ในแนวระนาบเพื่อไปยังเป้าหมายของวิธี IK	86
ภาพที่ 4-20 เวลาการฝึกสอนในแต่ละวิธีของการควบคุมแขนหุ่นยนต์ในแนวระนาบ	86
ภาพที่ 4-21 เวลาการทดสอบในแต่ละวิธีของการควบคุมแขนหุ่นยนต์ในแนวระนาบ	87
ภาพที่ 4-22 อัตราความสำเร็จในแต่ละวิธีของการควบคุมแขนหุ่นยนต์ในพิกัดฉาก 3 มิติ.....	88
ภาพที่ 4-23 การเคลื่อนที่ของแขนหุ่นยนต์ในพิกัดฉาก 3 มิติเพื่อไปยังเป้าหมายของวิธี MBRL(GPR).....	88
ภาพที่ 4-24 การเคลื่อนที่ของแขนหุ่นยนต์ในพิกัดฉาก 3 มิติเพื่อไปยังเป้าหมายของวิธี MBRL(GPR)+AN.....	89
ภาพที่ 4-25 การเคลื่อนที่ของแขนหุ่นยนต์ในพิกัดฉาก 3 มิติเพื่อไปยังเป้าหมายของวิธี IK.....	89
ภาพที่ 4-26 เวลาการฝึกสอนในแต่ละวิธีของการควบคุมแขนหุ่นยนต์ในพิกัดฉาก 3 มิติ.....	90
ภาพที่ 4-27 เวลาการทดสอบในแต่ละวิธีของการควบคุมแขนหุ่นยนต์ในพิกัดฉาก 3 มิติ	90
ภาพที่ 4-28 การเคลื่อนที่ในแต่ละครั้งของการจับและวางวัตถุจากมุมมองกล้อง OAK-D.....	91
ภาพที่ 4-29 การเคลื่อนที่ในแต่ละครั้งของการจับและวางวัตถุจากมุมมองด้านข้าง	92
ภาพที่ 4-30 รูปแบบการจับวัตถุ.....	93
ภาพที่ 4-31 รูปแบบการวางวัตถุ.....	93



107404525

บทที่ 1

บทนำ

ที่มาและความสำคัญของงานวิจัย

แขนหุ่นยนต์ (Robotic arm) เป็นหนึ่งในประเภทของหุ่นยนต์ที่ถูกออกแบบให้มีกลไกของแขนและข้อต่อคล้ายคลึงกับแขนของมนุษย์ ทำให้สามารถเคลื่อนไหวข้อต่อตามจุดต่าง ๆ ได้อย่างอิสระ เพื่อยังผลให้ปลายแขนหุ่นยนต์เคลื่อนที่ไปยังตำแหน่งที่ต้องการได้ โดยส่วนใหญ่แล้วแขนหุ่นยนต์มักจะพบเจอในแวดวงของอุตสาหกรรม ซึ่งถูกนำมาประยุกต์ใช้ในการหยิบจับวัตถุ การเคลื่อนย้ายวัตถุ การเชื่อม การประกอบ หรือการนำไปใช้งานในพื้นที่แคบ ๆ ที่มนุษย์ไม่สามารถเข้าไปทำงานได้ แต่ในปัจจุบันได้มีการนำแขนหุ่นยนต์มาประยุกต์ใช้งานในด้านอื่น ๆ มากยิ่งขึ้น ไม่เพียงแต่ด้านอุตสาหกรรม ดังเช่น ในด้านการแพทย์ที่มีการนำแขนหุ่นยนต์มาช่วยเหลือในการผ่าตัดเปลี่ยนข้อเข่าเทียม (Khlopas et al., 2018) และช่วยดูแลผู้ป่วยโรคกล้ามเนื้ออ่อนแรง (Rivera et al., 2019) ในด้านการเกษตรที่มีการออกแบบและพัฒนาหุ่นยนต์เพื่อใช้ในการเก็บเกี่ยวหน่อไม้ฝรั่งแบบอัตโนมัติ (Funami et al., 2020)

การควบคุมการเคลื่อนที่ของแขนหุ่นยนต์เป็นสิ่งสำคัญในการใช้งานหุ่นยนต์ โดยวิธีการที่นิยมนำมาใช้ควบคุมแขนหุ่นยนต์อย่างอัตโนมัติ คือ การเขียน โปรแกรมพีแอลซี (Programmable logic controller: PLC) ซึ่งเหมาะกับงานที่มีลักษณะการทำงานในรูปแบบซ้ำ ๆ เป็นเวลานาน แต่อย่างไรก็ตามหากต้องการนำแขนหุ่นยนต์ไปประยุกต์ใช้สำหรับงานที่ไม่ได้มีการทำงานซ้ำ ๆ เช่น การหยิบจับและวางวัตถุ (Pick-and-Place) ที่มีตำแหน่งเริ่มต้นของวัตถุและเป้าหมายรวมไปถึงมีความลึกที่เปลี่ยนแปลงอยู่ตลอดเวลา เช่น การช่วยเหลือผู้ป่วยที่เป็นอัมพาตในการหยิบสิ่งของต่าง ๆ หรือการหยิบสินค้าบนชั้นวางตามรายการสั่งซื้อของลูกค้าในร้านสะดวกซื้อ ก็จะเป็นเรื่องยากอย่างมาก เนื่องจากจะต้องมีการตั้งโปรแกรมการทำงานเป็นจำนวนมาก ซึ่งอาจทำให้พื้นที่ของหน่วยความจำไม่สามารถรองรับได้ ในปัจจุบันด้วยความก้าวหน้าของเทคโนโลยีจึงมีวิธีการที่ถูกนำเสนอในการแก้ไขปัญหา คือ การเรียนรู้ของเครื่อง (Machine learning) โดยมีงานวิจัยของ (Jiang et al., 2021) ได้นำเสนอวิธีการเรียนรู้แบบเสริมกำลังเชิงลึกที่มีพื้นฐานมาจากอัลกอริทึม DDPG แบบผสมผสานในการควบคุมหุ่นยนต์โดยมุ่งไปยังการทำรูปภาพของเป้าหมายให้มีประสิทธิภาพ งานวิจัยของ (Zhang & Zheng, 2022) ได้นำเสนอวิธีการเรียนรู้แบบเสริมกำลัง PPO ที่ถูกปรับปรุงสำหรับการควบคุมแขนหุ่นยนต์ 6 องศาอิสระ อีกทั้งงานวิจัยของ (Joshi et al., 2020) ยังได้นำเสนอวิธีการเรียนรู้แบบเสริมกำลังเชิงลึก Double deep Q-learning (DDQN) ร่วมกับ Grasp-



107404525

BUU-1Thesis 64910146 thesis / rev: 20062566 19:32:29 / seq: 122

Q-Network สำหรับการจับวัตถุของแขนหุ่นยนต์ Baxter แต่อย่างไรก็ตามวิธีการดังกล่าวจะเป็นการเรียนรู้แบบเสริมกำลัง โมเดลฟรี ซึ่งเป็นหนึ่งในประเภทของวิธีการเรียนรู้แบบเสริมกำลังที่มีข้อเสียคือ ต้องใช้จำนวนข้อมูลและเวลาในการฝึกสอนเป็นจำนวนมากในการแก้ไขปัญหา ในขณะที่วิธีการอีกประเภทหนึ่งสามารถแก้ไขปัญหาเหล่านี้ได้ คือ การเรียนรู้แบบเสริมกำลัง โมเดลเบส โดยมีงานวิจัยของ (Li et al., 2020) ที่นำเสนอวิธีการเรียนรู้แบบเสริมกำลัง โมเดลเบสในการควบคุมหุ่นยนต์ Baxter และงานวิจัยของ (Deisenroth et al., 2011) ที่นำเสนอวิธีการเรียนรู้แบบเสริมกำลัง โมเดลเบสสำหรับควบคุมแขนหุ่นยนต์ในปัญหาการวางซ้อนกล่อง (Block stacking task) ซึ่งวิธีการดังกล่าวสามารถลดจำนวนข้อมูลและเวลาในการฝึกสอนของการแก้ไขปัญหาได้ อีกทั้งยังให้ประสิทธิภาพการทำงานที่ดี ดังนั้นงานวิจัยนี้จึงได้นำเสนอขั้นตอนวิธีการเรียนรู้แบบเสริมกำลัง โมเดลเบสสำหรับการควบคุมแขนหุ่นยนต์ โดยทำงานร่วมกับการตรวจจับวัตถุด้วยกล้อง 3 มิติในการโยกย้ายวัตถุไปยังตำแหน่งเป้าหมายที่แตกต่างกัน

วัตถุประสงค์ของงานวิจัย

1. เพื่อประยุกต์ใช้ขั้นตอนวิธีการเรียนรู้แบบเสริมกำลัง โมเดลเบสสำหรับการควบคุมแขนหุ่นยนต์ร่วมกับการตรวจจับวัตถุด้วยกล้อง 3 มิติในการโยกย้ายวัตถุ
2. เพื่อใช้กล้อง 3 มิติในการตรวจจับพิกัดตำแหน่งของมือจับ วัตถุและเป้าหมายในพิกัดฉาก 3 มิติ และนำไปใช้เป็นข้อมูลการเรียนรู้สำหรับขั้นตอนวิธีการเรียนรู้แบบเสริมกำลัง โมเดลเบส

ขอบเขตของงานวิจัย

1. แขนหุ่นยนต์ที่ใช้ในการทดสอบเป็นแขนหุ่นยนต์แบบข้อต่อ (Articulated robot) ที่มีจำนวนองศาอิสระเท่ากับ 4
2. มือจับที่ติดตั้งในส่วนปลายแขนหุ่นยนต์มีลักษณะเป็น 2 นิ้วจับ
3. วัตถุที่นำมาใช้ทดสอบมีลักษณะเป็นรูปทรงกระบอก
4. เป้าหมายที่ใช้ในการวางวัตถุมีลักษณะเป็นรูปสี่เหลี่ยมจัตุรัส
5. การตรวจจับระยะความลึกด้วยกล้อง 3 มิติในพิกัดแกน z มีค่าตั้งแต่ 20 ซม. ขึ้นไป
6. ในการดำเนินการทดสอบจะมีการทดสอบกับปัญหาเชิงตัวเลขและปัญหาจริง

ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถประยุกต์ใช้ขั้นตอนวิธีการเรียนรู้แบบเสริมกำลัง โมเดลเบสร่วมกับการตรวจจับวัตถุด้วยกล้อง 3 มิติสำหรับการควบคุมแขนหุ่นยนต์ประเภทอื่นได้ เช่น แขนหุ่นยนต์คาร์ที



107404525

เจียน (Cartesian robot) แขนหุ่นยนต์ทรงกระบอกรอก (Cylindrical robotic arm) และหุ่นยนต์สคารา (SCARA robot)

2. เป็นแนวทางการศึกษาสำหรับผู้สนใจในขั้นตอนวิธีการเรียนรู้แบบเสริมกำลัง โมเดลเบส และสามารถนำไปประยุกต์ใช้สำหรับการแก้ไขปัญหาอื่น ๆ ได้



107404525

BUU-IThesis 64910146 thesis / recv: 20062566 19:32:29 / seq: 122

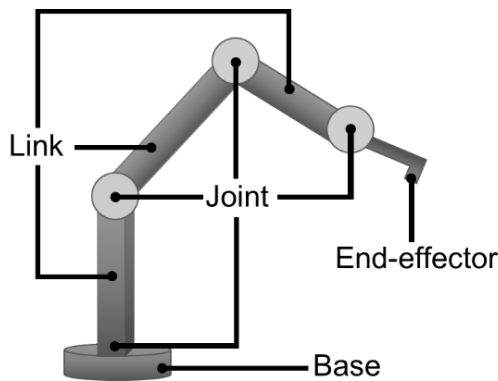
บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

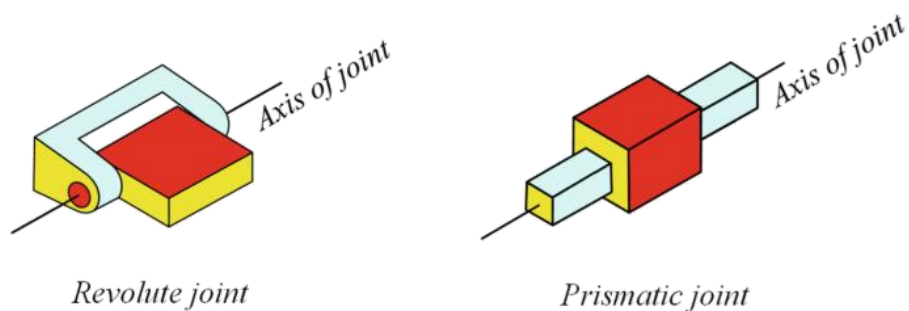
หุ่นยนต์แบบอนุกรม (SERIAL ROBOT)

หุ่นยนต์แบบอนุกรมเป็นหุ่นยนต์ที่มีลักษณะคล้ายคลึงกับแขนของมนุษย์ (Anthropomorphic arm) ซึ่งประกอบขึ้นจากการต่อกันของก้านต่อด้วยข้อต่อจากส่วนฐานไปยังส่วนปลายของแขนหุ่นยนต์ (พงศ์แสน พิทักษ์วัชร, 2557) ดังภาพที่ 2-1 เหมาะสำหรับงานหยิบจับและเคลื่อนย้ายวัตถุ งานที่มีอันตรายหรือมีความเสี่ยงสูงเกินกว่าที่มนุษย์จะสามารถปฏิบัติได้ โดยองค์ประกอบที่สำคัญของหุ่นยนต์แบบอนุกรมมีดังนี้

1. ฐาน (Base) เป็นส่วนที่ยึดติดกับพื้น (Ground) ทำหน้าที่รองรับแรงปฏิกิริยาที่เกิดขึ้นจากภายนอกและแรงพลศาสตร์ (Dynamics) ของหุ่นยนต์ อีกทั้งยังเป็นส่วนสำคัญที่มักใช้เป็นตำแหน่งของกรอบอ้างอิง (Base frame)
2. ก้านต่อ (Link) เป็นชิ้นส่วนสำคัญที่ประกอบกันขึ้นเป็นแขนหุ่นยนต์ ซึ่งเปรียบเสมือนท่อนแขนของมนุษย์
3. ข้อต่อ (Joint) เป็นจุดที่เกิดจากการประสานกันระหว่างก้านต่อตั้งแต่สองชิ้นเข้าด้วยกัน ทำให้แขนหุ่นยนต์สามารถเคลื่อนที่ได้ โดยทั่วไปข้อต่อที่ใช้ในแขนหุ่นยนต์จะมีอยู่ 2 ลักษณะ คือ ข้อต่อแบบหมุน (Revolute joint: R) และข้อต่อแบบเชิงเส้น (Prismatic joint: P) ดังภาพที่ 2-2
4. ปลายแขนหุ่นยนต์ (End-effector) เป็นส่วนที่ใช้สำหรับติดตั้งเครื่องมือเพื่อใช้ทำงานในลักษณะต่าง ๆ เช่น มือจับใช้สำหรับเคลื่อนย้ายวัตถุ หัวเชื่อมไฟฟ้าใช้สำหรับงานเชื่อมโลหะ



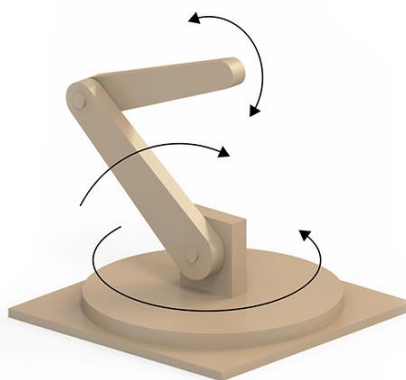
ภาพที่ 2-1 องค์ประกอบของหุ่นยนต์แบบอนุกรม (Khasale et al., 2019)



ภาพที่ 2-2 ลักษณะของข้อต่อแบบหมุนและข้อต่อแบบเชิงเส้น (Jazar, 2022)

1. แขนหุ่นยนต์แบบข้อต่อ (Articulated robot)

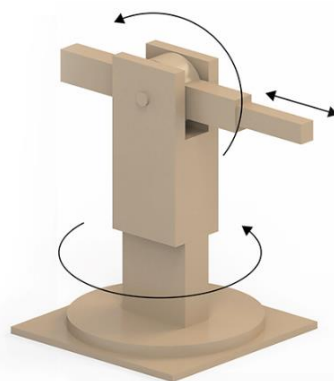
แขนหุ่นยนต์แบบข้อต่อเป็นหุ่นยนต์ที่มีลักษณะคล้ายคลึงกับแขนของมนุษย์มากที่สุด มีข้อต่อเป็นแบบหมุนทั้งหมดสามจุดขึ้นไป หรือมีโครงสร้างแบบ RRR ดังภาพที่ 2-3 สามารถนำไปใช้งานได้หลากหลายประเภท เช่น งานเคลื่อนย้ายวัตถุ งานเชื่อม งานพ่นสี



ภาพที่ 2-3 แขนหุ่นยนต์แบบข้อต่อ (Rootstrap, 2022)

2. แขนหุ่นยนต์ทรงกลม (Spherical robot)

แขนหุ่นยนต์ทรงกลมเป็นหุ่นยนต์ที่มีโครงสร้างแบบ RRP เคลื่อนที่ด้วยข้อต่อแบบหมุนสองจุดในส่วนฐานและไหล่ และในส่วนปลายแขนหุ่นยนต์เคลื่อนที่ด้วยข้อต่อแบบเชิงเส้นดังภาพที่ 2-4 ทำให้พื้นที่การทำงานจะมีลักษณะเป็นรูปทรงกลม ซึ่งเป็นที่มาของชื่อแขนหุ่นยนต์ เหมาะสำหรับงานประเภทเคลื่อนย้ายวัตถุ



ภาพที่ 2-4 แขนหุ่นยนต์ทรงกลม (Rootstrap, 2022)

3. แขนหุ่นยนต์สคารา (SCARA robot)

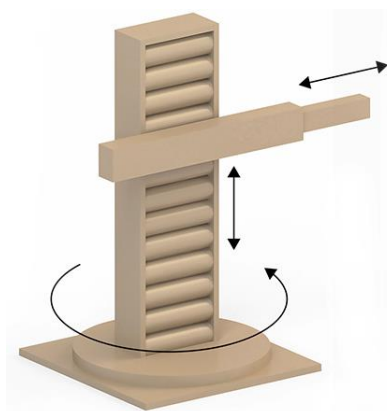
แขนหุ่นยนต์สคาราเป็นหุ่นยนต์ที่มีโครงสร้างแบบ RRP เหมือนกับแขนหุ่นยนต์ทรงกลมดังภาพที่ 2-5 แต่จะมีความแตกต่างในเชิงโครงสร้าง โดยแกนหมุนและแกนเคลื่อนที่ในแนวเชิงเส้นของทั้งสามข้อต่อจะขนานกันทั้งหมด นิยมนำไปใช้ในงานประกอบชิ้นส่วนอิเล็กทรอนิกส์ขนาดเล็ก



ภาพที่ 2-5 แขนหุ่นยนต์สคารา (Rootstrap, 2022)

4. แขนหุ่นยนต์ทรงกระบอก (Cylindrical robot)

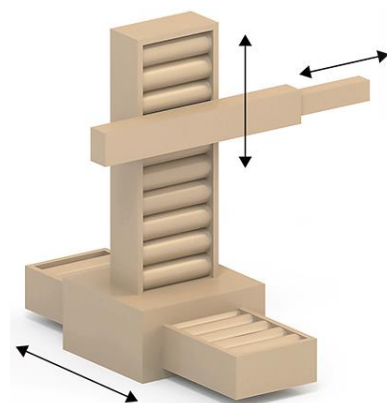
แขนหุ่นยนต์ทรงกระบอกเป็นหุ่นยนต์ที่มีโครงสร้างแบบ RPP โดยข้อต่อแรกเป็นแบบหมุน ข้อต่อที่สองและสามเป็นแบบเชิงเส้นดังภาพที่ 2-6 มีพื้นที่การทำงานเป็นรูปทรงกระบอก ซึ่งเป็นที่มาของชื่อแขนหุ่นยนต์ เหมาะสำหรับงานที่มีความซับซ้อนต่ำ และเน้นความรวดเร็วในการทำงาน



ภาพที่ 2-6 แขนหุ่นยนต์ทรงกระบอกรูป (Rootstrap, 2022)

5. แขนหุ่นยนต์คาร์ทีเซียน (Cartesian robot)

แขนหุ่นยนต์คาร์ทีเซียนเป็นหุ่นยนต์ที่มีโครงสร้างแบบ PPP หรือแบบเชิงเส้นทั้งสามข้อ ต่อด้วยภาพที่ 2-7 มีพื้นที่การทำงานเป็นรูปสี่เหลี่ยม เหมาะสำหรับงานเคลื่อนย้ายวัตถุ งานประกอบชิ้นงานบนโต๊ะ



ภาพที่ 2-7 แขนหุ่นยนต์คาร์ทีเซียน (Rootstrap, 2022)

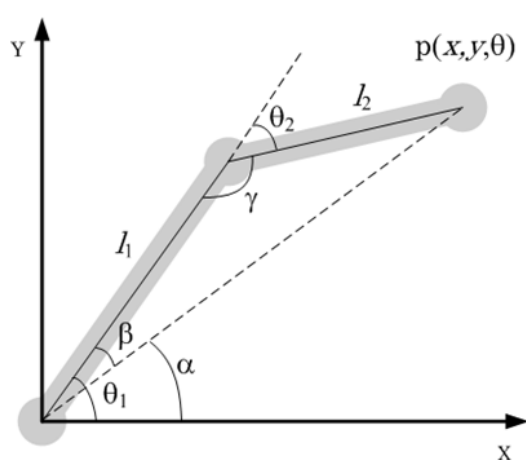
จลนศาสตร์ (KINEMATICS)

จลนศาสตร์เป็นการศึกษาการเคลื่อนที่ของวัตถุโดยไม่คำนึงถึงแรง (Force) ที่มากระทำ ให้อัตถุเกิดการเคลื่อนที่ แต่จะพิจารณาถึงตำแหน่ง (Position) การกระจัด (Displacement) ความเร็ว (Velocity) และความเร่ง (Acceleration) ซึ่งจลนศาสตร์ของหุ่นยนต์จะเป็นการประยุกต์ใช้เรขาคณิต

(Geometry) ในการศึกษาความสัมพันธ์ระหว่างมุมข้อต่อและพิกัดปลายแขนหุ่นยนต์ ที่เป็นพื้นฐานสำหรับการควบคุมแขนหุ่นยนต์ (Kucuk & Bingul, 2006)

1. จลนศาสตร์แบบไปข้างหน้า (Forward kinematics: FK)

จลนศาสตร์แบบไปข้างหน้าเป็นการหาค่าตำแหน่งของปลายแขนหุ่นยนต์เมื่อทราบค่าของมุมข้อต่อต่าง ๆ โดยสมการที่ 2-1 และ 2-2 (Mistakes., 2022) เป็นตัวอย่างในการคำนวณหาตำแหน่งปลายแขนหุ่นยนต์สองชั้นต่อ โยง ซึ่งตัวแปรในการคำนวณดังภาพที่ 2-8



ภาพที่ 2-8 แขนหุ่นยนต์สองชั้นต่อ โยง (Mistakes., 2022)

$$x = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \quad (2-1)$$

$$y = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \quad (2-2)$$

2. จลนศาสตร์แบบผกผัน (Inverse kinematics: IK)

จลนศาสตร์แบบผกผันเป็นการคำนวณย้อนกลับของจลนศาสตร์แบบไปข้างหน้า โดยการกำหนดตำแหน่งปลายแขนหุ่นยนต์เพื่อคำนวณหามุมข้อต่อต่าง ๆ ซึ่งจะมีการคำนวณที่ยากกว่าจลนศาสตร์แบบไปข้างหน้า เนื่องจากคำตอบของปัญหาอาจมีได้หลายคำตอบ และในบางครั้งก็อาจไม่สามารถหาคำตอบได้ โดยสมการที่ 2-6 และ 2-11 (Rehiara, 2011) เป็นตัวอย่างในการคำนวณหามุมข้อต่อของแขนหุ่นยนต์สองชั้นต่อ โยง ซึ่งในการคำนวณจะนำกฎของโคไซน์ (Cosine rule) มาประยุกต์ในการวิเคราะห์เชิงเรขาคณิตดังสมการที่ 2-3

$$(x^2 + y^2) = l_1^2 + l_2^2 - 2l_1l_2 \cos(180 - \theta_2) \quad (2-3)$$

จาก $\cos(180 - \theta_2) = -\cos(\theta_2)$ นำมาแทนในสมการที่ 2-3 จะได้สมการที่ 2-4

$$(x^2 + y^2) = l_1^2 + l_2^2 + 2l_1l_2 \cos(\theta_2) \quad (2-4)$$

จากสมการที่ 2-4 จึงสามารถคำนวณหามุมข้อต่อที่สองได้ดังสมการที่ 2-6

$$\cos(\theta_2) = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2} \quad (2-5)$$

$$\theta_2 = \cos^{-1} \left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2} \right) \quad (2-6)$$

คำนวณหามุม β โดยใช้กฎของไซน์ (Sine rule) จะได้สมการที่ 2-8

$$\frac{\sin(\beta)}{l_2} = \frac{\sin(\gamma)}{\sqrt{x^2 + y^2}} \quad (2-7)$$

$$\beta = \sin^{-1} \left(\frac{l_2 \sin(\gamma)}{\sqrt{x^2 + y^2}} \right) \quad (2-8)$$

จาก $\sin(\gamma) = \sin(180 - \theta_2) = \sin(\theta_2)$ เมื่อแทนในสมการ 2-8 จะได้สมการที่ 2-9

$$\beta = \sin^{-1} \left(\frac{l_2 \sin(\theta_2)}{\sqrt{x^2 + y^2}} \right) \quad (2-9)$$

คำนวณหามุม α โดยใช้อัตราส่วนของตรีโกณมิติ (Trigonometry ratios) จะได้สมการที่ 2-10

$$\alpha = \tan^{-1}\left(\frac{y}{x}\right) \quad (2-10)$$

จาก $\theta_1 = \beta + \alpha$ ดังนั้นจึงสามารถคำนวณหามุมข้อต่อที่หนึ่งได้ดังสมการที่ 2-11

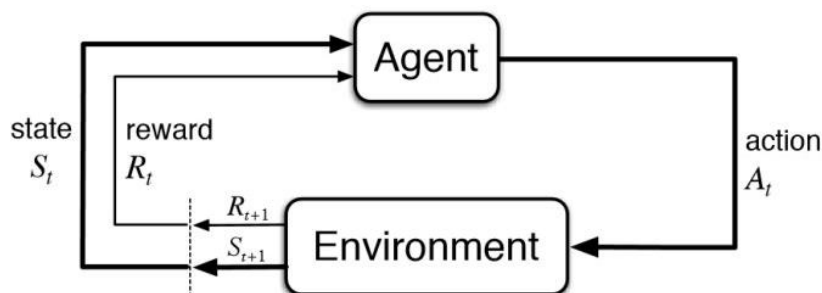
$$\theta_1 = \sin^{-1}\left(\frac{l_2 \sin(\theta_2)}{\sqrt{x^2 + y^2}}\right) + \tan^{-1}\left(\frac{y}{x}\right) \quad (2-11)$$

การเรียนรู้แบบเสริมกำลัง (REINFORCEMENT LEARNING: RL)

การเรียนรู้แบบเสริมกำลังคือหนึ่งในประเภทของการเรียนรู้ของเครื่อง (Machine learning) ที่มีการฝึกสอนโดยให้คอมพิวเตอร์เรียนรู้จากการมีปฏิสัมพันธ์ (Interaction) กับสิ่งแวดล้อม (Environment) โดยการลองผิดลองถูก (Trial and error) ว่าการกระทำไหนดีหรือไม่ดีกับสภาพแวดล้อมที่เกิดขึ้น ซึ่งในการกระทำแต่ละครั้งก็จะมีให้คะแนน (Reward) เพื่อเป็นตัวบ่งบอกว่าคอมพิวเตอร์ควรปรับปรุงการตัดสินใจการกระทำเช่นไร เพื่อให้ได้รางวัลรวม (Cumulative reward) ในระยะยาวมากที่สุด (Kaelbling et al., 1996) โดยการเรียนรู้แบบเสริมกำลังจะประกอบไปด้วย 5 องค์ประกอบหลัก ดังนี้

1. ตัวแทน (Agent) คือ ผู้ที่จะตัดสินใจว่าควรกระทำสิ่งใดภายใต้สถานการณ์ที่เกิดขึ้น
2. สิ่งแวดล้อม (Environment) คือ สภาพแวดล้อมที่กำหนดให้ตัวแทนมีปฏิสัมพันธ์กัน
3. สถานะ (State) คือ สถานการณ์ของสภาพแวดล้อมที่ตัวแทนสามารถรับรู้ได้
4. การกระทำ (Action) คือ การกระทำที่เกิดจากการตัดสินใจของตัวแทน
5. รางวัล (Reward) คือ คะแนนที่ได้จากการกระทำของตัวแทนต่อสถานการณ์ที่เกิดขึ้น

จากองค์ประกอบหลักของการเรียนรู้แบบเสริมกำลัง จึงสามารถอธิบายความสัมพันธ์ของแต่ละองค์ประกอบได้ดังภาพที่ 2-9 โดยเริ่มต้นจากตัวแทนที่สามารถรับรู้สถานการณ์หรือสถานะปัจจุบันที่เกิดขึ้นในสภาพแวดล้อมได้ และจะตัดสินใจที่จะกระทำบางอย่างต่อสถานการณ์ที่เกิดขึ้น จากนั้นจะถูกประเมินผลลัพธ์จากการกระทำของตัวแทนด้วยการให้คะแนน และจะถูกเปลี่ยนผ่านไปสู่สถานะถัดไป

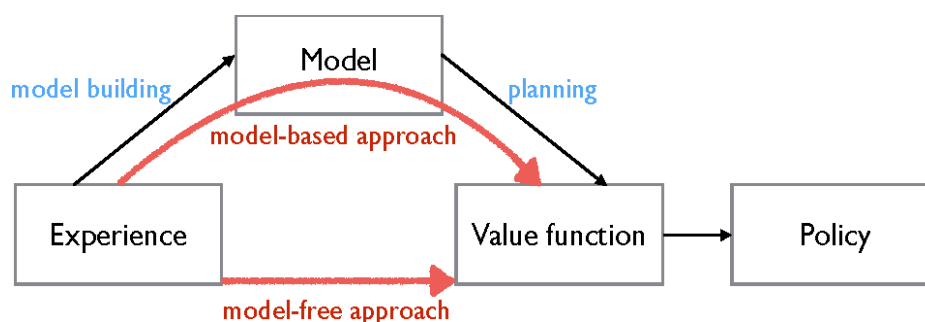


ภาพที่ 2-9 องค์ประกอบของการเรียนรู้แบบเสริมกำลัง (Shweta, 2018)

การแบ่งประเภทของขั้นตอนวิธีการเรียนรู้แบบเสริมกำลัง (Reinforcement learning algorithm) ตามการจำลองสิ่งแวดล้อมสามารถแบ่งได้เป็น 2 ประเภท ดังนี้

1. การเรียนรู้แบบเสริมกำลัง โมเดลฟรี (Model-free reinforcement learning: MFRL) เป็นขั้นตอนวิธีการเรียนรู้ที่ใช้หลักการลองผิดลองถูกในการกระทำบางสิ่งบางอย่างต่อสภาพแวดล้อม ซึ่งไม่จำเป็นต้องรู้ถึงกฎเกณฑ์หรือกลไกของสภาพแวดล้อมว่าเป็นเช่นไร เมื่อตัวแทนมีประสบการณ์จากการลองผิดลองถูกแล้ว ตัวแทนก็จะสามารถปรับปรุงการตัดสินใจในการกระทำต่อสถานะต่าง ๆ ของสภาพแวดล้อมเพื่อให้ได้รางวัลรวมระยะยาวมากที่สุด (Degris et al., 2012)

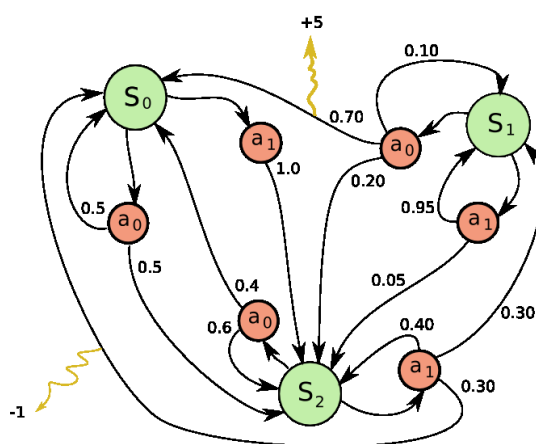
2. การเรียนรู้แบบเสริมกำลัง โมเดลเบส (Model-based reinforcement learning: MBRL) เป็นขั้นตอนการการเรียนรู้แบบเสริมกำลังที่มีการจำลองสภาพแวดล้อมขึ้นมา เพื่อเรียนรู้และทำความเข้าใจถึงกฎเกณฑ์หรือกลไกของสภาพแวดล้อม เพื่อที่จะสามารถวางแผนการตัดสินใจได้ว่าควรกระทำสิ่งใดในสถานะต่าง ๆ เพื่อให้ได้รางวัลรวมระยะยาวมากที่สุด (Moerland et al., 2020)



ภาพที่ 2-10 ความแตกต่างระหว่าง MFRL และ MBRL (Gireesh, 2020)

1. กระบวนการตัดสินใจแบบมาร์คอฟ (Markov Decision Process: MDP)

MDP เป็นกรอบงาน (Framework) ทางคณิตศาสตร์ที่ใช้แสดงถึงรูปแบบของปัญหาที่สามารถใช้การเรียนรู้แบบเสริมกำลังในการแก้ปัญหาได้ โดยจะเป็นการจำลองถึงสถานะต่าง ๆ ที่อยู่ในสภาพแวดล้อม และจะมีการกระทำต่าง ๆ ในแต่ละสถานะที่เมื่อกระทำไปแล้วจะมีโอกาสที่จะได้รับรางวัลและความน่าจะเป็นในการเปลี่ยนจากสถานะหนึ่ง (Transition Probability) ไปยังอีกสถานะหนึ่งที่แตกต่างกัน (Otterlo & Wiering, 2012) ลักษณะของปัญหาแบบ MDP สามารถอธิบายได้ดังภาพที่ 2-11



ภาพที่ 2-11 Markov Decision Process (Sagi, 2020)

จากภาพที่ 2-11 จะพบว่าสถานะในสภาพแวดล้อม จะมีอยู่ทั้งหมด 3 สถานะ และในแต่ละสถานะก็จะมีกระทำให้เลือกทำอย่างละ 2 การกระทำ ซึ่งในแต่ละการกระทำก็จะมีโอกาสที่จะได้รับรางวัล และความน่าจะเป็นในการเปลี่ยนจากสถานะหนึ่งไปยังอีกสถานะหนึ่งที่แตกต่างกันไป โดยหากอยู่ในสถานะที่ 0 (S_0) แล้วเลือกการกระทำที่ 0 (a_0) ก็มีความน่าจะเป็น 50% ที่จะเปลี่ยนจากสถานะที่ 0 ไปยังสถานะที่ 2 (S_2) และไม่ได้รางวัล อีกทั้งยังมีความน่าจะเป็น 50% ที่ยังอยู่ในสถานะเดิม และไม่ได้รางวัล ซึ่งเป้าหมายหลักของการแก้ปัญหาแบบ MDP จะเป็นการสร้างระบบการตัดสินใจที่เรียกว่า ฟังก์ชันนโยบาย (Policy function) ซึ่งเป็นฟังก์ชันที่รับค่าสถานะและให้ผลลัพธ์เป็นการกระทำที่ดีที่สุดออกมา เพื่อให้ได้รางวัลรวมในระยะยาวมีค่ามากที่สุด ฟังก์ชันนโยบายและฟังก์ชันวัตถุประสงค์ของปัญหา MDP จะแสดงดังสมการที่ 2-12 และ 2-13 ตามลำดับ

$$a_t^* = \pi(s_t) \quad (2-12)$$

$$G_t = \sum_{t=1}^{\infty} \gamma^t r_t \quad (2-13)$$

โดยที่ a_t^* คือ การกระทำที่ดีที่สุด ณ เวลาที่ t (Optimal action)

π คือ ฟังก์ชันนโยบาย (Policy function)

G_t คือ รางวัลรวมระยะยาว ณ เวลาที่ t (Cumulative reward)

γ^t คือ ปัจจัยส่วนลด ณ เวลาที่ t (Discount factor)

r_t คือ รางวัล ณ เวลาที่ t (Reward)

2. การเรียนรู้แบบคิว (Q-learning)

การเรียนรู้แบบคิวเป็นหนึ่งในขั้นตอนวิธีการพื้นฐานของการเรียนรู้แบบเสริมกำลัง โมเดลฟรี (ภาพที่ 2-12) ที่ใช้ในการแก้ปัญหาแบบ MDP ที่มีสถานะและการกระทำแบบไม่ต่อเนื่อง (Discrete action and state spaces) โดยหลักการของวิธีการเรียนรู้แบบคิวจะใช้ค่าคิว (Q-Value) ที่อยู่ในตารางคิว (Q-Table) มาสร้างเป็นนโยบายเพื่อบ่งบอกว่าควรกระทำสิ่งใดในสถานะปัจจุบัน เพื่อให้ได้รางวัลรวมในระยะยาวมากที่สุด (Watkins & Dayan, 1992) นอกจากนี้ยังได้มีการพัฒนาการเรียนรู้แบบคิวให้เป็นการเรียนรู้แบบเสริมกำลัง โมเดลเบสที่มีชื่อว่า ไดนาคิว (Dyna-Q) อีกด้วย (Peng et al., 2018) โดยขั้นตอนของวิธีการเรียนรู้แบบคิวมีดังนี้

Algorithm 1: Q-Learning

Result: Off-policy control for estimating $\pi \simeq \pi_*$

Parameters: step size $\alpha \in (0, 1]$, small $\epsilon > 0$

Initialize $Q(s, a) \forall s \in \mathcal{S}^+, a \in \mathcal{A}(s)$ arbitrarily.

Set $Q(\text{terminal}, \cdot) = 0$

for each episode do

 Initialize S ;

for $step = 0$ **to** T **do**

 Choose A from S using policy derived from Q (e.g., ϵ -greedy);

 Take action A , observe R, S' ;

$Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$;

$S \leftarrow S'$;

end

end

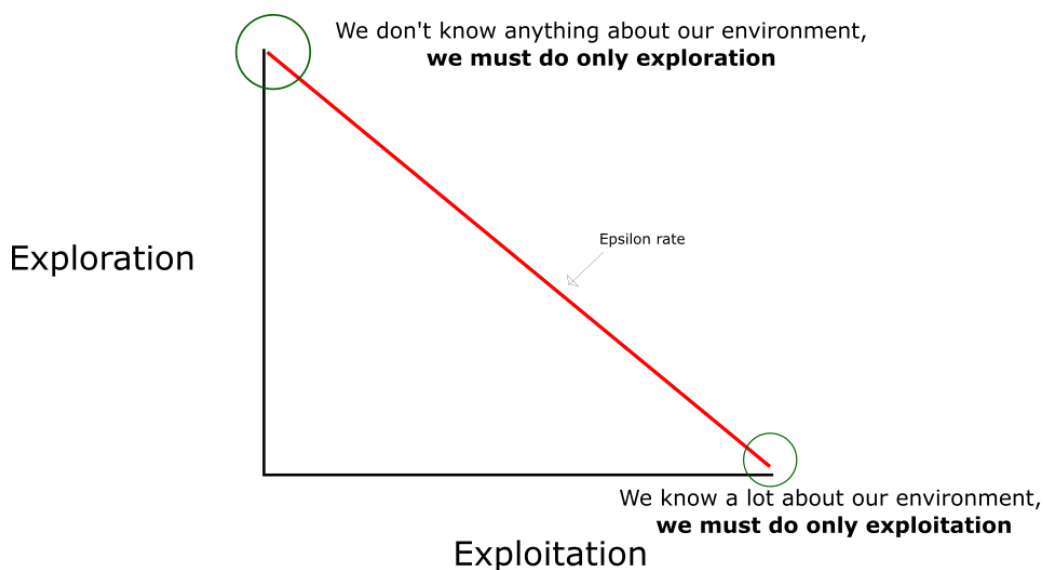
ภาพที่ 2-12 รหัสเทียม (Pseudocode) การทำงานของการเรียนรู้แบบคิว (Mateo, 2020)

2.1 กำหนดรอบของการทำซ้ำ (Episode), กำหนดจำนวนครั้ง (Step) ของแต่ละรอบการทำซ้ำ และสร้างตารางคิวคังตารางที่ 2-1 โดยจำนวนของแถวและหลักของตารางคิว จะมีค่าเท่ากับ จำนวนสถานะคูณกับจำนวนของการกระทำ จากนั้นจึงกำหนดค่าคิว เริ่มต้นในแต่ละช่องเป็นค่าจากการสุ่มหรือมีค่าเท่ากับศูนย์

ตารางที่ 2-1 ตารางคิว (Baeldung, 2023)

States	Actions			
	a_0	a_1	...	a_m
S_0	$Q(S_0, a_0)$	$Q(S_0, a_1)$...	$Q(S_0, a_m)$
S_1	$Q(S_1, a_0)$	$Q(S_1, a_1)$...	$Q(S_1, a_m)$
\vdots	\vdots	\vdots	\vdots	\vdots
S_n	$Q(S_n, a_0)$	$Q(S_n, a_1)$...	$Q(S_n, a_m)$

2.2 ใช้หลักการของ Epsilon-Greedy ดังภาพที่ 2-13 ในการเลือกการกระทำต่อสถานะปัจจุบัน ระหว่างการกระทำแบบสุ่ม (Explore) ด้วยความน่าจะเป็น ϵ (มีค่าอยู่ระหว่าง 0-1) และการกระทำที่ดีที่สุดตามนโยบาย (Exploit) ด้วยความน่าจะเป็น $1-\epsilon$ ซึ่งในช่วงแรกของการเรียนรู้จะกำหนดให้ ϵ มีค่าเท่ากับหนึ่ง กล่าวคือให้เลือกการกระทำแบบสุ่ม 100% จากนั้นจะค่อย ๆ ปรับค่าให้ลดลง วิธีการนี้จะทำให้ตัวแทนมีการเลือกการกระทำทั้งสองอย่างเหมาะสมเพื่อให้รางวัลรวมในระยะยาวมีค่าสูงที่สุด



ภาพที่ 2-13 หลักการของ Epsilon-Greedy (Thomas, 2018)

2.3 นำการกระทำที่ถูกเลือกไปกระทำกับสถานะปัจจุบันในสภาพแวดล้อม จากนั้นจะได้รับรางวัลจากการกระทำและสถานะถัดไป

2.4 อัปเดตค่าคิว โดยใช้สมการเบลแมน (Bellman Equation) แสดงดังสมการที่ 2-14

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha(r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)) \quad (2-14)$$

โดยที่ $Q(s_t, a_t)$ คือ ค่าคิวในสถานะและการกระทำ ณ เวลาที่ t

α คือ อัตราการเรียนรู้ (Learning rate)

r_t คือ รางวัล ณ เวลาที่ t (Reward)


γ คือ ปัจจัยส่วนลด (Discount factor)

$\max_{a_{t+1}} Q(s_{t+1}, a_{t+1})$ คือ การกระทำ ณ เวลา $t+1$ ที่ทำให้มีค่าคิวมากที่สุด

2.5 กำหนดให้สถานะถัดไปเป็นสถานะปัจจุบัน เพื่อไปสู่การทำงานในครั้งถัดไปของแต่ละรอบการทำงานซ้ำ จากนั้นกลับไปยังขั้นตอนที่ 2-4 จนกว่าจะครบจำนวนรอบการทำงานซ้ำ

ตัวอย่างของการใช้นโยบายจากตารางคิว ที่ได้จากการเรียนรู้แบบคิว สำหรับปัญหาการให้หุ่นยนต์เคลื่อนที่ไปยังเป้าหมายในเขาวงกตดังภาพที่ 2-14 ที่มี 11 สถานะและ 4 การกระทำ (Jaiswal, 2017) จะพบว่าเริ่มต้นเมื่อหุ่นยนต์อยู่ในสถานะที่ 8 หุ่นยนต์จะเคลื่อนที่ขึ้นไปยังสถานะที่

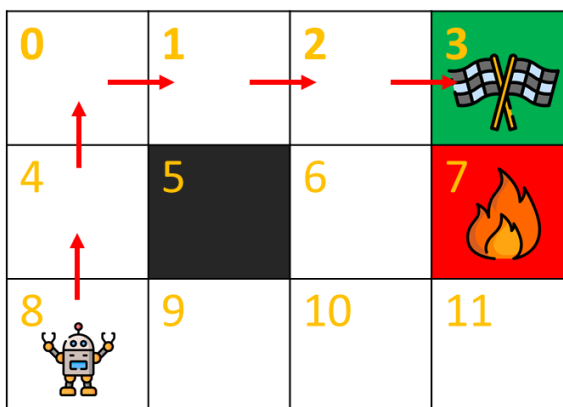
4 เนื่องจากในสถานะที่ 4 และการกระทำขึ้น (Up) จะมีค่าควมมากที่สุดดังตารางที่ 2-2 จากนั้นจะเคลื่อนที่ต่อไปยังสถานะที่ 0, 1, 2 และ 3 (เป้าหมาย) ตามลำดับ ดังภาพที่ 2-15

0	1	2	3 
4	5 	6	7 
8 	9	10	11

ภาพที่ 2-14 ปัญหาการให้หุ่นยนต์เคลื่อนที่ไปยังเป้าหมาย (Jaiswal, 2017)

ตารางที่ 2-2 ตารางควมของปัญหาการให้หุ่นยนต์เคลื่อนที่ไปยังเป้าหมาย

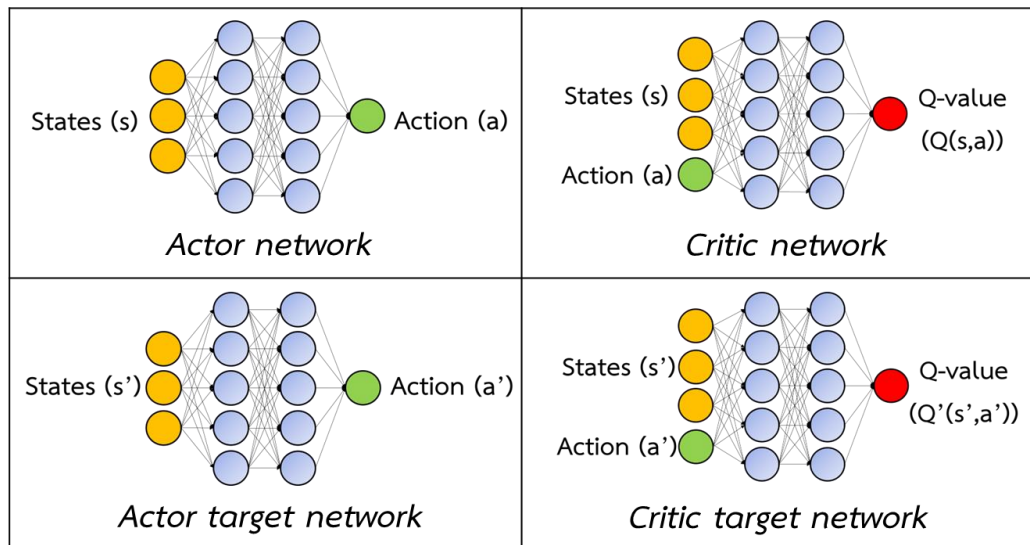
States	Actions			
	Up	Down	Left	Right
0	-0.25	-0.27	-0.25	9.60
1	-0.17	-0.85	-0.24	9.80
2	-0.09	-0.09	-0.16	10.00
3	0.00	0.00	0.00	0.00
4	9.32	-0.44	-0.34	-0.85
5	0.00	0.00	0.00	0.00
6	9.76	-0.25	-0.85	-8.50
7	0.00	0.00	0.00	0.00
8	8.58	-0.42	-0.42	-0.38
9	-0.85	-0.34	-0.32	5.53
10	8.96	-0.25	-0.28	-0.24
11	-8.50	-0.25	-0.25	-0.25



ภาพที่ 2-15 การเคลื่อนที่ตามนโยบายสำหรับปัญหาการให้หุ่นยนต์เคลื่อนที่ไปยังเป้าหมาย

3. Deep Deterministic Policy Gradient (DDPG)

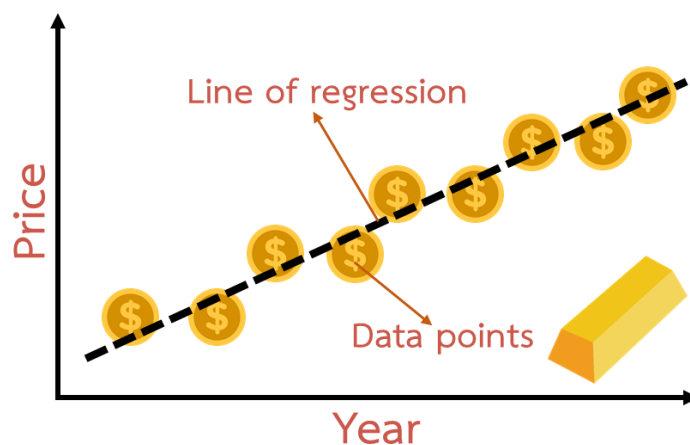
DDPG เป็นขั้นตอนวิธีการเรียนรู้แบบเสริมกำลังเชิงลึก โมเดลฟรีที่เกิดจากการนำแนวคิดของ DPG (Deterministic Policy Gradient) และ DQN (Deep Q-Network) มารวบรวมเข้าด้วยกัน เพื่อใช้สำหรับการแก้ปัญหาที่มีการกระทำแบบต่อเนื่อง (Lillicrap et al., 2015) โดยหลักการของ DDPG จะเป็นการนำโครงข่ายประสาทเทียม 4 โครงข่ายประสาท ซึ่งประกอบไปด้วยโครงข่ายนักแสดง (Actor network), โครงข่ายนักวิจารณ์ (Critic network, โครงข่ายเป้าหมายนักแสดง (Actor target network) และโครงข่ายเป้าหมายนักวิจารณ์ (Critic target network) ดังภาพที่ 2-16 มาทำงานร่วมกันเพื่อให้ได้นโยบายที่ดีที่สุดของปัญหา ซึ่งในการเรียนรู้ของ DDPG จะเรียนรู้จากการใช้รีเพลย์บัฟเฟอร์ (Replay buffer) ที่เสมือนเป็นชุดข้อมูลขนาดใหญ่ที่ประกอบไปด้วยค่าของสถานะ, การกระทำ, รางวัลและสถานะถัดไป ซึ่งถูกเก็บรวบรวมมาจากการให้ตัวแทนได้กระทำต่อสถานะปัจจุบัน และได้รับรางวัลจากการกระทำที่ได้ทำไป จนกระทั่งไปยังสถานะถัดไปของแต่ละครั้งการเคลื่อนที่ในแต่ละรอบการทำงาน มาทำการสุ่มเป็นชุดข้อมูลขนาดย่อย และนำชุดข้อมูลขนาดย่อยนั้นมาอัปเดตค่าน้ำหนักของโครงข่ายนักวิจารณ์และนักแสดง ซึ่งโครงข่ายนักวิจารณ์จะใช้การสูญเสียของนักวิจารณ์ (Critic loss) ที่เป็นการคำนวณหาค่าเฉลี่ยความผิดพลาดกำลังสอง (Mean squared error) ระหว่างค่าวิเป้าหมายที่คำนวณมาจากโครงข่ายเป้าหมายนักแสดงและโครงข่ายเป้าหมายนักวิจารณ์กับค่าวิที่ได้จากโครงข่ายนักวิจารณ์มาทำการอัปเดตค่าน้ำหนัก แต่ในส่วนของโครงข่ายนักแสดงจะใช้การสูญเสียของนักแสดง (Actor loss) ที่คำนวณมาจากค่าเฉลี่ยของค่าวิที่ได้จากโครงข่ายนักวิจารณ์สำหรับการกระทำของโครงข่ายนักแสดง และสุดท้ายจึงนำค่าน้ำหนักของโครงข่ายนักวิจารณ์และนักแสดงที่ถูกอัปเดตแล้วมาทำการอัปเดตค่าน้ำหนักของโครงข่ายเป้าหมายนักวิจารณ์และโครงข่ายเป้าหมายนักแสดงตามลำดับ



ภาพที่ 2-16 โครงข่ายประสาทเทียมที่ใช้ใน DDPG (Gupta et al., 2021)

ขั้นตอนวิธีการเรียนรู้ของเครื่องสำหรับปัญหาการถดถอย (MACHINE LEARNING ALGORITHMS FOR REGRESSION PROBLEM)

ขั้นตอนวิธีการเรียนรู้ของเครื่องสำหรับการถดถอยเป็นวิธีการที่ใช้สำหรับการทำนายค่าเชิงตัวเลข ซึ่งได้จากการนำข้อมูลเข้าไปฝึกสอน โดยการประมาณความสัมพันธ์ระหว่างค่าตัวแปรนำเข้า (Input variable: IV) และค่าตัวแปรส่งออก (Output variable: OV) เช่น การทำนายราคาขึ้น-ลงของบิทคอยน์ และการทำนายราคาทองคำในอนาคตดังภาพที่ 2-17



ภาพที่ 2-17 การทำนายราคาทองคำในอนาคต

1. การถดถอยของกระบวนการเกาส์เซียน (Gaussian process regression: GPR)

การถดถอยของกระบวนการเกาส์เซียนเป็นขั้นตอนวิธีการที่ใช้สำหรับแก้ปัญหาการถดถอยของข้อมูลที่ไม่เป็นเชิงเส้นหรือมีความซับซ้อน โดยมีสมการที่สำคัญคือฟังก์ชันความแปรปรวนร่วมที่เป็นการประมาณความแปรปรวนระหว่างข้อมูลฝึกสอนด้วยฟังก์ชันความแปรปรวนเฉพาะ เพื่อใช้ประมาณค่าความไม่แน่นอนสำหรับการทำนาย (Williams & Rasmussen, 1995) ซึ่งเป็นประโยชน์ในสถานการณ์ที่เราต้องการทราบว่าแบบจำลองมีค่าความเชื่อมั่นเพียงใดในการทำนาย โดยฟังก์ชันที่นิยมใช้งาน คือ ฟังก์ชันเอกซ์โพเนนเชียลกำลังสอง (Squared Exponential Function) แสดงดังสมการที่ 2-15

$$k(x, x') = \sigma_f^2 \exp\left[\frac{-(x-x')^2}{2l^2}\right] \quad (2-15)$$

โดยที่ $k(x, x')$ คือ ฟังก์ชันความแปรปรวนเฉพาะระหว่าง x กับ x'
 x, x' คือ คุณลักษณะของข้อมูล
 σ_f^2 คือ ค่าความแปรปรวน
 l คือ ค่าความยาวสเกล (Length scale)

ในทางปฏิบัติแล้ว ข้อมูลมักมีสัญญาณรบกวน (Noise) ซึ่งอาจเกิดจากการวัดหรือสิ่งแวดล้อม ดังนั้นจึงสามารถพิจารณาความสัมพันธ์ของข้อมูลได้ดังสมการที่ 2-16

$$y = f(x) + N(0, \sigma_n^2) \quad (2-16)$$

โดยที่ y คือ ค่าที่ต้องการทำนาย
 $f(x)$ คือ ฟังก์ชันแฝง (Latent function)
 $N(0, \sigma_n^2)$ คือ สัญญาณรบกวนแบบเกาส์เซียน
 σ_n^2 คือ ค่าความแปรปรวนในสัญญาณรบกวนแบบเกาส์เซียน

เมื่อสัญญาณรบกวนถูกพิจารณา ฟังก์ชันความแปรปรวนร่วมในสมการที่ 2-15 จึงเปลี่ยนเป็นดังสมการที่ 2-17

$$k(x, x') = \sigma_f^2 \exp\left[-\frac{(x-x')^2}{2l^2}\right] + \sigma_n^2 \delta(x, x') \quad (2-17)$$

โดยที่ $\delta(x, x')$ คือ Kronecker delta function ซึ่งจะเท่ากับหนึ่ง หาก x, x' มีค่าเท่ากัน และเท่ากับศูนย์ หากเงื่อนไขเป็นอย่างอื่น

สมมติให้ $\{y\}$ คือ ชุดข้อมูลจำนวน n ตัว และ y^* คือ ค่าที่ต้องการทำนายด้วยตัวแปรอินพุต x ในตอนเริ่มต้นต้องคำนวณฟังก์ชันความแปรปรวนดังสมการที่ 2-17 ของชุดข้อมูลทั้งหมดที่เป็นไปได้ ซึ่งจะได้อเมทริกซ์ดังสมการที่ 2-18 และ 2-19 ดังนี้

$$[K] = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \dots & k(x_n, x_n) \end{bmatrix} \quad (2-18)$$

$$[K_*] = [k(x_*, x_1) \quad k(x_*, x_2) \quad \dots \quad k(x_*, x_n)], \quad [K_{**}] = [k(x_*, x_*)] \quad (2-19)$$

โดยการจำลองแบบกระบวนการเกาส์เซียน ข้อมูลที่มีจะสามารถแทนด้วยตัวอย่างที่สร้างจากการแจกแจงแบบเกาส์เซียนหลายตัวแปร (Gaussian normal distribution) ซึ่งแสดงได้ดังสมการที่ 2-20

$$\begin{bmatrix} \{y\} \\ y_* \end{bmatrix} \approx N\left(\{0\}, \begin{bmatrix} K & K_*^T \\ K_* & K_{**} \end{bmatrix}\right) \quad (2-20)$$

ทั้งนี้จะมุ่งความสนใจไปที่ความน่าจะเป็นเงื่อนไข (Conditional probability) $p(y^*|\{y\})$ ซึ่งมีความหมายว่า “เมื่อมีชุดข้อมูล $\{y\}$ ความเป็นไปได้ที่ทำนายได้ค่าเป็น y^* เป็นเท่าใด” โดยความน่าจะเป็นนี้จะขึ้นอยู่กับผลการแจกแจงแบบเกาส์เซียนที่แสดงดังสมการที่ 2-21

$$y^*|\{y\} \approx N\left([K_*][K]^{-1}\{y\}, [K_{**}] - [K_*][K]^{-1}[K_*]^T\right) \quad (2-21)$$

ค่าที่ดีที่สุดของการประมาณ y^* คือ ค่าเฉลี่ยของการแจกแจงดังกล่าว ดังสมการที่ 2-22

$$\bar{y}^* = [K]^* [K]^{-1} \{y\} \quad (2-22)$$

ขณะที่ความไม่แน่นอนของการประมาณ สามารถอธิบายได้ด้วยค่าความแปรปรวน ดังสมการที่ 2-23

$$\text{var}(y_*) = [K_{**}] - [K_*][K]^{-1}[K_*] \quad (2-23)$$

ทั้งนี้จะต้องหาค่าตัวแปรตัดสินใจ $\{\theta\} = \{l, \sigma_f, \sigma_v\}$ ที่เหมาะสมที่สุด โดยมีฟังก์ชันวัตถุประสงค์ที่กำหนดตามทฤษฎีของเบย์ (Bayes' theorem) ซึ่งแสดงด้วยค่าลอการิทึมความคล้าย (Likelihood logarithm) ดังสมการที่ 2-24 เมื่อได้ค่าตัวแปรตัดสินใจแล้ว จึงถือเป็นการเสร็จสิ้นของการจำลองโมเดล

$$\log p(\{y\} | \{x\}, \{\theta\}) = -\frac{1}{2} \{y\}^T [K]^{-1} \{y\} - \frac{1}{2} \log(\det([K])) - \frac{n}{2} \log 2\pi \quad (2-24)$$

2. ซัพพอร์ตเวกเตอร์สำหรับการถดถอย (Support vector regression: SVR)

ซัพพอร์ตเวกเตอร์แมชชีนสำหรับการถดถอยเป็นการนำเอาเทคนิคของซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine: SVM) ที่ใช้สำหรับการจำแนกกลุ่มข้อมูล (Classification) ด้วยวิธีการหาระนาบตัดสินใจ (Decision hyperplane) ที่เหมาะสมที่สุดมาประยุกต์ใช้สำหรับการประมาณความสัมพันธ์ระหว่างเวกเตอร์ข้อมูลนำเข้า (Input vector) และตัวแปรส่งออก (Output variable) ด้วยฟังก์ชันเชิงเส้น $f(x)$ (Vapnik, 1999) ซึ่งแสดงดังสมการที่ 2-25

$$f(x) = w^T x_i + b \quad (2-25)$$

โดยที่ $f(x)$ คือ ฟังก์ชันเชิงเส้นที่ใช้แทนระนาบตัดสินใจ

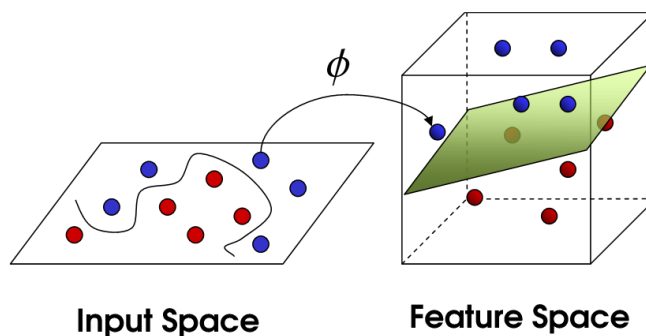
w คือ ค่าเวกเตอร์น้ำหนัก (Weight vector)

b คือ ค่าโน้มน้ำหนัก (Bias)

โดยทั่วไปแล้ว สถาปัตยกรรมของ SVR ได้ออกแบบให้มีการแปลง (Mapping) ข้อมูลจากปริภูมิอินพุต (Input space) ไปเป็นปริภูมิที่สูงขึ้น ซึ่งเรียกว่า ปริภูมิลักษณะเด่น (Feature space) ด้วยฟังก์ชันเคอร์เนล (Kernel function) เพื่อให้สามารถหาฟังก์ชันเชิงเส้นที่เหมาะสมสำหรับ

สมการระนาบตัดสินใจได้ กล่าวคือ แปลงจาก x_i เป็น $K(x_i, x_j)$ ดังภาพที่ 2-18 ทำให้ฟังก์ชันเชิงเส้นในสมการที่ 2-25 เปลี่ยนเป็นดังสมการที่ 2-26

$$f(x) = w^T K(x_i, x_j) + b \quad (2-26)$$



ภาพที่ 2-18 การแมปข้อมูลในปริภูมิอินพุตไปเป็นปริภูมิลักษณะเด่น (Wilimitis, 2018)

ซึ่งฟังก์ชันเคอร์เนลที่นิยมใช้งานใน SVR จะมีดังนี้

1. เคอร์เนลเชิงเส้น (Linear kernel)

$$K(x_i, x_j) = x_i^T x_j \quad (2-27)$$

2. เคอร์เนลโพลิโนเมียล (Polynomial kernel)

$$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d \quad (2-28)$$

3. ฟังก์ชันเรเดียลเบสิส (Radial Basis Function: RBF)

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \quad (2-29)$$

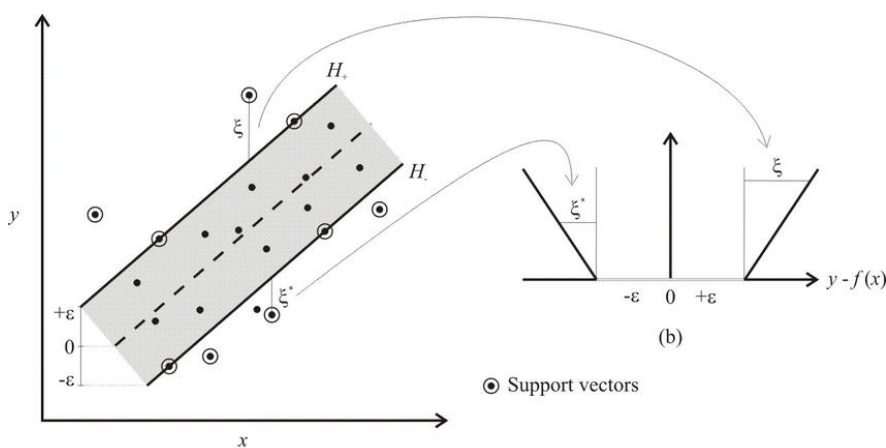
4. เคอร์เนลซิกมอยด์ (Sigmoid kernel)

$$K(x_i, x_j) = \tanh(\gamma x_i^T x_j) + r \quad (2-30)$$

ในการสร้างสมการระนาบตัดสินใจจะพิจารณาค่าความคลาดเคลื่อนที่ยอมรับได้ ซึ่งอยู่ในรูปของฟังก์ชันการสูญเสีย (Loss function) โดยฟังก์ชันที่นิยมนำมาใช้งาน คือ ฟังก์ชันการสูญเสียแบบเอปซิลอนอินเซนซิทีฟ (\mathcal{E} -insensitive loss function) (ภาพที่ 2-19) ดังสมการที่ 2-31 (Vapnik, 1999)

$$L(y_i, f(x_i)) = \begin{cases} 0 & \text{if } |y_i - f(x_i)| \leq \varepsilon \\ |y_i - f(x_i)| - \varepsilon & \text{otherwise} \end{cases} \quad (2-31)$$

โดยที่ $L(y_i, f(x_i))$ คือ ความคลาดเคลื่อนที่เกิดขึ้นระหว่าง y_i และ $f(x_i)$
 ε คือ ค่าความคลาดเคลื่อนที่ยอมรับได้



ภาพที่ 2-19 ฟังก์ชันการสูญเสียแบบเอปซิลอนอินเซนซิทีฟ (Lins et al., 2010)

จากการพิจารณาค่าความคลาดเคลื่อนที่ยอมรับได้นี้ จึงทำให้เกิดเส้นขอบหรือระนาบสองระนาบที่ขนานกับระนาบตัดสินใจ โดยจุดข้อมูลที่อยู่บนระนาบทั้งสองนี้จะเรียกว่า “ซัพพอร์ตเวกเตอร์ (Support vector)” ซึ่งจะเป็นตัวแทนของกลุ่มข้อมูลที่ใช้สำหรับฝึกสอนทั้งหมด นอกจากนี้ยังมีการพิจารณาถึงค่าความคลาดเคลื่อนที่อยู่นอกเหนือระนาบทั้งสอง ซึ่งจะเรียกว่าตัว

แปรส่วนขาด (Slack variable: ξ) ดังนั้นการหาค่า w ที่เหมาะสมที่สุดของสมการระนาบตัดสินใจ จะเป็นการหาค่า Norm) ที่น้อยที่สุดของ w ดังสมการที่ 2-32 ซึ่งอยู่ภายใต้เงื่อนไขของสมการที่ 2-33 (Bagheripour et al., 2015)

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i - \xi_i^*) \quad (2-32)$$

$$\left. \begin{aligned} y_i - (w^T K(x_i, x_j)) - b &\leq \varepsilon + \xi_i \\ (w^T K(x_i, x_j)) - b - y_i &\leq \varepsilon + \xi_i \\ \xi_i, \xi_i^* &\geq 0 \end{aligned} \right\} \quad (2-33)$$

หลักการแก้สมการที่ 2-32 จะใช้ตัวคูณลากรองจ์ (Lagrange multipliers) ทำให้อยู่ในรูปของปัญหาควบคู่ (Dual problem) ดังสมการที่ 2-34 ภายใต้เงื่อนไขของสมการที่ 2-35

$$\max -\frac{1}{2} \sum_{i,j=1}^l (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K(x_i, x_j) + \varepsilon \sum_{i=1}^l (\alpha_i - \alpha_i^*) + \sum_{i=1}^l y_i (\alpha_i - \alpha_i^*) \quad (2-34)$$

$$\sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0, \quad \alpha_i, \alpha_i^* \in [0, C] \quad (2-35)$$

โดยที่ α_i, α_i^* คือ ตัวคูณลากรองจ์
 C คือ ค่าคงที่

ดังนั้นจึงสามารถหาค่าของ w ได้ดังสมการที่ 2-36

$$w = \sum_{i=1}^l (\alpha_i - \alpha_i^*) K(x_i, x_j) \quad (2-36)$$

และเมื่อแทนค่าลงในสมการที่ 2-26 จะได้สมการระนาบตัดสินใจดังสมการที่ 2-37

$$f(x) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) K(x_i, x_j) + b \quad (2-37)$$

3. โครงข่ายประสาทเทียม (Artificial Neural Network: ANN)

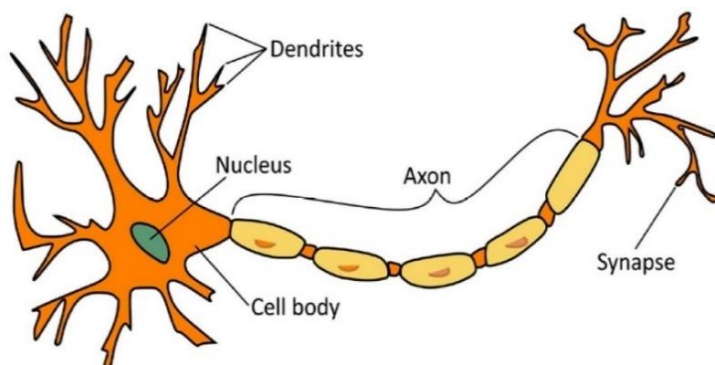
โครงข่ายประสาทเทียมเป็นแบบจำลองทางคณิตศาสตร์ที่ลอกเลียนแบบการทำงานของระบบประสาทของมนุษย์ให้มาอยู่บนเครื่องคอมพิวเตอร์ ทำให้คอมพิวเตอร์สามารถคิดวิเคราะห์ และประมวลผลได้ในลักษณะเดียวกับโครงข่ายประสาทของมนุษย์ (Grossi & Buscema, 2007)

3.1 โครงข่ายประสาทเทียมกับโครงข่ายประสาทในสมองของมนุษย์

แนวคิดเริ่มต้นของโครงข่ายประสาทเทียมนั้นมาจากการลอกเลียนแบบการทำงานของโครงข่ายประสาทในสมองของมนุษย์ โดยโครงข่ายประสาทในสมองของมนุษย์จะประกอบไปด้วยเซลล์ประสาท (Neurons) และจุดประสานประสาท (Synapses) ซึ่งลักษณะในการส่งสัญญาณประสาทก็จะเกิดจากการเชื่อมต่อกันระหว่างเซลล์ประสาทหลายพันล้านเซลล์ที่มาทำงานร่วมกันเป็นเครือข่าย โดยในแต่ละเซลล์ประสาทจะประกอบไปด้วยแขนงของเซลล์ประสาท (Dendrite) ที่ทำหน้าที่เสมือนเป็นหน่วยรับข้อมูลจากเซลล์ประสาทข้างเคียง จากนั้นก็จะถูกทำการประมวลผลข้อมูลด้วยนิวเคลียส (Nucleus) ที่อยู่ภายในตัวเซลล์ (Cell body) และส่งผ่านไปยังแกนของเซลล์ประสาท (Axon) ที่ทำหน้าที่เสมือนเป็นหน่วยส่งออกข้อมูลไปยังเซลล์ประสาทอื่น ๆ (Nwadiugwu, 2020) และในส่วนของโครงข่ายประสาทเทียมก็คือ แบบจำลองทางคณิตศาสตร์ที่เกิดจากการเชื่อมต่อกันของหน่วยประมวลผลย่อย ๆ หลาย ๆ หน่วยมารวมกันเป็นเครือข่ายและทำงานร่วมกัน ซึ่งสามารถเปรียบเทียบเซลล์ประสาทเทียมดังภาพที่ 2-21 กับเซลล์ประสาทในสมองของมนุษย์ดังภาพที่ 2-20 ได้ดังนี้ เพอร์เซ็ปตรอน (Perceptron) หรือ โหนด (Node) คือ ตัวเซลล์ (Cell body), ฟังก์ชันผลรวม (Summation function) และฟังก์ชันกระตุ้น (Activation function) คือ นิวเคลียส (Nucleus), ข้อมูลนำเข้า (Input) คือ แขนงของเซลล์ประสาท (Dendrite), ผลลัพธ์ (Output) คือ ข้อมูลที่ถูกส่งผ่านแกนของเซลล์ประสาท (Axon)



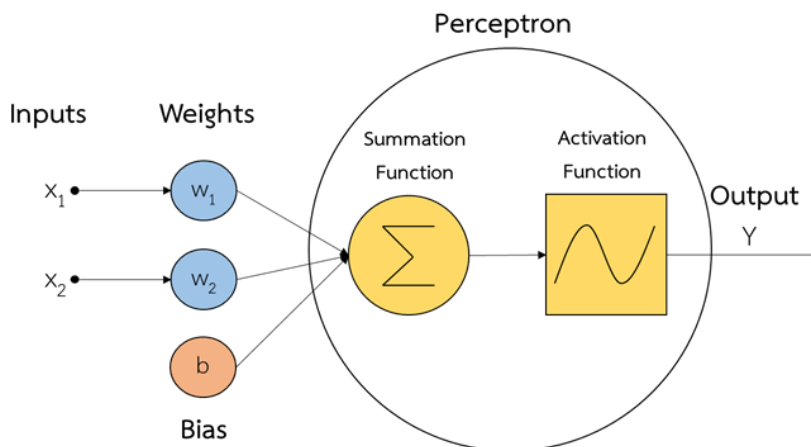
107404525



ภาพที่ 2-20 เซลล์ประสาทในสมองของมนุษย์ (Laubscher & Engelbrecht, 2019)

3.2 เพอร์เซ็ปตรอนชั้นเดียว (Single layer perceptron)

เพอร์เซ็ปตรอนชั้นเดียวเป็นหน่วยย่อยของโครงข่ายประสาทเทียมที่ประกอบไปด้วยเซลล์ประสาทเพียงหนึ่งเซลล์เท่านั้นดังภาพที่ 2-21 (กอบเกียรติ สระอุบล, 2565) ซึ่งองค์ประกอบหลักของเซลล์ประสาทเทียมมีดังนี้



ภาพที่ 2-21 เซลล์ประสาทเทียม (กอบเกียรติ สระอุบล, 2565)

3.2.1 ข้อมูลตัวแปรนำเข้า (Inputs: x) คือ คุณลักษณะของข้อมูลที่จะนำเข้าสู่เซลล์ประสาท ซึ่งเป็นข้อมูลในเชิงปริมาณหรือตัวเลข แต่หากข้อมูลที่นำเข้ามาเป็นรูปภาพหรือข้อมูลเชิงคุณภาพ ก็จะต้องแปลงข้อมูลให้แล้วเสร็จ ก่อนจะนำเข้าสู่เซลล์ประสาทเทียม

3.2.2 ค่าน้ำหนัก (Weights: w) คือ ค่าที่ใช้ในการกำหนดความสำคัญของข้อมูลที่ป้อนเข้าไปในเพอร์เซ็ปตรอนว่ามีผลต่อข้อมูลส่งออกมากน้อยเพียงใด

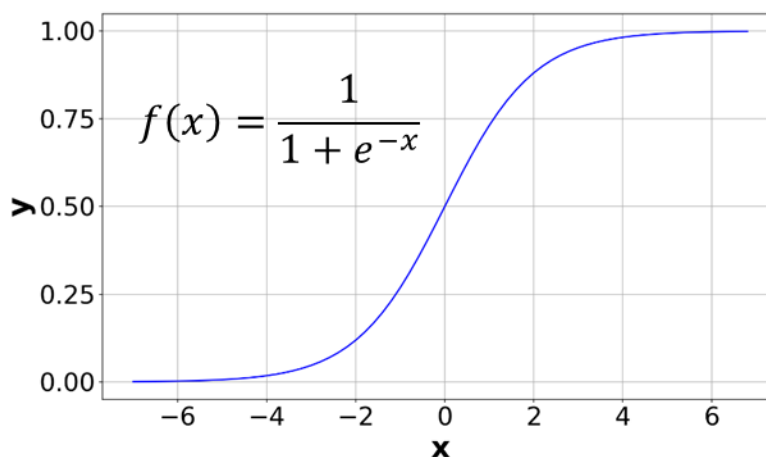
3.2.3 ไบแอส (Bias: b) คือ ค่าที่ใช้ในการช่วยปรับค่า เพื่อให้การคำนวณและการตัดสินใจในเพอร์เซ็ปตรอนมีความถูกต้องมากยิ่งขึ้น

3.2.4 ฟังก์ชันผลรวม (Summation function: S) คือ ฟังก์ชันที่เกิดจากการรวมกันของไบแอสและข้อมูลป้อนเข้าคูณกับค่าน้ำหนัก ซึ่งมีพื้นฐานมาจากสมการเส้นตรง (Linear Function) โดยจะแสดงดังสมการที่ 2-38

$$s = \sum_{i=1}^n w_i x_i + b \quad (2-38)$$

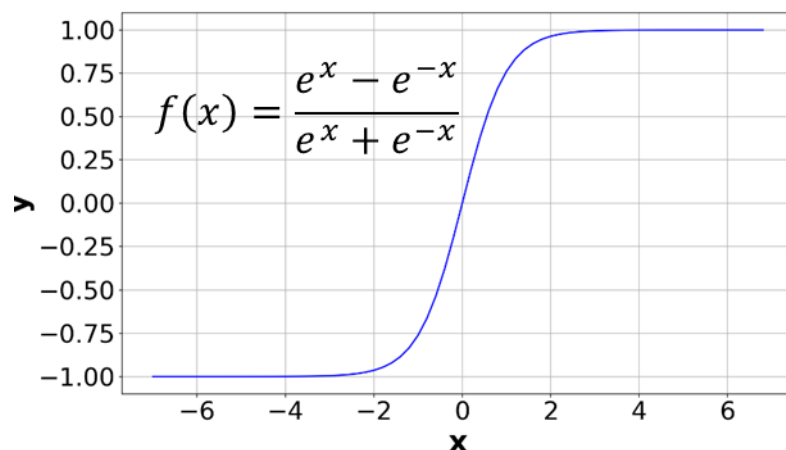
3.2.5 ฟังก์ชันกระตุ้น (Activation function) หรือฟังก์ชันการแปลง (Transfer function) คือ ฟังก์ชันที่รับผลลัพธ์จากฟังก์ชันผลรวมก่อนหน้า แล้วทำการแปลงค่าของข้อมูลเพื่อให้ได้ผลลัพธ์ตามที่ต้องการ ซึ่งมีหลากหลายรูปแบบให้เลือกใช้งานด้วยกัน ดังนี้

3.2.5.1 ฟังก์ชันซิกมอยด์ (Sigmoid function) หรือฟังก์ชันโลจิสติกส์ (Logistic function) เป็นฟังก์ชันที่มีผลลัพธ์จากการแปลงค่าข้อมูลอยู่ในช่วง $[0, 1]$ ดังภาพที่ 2-22



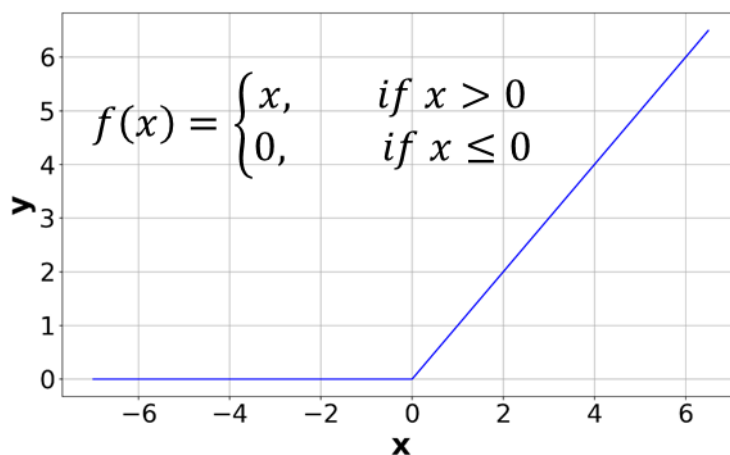
ภาพที่ 2-22 กราฟฟังก์ชันซิกมอยด์ (Van Rossum & Drake, 2009)

3.2.5.2 ฟังก์ชันไฮเพอร์โบลิกแทนเจนต์ (Hyperbolic tangent function) เป็นฟังก์ชันที่มีผลลัพธ์จากการแปลงค่าข้อมูลอยู่ในช่วง $[-1, 1]$ ดังภาพที่ 2-23



ภาพที่ 2-23 กราฟฟังก์ชันไฮเปอร์โบลิกแทนเจนต์ (Van Rossum & Drake, 2009)

3.2.5.3 ฟังก์ชันเรล (Rectifier Linear Unit: ReLU) เป็นฟังก์ชันที่มีผลลัพธ์อยู่ในช่วง $[0, x]$ โดยหากข้อมูลที่เข้ามาในฟังก์ชันมีค่ามากกว่าศูนย์ผลลัพธ์ที่ได้จะเท่ากับ x แต่หากข้อมูลที่นำเข้ามีค่าน้อยกว่าศูนย์ผลลัพธ์ที่ได้จะมีค่าเท่ากับศูนย์ดังภาพที่ 2-24

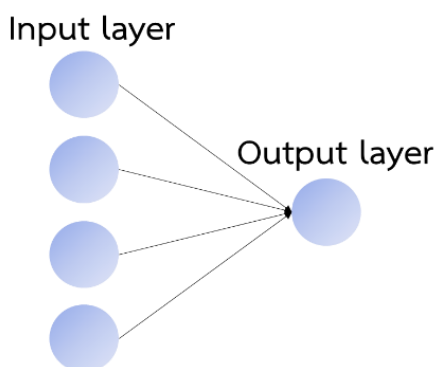


ภาพที่ 2-24 กราฟฟังก์ชันเรล (Van Rossum & Drake, 2009)

3.2.6 ข้อมูลตัวแปรส่งออก (Output) คือ ผลลัพธ์ที่ได้จากการคำนวณในเซลล์ประสาท หรือผลของการทำนาย

จากองค์ประกอบดังกล่าว จึงสามารถแบ่งโครงสร้างของชั้นการทำงาน (Layer) ได้เป็น 2 ชั้น (ภาพที่ 2-25) ดังนี้

1. ชั้นข้อมูลนำเข้า (Input layer) เป็นชั้นที่ทำหน้าที่ในการรับข้อมูลเข้ามาในเซลล์ประสาท และส่งข้อมูลต่อไปยังชั้นข้อมูลส่งออก
2. ชั้นข้อมูลส่งออก (Output layer) เป็นชั้นที่รับข้อมูลมาจากชั้นข้อมูลนำเข้า และทำการแสดงผลลัพธ์ที่ถูกประมวลผลโดยพอร์เซ็ปตรอน



ภาพที่ 2-25 ชั้นการทำงานของพอร์เซ็ปตรอนชั้นเดียว (กอบเกียรติ สระอุบล, 2565)

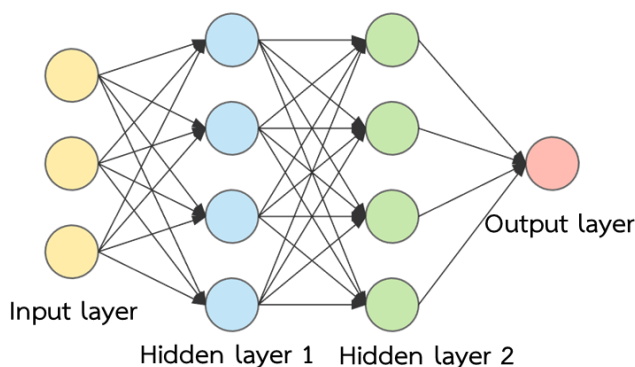
3.3 พอร์เซ็ปตรอนแบบหลายชั้น (Multi-layer perceptron: MLP)

พอร์เซ็ปตรอนแบบหลายชั้น คือ การนำเอาพอร์เซ็ปตรอนชั้นเดียวมาเชื่อมต่อกันเป็นเครือข่ายเพื่อทำงานร่วมกัน ซึ่งอาจกล่าวได้ว่าพอร์เซ็ปตรอนแบบหลายชั้นคือรูปแบบที่แท้จริงของโครงข่ายประสาทเทียม ดังนั้นจึงทำให้มีชั้นของการทำงานเพิ่มเข้ามาอีกหนึ่งชั้นคือ ชั้นแอบแฝง ซึ่งจะอยู่ระหว่างชั้นข้อมูลนำเข้าและชั้นข้อมูลส่งออกดังภาพที่ 2-26 โดยรายละเอียดของโครงสร้างชั้นการทำงานมีดังนี้

3.3.1 ชั้นข้อมูลนำเข้า (Input layer) เป็นชั้นที่ทำหน้าที่ในการรับข้อมูลเข้ามาในโครงข่ายประสาท และจากนั้นก็ส่งข้อมูลต่อไปยังชั้นแอบแฝง

3.3.2 ชั้นแอบแฝง (Hidden layer) เป็นชั้นที่ทำหน้าที่รับข้อมูลจากชั้นข้อมูลนำเข้า และทำการประมวลผลข้อมูลเพื่อส่งต่อไปยังชั้นข้อมูลส่งออก โดยในชั้นนี้สามารถที่จะมีได้มากกว่าหนึ่งชั้น และในแต่ละชั้นก็สามารถมีพอร์เซ็ปตรอนได้มากกว่าหนึ่งพอร์เซ็ปตรอน

3.3.3 ชั้นข้อมูลส่งออก (Output layer) เป็นชั้นที่ทำหน้าที่รับข้อมูลจากชั้นแอบแฝงในชั้นสุดท้าย และทำการแสดงผลลัพธ์ที่ได้จากการประมวลผลในโครงข่ายประสาทเทียม



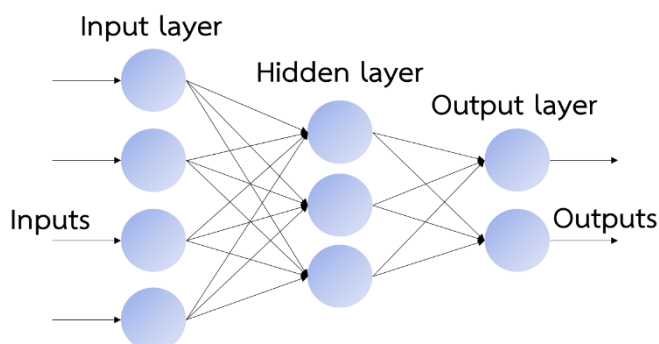
ภาพที่ 2-26 ชั้นการทำงานของเพอร์เซ็ปตรอนแบบหลายชั้น (Luigi, 2020)

3.4 การเรียนรู้แบบแพร่ย้อนกลับ

การเรียนรู้แบบแพร่ย้อนกลับเป็นหนึ่งในวิธีการเรียนรู้ในโครงข่ายประสาทเทียมที่เป็นการนำค่าผิดพลาดที่ได้จากการเปรียบเทียบของผลการทำนาย (Output) กับเป้าหมาย (Target) มาใช้ในการปรับค่าน้ำหนักและไบแอส (ขนาดูติ ประกอบผล, 2552) ซึ่งขั้นตอนการเรียนรู้แบบแพร่ย้อนกลับจะประกอบไปด้วย 2 ขั้นตอน ดังนี้

3.4.1 การแพร่เดินทาง (Forward propagation)

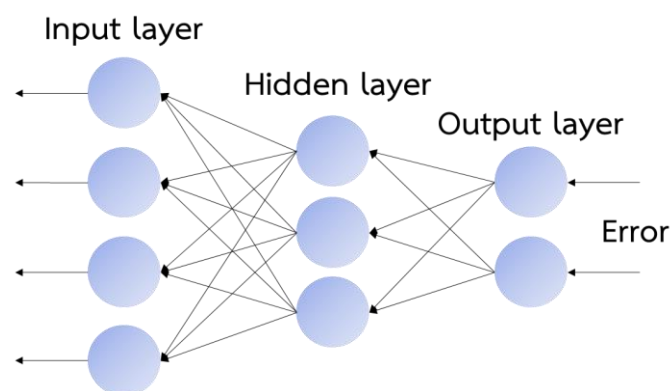
ขั้นตอนของการแพร่เดินทางจะเป็นขั้นตอนที่ดำเนินการไปข้างหน้า โดยในตอนเริ่มต้นเมื่อเซลล์ประสาทรับข้อมูลเข้ามาในชั้นของข้อมูลนำเข้าก็จะทำการสร้างน้ำหนักและไบแอสของโครงข่ายประสาทขึ้นมา ซึ่งเป็นค่าที่ได้จากการสุ่ม จากนั้นจึงส่งต่อไปยังชั้นข้อมูลส่งออกและแสดงผลลัพธ์ของการทำนายหากเป็นเพอร์เซ็ปตรอนชั้นเดียว แต่หากเป็นเพอร์เซ็ปตรอนแบบหลายชั้นก็จะส่งต่อไปยังชั้นแอบแฝงแล้วทำการประมวลผลข้อมูล และสุดท้ายจึงแสดงผลลัพธ์ของการทำนายหรือคำตอบในชั้นของข้อมูลส่งออกดังภาพที่ 2-27



ภาพที่ 2-27 การแพร่เดินทาง (Camuñas-Mesa et al., 2019)

3.4.2 การแพร่ย้อนกลับ (Back propagation)

การแพร่ย้อนกลับจะเป็นขั้นตอนของการปรับปรุงค่าน้ำหนักและไบแอส โดยเมื่อได้ผลลัพธ์จากการทำนายในชั้นข้อมูลส่งออกของขั้นตอนการแพร่เดินหน้าแล้ว ก็จะนำผลลัพธ์ที่ได้ (Output) มาทำการเปรียบเทียบกับค่าผลลัพธ์เป้าหมาย (Target) เพื่อหาค่าผิดพลาด (Error) จากนั้นจึงนำค่าผิดพลาดไปปรับปรุงค่าน้ำหนักและไบแอส โดยทำการแพร่ย้อนกลับดังภาพที่ 2-28 เมื่อแพร่ย้อนกลับไปถึงชั้นข้อมูลนำเข้าแล้ว จึงเริ่มต้นใหม่ในขั้นตอนของการแพร่เดินหน้าอีกครั้ง และจะดำเนินการไป-กลับเช่นนี้ จนกว่าจะได้ค่าน้ำหนักและไบแอสที่ทำให้ผลการทำนายและเป้าหมายมีค่าผิดพลาดน้อยที่สุดหรือสามารถยอมรับได้



ภาพที่ 2-28 การแพร่ย้อนกลับ (Camuñas-Mesa et al., 2019)

ขั้นตอนวิธีการหาค่าเหมาะสมที่สุด (OPTIMIZATION ALGORITHM)

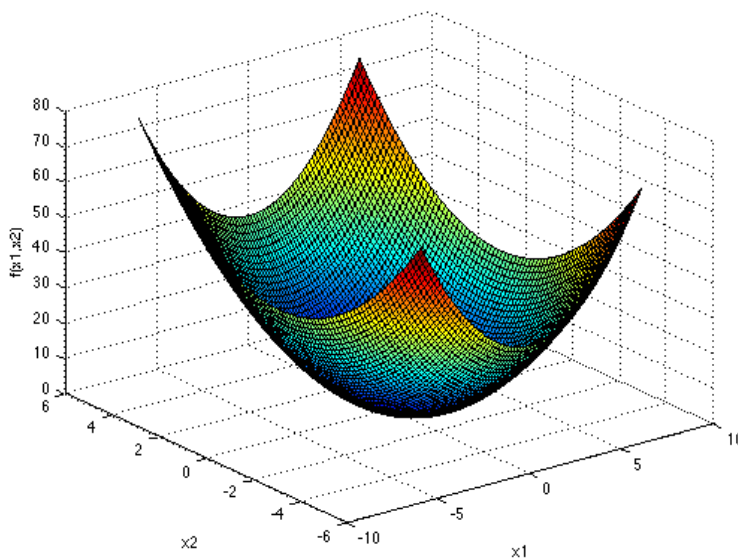
ขั้นตอนวิธีการหาค่าเหมาะสมที่สุด คือ ขั้นตอนวิธีการที่ใช้ในการหาคำตอบของปัญหาการหาค่าเหมาะสมที่สุดซึ่งก็คือ การหาตัวแปรออกแบบ (Design Variable) ที่ทำให้ฟังก์ชันวัตถุประสงค์ (Objective function) มีค่าเหมาะสมที่สุด กล่าวคือ มีค่าสูงสุด (Maximum) หรือค่าต่ำสุด (Minimum) (Yang, 2011) โดยตัวแปรออกแบบ หมายถึง ตัวแปรที่เป็นคำตอบของปัญหาการหาค่าเหมาะสมที่สุด ซึ่งจะมีค่าอยู่ภายในเซตที่ทำให้ฟังก์ชันวัตถุประสงค์มีค่าเหมาะสมที่สุด และฟังก์ชันวัตถุประสงค์ หมายถึง ฟังก์ชันที่ต้องการหาค่าสูงสุดหรือค่าต่ำสุด ซึ่งจะอยู่ในรูปแบบของสมการทางคณิตศาสตร์ที่ติดอยู่ในรูปของตัวแปรออกแบบ โดยจะแสดงดังสมการที่ 2-39

$$f(\mathbf{X}); \mathbf{X} = \begin{Bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{Bmatrix} \quad (2-39)$$

โดยที่ \mathbf{X} คือ เวกเตอร์ออกแบบ (Design vector)

ตัวอย่างของปัญหาการหาค่าเหมาะสมที่สุด เช่น ปัญหาของฟังก์ชันทรงกลม (Sphere function) ดังสมการที่ 2-40 ที่เป็นการหาค่าตัวแปรออกแบบ $x_1, x_2 \in [-\infty, \infty]$ ที่ทำให้ฟังก์ชันมีค่าต่ำสุด ซึ่งหากนำขั้นตอนวิธีการหาค่าเหมาะสมที่สุดมาใช้ในการแก้ปัญหา จะพบว่าคำตอบของปัญหา $x_1, x_2 = (0, 0)$ โดยที่ค่าของฟังก์ชันวัตถุประสงค์ $f(0, 0) = 0$ ดังภาพที่ 2-29

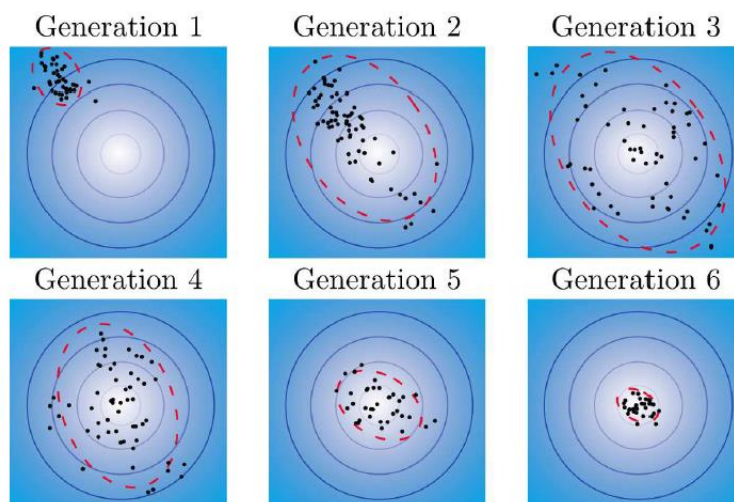
$$f(x_1, x_2) = x_1^2 + x_2^2 \quad (2-40)$$



ภาพที่ 2-29 ฟังก์ชันทรงกลม (Sonja & Derek, 2013)

1. วิธีการวิวัฒนาการการปรับตัวของเมทริกซ์ความแปรปรวนร่วม (Covariance matrix adaptation evolution strategy: CMA-ES)

วิธีการวิวัฒนาการการปรับตัวของเมทริกซ์ความแปรปรวนร่วมเป็นการใช้เทคนิคของเมทริกซ์ความแปรปรวนร่วมในการหาค่าตอบและเหมาะสำหรับปัญหาที่มีความสัมพันธ์แบบไม่เชิงเส้น อีกทั้งยังเป็นวิธีการหาค่าเหมาะสมที่สุดแบบสุ่มโดยไม่ใช้อนุพันธ์ (Derivative-free optimization) (Hansen, 2016) ซึ่งขั้นตอนในการหาค่าตอบของ CMA-ES (ภาพที่ 2-30) มีดังนี้ (โดยตัวแปรที่เขียนอยู่ในเครื่องหมาย {...} แทนเวกเตอร์ที่มีขนาดเท่ากับ $n \times 1$ โดย (n) เป็นจำนวนตัวแปรที่ทำการหาค่าเหมาะสมที่สุด ขณะที่ตัวแปรที่อยู่ในเครื่องหมาย [...] คือ เมทริกซ์ขนาด $n \times n$ และตัวแปรที่อยู่นอกเหนือจากที่กล่าวมาเป็นค่าสเกลาร์)



ภาพที่ 2-30 การหาค่าตอบของ CMA-ES (Tan et al., 2019)

1.1 กำหนดตัวแปรค่าคงที่ ได้แก่ ขนาดประชากรคำตอบ (Population size: λ) ขนาดตัวอย่าง (Sample size) จำนวนคำตอบรุ่นลูก (Number of offspring) และจำนวนคำตอบรุ่นพ่อแม่ (Parent number: μ) จำนวนจุดค้นหาที่ถูกเลือกในประชากร (Number of selected search points in the population) โดยที่ $\mu \leq \lambda$

1.2 กำหนดค่าเริ่มต้นของค่าเฉลี่ย (Mean: m) ขนาดก้าว (Step size: σ) เมทริกซ์ความแปรปรวน $[C] =$ เมทริกซ์เอกลักษณ์ (Identity matrix) $[I]$ เส้นทางวิวัฒนาการแบบไอโซทรอปิก (Isotropic evolution path) $\{p_s\} = \{0\}$ เส้นทางวิวัฒนาการแบบแอนไอโซทรอปิก (Anisotropic evolution path) $\{p_s\} = \{0\}$

1.3 สุ่มคำตอบใหม่แล้วประเมินสร้างคำตอบ x ใหม่จากการแจกแจงแบบปกติหลายตัวแปร (Multivariate normal distribution) ที่มีตำแหน่งค่าเฉลี่ย = $\{m\}$ และเมทริกซ์ความแปรปรวน $[V] = \sigma^2[C]$

1.4 คำนวณค่าความแข็งแรง (Fitness: f_i) ของ $\{x_i\}$ แต่ละตัวจากฟังก์ชันที่ต้องการหาค่าเหมาะสมที่สุด

1.5 จัดเรียง $\{x_1\}, \{x_2\}, \dots, \{x_n\}$ ตามค่าความแข็งแรง โดย $\{x_1\}$ จะให้ค่าความแข็งแรงมากที่สุดดังนั้นจึงเป็นคำตอบที่ดีที่สุด

1.6 อัปเดต $\{m\}$ โดยใช้ค่า $\{x_1\}, \{x_2\}, \dots, \{x_n\}$

1.7 อัปเดต $\{p_c\}$ โดยใช้ค่าปัจจุบันของ $\{p_s\}, \sigma, [C]$ และ $\{m\}-\{m'\}$

1.8 อัปเดต $\{p_s\}$ โดยใช้ค่าปัจจุบันของ $\{p_c\}, \sigma, \{m\}-\{m'\}$ และ $\{p_s\}$

1.9 อัปเดต $[C]$ โดยใช้ค่าปัจจุบันของ $[C], \{p_c\}, \{x_i\}-\{m'\}, \sigma$

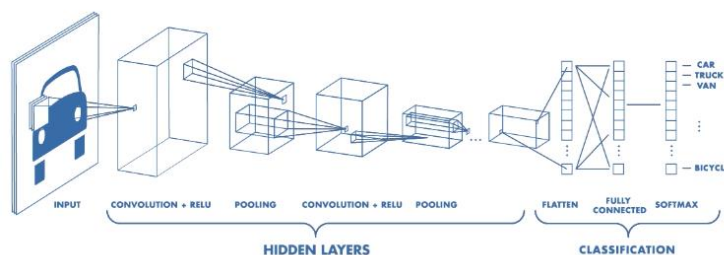
1.10 อัปเดต σ โดยใช้ค่าปัจจุบันของ σ และ $\{p_s\}$

1.11 สิ้นสุดการค้นหาคำตอบ เมื่อจำนวนรอบซึ่งนับจากขั้นตอนที่ 3-10 ครบตามค่าที่กำหนดไว้ และจะได้ $\{x_1\}$ เป็นผลลัพธ์ของการค้นหาคำตอบเหมาะสมที่สุดของ CMA-ES

โครงข่ายประสาทเทียมแบบสังวัตนาการ (CONVOLUTIONAL NEURAL NETWORK: CNN)

โครงข่ายประสาทเทียมแบบสังวัตนาการเป็นโครงข่ายประสาทที่จำลองมาจากระบบประสาทการมองเห็นของมนุษย์ที่มองวัตถุหรือภาพออกเป็นพื้นที่ย่อย ๆ แล้วทำการแยกแยะคุณลักษณะเด่นของพื้นที่ย่อย ๆ นั้นออกมา เช่น เส้นขอบ เส้นโค้ง ลวดลายและพื้นผิว จากนั้นก็จะนำกลุ่มลักษณะเด่นนี้มาผสมรวมกัน เพื่อจำแนกว่าสิ่งที่เห็นอยู่นั้นคืออะไรดังภาพที่ 2-31

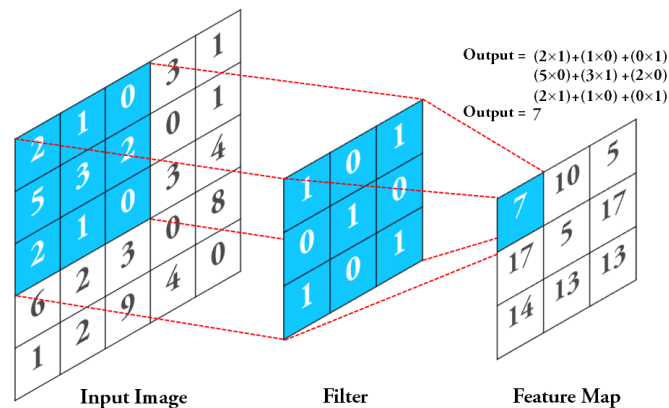
1. การทำงานของโครงข่ายประสาทเทียมแบบสังวัตนาการ



ภาพที่ 2-31 การทำงานของโครงข่ายประสาทเทียมแบบสังวัตนาการ (Sumit, 2018)

ในการทำงานของโครงข่ายประสาทเทียมแบบสังวัตนาการจะประกอบไปด้วย 4 ขั้นตอนหลักดังนี้

1.1 คอนโวลูชัน (Convolution) (ภาพที่ 2-32) มีจุดประสงค์เพื่อดึงคุณลักษณะเด่นที่ใช้ในการรู้จำวัตถุหรือรูปภาพออกมา เช่น การตรวจจับเส้นขอบของวัตถุ การลดจุดรบกวนในภาพ (Noise) และการทำให้ภาพคมชัดขึ้น โดยในขั้นตอนนี้จะเป็นการนำเอาตัวกรอง (Filter) หรือเคอร์เนล (Kernel) มาทาบบกับรูปภาพที่นำเข้ามา และทำการคูณกับค่าพิกเซล (Pixel) ของรูปภาพในรูปแบบของเมทริกซ์ (Matrix) แล้วนำค่าที่ได้มารวมกัน โดยจะเริ่มกระทำจากพิกเซลแรกของรูปภาพ และจะถูกลื่อน (Stride) ไปทาบบนพิกเซลอื่นในภาพทีละพิกเซลจนครบทุกพิกเซลในภาพ ซึ่งผลลัพธ์ที่ได้จากการทำคอนโวลูชันจะได้เมทริกซ์ชุดใหม่ที่เรียกว่า ฝังคุณลักษณะ (Feature map)

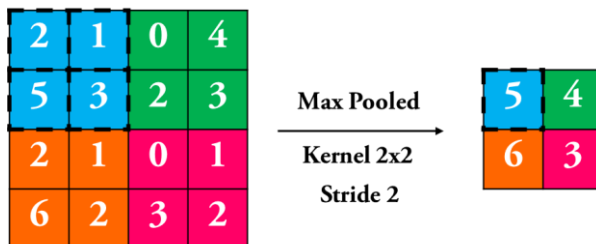


ภาพที่ 2-32 กระบวนการทำคอนโวลูชัน (Kumar, 2021)

1.2 การจัดความเป็นเชิงเส้น (ReLU) เป็นขั้นตอนของการนำฝังคุณลักษณะที่ได้จากการทำคอนโวลูชัน มาจัดความเป็นเชิงเส้น โดยการแปลงพิกเซลที่มีค่าติดลบให้มีค่าเป็นศูนย์ผ่านฟังก์ชันกระตุ้นที่มีชื่อว่า ฟังก์ชันเรลดู ดังนั้นจึงทำให้ค่าที่ได้จากการจัดความเป็นเชิงเส้นจะมีค่าเป็นบวกเท่านั้น

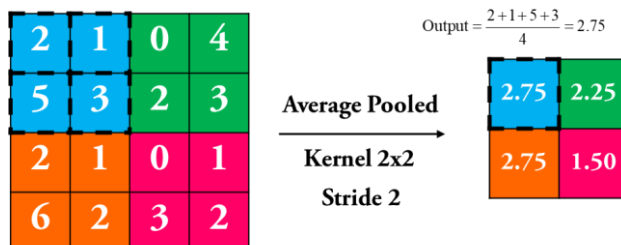
1.3 การพูลลิ่ง (Pooling) จะเป็นการลดมิติของฝังคุณลักษณะให้มีขนาดที่เล็กลง (Down sampling) แต่ยังคงรายละเอียดที่มีความสำคัญไว้อยู่ เนื่องจากในการจำแนกรูปภาพของโครงข่ายประสาทเทียมแบบสังวัตนาการไม่จำเป็นต้องใช้ภาพที่มีขนาดใหญ่หรือความคมชัดของภาพที่มาก แต่ต้องการเพียงคุณลักษณะเด่นของภาพ ก็สามารถที่จะจำแนกแยกแยะได้ว่าวัตถุหรือภาพที่เห็นคืออะไร โดยประเภทของการพูลลิ่ง สามารถแบ่งได้เป็น 2 ประเภทหลัก ๆ คือ

1.3.1 การพูลลิ่งจากค่าสูงสุด (Max pooling) เป็นการนำเอาเคอร์เนลมาทาบบกับฟังก์ชันคุณลักษณะ และหาค่าสูงสุดของการทาบบในแต่ละครั้งดังภาพที่ 2-33 โดยจะเริ่มทาบบจากพิกเซลแรกของฟังก์ชันคุณลักษณะ และทำการเลื่อน (Stride) ไปทาบบบนพิกเซลอื่นจนครบทุกพิกเซลในฟังก์ชันคุณลักษณะ



ภาพที่ 2-33 การพูลลิ่งจากค่าสูงสุด (Roman, 2020)

1.3.2 การพูลลิ่งจากค่าเฉลี่ย (Average pooling) เป็นการนำเอาเคอร์เนลมาทาบบกับฟังก์ชันคุณลักษณะ และหาค่าเฉลี่ยของการทาบบในแต่ละครั้งดังภาพที่ 2-34 โดยจะเริ่มทาบบจากพิกเซลแรกของฟังก์ชันคุณลักษณะ และทำการเลื่อนไปทาบบบนพิกเซลอื่นจนครบทุกพิกเซลในฟังก์ชันคุณลักษณะ

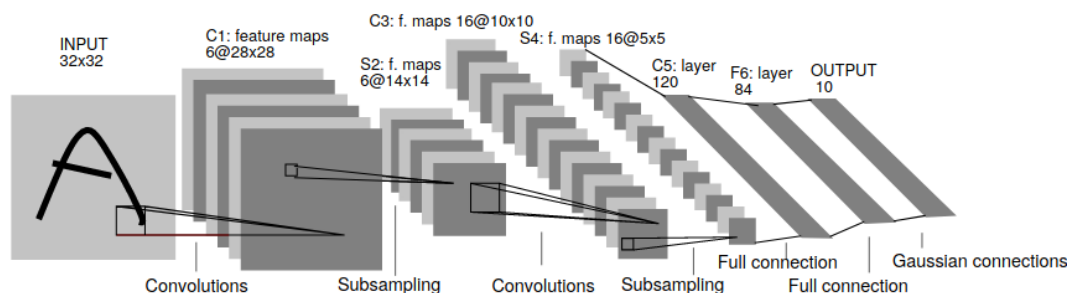


ภาพที่ 2-34 การพูลลิ่งจากค่าเฉลี่ย (Roman, 2020)

1.4 การเชื่อมต่อกันของแต่ละชั้นอย่างสมบูรณ์ (Fully connected layer) เป็นการนำฟังก์ชันคุณลักษณะที่ได้จากการทำคอนโวลูชันหรือการพูลลิ่ง มาทำการลดขนาดมิติให้เหลือเพียงหนึ่งมิติด้วยการแผ่ออก (Flatten) เพื่อให้ข้อมูลถูกเชื่อมต่อกันทั้งหมด จากนั้นจึงป้อนเข้าสู่ชั้นการทำงานของโครงข่ายประสาทเทียม เพื่อใช้ในการเรียนรู้และการทำงานนายผลรูปภาพ ซึ่งผลลัพธ์ของการทำนายจะเป็นประเภทของรูปภาพ พร้อมด้วยค่าที่แสดงถึงความเชื่อมั่นในการทำนาย (Confident)

2. สถาปัตยกรรมของโครงข่ายประสาทเทียมแบบสังวัตนาการ

2.1 โครงสร้างแบบเลเน็ต-5 (LeNet-5 architecture) เป็น CNN ที่ถูกนำเสนอโดย (LeCun et al., 1998) ดังภาพที่ 2-35 ซึ่งมีวัตถุประสงค์เพื่อใช้ในการตรวจจับตัวเลขลายมือที่มาจาก การเขียน โดยโครงสร้างของเลเน็ต-5 จะมีองค์ประกอบดังนี้



ภาพที่ 2-35 โครงสร้างแบบเลเน็ต-5 (LeCun et al., 1998)

2.1.1 Input: รูปภาพขนาด 32x32x1

2.1.2 Convolutional (C1): จำนวน 6 Feature Map, Filter ขนาด 5x5, Stride = 1, Input ขนาด 32x32x1, Output ขนาด 28x28x6

2.1.3 Subsampling (S2): จำนวน 6 Feature Map (Average Pooling), Kernel ขนาด 2x2, Stride = 2, Input ขนาด 28x28x6, Output ขนาด 14x14x6

2.1.4 Convolutional (C3): จำนวน 16 Feature Map, Filter ขนาด 5x5, Stride = 1, Input ขนาด 14x14x6, Output ขนาด 10x10x16

2.1.5 Subsampling Layer 4 (S4): จำนวน 16 Feature Map (Average Pooling), Kernel ขนาด 2x2, Stride = 2, Input ขนาด 10x10x16, Output ขนาด 5x5x16

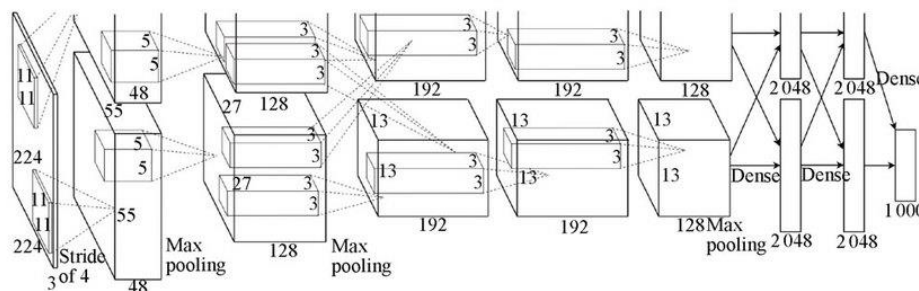
2.1.6 Fully Connected (C5): จำนวน 120 โหนด

2.1.7 Fully Connected (F6): จำนวน 84 โหนด

2.1.8 Output: จำนวน 10 โหนด ใช้สำหรับจำแนกตัวเลข 0-9

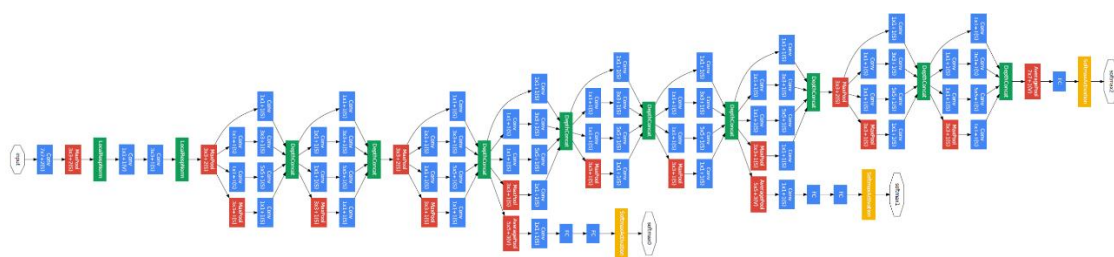
2.2 โครงสร้างแบบอเล็กเน็ต (AlexNet architecture) เป็น CNN ที่มีลักษณะ โครงสร้างคล้ายกับเลเน็ต-5 แต่จะมีขนาดใหญ่กว่าและสามารถจำแนกประเภทได้ถึง 1000 ประเภท ซึ่งถูกนำเสนอโดย (Krizhevsky et al., 2012) ดังภาพที่ 2-36 โดยในปีนั้นอเล็กเน็ต ได้รับรางวัลชนะเลิศในการแข่งขัน ILSVRC-2012 ที่ใช้ข้อมูลจากอิมเมจเน็ต (ImageNet) ซึ่งมีผลการ

ทดสอบค่าผิดพลาด Top-5 error rate อยู่ที่ 15.3 % โครงสร้างของ อเล็กเน็ตจะประกอบไปด้วยชั้นคอนโวลูชันจำนวน 5 ชั้น และมีชั้นการเชื่อมต่อกันอย่างสมบูรณ์ 3 ชั้น



ภาพที่ 2-36 โครงสร้างแบบอเล็กเน็ต (Krizhevsky et al., 2012)

2.3 โครงสร้างแบบกูเกิลเน็ต (GoogLeNet architecture) เป็น CNN ที่มีความลึก 22 ชั้น ซึ่งถูกนำเสนอโดยนักวิจัยของกูเกิล (Szegedy et al., 2015) ดังภาพที่ 2-37 และได้รับรางวัลชนะเลิศในการแข่งขัน ILSVRC-2014 โดยมีผลการทดสอบค่าผิดพลาด Top-5 error rate อยู่ที่ 6.67 % ซึ่งน้อยกว่าโครงสร้างแบบอเล็กเน็ตที่เป็นผู้ชนะเลิศการแข่งขัน ILSVRC ในปี 2012 และ โครงสร้างแบบแซดเอฟเน็ต (ZF-Net) ในปี 2013

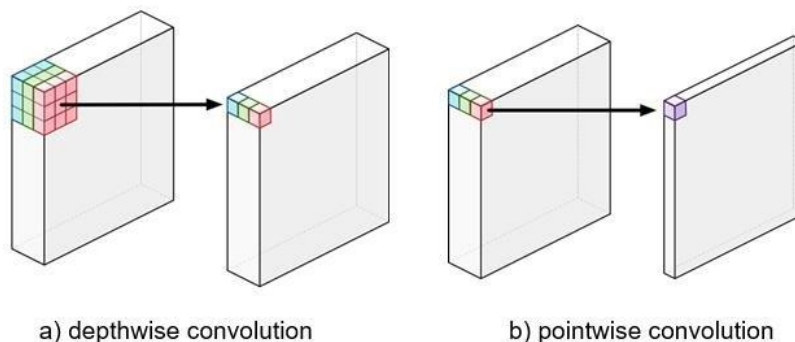


ภาพที่ 2-37 โครงสร้างแบบกูเกิลเน็ต (Szegedy et al., 2015)

2.4 โครงสร้างแบบโมบายเน็ต (MobileNet architecture) เป็น CNN ที่ถูกนำเสนอโดย (Howard et al., 2017) ซึ่งเป็นวิศวกรที่พัฒนาด้าน AI ของ Google โดยโมบายเน็ตได้ถูกออกแบบให้มีขนาดเล็ก เพื่อเน้นใช้งานสำหรับอุปกรณ์พกพา ทำให้สามารถประมวลผลได้อย่างรวดเร็วยิ่งขึ้น แต่ในเรื่องของประสิทธิภาพจะมีค่าลดลงเมื่อเทียบกับโครงสร้างในแบบอื่น ทั้งนี้เนื่องจากการทำคอนโวลูชันของโมบายเน็ตจะมีลักษณะที่แตกต่างจากการทำคอนโวลูชันแบบปกติ โดยจะเป็นการทำคอนโวลูชัน 2 ครั้ง คือ การทำคอนโวลูชันเชิงลึก (Depthwise convolution)



และการทำคอนโวลูชันเชิงจุด (Pointwise convolution) ดังภาพที่ 2-38 ซึ่งการทำคอนโวลูชันเชิงลึก หากภาพที่นำเข้ามาเป็นภาพสี ก็จะทำให้การแยกออกเป็นหนึ่งสีต่อหนึ่งตัวกรอง จากนั้นจึงนำมา รวมกันด้วยการทำคอนโวลูชันเชิงจุด แต่การทำคอนโวลูชันแบบปกติจะเป็นการใช้ตัวกรองหนึ่งตัว ต่อภาพสีทั้งสามสี



ภาพที่ 2-38 การทำคอนโวลูชันเชิงลึกและการทำคอนโวลูชันเชิงจุด (Neha & Sridhar, 2021)

3. การตรวจจับวัตถุ (Object Detection)

การตรวจจับวัตถุเป็นการนำเอาเทคโนโลยีทางด้านคอมพิวเตอร์ที่เกี่ยวข้องกับการมองเห็นของคอมพิวเตอร์ (Computer vision) และการประมวลผลภาพ (Image processing) มาใช้ในการตรวจจับหรือค้นหาวัตถุ (Object) ที่เราสนใจในรูปภาพหรือวิดีโอ เช่น มนุษย์ สัตว์ สิ่งของและรถยนต์ โดยทั่วไปเมื่อพบวัตถุที่เราสนใจแล้ว ก็จะทำให้การตีกรอบ (Bounding box) รอบวัตถุนั้น และบอกถึงประเภทและตำแหน่งของวัตถุว่าอยู่ตรงส่วนไหนของภาพ ซึ่งวัตถุที่สนใจอาจมีมากกว่าหนึ่งประเภท และจำนวนของวัตถุที่สนใจอาจมีมากกว่าหนึ่งจำนวนก็ได้ดังภาพที่ 2-39

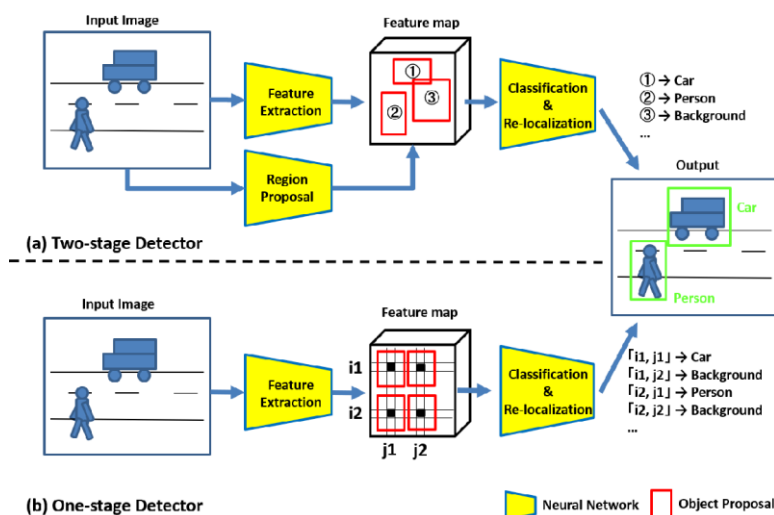


ภาพที่ 2-39 การตรวจจับวัตถุ (Prakhar, 2019)

โดยในการตรวจจับวัตถุสามารถแบ่งวิธีการตรวจจับตามลักษณะการประมวลผลได้เป็น 2 วิธี (ภาพที่ 2-40) ดังนี้

1. วิธีการประมวลผลแบบสองครั้ง (Two-stage detector) เป็นวิธีที่มีการแบ่งการประมวลผลเป็นสองขั้นตอน โดยการประมวลผลในขั้นแรกจะเป็นการดึงคุณลักษณะ (Feature extraction) ที่ใช้ในการรู้จำวัตถุออกมา ซึ่งจะได้เป็นฟังก์ชันลักษณะ อีกทั้งยังมีการค้นหาขอบเขต (Region) ที่คาดว่าจะมีวัตถุที่สนใจอยู่ ซึ่งสิ่งที่ได้ก็คือ ขอบเขตที่สนใจ (Region proposal) ที่มีขอบเขตตำแหน่ง (Localization) ของวัตถุ จากนั้นการประมวลผลในขั้นตอนที่สอง ก็จะเป็นการจำแนกว่าวัตถุนั้นคืออะไร โดยวิธีที่ใช้หลักการของการประมวลผลแบบสองครั้ง เช่น R-CNN, Fast-RCNN และ Faster-RCNN

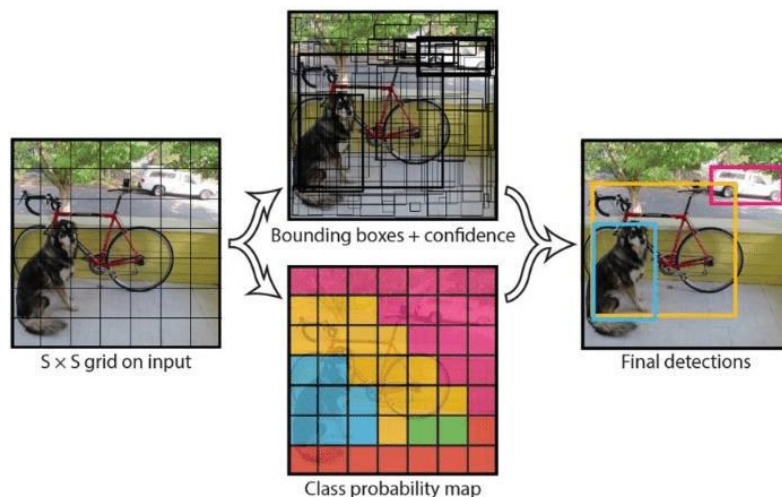
2. วิธีการประมวลผลแบบครั้งเดียว (Single-stage detector) เป็นวิธีการตรวจจับที่มีความรวดเร็วในการประมวลผลมากกว่าวิธีการประมวลผลแบบสองครั้ง เนื่องจากในการจำแนกและบ่งบอกตำแหน่งของวัตถุจะเป็นการประมวลผลเพียงขั้นตอนเดียวเท่านั้น ซึ่งโดยทั่วไปแล้วนิยมใช้กับการตรวจจับวัตถุแบบทันที (Real-time) เช่น การตรวจจับรถบนท้องถนน การตรวจจับผู้คนในเมือง โดยวิธีที่ใช้หลักการของการประมวลผลแบบครั้งเดียว ได้แก่ YOLO และ SSD



ภาพที่ 2-40 ความแตกต่างระหว่างวิธีการประมวลผลแบบสองครั้งและแบบครั้งเดียว (Phan & Yamamoto, 2020)

3.1 You Only Look Once (YOLO)

YOLO เป็นอัลกอริทึมการตรวจจับวัตถุที่มีการประมวลผลแบบครั้งเดียว ซึ่งถูกนำเสนอโดย (Szegedy et al., 2015) มีจุดเด่นในเรื่องของความเร็วและความแม่นยำสูง โดยขั้นตอนการทำงานของ YOLO (ภาพที่ 2-41) มีดังนี้



ภาพที่ 2-41 หลักการทำงานของ YOLO (Wai et al., 2019)

3.1.1 นำรูปภาพที่ป้อนเข้ามาแบ่งออกเป็นตารางกริด (Grid) ที่มีขนาดเท่ากับ $S \times S$

3.1.2 ระบุค่า (Label) ให้แต่ละตารางกริด ซึ่งเป็นค่าเวกเตอร์ ดังนี้

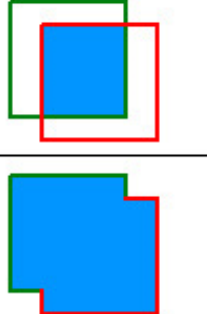
$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \end{bmatrix}$$

โดยที่ p_c คือ ค่าที่บ่งบอกว่ามีวัตถุที่สนใจอยู่ในตารางหรือไม่ ถ้ามีวัตถุค่า $p_c = 1$ แต่ถ้าไม่มีวัตถุค่า $p_c = 0$

b_x, b_y คือ ตำแหน่งกึ่งกลางของวัตถุ

b_h, b_w คือ ขนาดความสูงและความกว้างของวัตถุ
 c_1, c_2 คือ ค่าที่บ่งบอกว่าวัตถุอยู่ในประเภทใด

3.1.3 กำหนดค่า Intersection over Union (IoU) เนื่องจากผลลัพธ์ของการทำนาย อาจมีขอบเขตที่ทำนายออกมาเป็นจำนวนมาก ดังนั้นจึงต้องกำหนดหาขอบเขตที่เหมาะสมที่สุดด้วยค่า IoU ที่เกิดจากการนำพื้นที่ที่เกยทับกันหารด้วยพื้นที่รวมทั้งหมกระหว่างขอบเขตที่ทำนายได้และขอบเขตจริงดังภาพที่ 2-42

$$IOU = \frac{\text{area of overlap}}{\text{area of union}} = \frac{\text{Area of the intersection of the two boxes}}{\text{Area of the union of the two boxes}}$$


ภาพที่ 2-42 การคำนวณค่า IoU (Padilla et al., 2020)

งานวิจัยที่เกี่ยวข้อง

Kumar et al. (2017) ได้นำเสนอเทคนิค Soft Computing โดยใช้ขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm: GA) และแบบจำลองข้อต่อ (Model joints) ในการหาค่าเหมาะสมที่สุดการเคลื่อนที่ของแขนหุ่นยนต์ที่มีจำนวนองศาอิสระแตกต่างกัน ผลการวิจัยพบว่าเทคนิคที่นำเสนอสามารถหาค่าเหมาะสมที่สุดการเคลื่อนที่ของแขนหุ่นยนต์สำหรับค่าตัวแปรนำเข้าที่เป็นไปได้ทั้งหมด อีกทั้งยังสามารถไปถึงเป้าหมายได้ภายในระยะเวลาของการจำลอง

Guo et al. (2019) ได้นำเสนอขั้นตอนวิธีการเรียนรู้แบบเสริมกำลังเชิงลึก Deep deterministic policy gradient (DDPG) ในการประยุกต์ใช้สำหรับการแก้ปัญหากลศาสตร์แบบผกผันของแขนหุ่นยนต์ 5 องศาอิสระ ซึ่งจำลองด้วย Tensorflow และ Matplotlib เพื่อประเมินความแม่นยำของวิธีการที่นำเสนอ ผลการทดสอบพบว่าวิธีการที่นำเสนอสามารถคำนวณหามุมของทุกข้อต่อได้โดยมีข้อผิดพลาดอยู่ในช่วงที่ยอมรับได้ อีกทั้งยังหาค่าตอบได้ง่ายกว่าวิธีการคำนวณแบบดั้งเดิม



107404525

Dalla Libera et al. (2020) ได้นำเสนอกรอบการเรียนรู้ที่ไม่ใช้ข้อมูลสำหรับการเรียนรู้แบบเสริมกำลังโมเดลเบสซึ่งมีพื้นฐานจากการถดถอยของกระบวนการเกาส์เซียน และได้มีการเปรียบเทียบกับกรอบการถดถอยของกระบวนการเกาส์เซียนที่ใช้ข้อมูลและวิธีการที่ไม่ใช้ข้อมูลถึงพารามิเตอร์ โดยได้นำไปทดลองกับหุ่นยนต์จริง 2 ระบบที่พิจารณาเพียงตำแหน่ง ไม่พิจารณาถึงความเร็วและความเร่งที่เกิดขึ้น ผลการทดลองแสดงให้เห็นว่ากรอบการเรียนรู้ที่ได้นำเสนอมีประสิทธิภาพเหนือกว่าการถดถอยของกระบวนการเกาส์เซียนที่ใช้ข้อมูล และวิธีการที่ไม่ใช้ข้อมูลถึงพารามิเตอร์นั้นก็มีความแม่นยำเพียงพอสำหรับการแก้ปัญหาที่นำมาทดลอง

Joshi et al. (2020) ได้นำเสนอวิธีการเรียนรู้แบบเสริมกำลังเชิงลึก Double deep Q-learning (DDQN) ร่วมกับ Grasp-Q-Network สำหรับการแก้ไขปัญหาการจับวัตถุของแขนหุ่นยนต์ Baxter โดยใช้สัญญาณป้อนกลับจากมอเตอร์และกล้องหลายตัวในการตรวจจับวัตถุ ซึ่งมีการทดสอบทั้งปัญหาการจำลองและปัญหาจริง จากผลการทดสอบพบว่าวิธีการที่นำเสนอมีประสิทธิภาพที่เหนือกว่าขั้นตอนวิธีการ Q-learning และ Deep Q Learning (DQN) ด้วยความสำเร็จในการจับสูงสุด อีกทั้งยังมีความแม่นยำในการจับที่มากขึ้นเมื่อมีการใช้กล้องหลายตัวเปรียบเทียบกับการใช้กล้องตัวเดียวในการตรวจจับ

Jiang et al. (2021) ได้นำเสนอวิธีการเรียนรู้แบบเสริมกำลังเชิงลึกที่มีพื้นฐานมาจากอัลกอริทึม DDPG แบบขอมมาตรสำหรับการเรียนรู้ของแขนหุ่นยนต์โดยมุ่งไปยังการทำรูปภาพของเป้าหมายให้มีประสิทธิภาพ ซึ่งได้นำมาทดสอบกับปัญหาการวางและปัญหาการไปถึงที่เป็นการจำลอง ก่อนจะนำโมเดลที่ได้ฝึกสอนแล้วมาใช้กับแขนหุ่นยนต์จริง ผลการทดสอบพบว่าขั้นตอนวิธีการที่นำเสนอสามารถนำไปใช้กับปัญหาการไปถึงและการวางได้โดยมีอัตราความสำเร็จเท่ากับ 95% และ 98% ภายใต้อุปกรณ์ที่แตกต่างกัน และในการนำโมเดลที่ฝึกสอนแล้วมาใช้กับแขนหุ่นยนต์จริงก็พบว่าสามารถนำไปใช้งานได้โดยไม่ต้องฝึกสอนใหม่ แต่มีประสิทธิภาพที่ลดลงในระดับหนึ่ง

Sekkat et al. (2021) ได้นำเสนอขั้นตอนวิธีการเรียนรู้แบบเสริมกำลังเชิงลึก Deep Deterministic Policy Gradient (DDPG) ในการให้แขนหุ่นยนต์เคลื่อนที่เข้าไปจับวัตถุที่สนใจได้อย่างอัตโนมัติโดยใช้กล้องในการตรวจจับวัตถุ ซึ่งแขนหุ่นยนต์ที่นำมาใช้ในการทดสอบมีจำนวนองศาอิสระเท่ากับ 5 และมีการตรวจจับวัตถุโดยใช้ขั้นตอนวิธีการ YOLOv5 อีกทั้งยังมีการดึงค่าพิกัด 3 มิติของวัตถุด้วยขั้นตอนวิธีการ Backward projection เพื่อใช้ในการคำนวณมุมข้อต่อด้วยจลนศาสตร์แบบผกผัน และนำมาฝึกสอนด้วย DDPG ผลการทดสอบพบว่าวิธีการที่นำเสนอสามารถทำให้แขนหุ่นยนต์ไปถึงเป้าหมายได้หลังจากผ่านการฝึกสอนไป 400 Episodes ด้วยความแม่นยำ 95.5%

Li & Wang (2021) ได้นำเสนอขั้นตอนวิธีการเรียนรู้แบบเสริมกำลังเชิงลึกโมเดลฟรี DDPG เปรียบเทียบกับขั้นตอนวิธีการ NAF เพื่อใช้สำหรับการควบคุมแขนหุ่นยนต์ภายใต้ข้อจำกัด ลี้อย่างในการจับวัตถุที่มีรูปร่างไม่เฉพาะเจาะจง โดยทำการทดสอบกับแขนหุ่นยนต์ UR10 ที่เป็น ปัญหาแบบจำลอง ผลการทดสอบพบว่าขั้นตอนวิธีการ NAF ให้ประสิทธิภาพที่ดีกว่า DDPG เมื่อ ใช้ปริมาณข้อมูลที่เท่ากัน

Gomes et al. (2021) ได้นำเสนอวิธีการเรียนรู้แบบเสริมกำลังเชิงลึก Deep Q Learning (DQN) ที่ประยุกต์ใช้สำหรับแขนหุ่นยนต์ที่ทำงานร่วมกับมนุษย์ (Cobot) ในงานจับและวางวัตถุ โดยใช้กล้อง 3 มิติ ซึ่งได้ดำเนินการทดสอบผ่านปัญหาแบบจำลอง พร้อมทั้งมีการเปรียบเทียบ ประสิทธิภาพของโครงข่ายประสาทเทียมแบบสังวัตนาการ (CNN) ลีชนิดที่ใช้ในการประมาณค่า Q-Value ได้แก่ RexNext, MobileNet, MNASNet และ DenseNet ผลการทดสอบพบว่า MobileNet ให้ประสิทธิภาพที่ดีที่สุดด้วยความแม่นยำเฉลี่ย 84%

Rokbani et al. (2022) ได้นำเสนอขั้นตอนวิธีการ modified PSO (m-PSO) และ multi-objective modified PSO (MO-m-PSO) เพื่อแก้ปัญหาการลดขนาดแบบผกผันของแขนหุ่นยนต์ 5 องศาอิสระ โดยมุ่งเน้นไปที่ปัญหาการหาค่าเหมาะสมที่สุดที่มีหลายวัตถุประสงค์ที่เป็นการพิจารณา ทั้งตำแหน่งและทิศทางของปลายแขนหุ่นยนต์ อีกทั้งยังมีการเปรียบเทียบกับขั้นตอนวิธีการ MO-PSO, MO-JAYA และ MO-CSA ผลการทดสอบพบว่าขั้นตอนวิธีการ MO-m-PSO สามารถแก้ไข ปัญหาการลดขนาดแบบผกผันของแขนหุ่นยนต์ 5 องศาอิสระได้ โดยให้ผลลัพธ์ที่แม่นยำมากกว่า ขั้นตอนวิธีการ MO-PSO, MO-JAYA และ MO-CSA

Zhang & Zheng (2022) ได้นำเสนอขั้นตอนวิธีการเรียนรู้แบบเสริมกำลังเชิงลึก PPO ที่ ถูกปรับปรุง โดยมุ่งเน้นไปที่การกำหนดรางวัล (Reward) ของปัญหาการควบคุมแขนหุ่นยนต์ 6 องศาอิสระในการจับวัตถุโดยใช้กล้อง ซึ่งดำเนินการทดสอบผ่านปัญหาแบบจำลอง อีกทั้งยังมีการ เปรียบเทียบกับขั้นตอนวิธีการ PPO แบบดั้งเดิม ผลการทดสอบพบว่าวิธีการที่นำเสนอสามารถเข้าสู่ ลู่คำตอบได้อย่างรวดเร็วและมีความแม่นยำที่มากกว่าขั้นตอนวิธีการ PPO แบบดั้งเดิม



107404525

บทที่ 3

ระเบียบวิธีวิจัย

ในการดำเนินงานวิจัย จะเริ่มต้นจากการนำวิธี MBRL มาใช้ทดสอบสำหรับปัญหาการศึกษาเชิงตัวเลข (Numerical study) เพื่อทดสอบขั้นตอนวิธีการที่ได้นำเสนอในเบื้องต้น ก่อนจะนำไปใช้ทดสอบกับปัญหาจริง จากนั้นในการทดสอบกับปัญหาจริง จะมีการแบ่งการดำเนินงานออกเป็นสองส่วน คือ ส่วนที่หนึ่ง การฝึกสอน โมเดลการตรวจจับตำแหน่งวัตถุ (Object) ตำแหน่งเป้าหมาย (Target) และตำแหน่งมือจับ (Gripper) ของแขนหุ่นยนต์ เพื่อนำมาใช้เป็นข้อมูลในการเรียนรู้ของวิธี MBRL และส่วนที่สองจะเป็นการนำวิธี MBRL มาทดสอบกับแขนหุ่นยนต์จริง

อุปกรณ์ที่ใช้ในการทดสอบ

1. กล้อง 3 มิติ (3D Camera)

การตรวจจับพิกัดตำแหน่งวัตถุ ตำแหน่งเป้าหมาย และตำแหน่งมือจับของแขนหุ่นยนต์ จะประยุกต์ใช้กล้อง 3 มิติที่มีชื่อว่า OpenCV AI Kit-Depth หรือ OAK-D ซึ่งเป็นกล้องปัญญาประดิษฐ์เชิงพื้นที่ (Spatial AI camera) ที่สามารถมองเห็นและตีความวัตถุที่อยู่ตรงหน้าได้ โดยเรียกใช้โครงข่ายประสาทเทียมในการตรวจจับวัตถุจากกล้องสี (Color camera) ที่มีความละเอียดสูงถึง 4K ซึ่งอยู่ตรงกลางของกล้อง พร้อมทั้งบอกถึงคุณลักษณะของวัตถุที่กำลังตรวจจับได้จากกล้องแบบสเตอริโอสองตัว (Stereo cameras) ที่อยู่ด้านข้างดังภาพที่ 3-1



ภาพที่ 3-1 กล้อง OAK-D

2. แขนหุ่นยนต์ (Robotic arm)

สำหรับแขนหุ่นยนต์ที่นำมาใช้ในการทดสอบจะเป็นแขนหุ่นยนต์ 6 องศาอิสระ ที่มีชื่อว่า Braccio Robotic Arm แสดงดังภาพที่ 3-2 ซึ่งออกแบบมาให้ควบคุมผ่านบอร์ดอาคิโน (Arduino Board) เหมาะสำหรับนำมาใช้ในการศึกษา สามารถประกอบนำไปใช้ได้หลากหลายงาน เช่น การเคลื่อนย้ายสิ่งของ การติดตั้งกล้องและการติดตั้งแผงโซลาร์เซลล์ แต่ในงานวิจัยนี้จะติดตั้งมือจับ (Gripper) ที่มีลักษณะคล้ายกับนิ้วมือมนุษย์ 2 นิ้วจับ และทดสอบการควบคุมแขนหุ่นยนต์เพียง 4 องศาอิสระ เพื่อใช้ในการหยิบจับวัตถุและโยกย้ายไปยังตำแหน่งของเป้าหมายเพื่อวางวัตถุ



ภาพที่ 3-2 แขนหุ่นยนต์ Braccio

การออกแบบชุดการทดลอง

ในการออกแบบชุดการทดลองจะเป็นการนำเอาแขนหุ่นยนต์และกล้อง 3 มิติมาทำงานร่วมกันดังภาพที่ 3-3 โดยออกแบบให้กล้อง 3 มิติยึดติดกับโครงติดตั้งกล้องที่ทำจากท่อพีวีซี (Polyvinylchloride: PVC) เนื่องจากมีราคาถูก สามารถหาซื้อได้ง่ายและมีความแข็งแรง สามารถรองรับน้ำหนักของกล้อง 3 มิติได้ ซึ่งอยู่ด้านบนของแขนหุ่นยนต์ โดยมีระยะห่างระหว่างพื้นถึงหน้ากล้องเท่ากับ 125.5 ซม. เนื่องจากเป็นระยะความสูงที่มีความเหมาะสมและอยู่ในช่วงที่กล้องสามารถตรวจจับได้

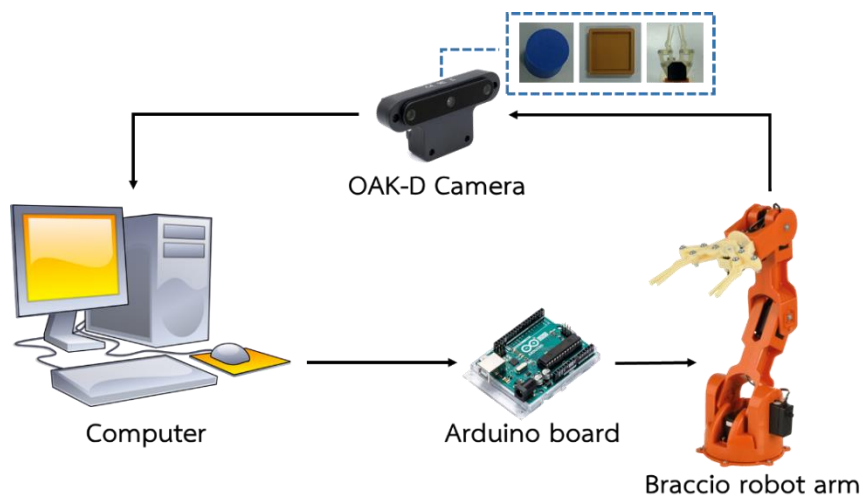


ภาพที่ 3-3 ชุดการทดลอง

การทำงานของชุดการทดลอง

ในส่วนการทำงานของชุดการทดลอง จะประกอบไปด้วย 3 ส่วน คือ ส่วนที่หนึ่ง คอมพิวเตอร์ที่ใช้ในการส่งค่าของมุมการหมุนไปยังบอร์ดอาคยโน้ผ่าน Serial port และทำหน้าที่ประมวลผลวิธี MBRL ส่วนที่สอง บอร์ดอาคยโน้ที่ทำหน้าที่รับข้อมูลจากคอมพิวเตอร์และแปลงค่าของมุมการหมุนเป็นสัญญาณทางไฟฟ้า เพื่อควบคุมการเคลื่อนที่ของแขนหุ่นยนต์ และส่วนที่สาม กล้อง OAK-D ที่ใช้ในการตรวจจับพิกัดตำแหน่งวัตถุ ตำแหน่งเป้าหมายและตำแหน่งมือจับของแขนหุ่นยนต์ เพื่อส่งค่าไปยังส่วนของการประมวลผลวิธี MBRL โดยใช้หลักการของ Client-Server ซึ่ง Client เปรียบเสมือนโปรแกรมการประมวลผลวิธี MBRL ที่ร้องขอข้อมูลจาก Server ที่เป็นโปรแกรมการตรวจจับ ซึ่งดำเนินการผ่านคอมพิวเตอร์เครื่องเดียวกันแสดงดังภาพที่ 3-4



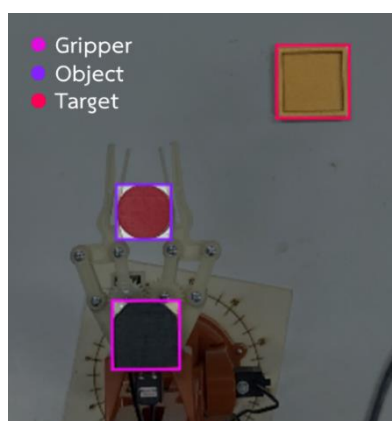


ภาพที่ 3-4 การทำงานของชุดการทดลอง

การสร้างโมเดลการตรวจจับวัตถุ

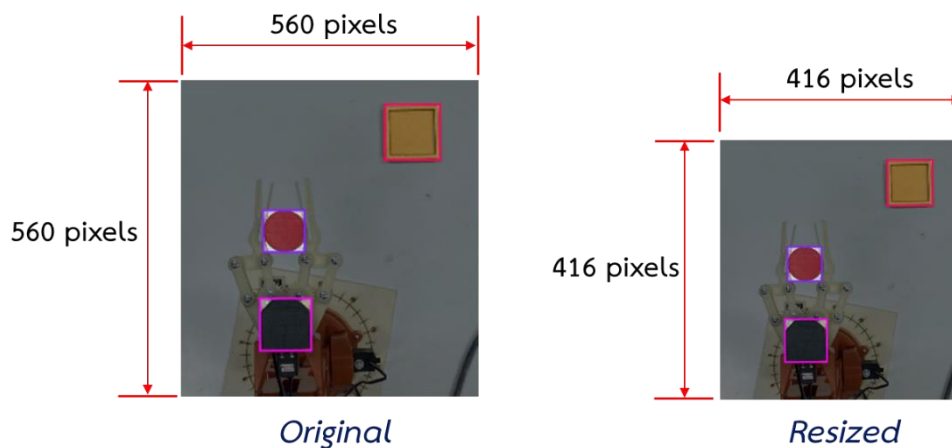
งานวิจัยนี้ได้ประยุกต์ใช้วิธี YOLO v5 ในการสร้างโมเดลการตรวจจับวัตถุ เนื่องจากมีการประมวลผลที่รวดเร็วและมีความแม่นยำสูง อีกทั้งยังเหมาะสำหรับการใช้งานกับกล้อง 3 มิติ ซึ่งขั้นตอนในการสร้างโมเดลการตรวจจับวัตถุมีดังนี้

1. จัดเตรียมรูปภาพของวัตถุ เป้าหมายและมือจับที่ใช้ในการตรวจจับ จากนั้นจึงทำการสร้างกรอบรูปภาพ (Bounding box) พร้อมระบุประเภท (Label image) ในรูปภาพด้วยโปรแกรม Roboflow ที่เป็นเครื่องมือสำหรับจัดเตรียมชุดข้อมูล แสดงดังภาพที่ 3-5



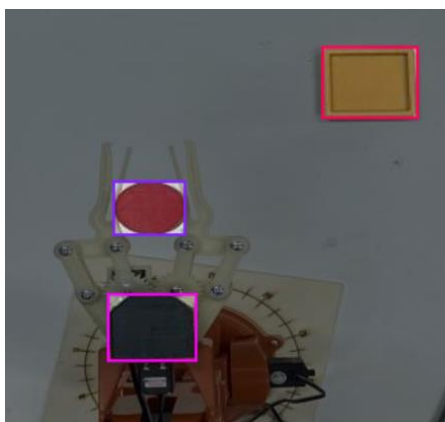
ภาพที่ 3-5 การสร้างกรอบรูปภาพและระบุประเภทของวัตถุที่สนใจ

2. ประมวลผลภาพ (Image processing) ด้วยการปรับลดขนาดภาพก่อนนำไปฝึกสอน เพื่อเพิ่มประสิทธิภาพและลดเวลาในการฝึกสอน แสดงดังภาพที่ 3-6



ภาพที่ 3-6 การปรับขนาดภาพ

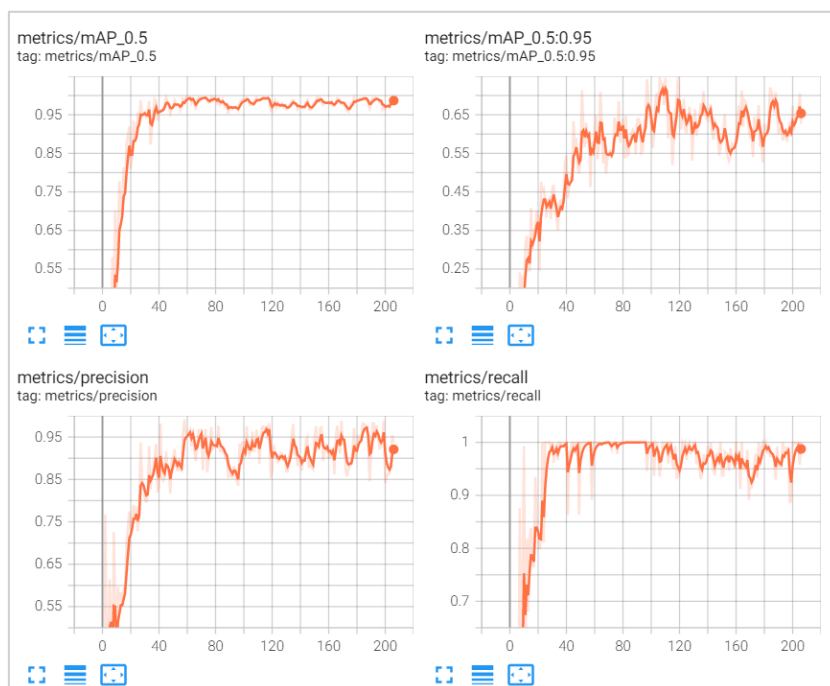
3. เพิ่มปริมาณข้อมูลภาพสอน (Data augmentation) เนื่องจากในบางครั้งข้อมูลรูปภาพที่นำมาฝึกสอนอาจมีน้อยเกินไป ซึ่งหากนำข้อมูลนี้ไปใช้ในการฝึกสอน โมเดลทันที อาจทำให้ประสิทธิภาพความแม่นยำของการตรวจจับนั้นมีค่าต่ำ ดังนั้นปริมาณของข้อมูลจึงเป็นสิ่งสำคัญที่จะส่งผลต่อประสิทธิภาพความแม่นยำในการตรวจจับ ซึ่งเทคนิคของการเพิ่มปริมาณข้อมูลภาพสอน เช่น การพลิกรูปภาพในทิศทางซ้าย/ขวา การหมุนรูปภาพซ้าย/ขวาและการย่อ/ขยายรูปภาพดังภาพที่ 3-7



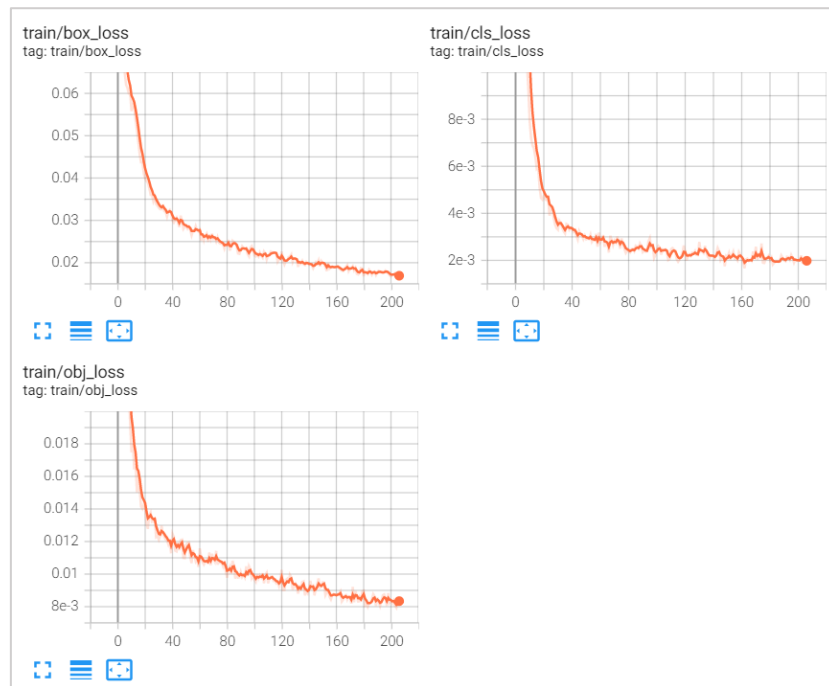
ภาพที่ 3-7 การเพิ่มปริมาณข้อมูลภาพสอนด้วยการขยายรูปภาพ

4. นำข้อมูลรูปภาพที่ได้มาทำการฝึกสอนด้วยวิธี YOLO v5 ใน Google Colab โดยก่อนการฝึกสอนสามารถกำหนดค่าของตัวแปรที่มีผลต่อประสิทธิภาพของโมเดลได้ เช่น ขนาดรูปภาพ (Image size) จำนวนข้อมูลที่ให้โมเดลคำนวณแต่ละครั้งของรอบการฝึกสอน (Batch size) จำนวนรอบการฝึกสอน (Epoch)

5. ประเมินผลโมเดลจากกราฟความแม่นยำและกราฟความผิดพลาดของโมเดลแสดงดังภาพที่ 3-8 และภาพที่ 3-9 เนื่องจากโมเดลที่ฝึกสอนเรียบร้อยแล้วอาจยังมีประสิทธิภาพที่ต่ำ ดังนั้นจึงควรดูผลการประเมินก่อนจะนำโมเดลไปใช้งาน โดยหากโมเดลยังมีประสิทธิภาพต่ำ จึงทำการฝึกสอนเพิ่ม หรืออาจฝึกสอนโมเดลใหม่โดยการเพิ่มจำนวนข้อมูลหรือกำหนดตัวแปรที่มีผลต่อประสิทธิภาพของโมเดลใหม่อีกครั้ง

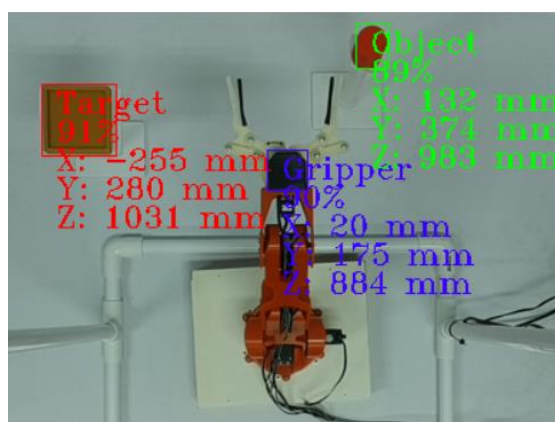


ภาพที่ 3-8 กราฟแสดงความแม่นยำในการฝึกสอนของแต่ละรอบ

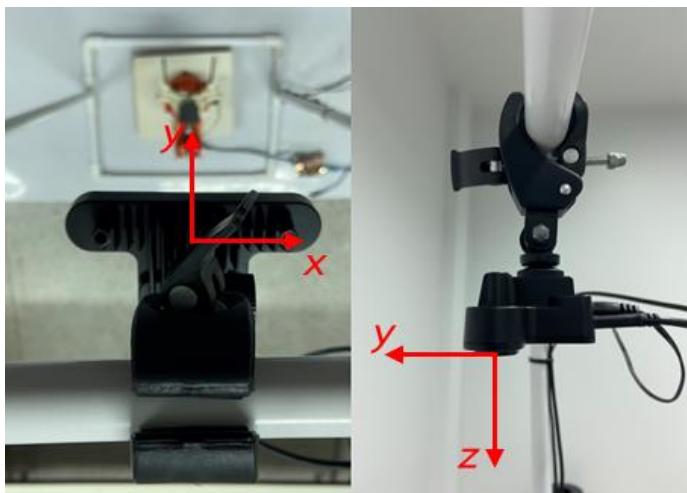


ภาพที่ 3-9 กราฟแสดงความผิดพลาดในการฝึกสอนของแต่ละรอบ

6. เมื่อฝึกสอนเสร็จแล้วจะได้ไฟล์โมเดลการตรวจจับที่มีชื่อว่า best.pt ซึ่งเป็นค่าน้ำหนักของโมเดล จากนั้นจึงดาวน์โหลดไฟล์นี้และนำไปแปลงเป็นไฟล์ .blob ด้วยโปรแกรม Luxonis เพื่อนำไปใช้กับกล้อง OAK-D โดยตัวอย่างการตรวจจับจะแสดงดังภาพที่ 3-10 ซึ่งรายละเอียดที่แสดงในรูปภาพการตรวจจับจะบอกถึงค่าความเชื่อมั่นของสิ่งที่ตรวจจับได้ และพิกัด x, y, z ที่วัดจากจุดกึ่งกลางของกรอบรูปภาพ ซึ่งอ้างอิงกับพิกัดจุดกำเนิดของกล้องดังภาพที่ 3-11

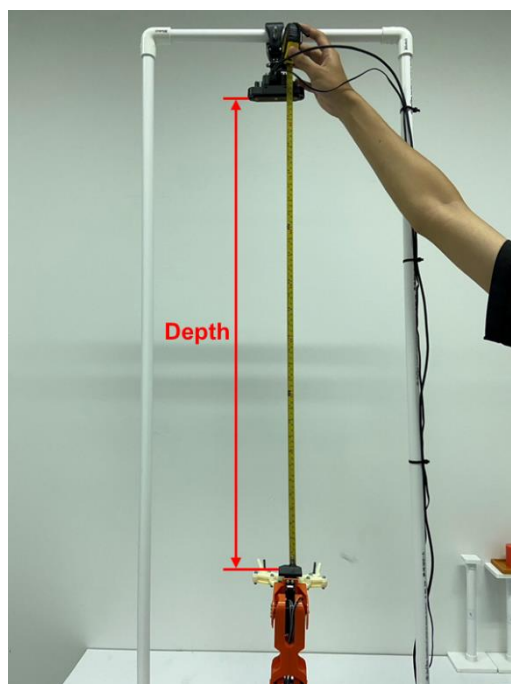


ภาพที่ 3-10 การตรวจจับตำแหน่งวัตถุ ตำแหน่งเป้าหมาย และตำแหน่งมือจับของแขนหุ่นยนต์



ภาพที่ 3-11 ระบบพิกัดของกล้อง OAK-D

โดยในพิกัดแกน z ได้มีการกำหนดหาความแตกต่างของระยะความลึกที่อ่านได้จากกล้อง 3 มิติและจากการวัดของมือจับ (ภาพที่ 3-12) ทั้งหมด 10 ตำแหน่ง แสดงดังตารางที่ 3-1 ซึ่งพบว่ามีความแตกต่างเฉลี่ย 9.45 มม. ทั้งนี้ค่าความแตกต่างที่ได้ก็จะเป็นค่าชดเชยของระยะความลึกในแกน z และถูกจัดการด้วยวิธี MBRL ต่อไป



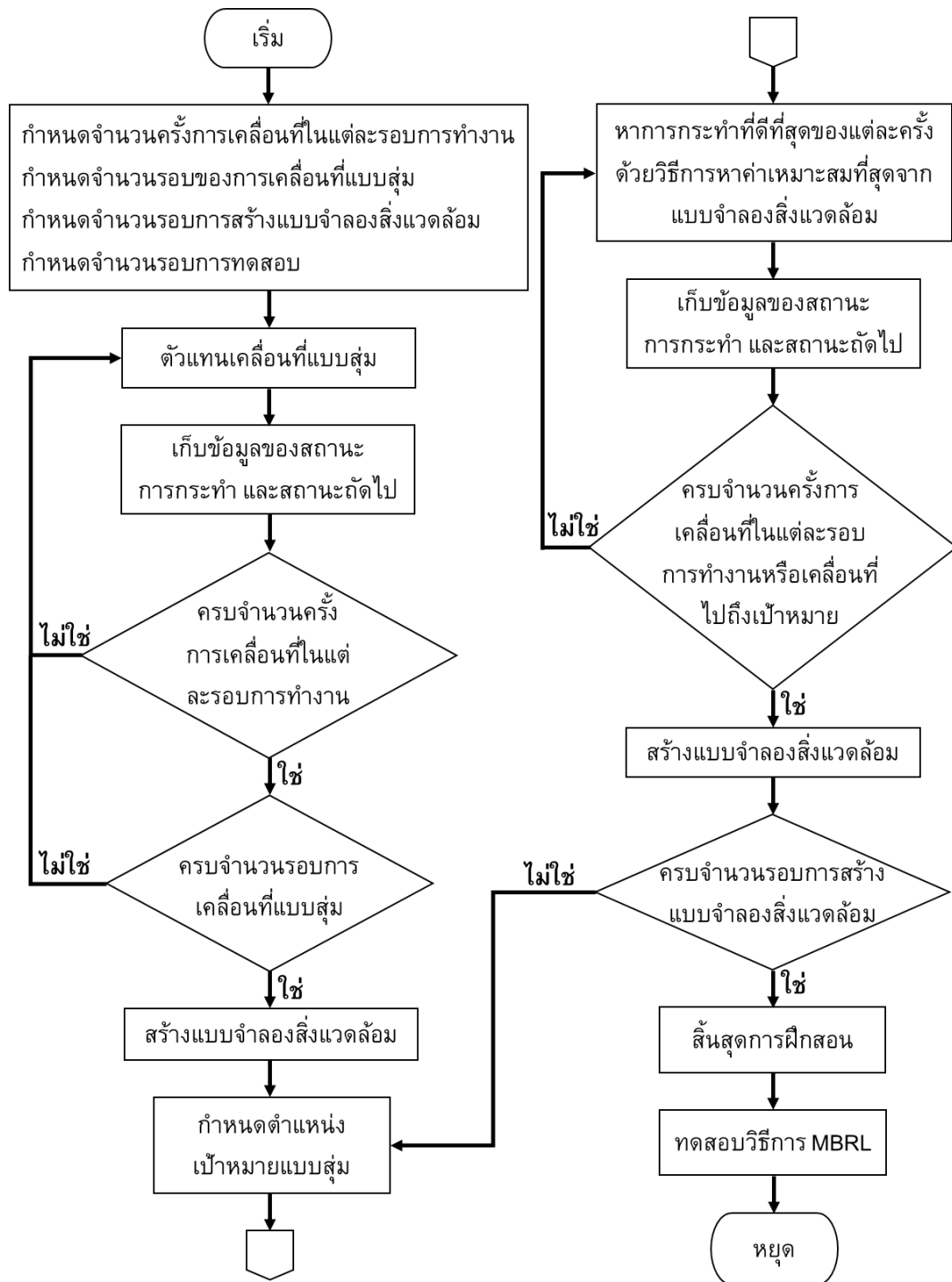
ภาพที่ 3-12 การวัดความลึกในแกน z ของมือจับด้วยตลับเมตร

ตารางที่ 3-1 ความแตกต่างของระยะความลึกในแกน z ที่ได้จากกล้อง 3 มิติและการวัดของมือจับ

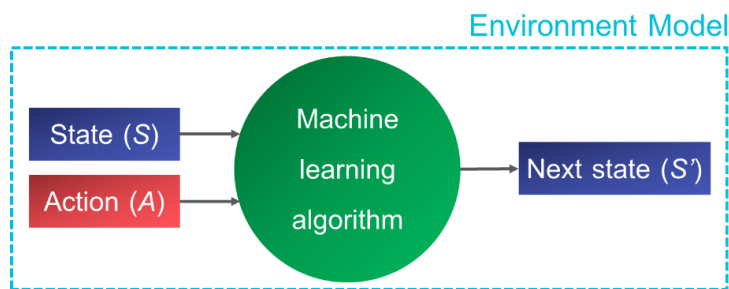
ครั้งที่	ระยะความลึกในแกน z จาก กล้อง 3 มิติ (มม.)	ระยะความลึกในแกน z จาก การวัด (มม.)	ความแตกต่างของระยะ ความลึกในแกน z (มม.)
1	873.82	864.00	9.82
2	902.00	893.00	9.00
3	930.00	920.00	10.00
4	934.76	925.00	9.76
5	939.69	930.00	9.69
6	944.69	935.00	9.69
7	960.09	951.00	9.09
8	992.00	983.00	9.00
9	1050.23	1041.00	9.23
10	1069.18	1060.00	9.18

การเรียนรู้แบบเสริมกำลังโมเดลเบส (MODEL-BASED REINFORCEMENT LEARNING)

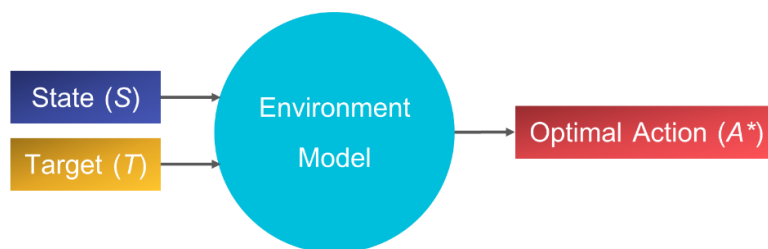
ในการฝึกสอน (Training) ของวิธี MBRL (ภาพที่ 3-13) จะเริ่มต้นจากการให้ตัวแทน (Agent) เคลื่อนที่แบบสุ่มในแต่ละครั้ง (Step) ของรอบการทำงาน (Episode) พร้อมเก็บข้อมูลของสถานะ (State: S) การกระทำ (Action: A) และสถานะถัดไป (Next state: S') จนครบจำนวนรอบที่กำหนดไว้ เพื่อนำมาใช้ในการสร้างแบบจำลองสิ่งแวดล้อม (Environment model) ด้วยเทคนิคการถดถอยการเรียนรู้ของเครื่อง (ภาพที่ 3-14) โดยมีค่าตัวแปรนำเข้า คือ สถานะและการกระทำ ค่าตัวแปรส่งออก คือ สถานะถัดไป จากนั้นในรอบการทำงานถัดไปจึงกำหนดตำแหน่งเป้าหมายที่ต้องการให้ตัวแทนเคลื่อนที่ไปถึงแบบสุ่มที่มีการกระจายตัวอย่างสม่ำเสมอ และนำขั้นตอนวิธีการหาค่าเหมาะสมที่สุดมาใช้ในการหาการกระทำที่ดีที่สุดของแต่ละครั้งการเคลื่อนที่จากแบบจำลองสิ่งแวดล้อม (ภาพที่ 3-15) พร้อมเก็บข้อมูลของสถานะ การกระทำและสถานะถัดไปจนครบจำนวนครั้งสูงสุดที่กำหนดไว้หรือเคลื่อนที่ไปถึงเป้าหมายในแต่ละรอบ แล้วนำข้อมูลที่ได้นำไปรวมกับชุดข้อมูลเดิมเพื่อสร้างแบบจำลองสิ่งแวดล้อมอีกครั้ง และทำซ้ำจนกระทั่งครบจำนวนรอบสูงสุดในการสร้างแบบจำลองสิ่งแวดล้อม จึงเสร็จสิ้นการฝึกสอน จากนั้นจึงนำวิธี MBRL มาใช้ในการทดสอบ (Test) ตามจำนวนรอบที่กำหนด เพื่อหาประสิทธิภาพของแบบจำลอง



ภาพที่ 3-13 แผนผังการทำงาน (Flowchart) ของวิธี MBRL

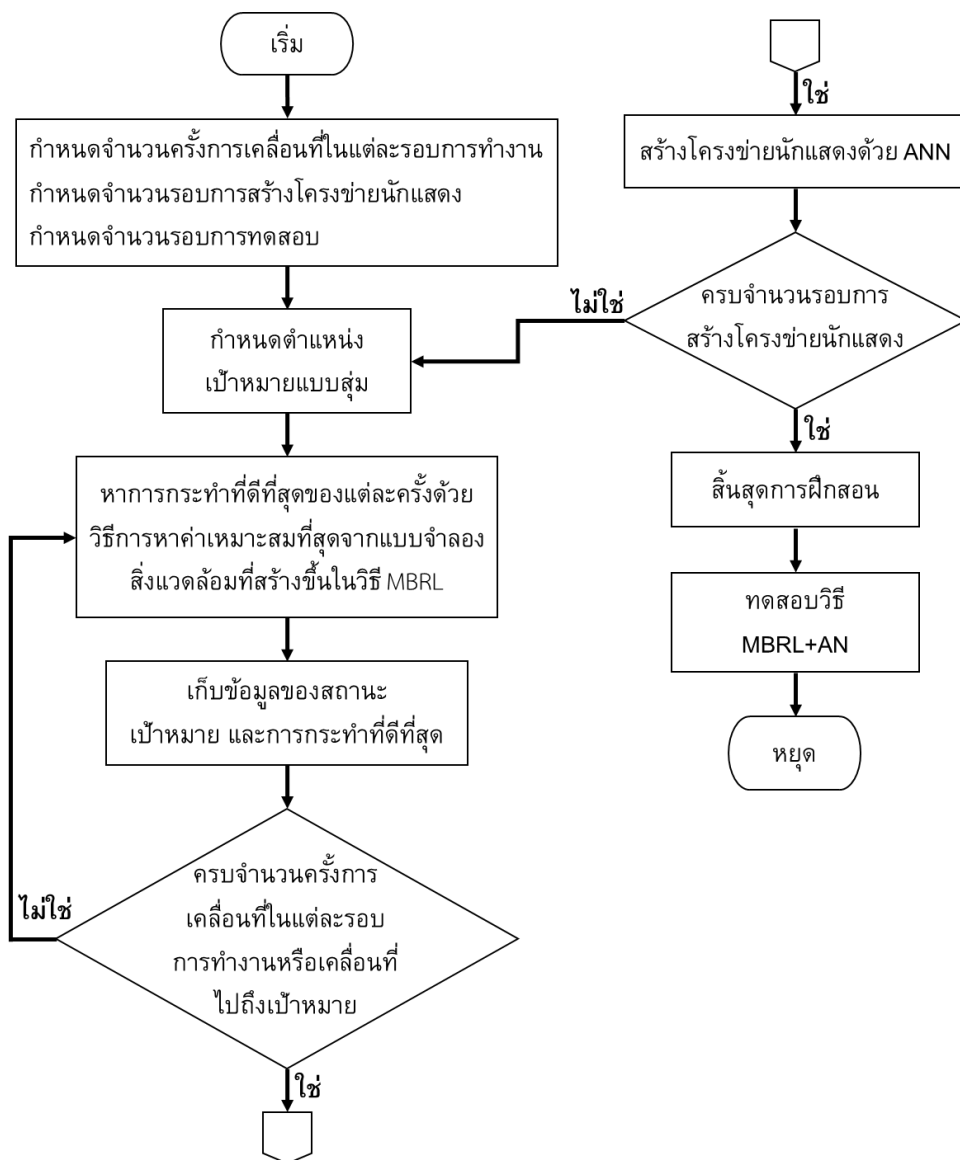


ภาพที่ 3-14 การสร้างแบบจำลองสิ่งแวดล้อมด้วยเทคนิคการถดถอยการเรียนรู้ของเครื่องในวิธี MBRL

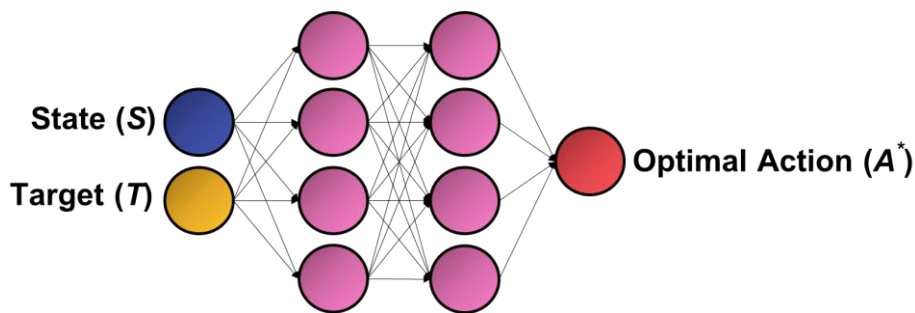


ภาพที่ 3-15 การหาการกระทำที่ดีที่สุดจากแบบจำลองสิ่งแวดล้อม

นอกจากนี้ยังมีการนำแบบจำลองสิ่งแวดล้อมที่สร้างเสร็จเรียบร้อยแล้วพร้อมทั้งขั้นตอนวิธีการหาค่าเหมาะสมที่สุดในวิธี MBRL มาใช้ในการสร้างโครงข่ายนักแสดง (Actor network: AN) เนื่องจากวิธี MBRL หากมีตัวแปรนำเข้าจำนวนมาก ก็จะใช้เวลาในการคำนวณหาการกระทำที่ดีที่สุดของแต่ละครั้งการเคลื่อนที่มากขึ้นตามไปด้วย ดังนั้นหากมีการสร้างเป็นโครงข่ายนักแสดง ก็จะทำให้ใช้เวลาในการเลือกการกระทำที่น้อยลง ส่งผลให้แขนหุ่นยนต์เคลื่อนที่ไปถึงเป้าหมายได้รวดเร็วยิ่งขึ้น ซึ่งในการสร้างโครงข่ายนักแสดง (ภาพที่ 3-16) จะฝึกสอนโดยเริ่มต้นจากการกำหนดตำแหน่งเป้าหมายที่ต้องการให้ตัวแทนเคลื่อนที่ไปถึงแบบสุ่มที่มีการกระจายตัวอย่างสม่ำเสมอ และนำขั้นตอนวิธีการหาค่าเหมาะสมที่สุดมาใช้ในการหาการกระทำที่ดีที่สุดของแต่ละครั้งการเคลื่อนที่จากแบบจำลองสิ่งแวดล้อมที่สร้างขึ้นในวิธี MBRL พร้อมเก็บข้อมูลของสถานะ เป้าหมายและการกระทำที่ดีที่สุด จนครบจำนวนครั้งสูงสุดที่กำหนดไว้หรือเคลื่อนที่ไปถึงเป้าหมายในแต่ละรอบ แล้วนำข้อมูลที่ได้อมาสร้างโครงข่ายนักแสดงด้วย ANN โดยมีค่าตัวแปรนำเข้า คือ สถานะ เป้าหมาย และค่าตัวแปรส่งออก คือ การกระทำที่ดีที่สุดดังภาพที่ 3-17 และทำซ้ำจนกระทั่งครบจำนวนรอบสูงสุดในการสร้างโครงข่ายนักแสดง จึงเสร็จสิ้นการฝึกสอน จากนั้นจึงนำวิธี MBRL+AN มาใช้ในการทดสอบ (Test) ตามจำนวนรอบที่กำหนด เพื่อหาประสิทธิภาพของแบบจำลอง



ภาพที่ 3-16 แผนผังการทำงาน (Flowchart) ของวิธี MBRL+AN



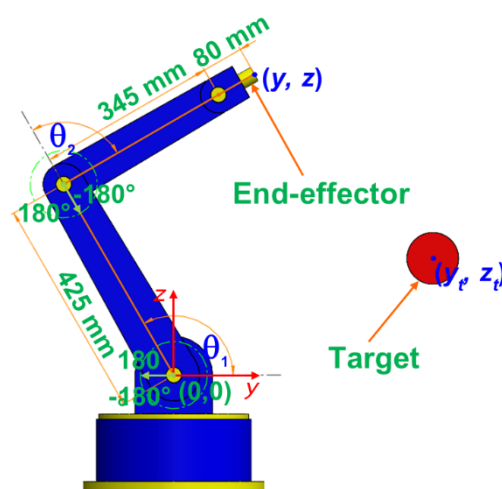
ภาพที่ 3-17 การสร้างแบบจำลอง ANN ในวิธี MBRL+AN

การศึกษาเชิงตัวเลข (NUMERICAL STUDY)

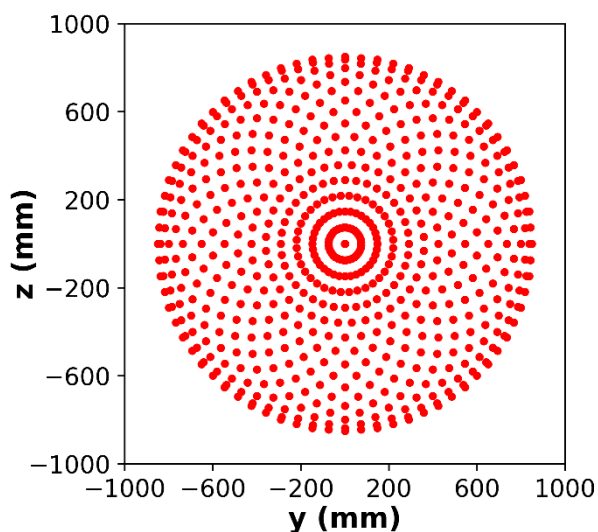
การศึกษาเชิงตัวเลขเป็นการนำวิธี MBRL มาใช้สำหรับการแก้ไขปัญหาเชิงตัวเลข เพื่อทดสอบการทำงานในเบื้องต้นว่ามีความเหมาะสมในการนำมาใช้แก้ปัญหาหรือไม่ ก่อนที่จะนำไปทดสอบกับปัญหาจริง โดยตัวแทน (Agent) ของปัญหาคือแขนหุ่นยนต์

1. ปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 2 องศาอิสระ (2-DOF Robotic arm reaching task)

ปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 2 องศาอิสระเป็นปัญหาที่มีวัตถุประสงค์คือการควบคุมการหมุนของมุมข้อต่อทั้ง 2 มุม เพื่อให้ปลายแขนหุ่นยนต์เคลื่อนที่ไปยังตำแหน่งเป้าหมายที่ต้องการ โดยในการทดสอบจะมีการเพิ่มสัญญาณรบกวนของการหมุนแบบเอกรูป (Uniform noise) ที่มีค่าเท่ากับ 0.0%, 2.5%, 5.0%, 7.5%, 10.0%, 15.0% และ 20.0% เนื่องจากในทางปฏิบัติแล้ว หากต้องการควบคุมให้แขนหุ่นยนต์หมุนไปยังมุมที่ต้องการ อาจมีปัจจัยที่ส่งผลให้มุมที่วัดได้จริง มีความคลาดเคลื่อนเกิดขึ้นได้ เช่น มอเตอร์ที่ใช้ในการหมุน สัญญาณควบคุมทางไฟฟ้า หากสั่งการควบคุมให้มุมข้อต่อหมุนไปยังมุม 40 องศา มุมที่เป็นไปได้จากการหมุนอาจมีค่าอยู่ระหว่าง 36 องศาถึง 44 องศาในกรณีที่มีสัญญาณรบกวนเท่ากับ 10.0% ดังนั้นจึงได้ทำการพิจารณาถึงสัญญาณรบกวนที่เกิดขึ้นต่อการหมุน เพื่อให้มีความใกล้เคียงกับในทางปฏิบัติมากที่สุด โดยในแต่ละครั้งการเคลื่อนที่ มุมข้อต่อทั้ง 2 มุมสามารถหมุนได้ระหว่าง -45 องศาถึง 45 องศา อีกทั้งยังมีขอบเขตการหมุนของทั้งสองมุมอยู่ระหว่าง -180 องศาถึง 180 องศาแสดงดังภาพที่ 3-18 ดังนั้นจึงทำให้มีตำแหน่งเป้าหมายที่เป็นไปได้แสดงดังภาพที่ 3-19



ภาพที่ 3-18 การจำลองปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 2-DOF



ภาพที่ 3-19 ตำแหน่งเป้าหมายที่เป็นไปได้ของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 2-DOF

และกำหนดให้สถานะในการเรียนรู้ของปัญหา คือ มุมข้อต่อและตำแหน่งพิกัดปลายแขนหุ่นยนต์ในระนาบ yz การกระทำ คือ การหมุนของมุมข้อต่อทั้ง 2 มุม และรางวัล คือ ระยะห่างระหว่างปลายแขนหุ่นยนต์กับเป้าหมายดังภาพที่ 3-20

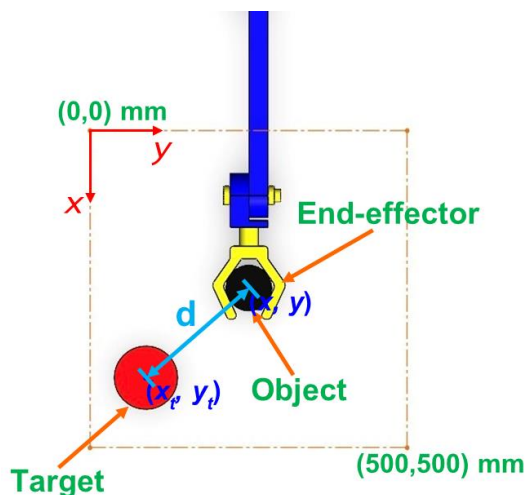
States (S)	θ_1, θ_2, y, z
Actions (A)	$\Delta\theta_1, \Delta\theta_2$
Reward (R)	$-d$

ภาพที่ 3-20 สถานะ การกระทำและรางวัลของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 2-DOF

2. ปัญหาการวางวัตถุ (Placing task)

ปัญหาการวางวัตถุเป็นปัญหาที่มีวัตถุประสงค์ คือ การควบคุมการเคลื่อนที่ในแนวเชิงเส้น (Linear motion) ของปลายแขนหุ่นยนต์ เพื่อนำวัตถุไปวางยังตำแหน่งเป้าหมายที่ต้องการ ซึ่งในแต่ละครั้งการเคลื่อนที่ จะสามารถเคลื่อนที่ได้ระหว่าง -30 ถึง 30 มม. ทั้งในแนวแกน x และแกน y อีกทั้งยังมีขอบเขตของการเคลื่อนที่และตำแหน่งเป้าหมายในแต่ละแกนอยู่ภายใน 500 มม. แสดงดังภาพที่ 3-21 และกำหนดให้สถานะในการเรียนรู้ของปัญหา คือ ตำแหน่งพิกัดปลายแขนหุ่นยนต์ใน

ระนาบ xy การกระทำ คือ การเคลื่อนที่ในแนวแกน x และแกน y และรางวัล คือ ระยะห่างระหว่างปลายแขนหุ่นยนต์กับเป้าหมายดังภาพที่ 3-22



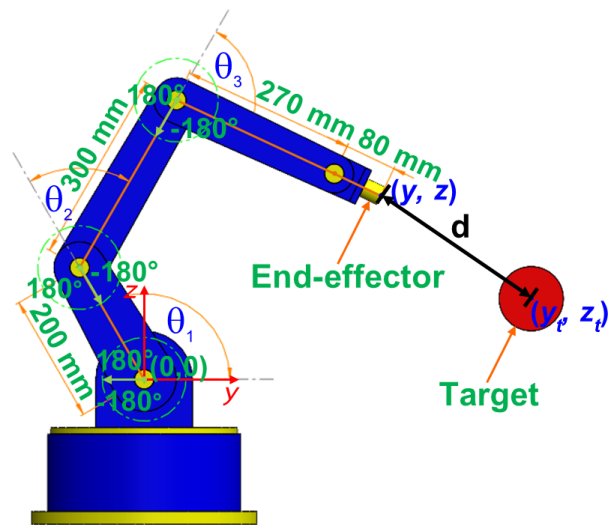
ภาพที่ 3-21 การจำลองปัญหาการวางวัตถุ

States (S)	x, y
Actions (A)	$\Delta x, \Delta y$
Reward (R)	$-d$

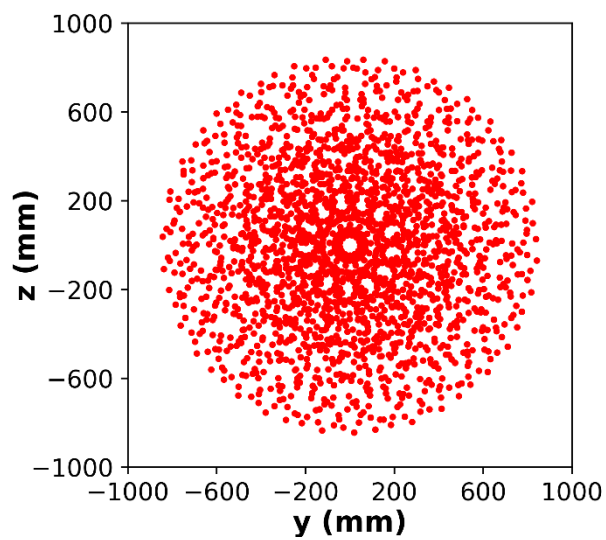
ภาพที่ 3-22 สถานะ การกระทำและรางวัลของปัญหาการวางวัตถุ

3. ปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 3 องศาอิสระ (3-DOF Robotic arm reaching task)

ปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 3 องศาอิสระจะมีวัตถุประสงค์ของปัญหาเหมือนกับปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 2 องศาอิสระ และจะทดสอบในกรณีที่มีสัญญาณรบกวนแบบเอกรูปของการหมุนเท่ากับ 0.0%, 5.0% และ 10.0% มุมข้อต่อทั้ง 3 มุมจะสามารถเคลื่อนที่ด้วยการหมุนระหว่าง -45 องศาถึง 45 องศาในแต่ละครั้งการเคลื่อนที่ และมีขอบเขตการหมุนอยู่ระหว่าง -180 องศาถึง 180 องศา แสดงดังภาพที่ 3-23 ดังนั้นจึงทำให้มีตำแหน่งเป้าหมายที่เป็นไปได้แสดงดังภาพที่ 3-24 และการกำหนดสถานะ การกระทำและรางวัลในการเรียนรู้จะแสดงดังภาพที่ 3-25



ภาพที่ 3-23 การจำลองปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 3-DOF



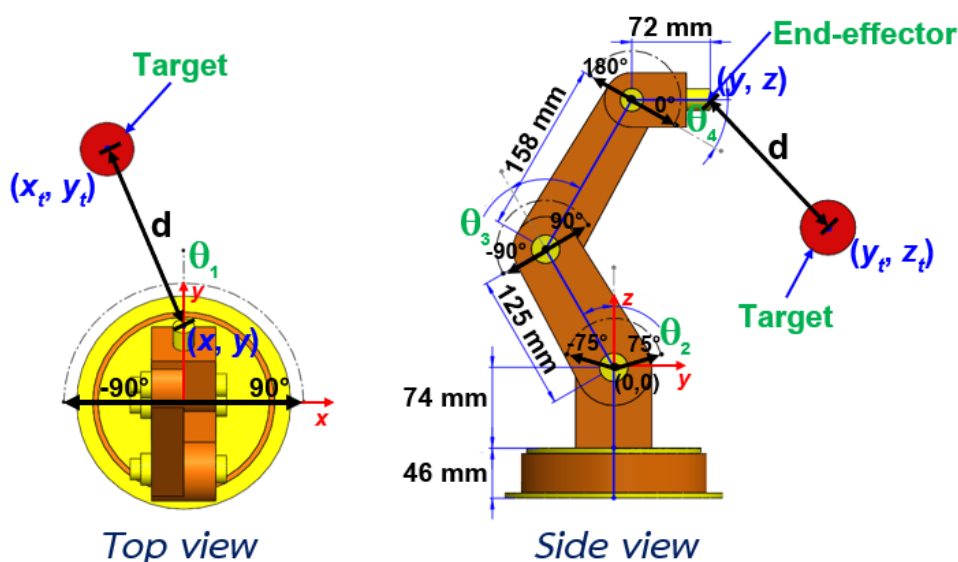
ภาพที่ 3-24 ตำแหน่งเป้าหมายที่เป็นไปได้ของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 3-DOF

States (S)	$\theta_1, \theta_2, \theta_3, y, z$
Actions (A)	$\Delta\theta_1, \Delta\theta_2, \Delta\theta_3$
Reward (R)	$-d$

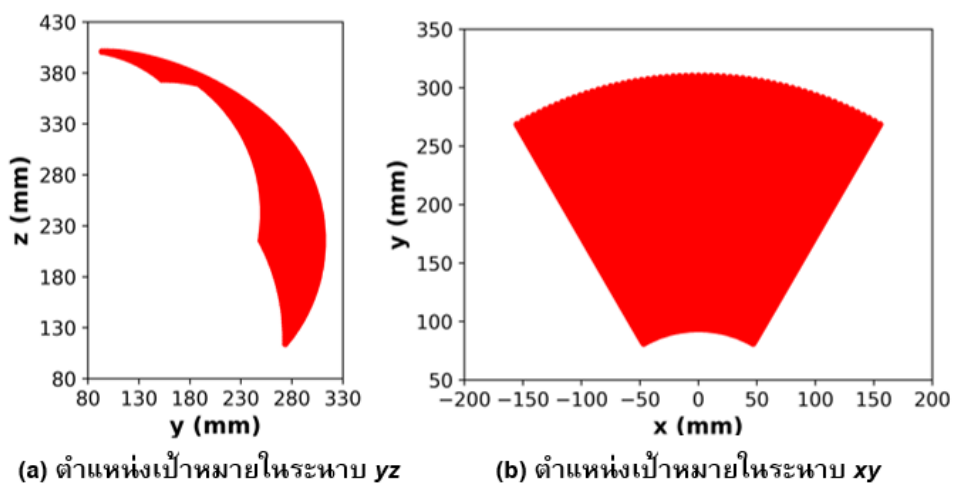
ภาพที่ 3-25 สถานะ การกระทำและรางวัลของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 3-DOF

4. ปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4 องศาอิสระ (4-DOF Robotic arm reaching task)

ปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4 องศาอิสระเป็นปัญหาที่จำลองให้มีความคล้ายคลึงกับปัญหาจริงมากที่สุด โดยมีวัตถุประสงค์ของปัญหาเหมือนกับปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 2 องศาอิสระและ 3 องศาอิสระ และจะทดสอบการเคลื่อนที่ทั้งในแนวระนาบ yz และในพิภคฉาก 3 มิติ โดยมีสัญญาณรบกวนแบบเอกรูปของการหมุนเท่ากับ 0.0%, 5.0% และ 10.0% มุมข้อต่อทั้ง 4 มุมจะสามารถเคลื่อนที่ด้วยการหมุนระหว่าง -20 องศาถึง 20 องศาในแต่ละครั้งการเคลื่อนที่ และมีขอบเขตการหมุนของมุมข้อต่อที่ 1 และมุมข้อต่อที่ 3 อยู่ระหว่าง -90 องศาถึง 90 องศา มุมข้อต่อที่ 2 อยู่ระหว่าง -75 องศาถึง 75 องศา มุมข้อต่อที่ 4 อยู่ระหว่าง 0 องศาถึง 180 องศา แสดงดังภาพที่ 3-26 อีกทั้งยังมีการกำหนดเงื่อนไขในการหมุนของมุมข้อต่อที่ 4 ให้ปลายแขนหุ่นยนต์ขนานกับพื้นในทุกครั้งของการเคลื่อนที่ ซึ่งจะมีความสัมพันธ์กับมุมข้อต่อที่ 2 และ 3 ดังนั้นจึงได้กำหนดขอบเขตของตำแหน่งเป้าหมายจากการหมุนของมุมข้อต่อที่ 2 ซึ่งมีค่าอยู่ระหว่างมุม 10 องศาถึง 40 องศา และมุมข้อต่อที่ 3 จะมีความสัมพันธ์กับมุมข้อต่อที่ 2 โดยมีค่าอยู่ระหว่างค่าลบของมุมข้อต่อที่ 2 ถึง 90 องศาแสดงดังภาพที่ 3-27 และการกำหนดสถานะ การกระทำ และรางวัลในการเรียนรู้ของการเคลื่อนที่ในแนวระนาบ yz และในพิภคฉาก 3 มิติ จะแสดงดังภาพที่ 3-28 และภาพที่ 3-29 ตามลำดับ



ภาพที่ 3-26 การจำลองปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4-DOF



ภาพที่ 3-27 ตำแหน่งเป้าหมายที่เป็นไปได้ของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4-DOF

States (S)	θ_2, θ_3, y, z
Actions (A)	$\Delta\theta_2, \Delta\theta_3$
Reward (R)	$-d$

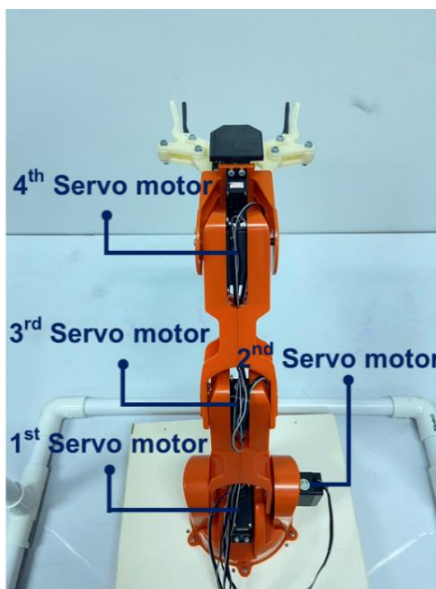
ภาพที่ 3-28 สถานะ การกระทำและรางวัลของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4-DOF กรณีเคลื่อนที่ในแนวระนาบ yz

States (S)	$\theta_1, \theta_2, \theta_3, x, y, z$
Actions (A)	$\Delta\theta_1, \Delta\theta_2, \Delta\theta_3$
Reward (R)	$-d$

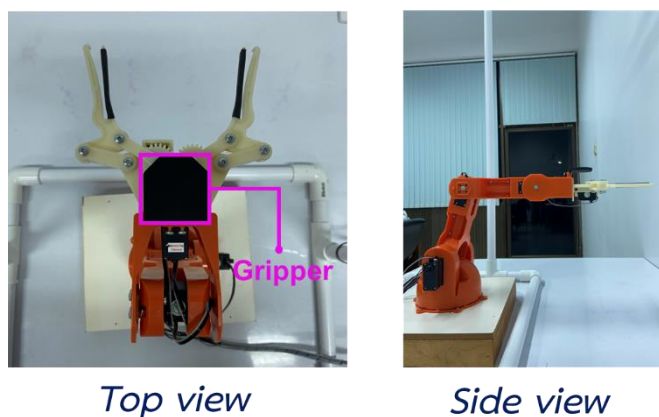
ภาพที่ 3-29 สถานะ การกระทำและรางวัลของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4-DOF กรณีเคลื่อนที่ในพิภพฉาก 3 มิติ

การทดสอบปัญหาจริง

การทดสอบปัญหาจริงเป็นการนำวิธี MBRL มาทดสอบผ่านชุดการทดลองที่ได้ออกแบบไว้ ซึ่งมุมข้อต่อที่ 1, 2, 3 และ 4 ของแขนหุ่นยนต์จะถูกขับเคลื่อนด้วยเซอร์โวมอเตอร์ตัวที่ 1, 2, 3 และ 4 ตามลำดับ แสดงดังภาพที่ 3-30 อีกทั้งในการตรวจจับตำแหน่งของมือจับแขนหุ่นยนต์ ยังได้กำหนดเงื่อนไขการหมุนของมุมข้อต่อที่ 4 ให้มือจับขนานกับพื้นในทุกครั้งของการเคลื่อนที่ เพื่อให้กล้อง OAK-D สามารถมองเห็นและตรวจจับตำแหน่งของมือจับได้ ซึ่งมีรูปทรงที่ใช้ในการตรวจจับ แสดงดังภาพที่ 3-31



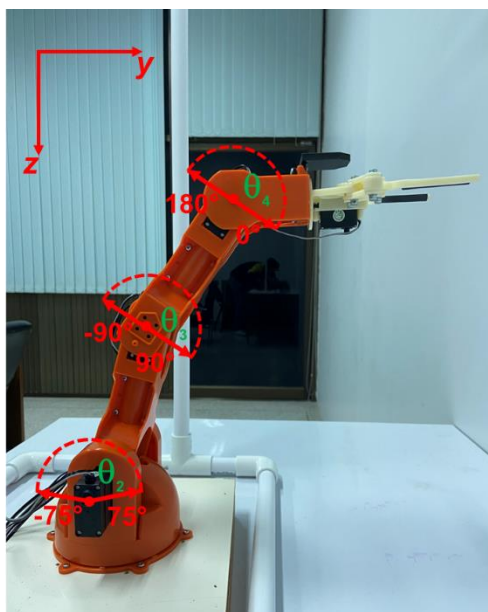
ภาพที่ 3-30 เซอร์โวมอเตอร์ที่ใช้ควบคุมการหมุนในแต่ละข้อต่อของแขนหุ่นยนต์



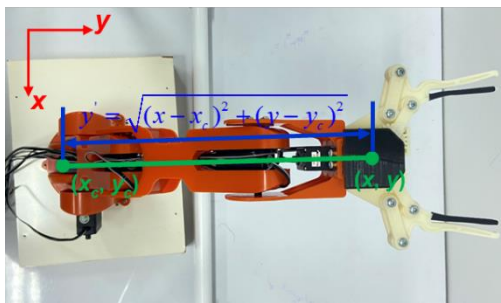
ภาพที่ 3-31 รูปทรงที่ใช้ในการตรวจจับตำแหน่งมือจับแขนหุ่นยนต์

1. การควบคุมแขนหุ่นยนต์ในแนวระนาบ

การควบคุมแขนหุ่นยนต์ในแนวระนาบเป็นการทดสอบการควบคุมการหมุนของมุมข้อต่อที่ 2, 3 และ 4 ในระนาบ yz เพื่อให้มือจับของแขนหุ่นยนต์ไปยังตำแหน่งเป้าหมายที่ต้องการ โดยในแต่ละครั้งการเคลื่อนที่ มุมข้อต่อทั้ง 3 มุมสามารถหมุนได้ระหว่าง -20 องศาถึง 20 องศา และมีขอบเขตการหมุนของมุมข้อต่อที่ 2 อยู่ระหว่าง -75 องศาถึง 75 องศา มุมข้อต่อที่ 3 อยู่ระหว่าง -90 องศาถึง 90 องศา และมุมข้อต่อที่ 4 อยู่ระหว่าง 0 องศาถึง 180 องศา แสดงดังภาพที่ 3-32 และกำหนดให้สถานะในการเรียนรู้ของปัญหา คือ มุมข้อต่อที่ 2 และ 3 ตำแหน่งมือจับในแกน z และระยะห่างระหว่างตำแหน่งมือจับกับจุดศูนย์กลางการหมุนของมุมข้อต่อที่ 1 ในระนาบ xy (ภาพที่ 3-33) เพื่อที่จะนำวิธี MBRL ที่ผ่านการฝึกสอนในหัวข้อนี้ไปใช้ทดสอบการจับและวางวัตถุ เมื่อมีการเคลื่อนที่ในพิสัย 3 มิติ การกระทำ คือ การหมุนของมุมข้อต่อที่ 2 และ 3 และรางวัล คือ ระยะห่างระหว่างปลายแขนหุ่นยนต์กับเป้าหมายดังภาพที่ 3-34 อีกทั้งขอบเขตของเป้าหมายจะถูกกำหนดโดยการหมุนของมุมข้อต่อที่ 2 ซึ่งมีค่าอยู่ระหว่างมุม 10 องศาถึง 40 องศา และมุมข้อต่อที่ 3 จะมีความสัมพันธ์กับมุมข้อต่อที่ 2 โดยมีค่าอยู่ระหว่างค่าลบของมุมข้อต่อที่ 2 ถึง 90 องศา



ภาพที่ 3-32 ขอบเขตการหมุนของมุมข้อต่อที่ 2, 3 และ 4 ในระนาบ yz



ภาพที่ 3-33 การคำนวณหาระยะห่างระหว่างตำแหน่งมือจับกับจุดศูนย์กลางการหมุนของข้อต่อที่ 1

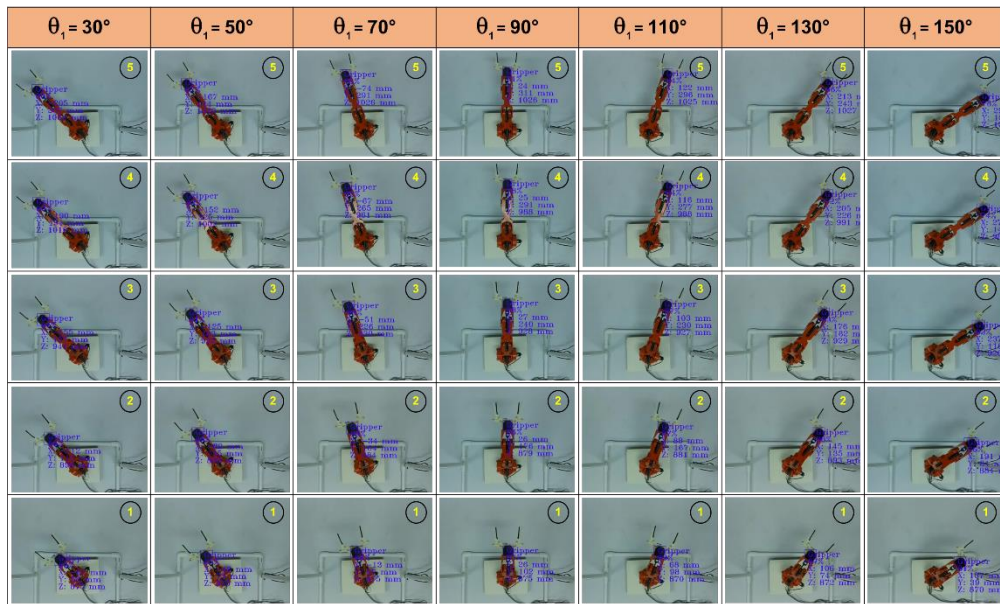
States (S)	$\theta_2, \theta_3, y', z$
Actions (A)	$\Delta\theta_2, \Delta\theta_3$
Reward (R)	$\sqrt{(y' - y_t')^2 + (z - z_t)^2}$

ภาพที่ 3-34 สถานะ การกระทำและรางวัลของการควบคุมแขนหุ่นยนต์ในแนวระนาบ

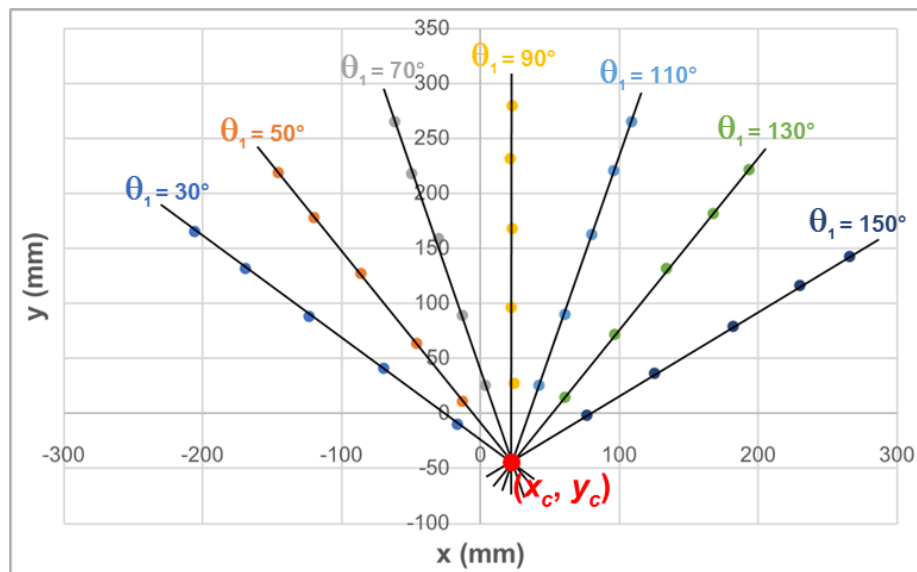
เนื่องจากการติดตั้งหุ่นยนต์ สิ่งที่สามารถตรวจจับได้มีเพียงตำแหน่งมือจับ ทำให้ไม่สามารถทราบได้ว่าตำแหน่งจุดศูนย์กลางการหมุนของข้อต่อที่ 1 อยู่ที่ใด ดังนั้นจึงทำการหาด้วยการหมุนข้อต่อที่ 1 ไปยังมุมต่าง ๆ และให้ข้อต่อที่ 2, 3 และ 4 เคลื่อนที่ไปยังตำแหน่งที่แตกต่างกันดังภาพที่ 3-35 พร้อมเก็บข้อมูลของพิกัดแกน x และแกน y เพื่อนำมาเขียนกราฟความสัมพันธ์ จากนั้นจึงหาสมการเส้นแนวโน้มในแต่ละเส้นของข้อต่อที่ 1 เพื่อหาจุดที่มีผลบวกของระยะห่างตั้งฉากกับเส้นแต่ละเส้นน้อยที่สุดดังสมการที่ 3.1 ซึ่งจุดนั้นก็จะเป็นตำแหน่งจุดศูนย์กลางการหมุนของข้อต่อที่ 1 ดังภาพที่ 3-36

$$d = \frac{|k + mx_c - y_c|}{\sqrt{1+m^2}} \quad (3.1)$$

โดยที่ d คือ ระยะห่างตั้งฉากระหว่างจุดกับเส้นแนวโน้มแต่ละเส้น
 k คือ จุดตัดในแกน y ของเส้นแนวโน้มแต่ละเส้น
 m คือ ความชันของเส้นแนวโน้มแต่ละเส้น
 x_c คือ จุดศูนย์กลางการหมุนในแกน x
 y_c คือ จุดศูนย์กลางการหมุนในแกน y



ภาพที่ 3-35 การหมุนมุมข้อต่อที่ 1 ไปยังมุมต่าง ๆ เพื่อหาดำแหน่งจุดศูนย์กลางการหมุน

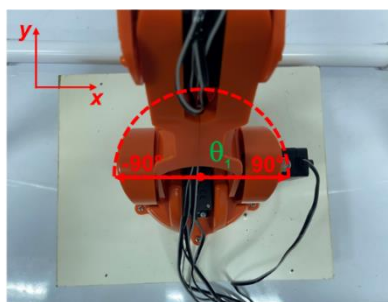


ภาพที่ 3-36 การหาดำแหน่งจุดศูนย์กลางการหมุนของมุมข้อต่อที่ 1

2. การควบคุมแขนหุ่นยนต์ในพิสัยจาก 3 เมตร

การควบคุมแขนหุ่นยนต์ในพิสัยจาก 3 เมตรเป็นการทดสอบการควบคุมการหมุนของมุมข้อต่อทั้ง 4 มุม เพื่อให้มือจับไปยังตำแหน่งของเป้าหมายที่ต้องการ โดยมีเงื่อนไขในการหมุนและการกำหนดสถานะ การกระทำและรางวัลในการเรียนรู้เหมือนกับปัญหาการศึกษาเชิงตัวเลขการไป

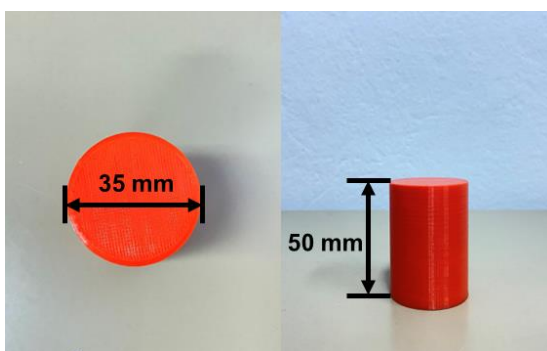
ถึงเป้าหมายของแขนหุ่นยนต์ 4 องศาอิสระ โดยขอบเขตการหมุนของมุมข้อต่อที่ 2, 3 และ 4 จะแสดงดังภาพที่ 3-32 และขอบเขตการหมุนของมุมข้อต่อที่ 1 จะแสดงดังภาพที่ 3-37 อีกทั้งขอบเขตของเป้าหมายจะถูกกำหนดโดยการหมุนของมุมข้อต่อที่ 1 ซึ่งมีค่าอยู่ระหว่างมุม -30 องศาถึง 30 องศา เนื่องจากเป็นช่วงที่กล้องสามารถตรวจจับพิกัดในแกน z ได้ โดยมีความคลาดเคลื่อนที่สามารถยอมรับได้



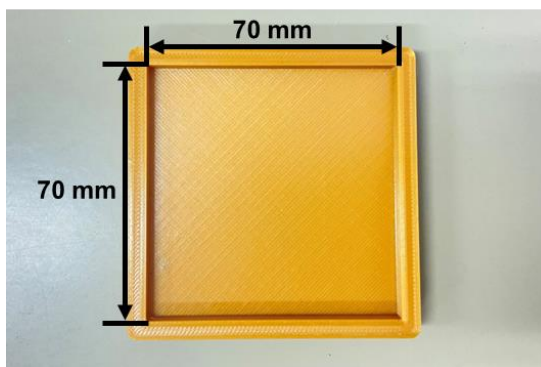
ภาพที่ 3-37 ขอบเขตการหมุนของมุมข้อต่อที่ 1 ในระนาบ xy

3. การจับและวางวัตถุ (Pick-and-place)

การจับและวางวัตถุเป็นการทดสอบการหยิบจับวัตถุและนำวัตถุไปวางยังเป้าหมายที่ต้องการ โดยการนำวิธี MBRL ที่ผ่านการฝึกสอนในแนวระนาบมาใช้ในการทดสอบ ซึ่งวัตถุที่ใช้สำหรับจับจะออกแบบให้เป็นรูปทรงกระบอกที่มีขนาดเส้นผ่านศูนย์กลางเท่ากับ 35 มม. และความสูงเท่ากับ 50 มม. ซึ่งแสดงดังภาพที่ 3-38 และออกแบบเป้าหมายให้เป็นรูปสี่เหลี่ยมจัตุรัสที่มีความยาวภายในด้านละ 70 มม. แสดงดังภาพที่ 3-39 อีกทั้งยังมีการออกแบบฐานที่ใช้วางวัตถุและเป้าหมายแสดงดังภาพที่ 3-40



ภาพที่ 3-38 วัตถุที่ใช้ในการหยิบจับ



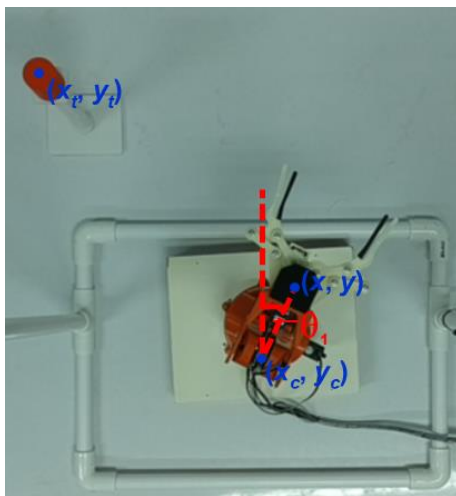
ภาพที่ 3-39 เป้าหมายที่ใช้ในการวางวัตถุ



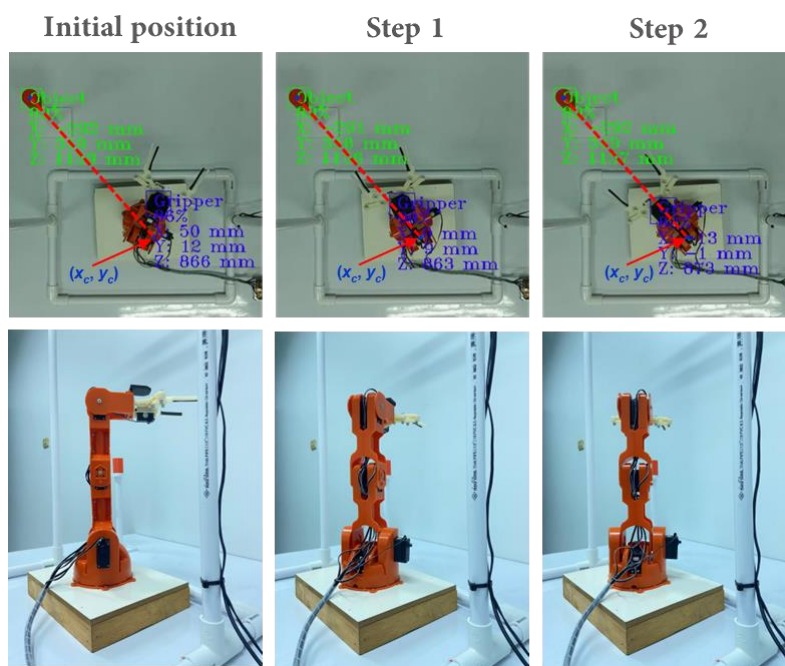
ภาพที่ 3-40 ฐานที่ใช้ในการวางวัตถุและเป้าหมาย

ในการหยิบจับและวางวัตถุจะต้องควบคุมให้มุมข้อต่อที่ 1 หมุนไปยังแนวเส้นตรงเดียวกันกับตำแหน่งที่ต้องการจับและวางวัตถุ แล้วจึงให้มุมข้อต่อที่ 2, 3 และ 4 เคลื่อนที่ไปยังตำแหน่งนั้น โดยการคำนวณหามุมที่ต้องเคลื่อนที่ไปในแต่ละครั้ง จะใช้หลักการคำนวณเชิงเรขาคณิต ที่แสดงดังสมการที่ 3.2 และใช้ตัวแปรในการคำนวณดังภาพที่ 3-41 ซึ่งตัวอย่างการเคลื่อนที่เพื่อไปยังแนวเส้นตรงเดียวกับตำแหน่งที่ต้องการจับวัตถุจะแสดงดังภาพที่ 3-42

$$\Delta\theta_1 = \theta_1 + \tan^{-1}\left(\frac{x-x_c}{y-y_c}\right) - \tan^{-1}\left(\frac{x_t-x_c}{y_t-y_c}\right) \quad (3.2)$$



ภาพที่ 3-41 การวิเคราะห์การหมุนของมุมข้อต่อที่ 1



ภาพที่ 3-42 การหมุนของมุมข้อต่อที่ 1 ไปยังแนวเส้นตรงเดียวกับตำแหน่งที่ต้องการจับวัตถุ



107404525

บทที่ 4

ผลการวิจัย

งานวิจัยนี้ได้นำวิธีการหาค่าเหมาะสมที่สุด CMA-ES มาใช้ในวิธี MBRL และดำเนินการทดสอบผ่านการเขียน โปรแกรมภาษาไพทอน (Python) ด้วยคอมพิวเตอร์ที่ใช้ GPU รุ่น NVIDIA GeForce MX150 ซึ่งการวัดประสิทธิภาพการทำงานของวิธี MBRL ในการศึกษาเชิงตัวเลข การทดสอบปัญหาจริงของการควบคุมแขนหุ่นยนต์ในแนวระนาบและในพิกัดฉาก 3 มิติ รวมถึงการจับและวางวัตถุ จะวัดด้วยอัตราความสำเร็จ (Success-rate) ที่คำนวณจากการทดสอบ (Test) การเคลื่อนที่ไปยังเป้าหมายตามจำนวนรอบที่กำหนด โดยหากระยะห่างระหว่างปลายแขนหุ่นยนต์หรือมือจับกับเป้าหมายมีค่าน้อยกว่าหรือเท่ากับ 10.00 มม. จึงถือว่าเคลื่อนที่ไปยังเป้าหมายได้สำเร็จ ซึ่งระยะห่างที่กำหนดขึ้นมานี้เป็นระยะที่วัตถุยังสามารถอยู่ภายในขอบเขตของเป้าหมายได้เมื่อเทียบกับจุดศูนย์กลางของเป้าหมาย และมีอัตราส่วนระหว่างวัตถุและเป้าหมายเป็น 1 : 4 ดังภาพที่ 3-38 และภาพที่ 3-39 ซึ่งสอดคล้องกับงานวิจัยก่อนหน้า (Jiang et al., 2021) โดยสังเกตจากรูปของมือจับวัตถุและเป้าหมาย พร้อมเปรียบเทียบกับวิธีจลนศาสตร์แบบผกผัน (IK) ที่เป็นการคำนวณในทางทฤษฎี ซึ่งงานวิจัยนี้ได้ใช้วิธีการหาค่าเหมาะสมที่สุดในการคำนวณหามุมของข้อต่อต่าง ๆ เมื่อทราบตำแหน่งพิกัดเป้าหมายในการเคลื่อนที่เพียงหนึ่งครั้ง อีกทั้งการทดสอบในทุกปัญหาจะไม่พิจารณาถึงความเร็วและความเร่งในการเคลื่อนที่ของแขนหุ่นยนต์เป็นสถานะในการเรียนรู้ ซึ่งเป็นทำนองเดียวกันกับงานวิจัยของ (Dalla Libera et al., 2020) โดยในทางปฏิบัติแล้ว ส่วนใหญ่เซนเซอร์วัดความเร็วและความเร่งมักจะไม่ถูกนำมาใช้งาน เนื่องจากมีต้นทุนที่สูง ทั้งนี้ในการศึกษาเชิงตัวเลข ความเร็วและความเร่งจะถูกคำนวณด้วยค่าเชิงตัวเลขจากตำแหน่งของปลายแขนหุ่นยนต์ ทำให้มีความแตกต่างกันระหว่างสัญญาณจริงกับสัญญาณ โดยการประมาณ ซึ่งค่าที่แตกต่างกันนี้จะส่งผลให้เกิดสัญญาณรบกวนกับตำแหน่งของปลายแขนหุ่นยนต์ รวมถึงในการควบคุมการเคลื่อนที่ของแขนหุ่นยนต์ในทางปฏิบัติ ก็อาจจะมีตำแหน่งที่คลาดเคลื่อนไปเนื่องจากแรงเฉื่อยที่เกิดขึ้นเนื่องจากความเร็วและความเร่ง รวมถึงความคลาดเคลื่อนเนื่องจากเซ็นเซอร์วัดตำแหน่ง ซึ่งความคลาดเคลื่อนดังกล่าวนี้ก็ถือเป็นสัญญาณรบกวนต่อตำแหน่งของปลายแขนหุ่นยนต์เช่นเดียวกัน ดังนั้นงานวิจัยนี้จึงพิจารณาทั้งกรณีศึกษาที่ไม่มีสัญญาณรบกวนและมีสัญญาณรบกวนเพื่อให้สอดคล้องกับการประยุกต์ใช้ต่อไป



107404525

BUU-IThesis 64910146 thesis / recv : 20062566 19:32:29 / seq: 122

ผลการทดสอบการศึกษาเชิงตัวเลข

1. ผลการทดสอบปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 2 องศาอิสระ

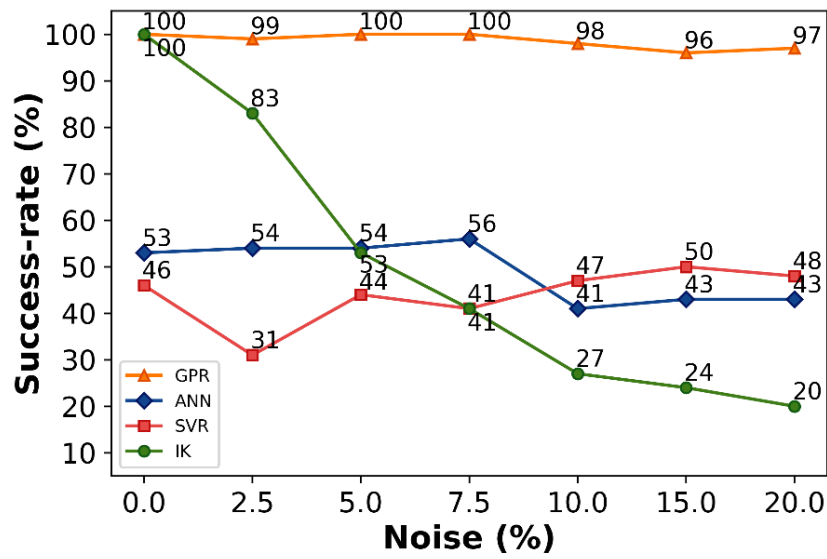
ในการทดสอบปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 2 องศาอิสระ ได้กำหนดเงื่อนไขการฝึกสอนของวิธี MBRL โดยกำหนดให้จำนวนครั้งการเคลื่อนที่ในแต่ละรอบการทำงานมีค่าเท่ากับ 7 ครั้ง จำนวนรอบการทำงานของการเคลื่อนที่แบบสุ่ม จำนวนรอบการสร้างแบบจำลองสิ่งแวดล้อม และจำนวนรอบการทดสอบเท่ากับ 1, 250 และ 100 รอบตามลำดับ อีกทั้งตำแหน่งมุมข้อต่อเริ่มต้นจะเกิดจากการสุ่มในทุกรอบการทำงาน โดยการทดสอบได้นำวิธี IK และเทคนิคการถอดอยการเรียนรู้ของเครื่อง 3 เทคนิคที่ใช้ในการสร้างแบบจำลองสิ่งแวดล้อมของวิธี MBRL มาเปรียบเทียบประสิทธิภาพการทำงาน คือ การถอดอยของกระบวนการเกาส์เซียน (GPR) ซัพพอร์ตเวกเตอร์แมชชีนสำหรับการถอดอย (SVR) และโครงข่ายประสาทเทียม (ANN) ซึ่งรายละเอียดของเทคนิค GPR จะใช้ฟังก์ชันค่าคงที่ (Constant kernel) ฟังก์ชันเรเดียลเบสซิส (Radial basis function) และฟังก์ชันสีขาว (White kernel) เป็นฟังก์ชันเคอร์เนล เทคนิค ANN จะใช้จำนวนชั้นแอบแฝงเท่ากับ 6 ชั้นและในแต่ละชั้นมีจำนวน 30 โหนด อีกทั้งยังใช้ฟังก์ชันกระตุ้น คือ ฟังก์ชันเรลู (ReLU) ในชั้นแอบแฝงและฟังก์ชันเชิงเส้น (Linear) ในชั้นข้อมูลส่งออก และเทคนิค SVR ใช้ฟังก์ชันเรเดียลเบสซิสเป็นฟังก์ชันเคอร์เนล

จากการทดสอบเมื่อนำอัตราความสำเร็จในทุกสัญญาณรบกวนของแต่ละเทคนิคมาเปรียบเทียบกันจะได้ดังภาพที่ 4-1 ซึ่งพบว่าเทคนิค GPR จะให้อัตราความสำเร็จสูงที่สุดในทุกสัญญาณรบกวน ซึ่งมีค่าเท่ากับ 97-100% เนื่องจากพื้นฐานของเทคนิค GPR จะสามารถจัดการกับสัญญาณรบกวนได้ดี แม้สัญญาณรบกวนเพิ่มขึ้น อีกทั้งยังมีค่ามากกว่าการศึกษาก่อนหน้านี้ (Jiang et al., 2021) ที่มีอัตราความสำเร็จเพียง 95% แต่ในเทคนิค ANN และ SVR จะมีอัตราความสำเร็จใกล้เคียงกันในทุกสัญญาณรบกวนซึ่งมีค่าเท่ากับ 41-56% และ 31-50% ตามลำดับ เนื่องจากการเพิ่มขึ้นของสัญญาณรบกวนไม่ได้ส่งผลถึงความแม่นยำของแบบจำลอง จากภาพที่ 4-2 ถึงภาพที่ 4-4 จะแสดงตัวอย่างระยะห่างระหว่างปลายแขนหุ่นยนต์และเป้าหมายในแต่ละรอบการทำงานกรณีสัญญาณรบกวนเท่ากับ 0.0% ของเทคนิค GPR, ANN และ SVR ตามลำดับ

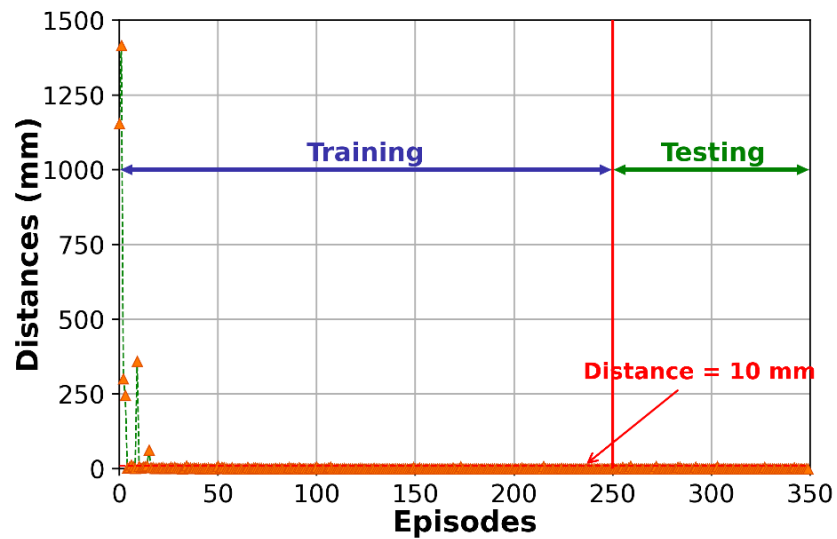


107404525

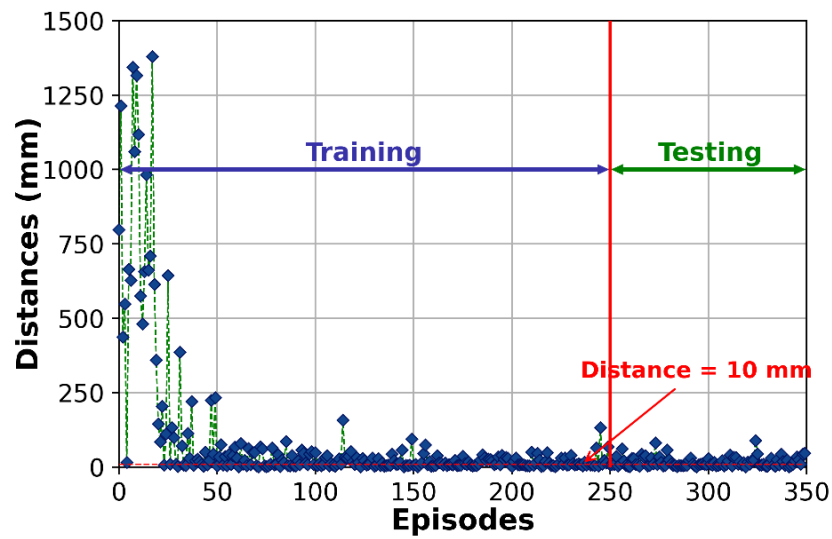
BUU-1Thesis 64910146 thesis / recv: 20062566 19:32:29 / seq: 122



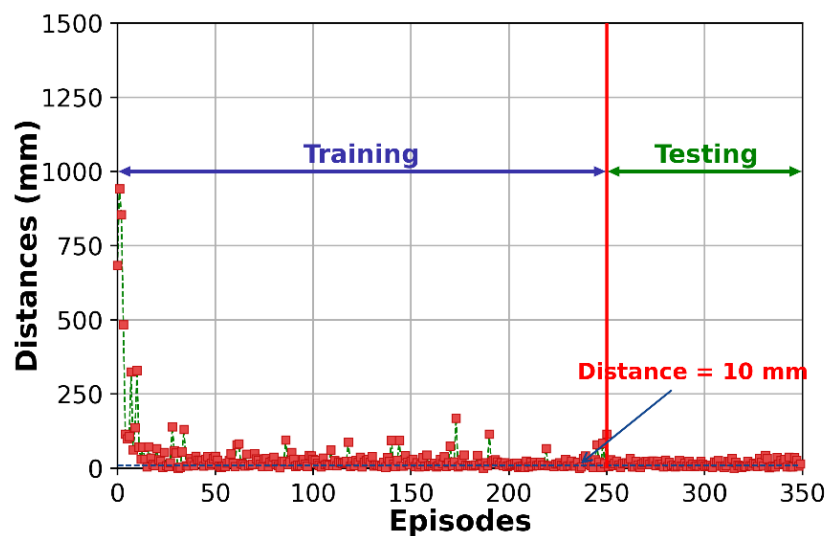
ภาพที่ 4-1 การเปรียบเทียบอัตราความสำเร็จในแต่ละเทคนิคเมื่อสัญญาณรบกวนเพิ่มขึ้นของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 2-DOF



ภาพที่ 4-2 ระยะห่างระหว่างปลายแขนหุ่นยนต์และเป้าหมายในแต่ละรอบของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 2-DOF จากเทคนิค GPR กรณีสัญญาณรบกวน = 0.0%



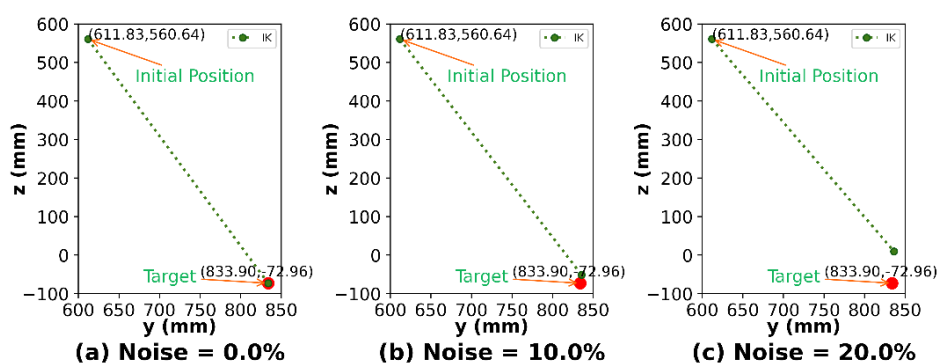
ภาพที่ 4-3 ระยะห่างระหว่างปลายแขนหุ่นยนต์และเป้าหมายในแต่ละรอบของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 2-DOF จากเทคนิค ANN กรณีสัญญาณรบกวน = 0.0%



ภาพที่ 4-4 ระยะห่างระหว่างปลายแขนหุ่นยนต์และเป้าหมายในแต่ละรอบของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 2-DOF จากเทคนิค SVR กรณีสัญญาณรบกวน = 0.0%

สำหรับวิธี IK พบว่าอัตราความสำเร็จมีค่าลดลงเมื่อสัญญาณรบกวนเพิ่มขึ้น ซึ่งต่ำสุดถึง 20% ในกรณีที่มีสัญญาณรบกวนเท่ากับ 20.0% เนื่องจากการเพิ่มขึ้นของสัญญาณรบกวน ทำให้ข้อต่อมีความคลาดเคลื่อนในการหมุนมากยิ่งขึ้น จึงส่งผลให้ปลายแขนหุ่นยนต์ไปถึงตำแหน่งเป้าหมาย

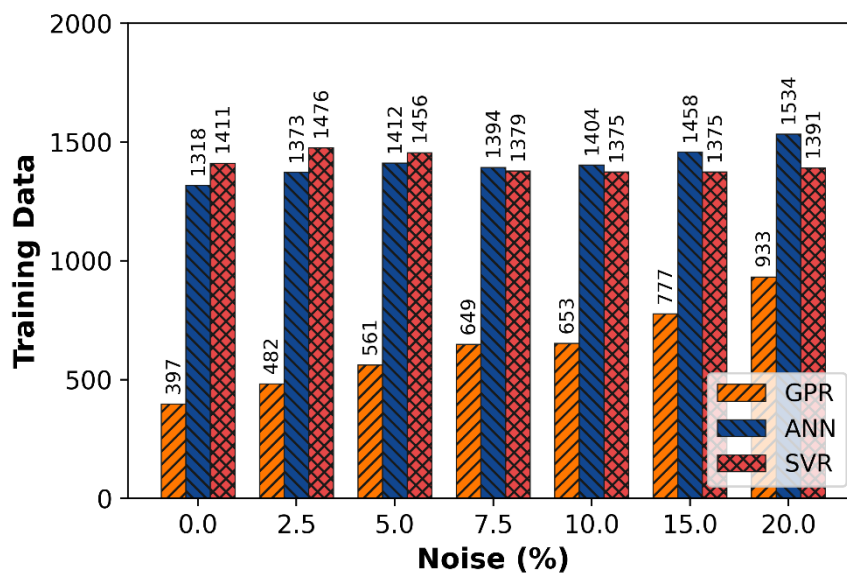
ได้ยากขึ้น ตัวอย่างเช่น หากตำแหน่งพิกัดเป้าหมายเท่ากับ (833.90, -72.96) มม. กรณีที่ไม่มีสัญญาณรบกวนจะสามารถคำนวณหามุมข้อต่อที่ 1 และ 2 ได้เท่ากับ -15.0 และ 20.0 องศาตามลำดับ และสามารถหมุนไปยังตำแหน่งเป้าหมายได้สำเร็จ แต่หากมีสัญญาณรบกวนเท่ากับ 20% อาจจะทำให้แขนหุ่นยนต์หมุนไปยังมุม -9.6 และ 20.6 องศาที่มีพิกัดเท่ากับ (836.27, 9.68) มม. ซึ่งไม่สามารถไปถึงเป้าหมายได้ เนื่องจากมีระยะห่างระหว่างปลายแขนหุ่นยนต์กับเป้าหมายเท่ากับ 82.67 มม. โดยตัวอย่างของเส้นทางการเคลื่อนที่สำหรับวิธี IK กรณีสัญญาณรบกวนเท่ากับ 0.0%, 10.0% และ 20.0% จะแสดงดังภาพที่ 4-5



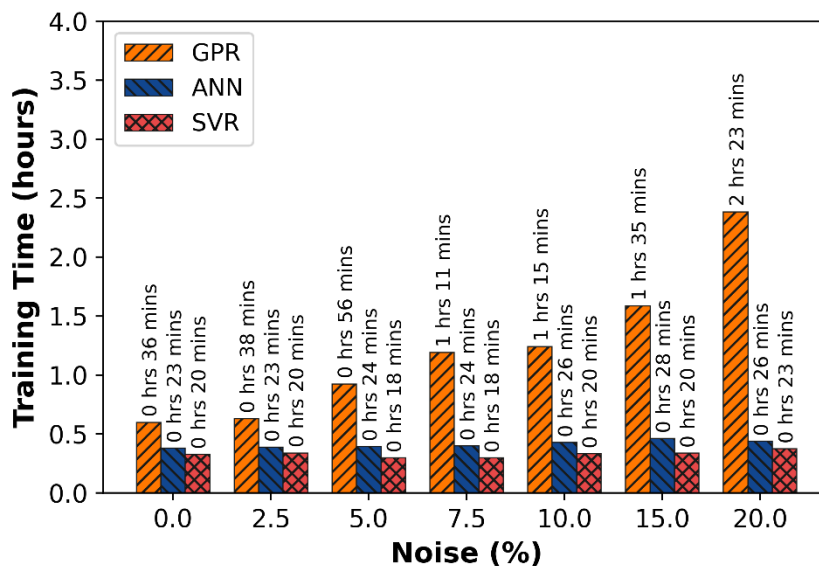
ภาพที่ 4-5 เส้นทางการเคลื่อนที่ของวิธี IK ในปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 2-DOF

อีกทั้งเมื่อเปรียบเทียบข้อมูลที่ใช้ในการฝึกสอนของแต่ละเทคนิคดังภาพที่ 4-6 จะพบว่าเทคนิค GPR จะใช้จำนวนข้อมูลที่น้อยกว่า ANN, SVR และการศึกษาก่อนหน้า (Jiang et al., 2021) ที่ใช้ข้อมูลในการฝึกสอนมากถึง 360,000 ข้อมูล แต่จำนวนข้อมูลจะมีค่าเพิ่มขึ้นเมื่อสัญญาณรบกวนมากขึ้น เนื่องจากการเพิ่มขึ้นของสัญญาณรบกวนจะทำให้เคลื่อนที่ไปถึงเป้าหมายได้ยากยิ่งขึ้น ดังนั้นจึงต้องใช้จำนวนครั้งในการเคลื่อนที่แต่ละรอบมากขึ้นซึ่งส่งผลต่อจำนวนข้อมูลในการฝึกสอน แต่อย่างไรก็ตามเมื่อพิจารณาถึงเวลาในการฝึกสอน (Training time) ของแต่ละเทคนิคดังภาพที่ 4-7 จะพบว่าเทคนิค GPR จะใช้เวลาการฝึกสอนมากที่สุดในทุกสัญญาณรบกวน ซึ่งมีค่าถึง 2 ชั่วโมง 23 นาที ในกรณีที่สัญญาณรบกวนเท่ากับ 20.0% เนื่องจากเทคนิค GPR จะใช้เมทริกซ์ความแปรปรวนร่วมในการคำนวณเพื่อสร้างแบบจำลอง หากสัญญาณรบกวนเพิ่มขึ้น ความยากในการคำนวณจึงยากขึ้นไปด้วย แต่ในเทคนิค ANN และ SVR มีเวลาในการฝึกสอนใกล้เคียงกันในทุกสัญญาณรบกวน ซึ่งมากที่สุดเพียง 28 นาทีในกรณีที่สัญญาณรบกวนเท่ากับ 15.0% และ 23 นาทีในกรณีที่สัญญาณรบกวนเท่ากับ 20.0% ตามลำดับ เนื่องจากมีพื้นฐานในการคำนวณที่รวดเร็ว และสัญญาณรบกวนที่เพิ่มขึ้นไม่ได้มีผลต่อความยากในการคำนวณ แต่เมื่อนำเทคนิค GPR มา

เปรียบเทียบกับการศึกษาก่อนหน้านี้ (Jiang et al., 2021) พบว่าเทคนิค GPR จะยังใช้เวลาการฝึกสอนที่น้อยกว่าการศึกษาก่อนหน้านี้ ซึ่งใช้เวลามากถึง 15 ชั่วโมง 24 นาที



ภาพที่ 4-6 ข้อมูลการฝึกสอนในแต่ละเทคนิคเมื่อสัญญาณรบกวนเพิ่มขึ้นของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 2-DOF



ภาพที่ 4-7 เวลาการฝึกสอนในแต่ละเทคนิคเมื่อสัญญาณรบกวนเพิ่มขึ้นของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 2-DOF

จากผลการทดสอบของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 2 องศาอิสระพบว่าวิธี MBRL ที่ใช้เทคนิค GPR จะให้อัตราความสำเร็จสูงที่สุด แม้ว่าจะใช้เวลาฝึกสอนมากที่สุด แต่ในช่วงที่มีสัญญาณรบกวนต่ำ ๆ ก็ใช้เวลาที่ใกล้เคียงกับเทคนิค ANN และ SVR อีกทั้งในทางปฏิบัติแล้ว หากแขนหุ่นยนต์สามารถไปถึงเป้าหมายได้ ก็ถือว่าบรรลุวัตถุประสงค์ของปัญหา ดังนั้นจึงนำวิธี MBRL ที่ใช้เทคนิค GPR มาใช้ในการทดสอบสำหรับปัญหาถัดไป

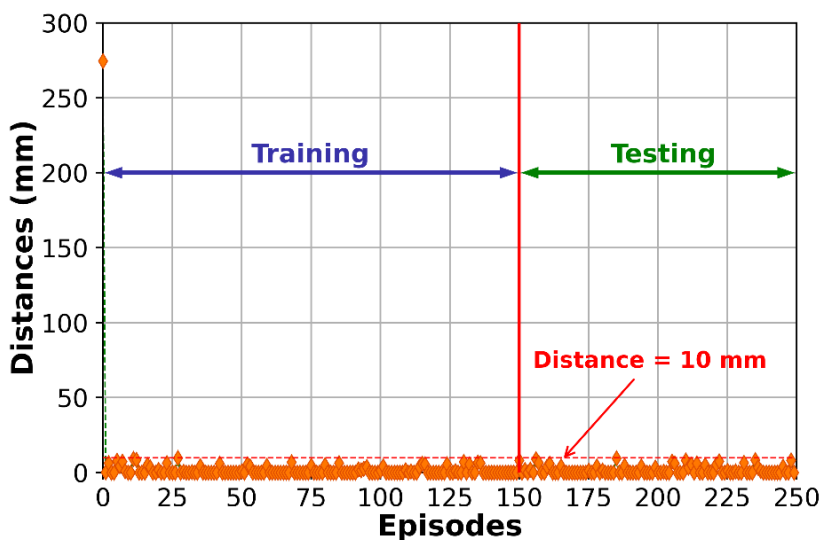
2. ผลการทดสอบปัญหาการวาง

ในการทดสอบปัญหาการวาง ได้กำหนดเงื่อนไขการฝึกสอนของวิธี MBRL โดยกำหนดให้จำนวนครั้งการเคลื่อนที่ในแต่ละรอบการทำงานมีค่าเท่ากับ 17 ครั้ง จำนวนรอบการทำงานของการเคลื่อนที่แบบสุ่ม จำนวนรอบการสร้างแบบจำลองสิ่งแวดล้อม และจำนวนรอบการทดสอบเท่ากับ 1, 150 และ 100 รอบตามลำดับ อีกทั้งตำแหน่งเริ่มต้นของปลายแขนหุ่นยนต์ในทุกรอบการทำงานจะมีพิกัดอยู่ที่ (250, 250) มม.

จากการทดสอบปัญหาการวางจะได้ผลการทดสอบดังแสดงในตารางที่ 4-1 ซึ่งพบว่าข้อมูลและเวลาที่ใช้ในการฝึกสอนจะมีค่าน้อยกว่าการศึกษาก่อนหน้านี้ (Jiang et al., 2021) ที่มีค่ามากถึง 270,000 ข้อมูลและ 14 ชั่วโมง 1 นาทีตามลำดับ อีกทั้งยังมีอัตราความสำเร็จที่มากกว่าการศึกษาก่อนหน้านี้ที่มีค่าเท่ากับ 98% ซึ่งจากภาพที่ 4-8 จะแสดงระยะห่างระหว่างปลายแขนหุ่นยนต์และเป้าหมายในแต่ละรอบการทำงาน

ตารางที่ 4-1 ผลการทดสอบปัญหาการวางด้วยวิธี MBRL

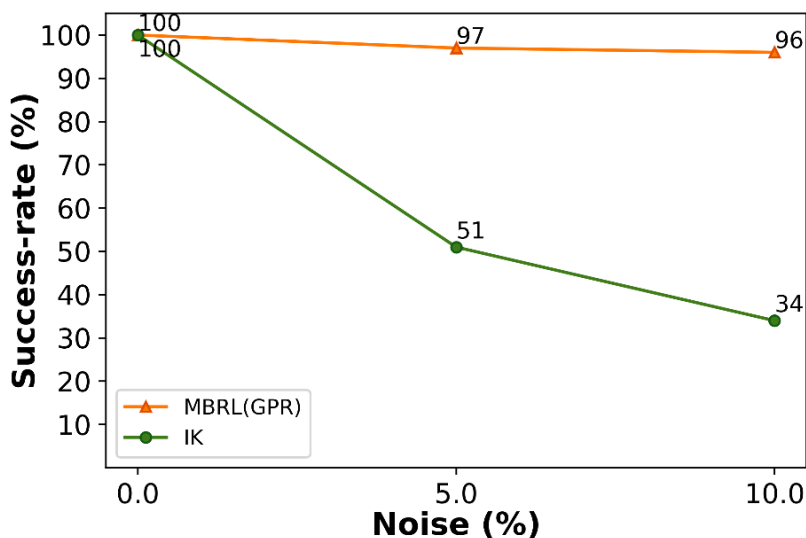
ตัวแปร	ค่า
อัตราความสำเร็จ	100%
ข้อมูลในการฝึกสอน	1405
เวลาในการฝึกสอน	3 ชั่วโมง 20 นาที



ภาพที่ 4-8 ระยะห่างระหว่างปลายแขนหุ่นยนต์และเป้าหมายในแต่ละรอบของปัญหาการวาง

3. ผลการทดสอบปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 3 องศาอิสระ

ในการทดสอบปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 3 องศาอิสระ ได้กำหนดเงื่อนไขการฝึกสอนของวิธี MBRL เหมือนกับปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 2 องศาอิสระ ซึ่งจากการทดสอบเมื่อเปรียบเทียบอัตราความสำเร็จในทุกสถานการณ์ระหว่างวิธี MBRL(GPR) กับวิธี IK ดังภาพที่ 4-9 จะพบว่าอัตราความสำเร็จที่ได้จากวิธี MBRL(GPR) จะมีค่ามากกว่าวิธี IK ในทุกสถานการณ์ แม้ในกรณีที่มีสถานการณ์ยากเท่ากับ 10.0% ยังให้อัตราความสำเร็จสูงถึง 96% ในขณะที่วิธี IK มีอัตราความสำเร็จเพียง 34% และเมื่อพิจารณาถึงข้อมูลและเวลาในการฝึกสอนดังตารางที่ 4-2 จะพบว่ามีค่าเพิ่มขึ้นตามสถานการณ์ที่ยากขึ้น ซึ่งเป็นไปในการทำงานเดียวกันกับปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 2 องศาอิสระ



ภาพที่ 4-9 การเปรียบเทียบอัตราความสำเร็จเมื่อสัญญาณรบกวนเพิ่มขึ้นของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 3-DOF

ตารางที่ 4-2 ผลการทดสอบปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 3-DOF

ตัวแปร	สัญญาณรบกวน (%)		
	0.0	5.0	10.0
อัตราความสำเร็จ	100%	97%	96%
ข้อมูลในการฝึกสอน	654	995	1293
เวลาในการฝึกสอน	0 ชั่วโมง 57 นาที	1 ชั่วโมง 37 นาที	2 ชั่วโมง 27 นาที

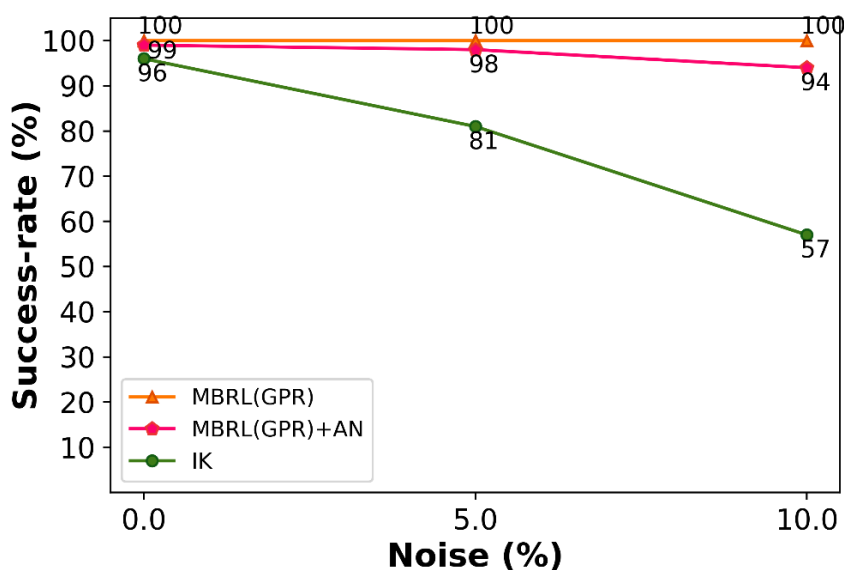
4. ผลการทดสอบปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4 องศาอิสระ

ในการทดสอบปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4 องศาอิสระ ได้กำหนดเงื่อนไขการฝึกสอนของวิธี MBRL โดยให้จำนวนครั้งการเคลื่อนที่ในแต่ละรอบการทำงานมีค่าเท่ากับ 6 ครั้ง จำนวนรอบการทำงานของการเคลื่อนที่แบบสุ่ม จำนวนรอบการสร้างแบบจำลองสิ่งแวดล้อม และจำนวนรอบการทดสอบเท่ากับ 1, 150 และ 100 รอบตามลำดับ อีกทั้งตำแหน่งมุมข้อต่อเริ่มต้นของแต่ละข้อต่อจะมีค่าเท่ากับศูนย์องศาในทุกรอบการทำงาน นอกจากนี้ยังมีการนำวิธี MBRL มาใช้ในการสร้างโครงข่ายนักแสดง (MBRL+AN) โดยกำหนดเงื่อนไขการฝึกสอนให้มีจำนวนรอบการสร้างโครงข่ายนักแสดงเท่ากับ 150 รอบ และมีรายละเอียดของโครงข่ายประสาท

เทียมนี่ใช้ คือ จำนวนชั้นแอบแฝงเท่ากับ 3 ชั้นและในแต่ละชั้นมีจำนวน 32 โหนด อีกทั้งยังใช้ ฟังก์ชันกระตุ้น คือ ฟังก์ชัน LeakyReLU ในชั้นแอบแฝงและฟังก์ชัน Tanh ในชั้นข้อมูลส่งออก

4.1 ผลการทดสอบปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4 องศาอิสระในแนวระนาบ

จากการทดสอบเมื่อเปรียบเทียบอัตราความสำเร็จในทุกสถานการณ์ระหว่างวิธี IK, วิธี MBRL(GPR) และวิธี MBRL(GPR)+AN ดังภาพที่ 4-10 พบว่าวิธี MBRL(GPR) จะให้อัตราความสำเร็จมากที่สุดในทุกสถานการณ์ซึ่งมีค่าเท่ากับ 100% รองลงมาคือวิธี MBRL(GPR)+AN และวิธี IK ตามลำดับ และเมื่อพิจารณาถึงข้อมูลและเวลาในการฝึกสอนของวิธี MBRL(GPR) ดังตารางที่ 4-3 จะพบว่าข้อมูลในการฝึกสอนมีค่าเพิ่มขึ้นเมื่อสถานการณ์มีค่ามากขึ้น แต่เวลาในการฝึกสอนมีค่าใกล้เคียงกัน ทั้งนี้เนื่องจากขอบเขตของเป้าหมายถูกจำกัดด้วยการหมุนของมุมข้อต่อที่ 4 ที่ปลายแขนหุ่นยนต์ต้องขนานกับพื้นในทุกครั้งของการเคลื่อนที่ ทำให้เมื่อสถานการณ์มีค่าเพิ่มขึ้นจึงไม่ส่งผลต่อความยากในการคำนวณ



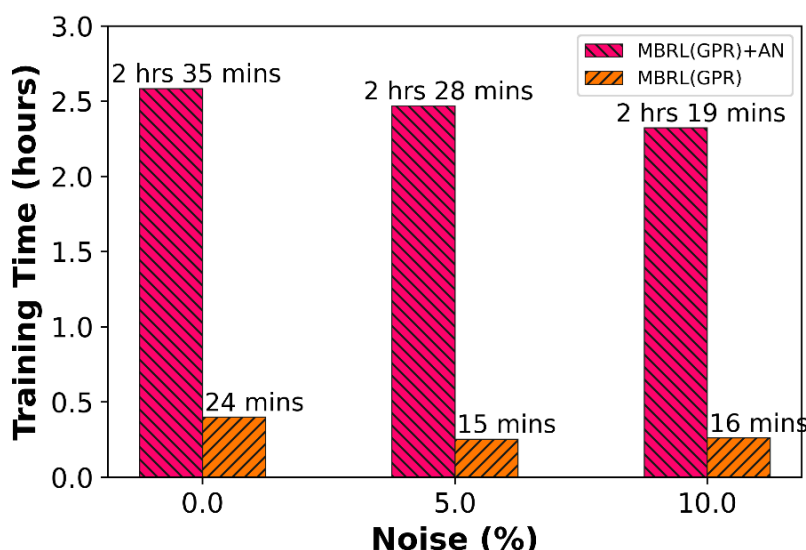
ภาพที่ 4-10 การเปรียบเทียบอัตราความสำเร็จในแต่ละวิธีเมื่อสถานการณ์เพิ่มขึ้นของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4-DOF ในแนวระนาบ

ตารางที่ 4-3 ผลการทดสอบปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4-DOF ในแนวระนาบด้วยวิธี MBRL(GPR)

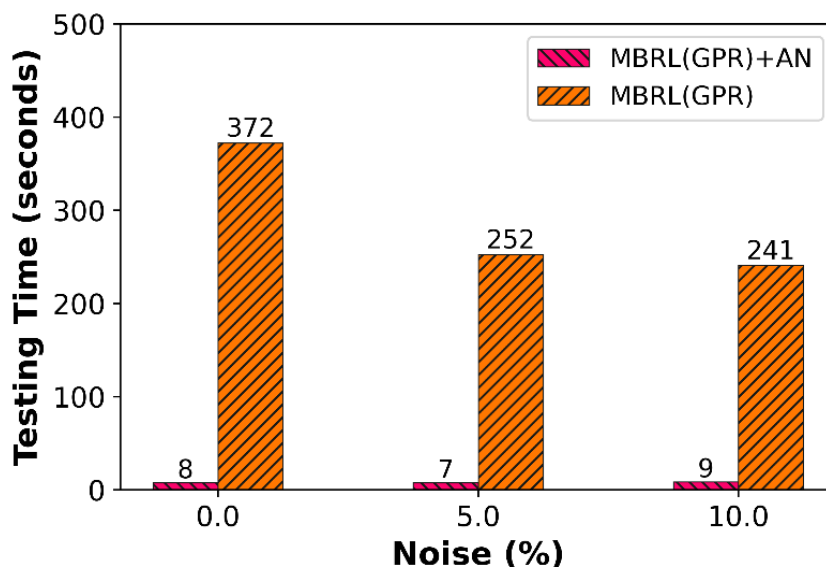
ตัวแปร	สัญญาณรบกวน (%)		
	0.0	5.0	10.0
อัตราความสำเร็จ	100%	100%	100%
ข้อมูลในการฝึกสอน	567	581	589
เวลาในการฝึกสอน	24 นาที	15 นาที	16 นาที

อีกทั้งเมื่อเปรียบเทียบเวลาการฝึกสอนในทุกสัญญาณรบกวนระหว่างวิธี

MBRL(GPR) และวิธี MBRL(GPR)+AN ดังภาพที่ 4-11 จะพบว่าวิธี MBRL(GPR)+AN ใช้เวลาในการฝึกสอนที่มากกว่าวิธี MBRL(GPR) ในทุกสัญญาณรบกวนซึ่งสูงถึง 2 ชั่วโมง 35 นาทีในกรณีสัญญาณรบกวนเท่ากับ 0.0% แต่อย่างไรก็ตามเมื่อพิจารณาถึงเวลาการทดสอบดังภาพที่ 4-12 จะพบว่าวิธี MBRL(GPR)+AN ใช้เวลาในการทดสอบที่น้อยกว่าวิธี MBRL(GPR) ซึ่งเป็นไปตามหลักการที่ได้กล่าวไว้ในการออกแบบวิธีการ MBRL(GPR)+AN โดยมีค่าต่ำสุดเท่ากับ 7 วินาทีในกรณีสัญญาณรบกวนเท่ากับ 5.0% ขณะที่วิธี MBRL(GPR) ใช้เวลาน้อยที่สุดเท่ากับ 241 วินาทีในกรณีสัญญาณรบกวนเท่ากับ 10.0%



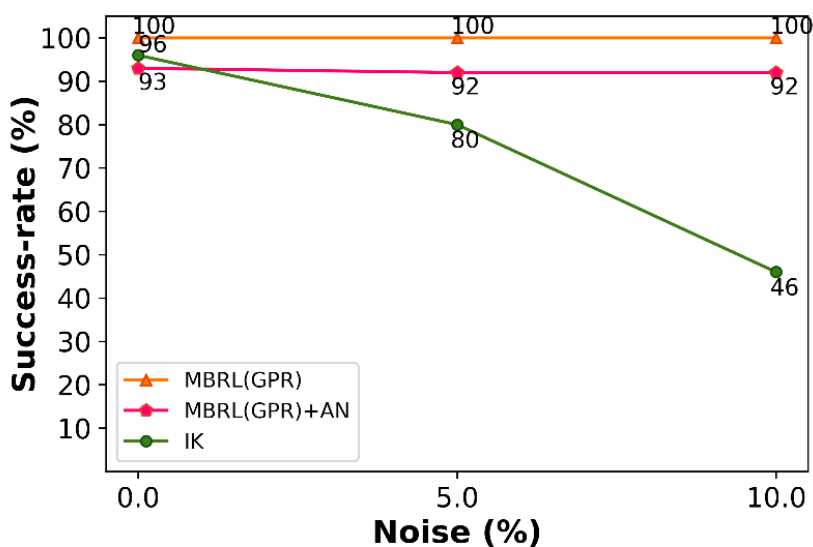
ภาพที่ 4-11 การเปรียบเทียบเวลาการฝึกสอนในแต่ละวิธีเมื่อสัญญาณรบกวนเพิ่มขึ้นของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4-DOF ในแนวระนาบ



ภาพที่ 4-12 การเปรียบเทียบเวลาการทดสอบในแต่ละวิธีเมื่อสัญญาณรบกวนเพิ่มขึ้นของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4-DOF ในแนวระนาบ

4.2 ผลการทดสอบปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4 องศาอิสระในพิกัดฉาก 3 มิติ

จากการทดสอบเมื่อเปรียบเทียบอัตราความสำเร็จในทุกสัญญาณรบกวนระหว่างวิธี IK, วิธี MBRL(GPR) และวิธี MBRL(GPR)+AN ดังภาพที่ 4-13 พบว่าวิธี MBRL(GPR) จะยังคงให้อัตราความสำเร็จสูงที่สุดในทุกสัญญาณรบกวนเท่ากับ 100% รองลงมาคือวิธี MBRL(GPR)+AN และวิธี IK ตามลำดับ และมีความสัมพันธ์ระหว่างสัญญาณรบกวนกับข้อมูลและเวลาในการฝึกสอนเป็นไปในทำนองเดียวกันกับการควบคุมการเคลื่อนที่ในแนวระนาบดังตารางที่ 4-4

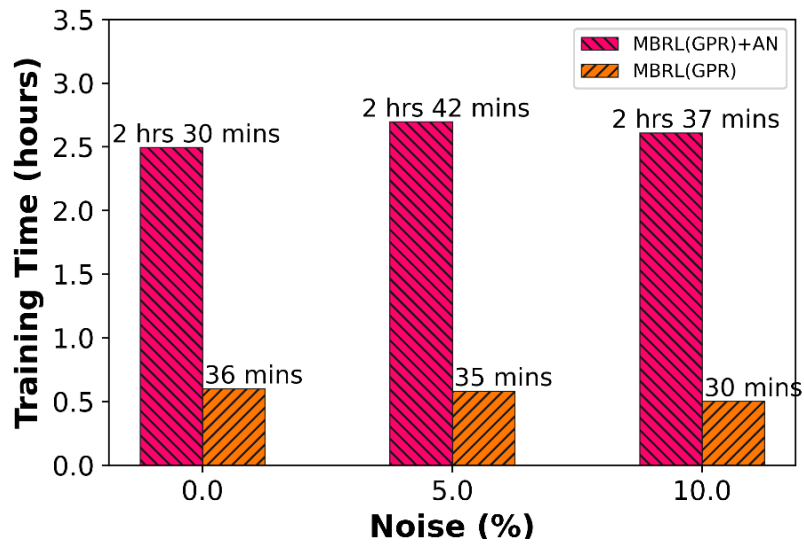


ภาพที่ 4-13 การเปรียบเทียบอัตราความสำเร็จในแต่ละวิธีเมื่อสัญญาณรบกวนเพิ่มขึ้นของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4-DOF ในแนวระนาบ

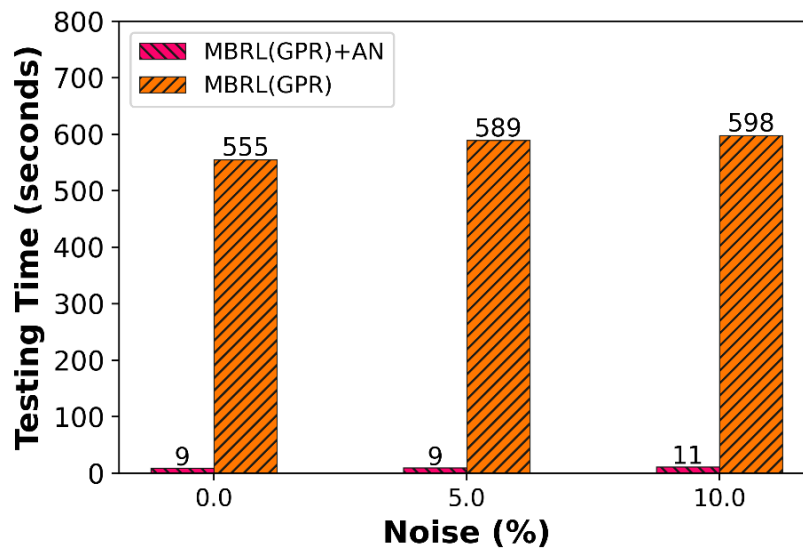
ตารางที่ 4-4 ผลการทดสอบปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4-DOF ในพิกัดฉาก 3 มิติ ด้วยวิธี MBRL(GPR)

ตัวแปร	สัญญาณรบกวน (%)		
	0.0	5.0	10.0
อัตราความสำเร็จ	100%	100%	100%
ข้อมูลในการฝึกสอน	590	624	653
เวลาในการฝึกสอน	36 นาที	35 นาที	30 นาที

อีกทั้งเมื่อเปรียบเทียบเวลาการฝึกสอนในทุกสัญญาณรบกวนระหว่างวิธี MBRL(GPR) และวิธี MBRL(GPR)+AN ดังภาพที่ 4-14 จะพบว่าวิธี MBRL(GPR)+AN ใช้เวลาในการฝึกสอนที่มากกว่าวิธี MBRL(GPR) โดยมีค่ามากที่สุดเท่ากับ 2 ชั่วโมง 42 นาทีในกรณีที่มีสัญญาณรบกวนเท่ากับ 5.0% แต่อย่างไรก็ตามเมื่อพิจารณาถึงเวลาการทดสอบดังภาพที่ 4-15 จะพบว่าวิธี MBRL(GPR)+AN ใช้เวลาการทดสอบที่น้อยกว่าวิธี MBRL(GPR) ซึ่งเป็นไปในทำนองเดียวกันกับการควบคุมการเคลื่อนที่ในแนวระนาบ โดยมีค่าต่ำสุดเท่ากับ 9 วินาทีในกรณีที่มีสัญญาณรบกวนเท่ากับ 0.0% และ 5.0% ในขณะที่วิธี MBRL(GPR) ใช้เวลาต่ำสุดเท่ากับ 555 วินาทีในกรณีที่มีสัญญาณรบกวนเท่ากับ 0.0%



ภาพที่ 4-14 การเปรียบเทียบเวลาการฝึกสอนในแต่ละวิธีเมื่อสัญญาณรบกวนเพิ่มขึ้นของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4-DOF ในพิกัดฉาก 3 มิติ



ภาพที่ 4-15 การเปรียบเทียบเวลาการทดสอบในแต่ละวิธีเมื่อสัญญาณรบกวนเพิ่มขึ้นของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4-DOF ในพิกัดฉาก 3 มิติ



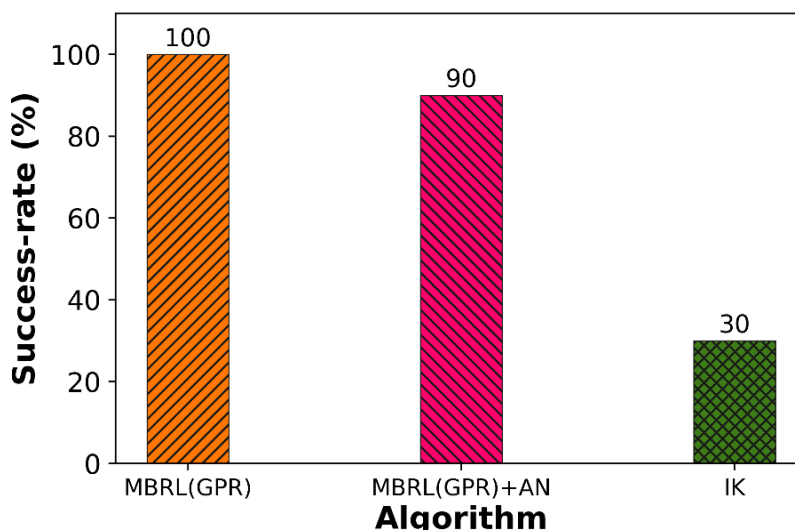
107404525

ผลการทดสอบปัญหาจริง

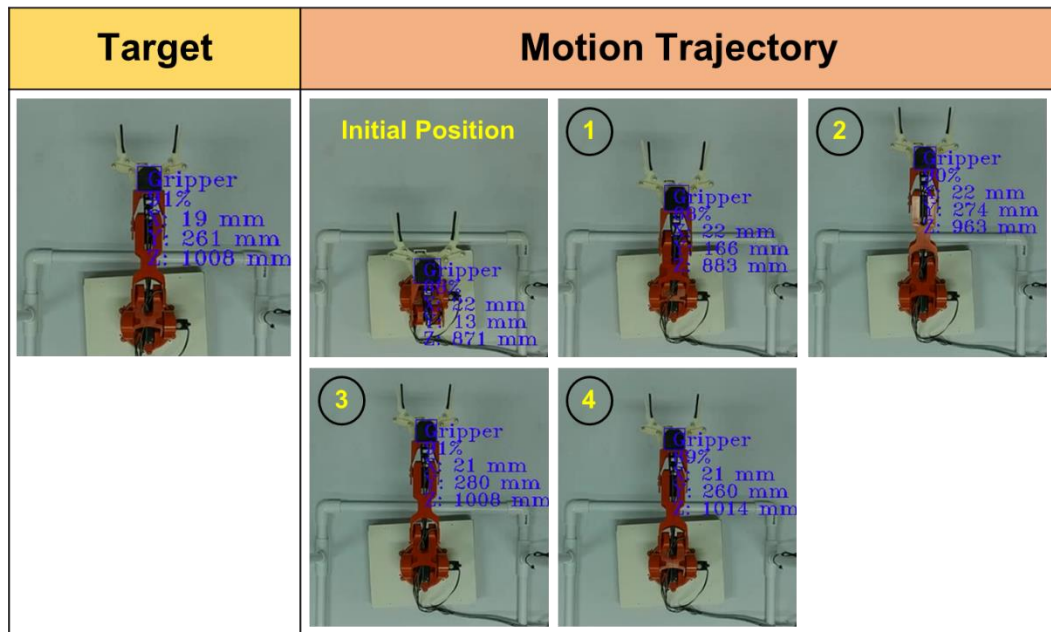
1. ผลการทดสอบการควบคุมแขนหุ่นยนต์ในแนวระนาบ

ในการทดสอบการควบคุมแขนหุ่นยนต์ในแนวระนาบ ได้กำหนดเงื่อนไขการฝึกสอนของวิธี MBRL โดยให้จำนวนครั้งการเคลื่อนที่ในแต่ละรอบการทำงานมีค่าเท่ากับ 6 ครั้ง จำนวนรอบการทำงานของการเคลื่อนที่แบบสุ่ม จำนวนรอบการสร้างแบบจำลองสิ่งแวดล้อม และจำนวนรอบการทดสอบเท่ากับ 1, 100 และ 30 รอบตามลำดับ อีกทั้งตำแหน่งมุมข้อต่อเริ่มต้นของแต่ละข้อต่อจะมีค่าเท่ากับศูนย์องศาในทุกรอบการทำงาน นอกจากนี้ยังมีการฝึกสอนด้วยวิธี MBRL(GPR)+AN โดยกำหนดเงื่อนไขการฝึกสอนให้มีจำนวนรอบการสร้างโครงข่ายนักแสดงเท่ากับ 150 รอบ และมีรายละเอียดของโครงข่ายประสาทเทียมที่ใช้คือ จำนวนชั้นแอบแฝงเท่ากับ 3 ชั้น และในแต่ละชั้นมีจำนวน 32 โหนด อีกทั้งยังใช้ฟังก์ชันกระตุ้น คือ ฟังก์ชัน LeakyReLU ในชั้นแอบแฝง และฟังก์ชัน Tanh ในชั้นข้อมูลส่งออก

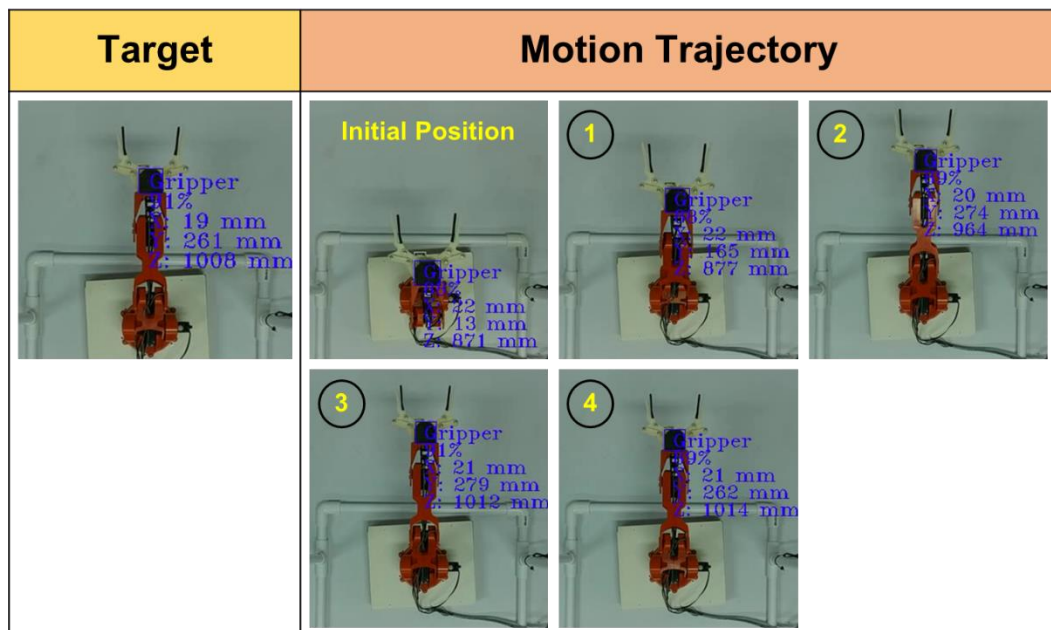
จากการทดสอบเมื่อนำวิธี MBRL(GPR) มาเปรียบเทียบกับอัตราความสำเร็จกับวิธี MBRL(GPR)+AN และวิธี IK ดังภาพที่ 4-16 พบว่าวิธี MBRL(GPR) จะให้อัตราความสำเร็จมากที่สุด โดยมีค่าเท่ากับ 100% รองลงมาคือ วิธี MBRL(GPR)+AN และวิธี IK ที่มีค่าเท่ากับ 90% และ 30% ตามลำดับ ซึ่งเป็นไปในทำนองเดียวกันกับการศึกษาเชิงตัวเลข จากภาพที่ 4-17 ถึงภาพที่ 4-19 จะแสดงตัวอย่างการเคลื่อนที่ในแต่ละครั้งของแขนหุ่นยนต์เพื่อไปยังเป้าหมายในแนวระนาบของวิธี MBRL(GPR), MBRL(GPR)+AN และวิธี IK ตามลำดับ



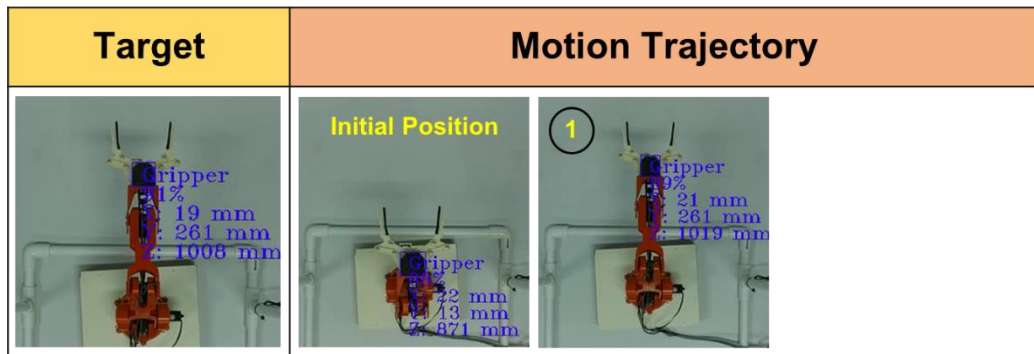
ภาพที่ 4-16 อัตราความสำเร็จในแต่ละวิธีของการควบคุมแขนหุ่นยนต์ในแนวระนาบ



ภาพที่ 4-17 การเคลื่อนที่ของแขนหุ่นยนต์ในแนวระนาบเพื่อไปยังเป้าหมายของวิธี MBRL(GPR)

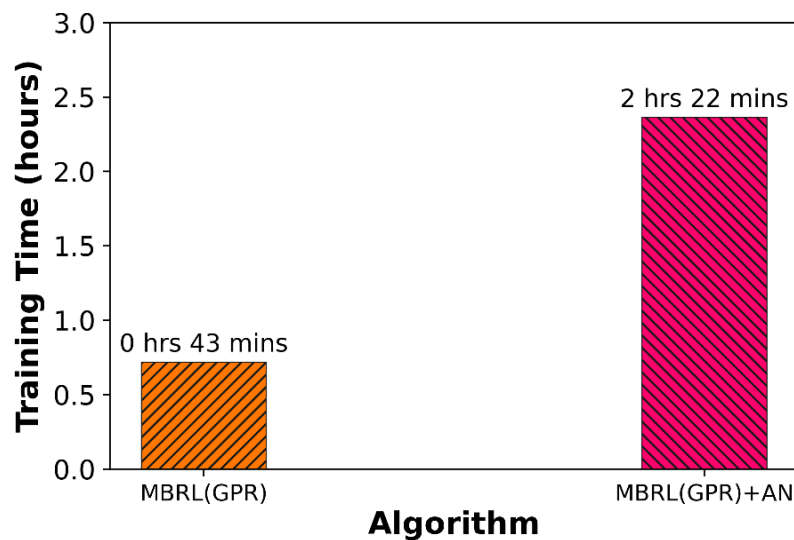


ภาพที่ 4-18 การเคลื่อนที่ของแขนหุ่นยนต์ในแนวระนาบเพื่อไปยังเป้าหมายของวิธี MBRL(GPR)+AN

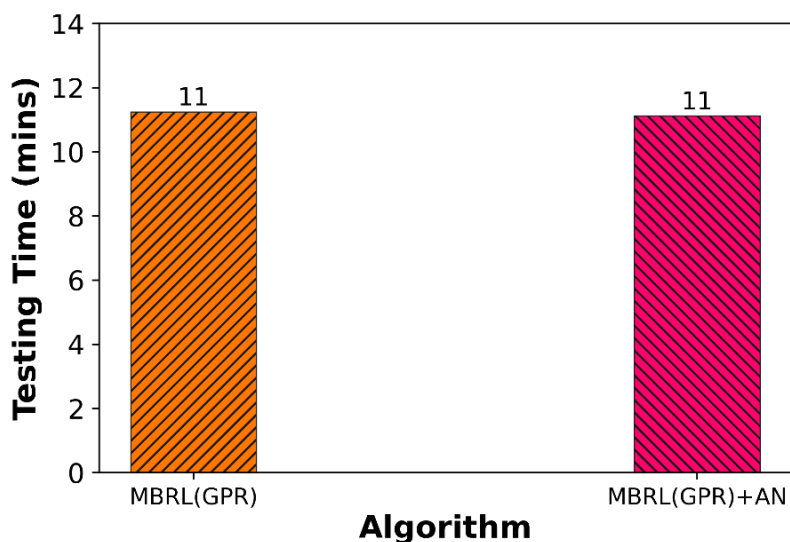


ภาพที่ 4-19 การเคลื่อนที่ของแขนหุ่นยนต์ในแนวระนาบเพื่อไปยังเป้าหมายของวิธี IK

อีกทั้งเมื่อพิจารณาเวลาในการฝึกสอนของวิธี MBRL(GPR) และวิธี MBRL(GPR)+AN ดังภาพที่ 4-20 พบว่าวิธี MBRL(GPR) จะใช้เวลาในการฝึกสอนที่น้อยกว่าวิธี MBRL(GPR)+AN ซึ่งมีค่าเท่ากับ 43 นาที ในขณะที่วิธี MBRL(GPR)+AN มีค่าเท่ากับ 2 ชั่วโมง 22 นาที และเมื่อเปรียบเทียบเวลาในการทดสอบของทั้งสองวิธีดังภาพที่ 4-21 จะพบว่าใช้เวลาในการทดสอบที่เท่ากัน คือ 11 นาที ซึ่งไม่เป็นไปในทำนองเดียวกันกับการศึกษาเชิงตัวเลข



ภาพที่ 4-20 เวลาการฝึกสอนในแต่ละวิธีของการควบคุมแขนหุ่นยนต์ในแนวระนาบ

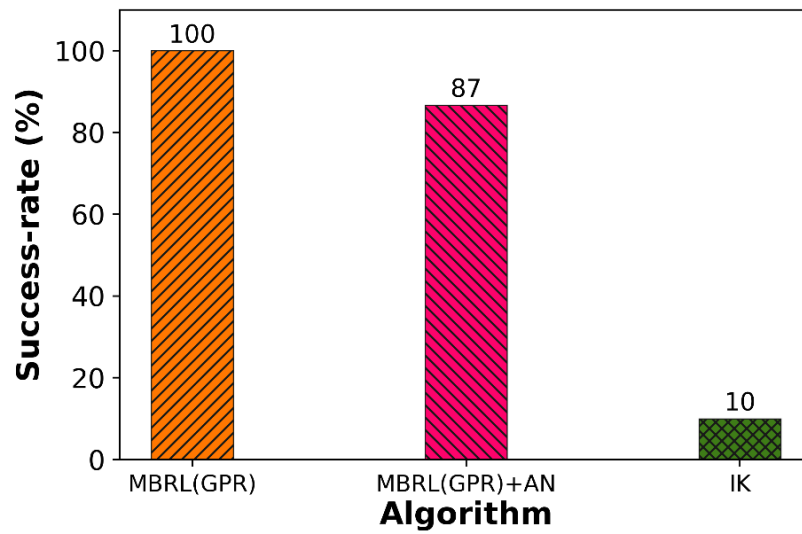


ภาพที่ 4-21 เวลาการทดสอบในแต่ละวิธีของการควบคุมแขนหุ่นยนต์ในแนวระนาบ

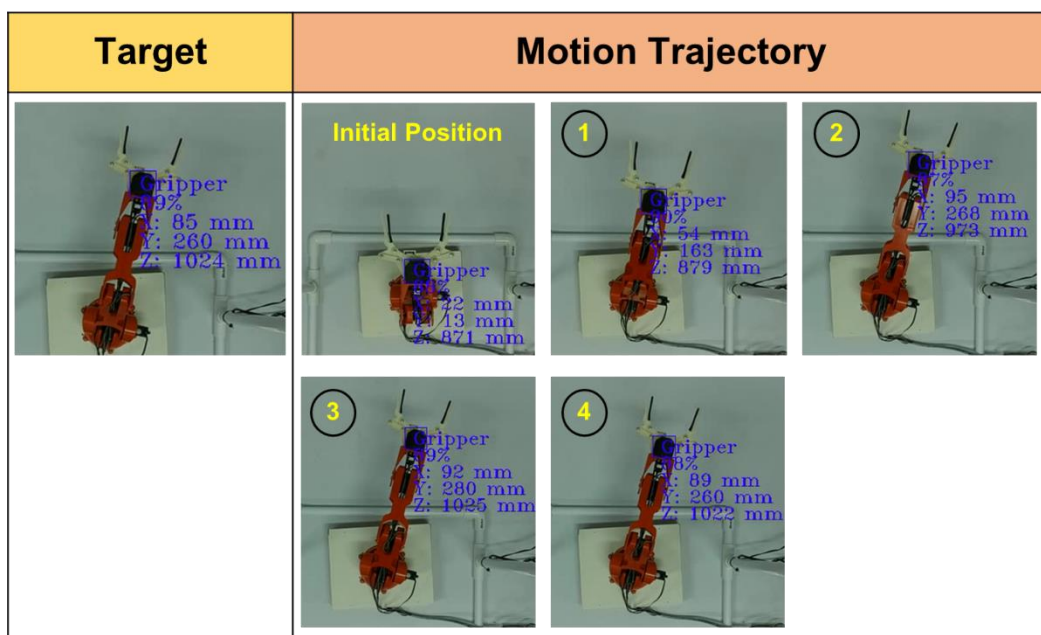
2. ผลการทดสอบการควบคุมแขนหุ่นยนต์ในพิกัดฉาก 3 มิติ

ในการทดสอบการควบคุมแขนหุ่นยนต์ในพิกัดฉาก 3 มิติ ได้กำหนดเงื่อนไขการฝึกสอนของวิธี MBRL เหมือนกับการควบคุมแขนหุ่นยนต์ในแนวระนาบ อีกทั้งยังมีการฝึกสอนด้วยวิธีการ MBRL(GPR)+AN โดยมีเงื่อนไขและรายละเอียดการฝึกสอนเช่นเดียวกัน

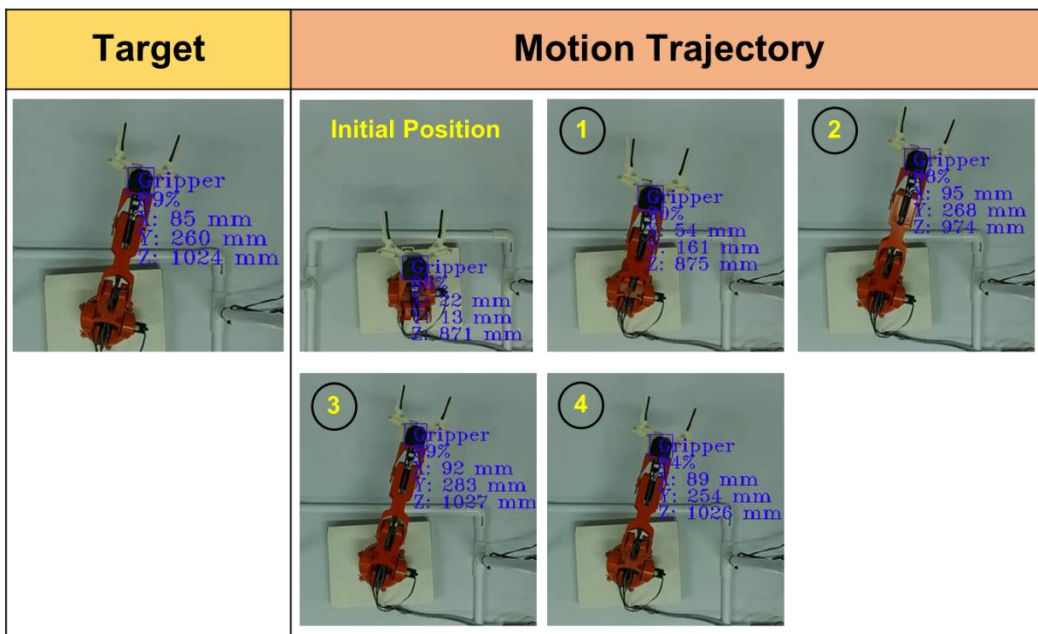
จากการทดสอบเมื่อนำวิธี MBRL(GPR) มาเปรียบเทียบกับอัตราความสำเร็จกับวิธี MBRL(GPR)+AN และวิธี IK ดังภาพที่ 4-22 พบว่าวิธี MBRL(GPR) จะยังคงให้อัตราความสำเร็จสูงที่สุด ซึ่งมีค่าเท่ากับ 100% รองลงมาคือ วิธี MBRL(GPR)+AN และวิธี IK ที่มีค่าเท่ากับ 87% และ 10% ตามลำดับ ซึ่งเป็นไปในทำนองเดียวกันกับการศึกษาเชิงตัวเลข จากภาพที่ 4-23 ถึงภาพที่ 4-25 จะแสดงตัวอย่างการเคลื่อนที่ในแต่ละครั้งของแขนหุ่นยนต์เพื่อไปยังเป้าหมายในพิกัดฉาก 3 มิติของวิธี MBRL(GPR), MBRL(GPR)+AN และวิธี IK ตามลำดับ



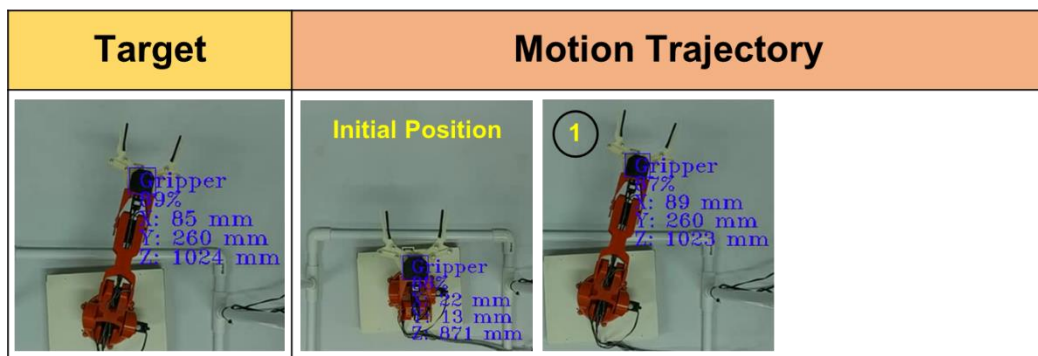
ภาพที่ 4-22 อัตราความสำเร็จในแต่ละวิธีของการควบคุมแขนหุ่นยนต์ในพิกัดฉาก 3 มิติ



ภาพที่ 4-23 การเคลื่อนที่ของแขนหุ่นยนต์ในพิกัดฉาก 3 มิติเพื่อไปยังเป้าหมายของวิธี MBRL(GPR)

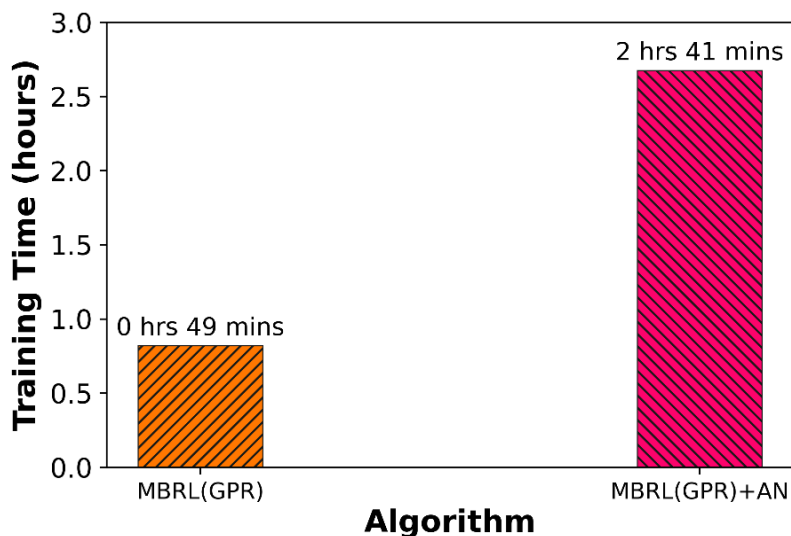


ภาพที่ 4-24 การเคลื่อนที่ของแขนหุ่นยนต์ในพิกัดฉาก 3 มิติเพื่อไปยังเป้าหมายของวิธี MBRL(GPR)+AN

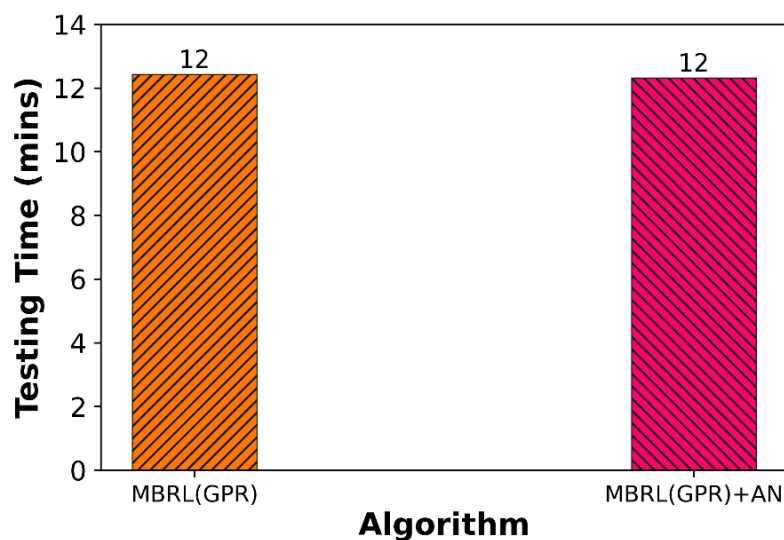


ภาพที่ 4-25 การเคลื่อนที่ของแขนหุ่นยนต์ในพิกัดฉาก 3 มิติเพื่อไปยังเป้าหมายของวิธี IK

อีกทั้งเมื่อเปรียบเทียบเวลาการฝึกสอนของวิธี MBRL(GPR) และวิธี MBRL(GPR)+AN ดังภาพที่ 4-26 พบว่าวิธี MBRL(GPR) จะใช้เวลาการฝึกสอนที่น้อยกว่าวิธี MBRL(GPR)+AN โดยมีค่าเท่ากับ 49 นาที ในขณะที่วิธี MBRL(GPR)+AN มีค่าเท่ากับ 2 ชั่วโมง 41 นาที และเมื่อพิจารณาเวลาในการทดสอบดังภาพที่ 4-27 พบว่าทั้งสองวิธีจะใช้เวลาในการทดสอบที่เท่ากัน คือ 12 นาที ซึ่งไม่เป็นไปในทำนองเดียวกันกับการศึกษาเชิงตัวเลข



ภาพที่ 4-26 เวลาการฝึกสอนในแต่ละวิธีของการควบคุมแขนหุ่นยนต์ในพิกัดฉาก 3 มิติ

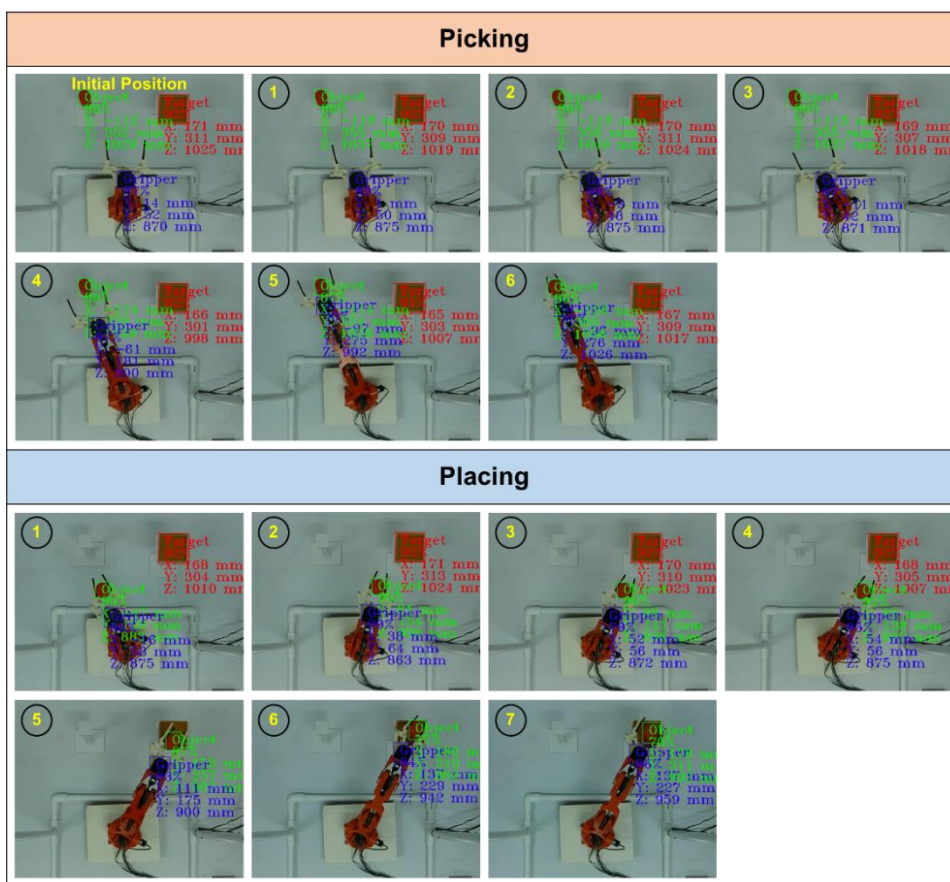


ภาพที่ 4-27 เวลาการทดสอบในแต่ละวิธีของการควบคุมแขนหุ่นยนต์ในพิกัดฉาก 3 มิติ

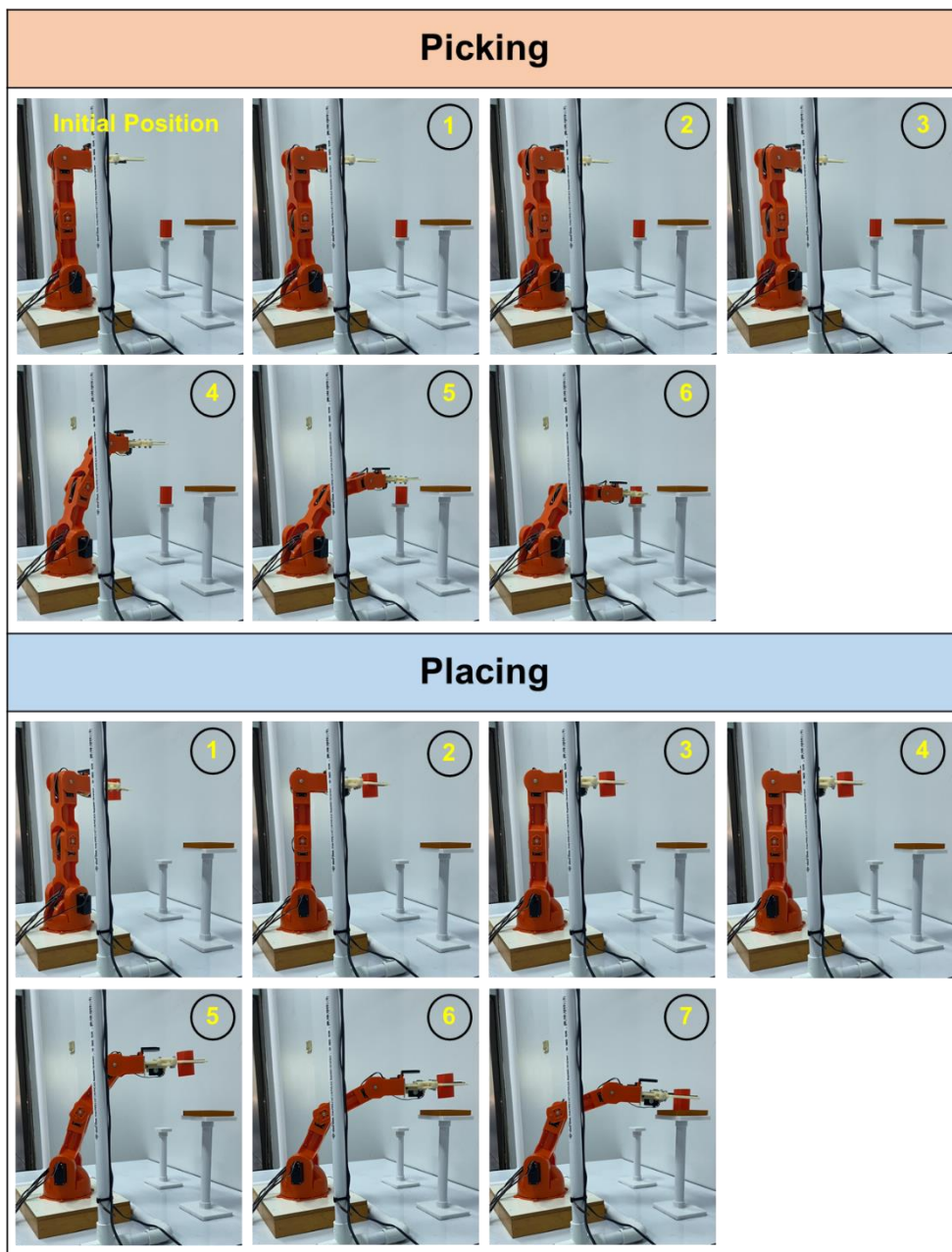
3. ผลการทดสอบการจับและวางวัตถุ

ในการทดสอบการจับและวางวัตถุได้นำวิธี MBRL ที่ฝึกสอนในแนวระนาบ มาใช้ในการทดสอบ โดยได้ออกแบบความสูงของฐานที่ใช้ในการวางวัตถุและเป้าหมายให้มีความสูงแตกต่างกันทั้งหมด 6 ความสูง คือ 165 มม., 200 มม., 215 มม., 230 มม., 260 มม. และ 285 มม. ซึ่งอยู่ภายในขอบเขตของการฝึกสอน จากนั้นจึงทดสอบโดยการกำหนดตำแหน่งของวัตถุที่ต้องการจับ

และตำแหน่งเป้าหมายที่ต้องการวางด้วยการสุ่มจากความสูงของฐานที่ออกแบบไว้ และให้แขนหุ่นยนต์เคลื่อนที่ไปยังพิกัดที่ตรวจจับได้โดยการชดเชยค่าในการจับและวางวัตถุทั้งหมด 36 รอบ ซึ่งผลที่ได้จากการทดสอบพบว่าแขนหุ่นยนต์สามารถจับและวางวัตถุได้สำเร็จ 100% และในแต่ละรอบจะใช้เวลาประมาณ 1 นาที 30 วินาที ภาพที่ 4-28 ถึงภาพที่ 4-29 จะแสดงตัวอย่างการจับและวางวัตถุผ่านมุมมองของกล้อง OAK-D และมุมมองด้านข้างตามลำดับ แต่จะพบว่า การจับวัตถุในแต่ละรอบจะมีการจับในตำแหน่งที่แตกต่างกันไป ซึ่งตำแหน่งการจับที่ดีที่สุดคือ กึ่งกลางของวัตถุ เนื่องจากสามารถจับได้แน่นและมั่นคง แต่หากจับในตำแหน่งด้านบนหรือด้านล่างของวัตถุอาจทำให้วัตถุตกหล่น หรือมีการพลิกได้ในระหว่างการเคลื่อนที่ไปยังตำแหน่งเป้าหมาย ภาพที่ 4-30 จะแสดงรูปแบบของการจับวัตถุในแต่ละแบบ อีกทั้งการวางวัตถุในแต่ละรอบ ก็จะมีตำแหน่งที่ต่างกันออกไปเช่นเดียวกัน โดยในบางครั้งอาจมีการเคลื่อนที่เข้าไปกระแทกระหว่างวัตถุและเป้าหมายก่อนวางวัตถุ ซึ่งอาจทำให้เกิดการเสียหายได้ หรือการวางวัตถุด้วยระยะห่างระหว่างวัตถุและเป้าหมายที่สูงเกินไปดังภาพที่ 4-31 ซึ่งอาจทำให้วัตถุล้มลงหรือเกิดความเสียหายได้เช่นกัน



ภาพที่ 4-28 การเคลื่อนที่ในแต่ละครั้งของการจับและวางวัตถุจากมุมมองกล้อง OAK-D

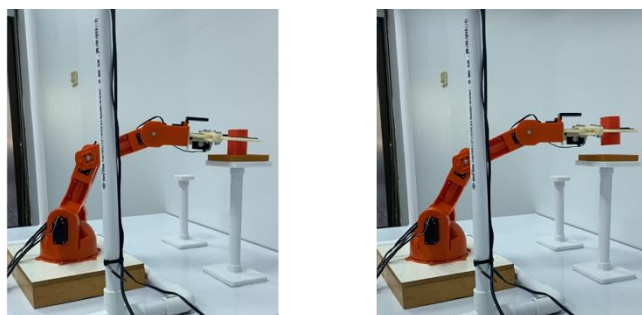


ภาพที่ 4-29 การเคลื่อนที่ในแต่ละครั้งของการจับและวางวัตถุจากมุมมองด้านข้าง



(a) การจับตรงกึ่งกลางวัตถุ (b) การจับด้านล่างวัตถุ (c) การจับด้านบนวัตถุ

ภาพที่ 4-30 รูปแบบการจับวัตถุ



(a) การวางในตำแหน่งที่เหมาะสม (b) การวางในตำแหน่งที่สูงเกินไป

ภาพที่ 4-31 รูปแบบการวางวัตถุ

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

สรุปผลการวิจัย

งานวิจัยนี้ได้แบ่งการทดสอบออกเป็นสองส่วน คือ การศึกษาเชิงตัวเลข 4 ปัญหาและการทดสอบปัญหาจริง 3 ปัญหา ซึ่งการศึกษาเชิงตัวเลขของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 2 องศาอิสระที่มีการเพิ่มสัญญาณรบกวนเท่ากับ 0.0%, 2.5%, 5.0%, 7.5%, 10.0%, 15.0% และ 20.0% ได้นำวิธีจลนศาสตร์แบบผกผัน (Inverse kinematics: IK) มาเปรียบเทียบกับเทคนิคการถดถอยการเรียนรู้ของเครื่อง 3 เทคนิคที่ใช้ในวิธี MBRL ได้แก่ การถดถอยของกระบวนการเกาส์เซียน (Gaussian process regression: GPR) ซัพพอร์ตเวกเตอร์แมชชีนสำหรับการถดถอย (Support vector regression: SVR) และโครงข่ายประสาทเทียม (Artificial neural network: ANN) จากการทดสอบพบว่าเทคนิค GPR จะให้อัตราความสำเร็จสูงที่สุดในทุกสัญญาณรบกวนโดยมีค่าเท่ากับ 97-100% แต่ในเทคนิค ANN และ SVR จะมีอัตราความสำเร็จใกล้เคียงกันในทุกสัญญาณรบกวนซึ่งมีค่าเท่ากับ 41-56% และ 31-50% ตามลำดับ และวิธี IK จะมีอัตราความสำเร็จที่ลดลงเมื่อสัญญาณรบกวนเพิ่มขึ้น ซึ่งต่ำสุด 20% ในส่วนของข้อมูลการฝึกสอน เทคนิค GPR จะใช้จำนวนข้อมูลที่น้อยกว่าเทคนิค ANN และ SVR แต่มีค่าเพิ่มขึ้นเมื่อสัญญาณรบกวนมากขึ้น ในส่วนของเวลาในการฝึกสอน เทคนิค GPR จะใช้เวลาในการฝึกสอนมากที่สุดในทุกสัญญาณรบกวน แต่เทคนิค ANN และ SVR จะมีเวลาการฝึกสอนใกล้เคียงกันในทุกสัญญาณรบกวน แม้ว่าเทคนิค GPR จะใช้เวลาในการฝึกสอนมากที่สุด แต่ก็ถือว่ามีความเหมาะสมกว่าเทคนิคอื่น ซึ่งมีอัตราความสำเร็จเฉลี่ยเพียง 50% ดังนั้นจึงนำเทคนิค GPR ไปใช้ทดสอบกับปัญหาถัดไป ซึ่งในการศึกษาเชิงตัวเลขของปัญหาการวาง พบว่ามีอัตราความสำเร็จเท่ากับ 100%

การศึกษาเชิงตัวเลขของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 3 องศาอิสระที่มีการเพิ่มสัญญาณรบกวนเท่ากับ 0.0%, 5.0% และ 10.0% เมื่อนำวิธี MBRL เปรียบเทียบกับวิธี IK พบว่าอัตราความสำเร็จที่ได้จากวิธี MBRL จะมีค่ามากกว่าวิธี IK ในทุกสัญญาณรบกวนซึ่งเท่ากับ 96-100%

การศึกษาเชิงตัวเลขของปัญหาการไปถึงเป้าหมายของแขนหุ่นยนต์ 4 องศาอิสระที่มีการเพิ่มสัญญาณรบกวนเท่ากับ 0.0%, 5.0% และ 10.0% ได้มีการนำวิธี MBRL(GPR) เปรียบเทียบกับวิธี IK และวิธี MBRL(GPR)+Actor Network (AN) ที่เป็นการนำวิธี MBRL มาใช้ในการสร้างโครงข่ายนักแสดง กรณีการควบคุมการเคลื่อนที่ในแนวระนาบพบว่าวิธี MBRL(GPR) จะให้อัตรา



107404525

BUU-IThesis 64910146 thesis / rev: 20062566 19:32:29 / seq: 122

ความสำเร็จมากที่สุดในทุกสัญญาณรบกวนซึ่งเท่ากับ 100% รองลงมาคือ วิธี MBRL(GPR)+AN และวิธี IK ที่มีค่าเท่ากับ 94-99% และ 57-96% ตามลำดับ อีกทั้งในส่วนของเวลาการฝึกสอนวิธี MBRL(GPR) จะใช้เวลาการฝึกสอนที่น้อยกว่าวิธี MBRL(GPR)+AN ในทุกสัญญาณรบกวน แต่เมื่อพิจารณาถึงเวลาการทดสอบพบว่าวิธี MBRL(GPR) จะใช้เวลาการทดสอบที่มากกว่าวิธี MBRL(GPR)+AN และในกรณีการควบคุมการเคลื่อนที่ในพิกัดฉาก 3 มิติพบว่าวิธี MBRL(GPR) ยังคงให้อัตราความสำเร็จสูงที่สุดในทุกสัญญาณรบกวนซึ่งเท่ากับ 100% รองลงมาคือ วิธี MBRL(GPR)+AN และวิธี IK ที่มีค่าเท่ากับ 92-93% และ 46-96% ตามลำดับ อีกทั้งในส่วนของเวลาการฝึกสอนวิธี MBRL(GPR) จะใช้เวลาการฝึกสอนที่น้อยกว่าวิธี MBRL(GPR)+AN ในทุกสัญญาณรบกวน แต่จะใช้เวลาการทดสอบที่มากกว่าวิธี MBRL(GPR)+AN ซึ่งเป็นไปในทำนองเดียวกันกับการควบคุมการเคลื่อนที่ในแนวระนาบ

ในส่วนการทดสอบปัญหาจริงได้นำวิธี MBRL(GPR) มาทำงานร่วมกับการตรวจจับวัตถุ จากกล้อง 3 มิติ โดยกรณีการควบคุมแขนหุ่นยนต์ในแนวระนาบ เมื่อเปรียบเทียบวิธี MBRL(GPR) วิธี MBRL(GPR)+AN และวิธี IK พบว่าวิธี MBRL(GPR) จะให้อัตราความสำเร็จสูงที่สุด โดยมีค่าเท่ากับ 100% รองลงมาคือ วิธี MBRL(GPR)+AN และวิธี IK ที่มีค่าเท่ากับ 90% และ 30% ตามลำดับ ซึ่งเป็นไปในทำนองเดียวกันกับการศึกษาเชิงตัวเลข และเมื่อพิจารณาถึงเวลาการฝึกสอนพบว่าวิธี MBRL(GPR) จะใช้เวลาในการฝึกสอนที่น้อยกว่าวิธี MBRL(GPR)+AN ส่วนเวลาในการทดสอบจะใช้เวลาที่เท่ากัน คือ 11 นาที

การทดสอบปัญหาจริงของการควบคุมแขนหุ่นยนต์ในพิกัดฉาก 3 มิติเมื่อเปรียบเทียบวิธี MBRL(GPR) กับวิธี MBRL(GPR)+AN และวิธี IK พบว่าวิธี MBRL(GPR) ยังคงให้อัตราความสำเร็จสูงที่สุดเท่ากับ 100% รองลงมาคือ วิธี MBRL(GPR)+AN และวิธี IK ที่มีค่าเท่ากับ 87% และ 10% ตามลำดับ ซึ่งเป็นไปในทำนองเดียวกันกับการศึกษาเชิงตัวเลข อีกทั้งวิธี MBRL(GPR) ยังใช้เวลาการฝึกสอนที่น้อยกว่าวิธี MBRL(GPR)+AN และใช้เวลาการทดสอบที่เท่ากัน คือ 12 นาที

การทดสอบปัญหาการจับและวางวัตถุ จะนำวิธี MBRL(GPR) ที่ฝึกสอนในการควบคุมแขนหุ่นยนต์ในแนวระนาบมาใช้ในการทดสอบ โดยจากการทดสอบจับและวางวัตถุจำนวน 36 รอบ พบว่าแขนหุ่นยนต์สามารถจับและวางวัตถุได้ 100% และในแต่ละรอบจะใช้เวลาประมาณ 1 นาที 30 วินาที แต่ในการจับวัตถุจะมีตำแหน่งการจับที่แตกต่างกันออกไป ซึ่งอาจจับในตำแหน่งตรงกลางวัตถุ ด้านล่างวัตถุและด้านบนของวัตถุ ส่วนการวางวัตถุอาจมีการวางวัตถุที่สูงเกินไป หรือการเคลื่อนที่เข้าไปกระแทกระหว่างวัตถุกับเป้าหมายแล้วจึงวางวัตถุ ดังนั้นวิธี MBRL(GPR) จึงมีความเหมาะสมในการนำไปใช้จับและวางวัตถุ ร่วมกับการตรวจจับตำแหน่งมือจับ วัตถุและเป้าหมายด้วยกล้อง 3 มิติ

ข้อเสนอแนะ

ควรมีกว้าง 3 มิติชนิดอื่นที่สามารถตรวจจับพิกัดความลึกในแกน z ได้น้อยกว่า 20 ซม. มาติดตั้งในบริเวณใกล้กับมือจับของแขนหุ่นยนต์ เพื่อให้แขนหุ่นยนต์สามารถหมุนได้อย่างอิสระ และมีลักษณะในการจับและวางวัตถุที่หลากหลายมากยิ่งขึ้น อีกทั้งปรับปรุงการตรวจจับระยะในพิกัดแกน z ของกล้อง 3 มิติให้มีความแม่นยำมากขึ้น เพื่อให้ในการนำไปฝึกสอนวิธี MBRL มีประสิทธิภาพการทำงานที่ดีขึ้น โดยสามารถจับและวางวัตถุในขอบเขตที่มากยิ่งขึ้น และวัตถุที่นำมาใช้ทดสอบอาจมีลักษณะรูปร่างอื่น เช่น ทรงลูกบาศก์ ทรงกลม เพื่อให้มีความหลากหลายในการนำไปประยุกต์ใช้งาน รวมถึงการนำวิธี MBRL มาประยุกต์ใช้ในการฝึกสอนกับหุ่นยนต์ที่มีลักษณะรอบกวนในการหมุนหรือมีจำนวนองศาอิสระที่มาก ซึ่งยากต่อการคำนวณหาจลนศาสตร์แบบผกผัน เช่น หุ่นยนต์ฮิวแมนนอยด์ (Humanoid robot) หุ่นยนต์สี่ขา (Quadruped robot) และ หุ่นยนต์หกขา (Hexapod robot)



107404525

บรรณานุกรม

- Baeldung. (2023). *Epsilon-Greedy Q-learning*. Retrieved April 15, 2023, from <https://www.baeldung.com/cs/epsilon-greedy-q-learning>
- Bagheripour, P., Gholami, A., Asoodeh, M., & Vaezzadeh-Asadi, M. (2015). Support vector regression based determination of shear wave velocity. *Journal of Petroleum Science and Engineering*, 125, 95-99.
- Camuñas-Mesa, L. A., Linares-Barranco, B., & Serrano-Gotarredona, T. (2019). Neuromorphic spiking neural networks and their memristor-CMOS hardware implementations. *Materials*, 12(17), 2745.
- Dalla Libera, A., Romeres, D., Jha, D. K., Yerazunis, B., & Nikovski, D. (2020). Model-based reinforcement learning for physical systems without velocity and acceleration measurements. *IEEE Robotics and Automation Letters*, 5(2), 3548-3555.
- Degrís, T., Pilarski, P. M., & Sutton, R. S. (2012). Model-free reinforcement learning with continuous action in practice. *2012 American Control Conference (ACC)* (pp. 2177-2182). IEEE.
- Deisenroth, M. P., Rasmussen, C. E., & Fox, D. (2011). Learning to control a low-cost manipulator using data-efficient reinforcement learning. *Robotics: Science and Systems VII*, 7, 57-64.
- Funami, Y., Kawakura, S., & Tadano, K. (2020). Development of a Robotic Arm for Automated Harvesting of Asparagus. *European Journal of Agriculture and Food Sciences*, 2(1).
- Gireesh, N. (2020). *Model-Based Offline Reinforcement Learning (MOREL)*. Retrieved January 12, 2022, from <https://medium.com/analytics-vidhya/model-based-offline-reinforcement-learning-morel-f5cd991d9fd5>
- Gomes, N. M., Martins, F. N., Lima, J., & Wörtche, H. (2021). Deep Reinforcement Learning Applied to a Robotic Pick-and-Place Application. *International Conference on Optimization, Learning Algorithms and Applications* (pp. 251-265). Springer.
- Grossi, E., & Buscema, M. (2007). Introduction to artificial neural networks. *European journal of gastroenterology & hepatology*, 19(12), 1046-1054.
- Guo, Z., Huang, J., Ren, W., & Wang, C. (2019). A reinforcement learning approach for inverse kinematics of arm robot. *Proceedings of the 2019 4th International Conference on*

Robotics, Control and Automation (pp. 95-99).

- Gupta, S., Singal, G., & Garg, D. (2021). Deep reinforcement learning techniques in diversified domains: a survey. *Archives of Computational Methods in Engineering*, 28(7), 4715-4754.
- Hansen, N. (2016). The CMA evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*.
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.
- Jaiswal, P. (2017). *Getting Started with Reinforcement Q Learning*. Retrieved July 13, 2022, from <https://towardsdatascience.com/getting-started-with-reinforcement-q-learning-77499b1766b6>
- Jazar, R. N. (2022). Introduction. In *Theory of Applied Robotics: Kinematics, Dynamics, and Control* (pp. 1-34). Springer International Publishing. https://doi.org/10.1007/978-3-030-93220-6_1
- Jiang, R., Wang, Z., He, B., Zhou, Y., Li, G., & Zhu, Z. (2021). A data-efficient goal-directed deep reinforcement learning method for robot visuomotor skill. *Neurocomputing*, 462, 389-401.
- Joshi, S., Kumra, S., & Sahin, F. (2020). Robotic grasping using deep reinforcement learning. *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)* (pp. 1461-1466). IEEE.
- Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4, 237-285.
- Khasale, A. D., Kolhe, A. S., Korde, A. A., & Bathe, A. (2019). Introduction to Pneumatic Robotic Arm. *International Journal of Research in Engineering, Science and Management*, 2(1), 688-691.
- Khlopas, A., Sodhi, N., Sultan, A. A., Chughtai, M., Molloy, R. M., & Mont, M. A. (2018). Robotic arm-assisted total knee arthroplasty. *The Journal of arthroplasty*, 33(7), 2002-2006.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.
- Kucuk, S., & Bingul, Z. (2006). *Robot kinematics: Forward and inverse kinematics*. INTECH Open Access Publisher London, UK.
- Kumar, A. (2021). *Real-World Applications of Convolutional Neural Networks*. Retrieved April 22,

- 2022, from <https://vitalflux.com/real-world-applications-of-convolutional-neural-networks/>
- Kumar, S., Rani, K., & Banga, V. (2017). Robotic arm movement optimization using soft computing. *International Journal of Robotics and Automation (IJRA)*, 16(1), 1-14.
- Laubscher, R., & Engelbrecht, Q. (2019). Application of machine learning algorithms in boiler plant root cause analysis: a case study on an industrial scale biomass unit co-firing sugar cane bagasse and furfural residue at excessive final steam temperatures. *International Sugar Journal*, 121(1445), 362-369.
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- Li, K., & Wang, K. (2021). Robot arm simulation based on model-free reinforcement learning. *2021 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)* (pp. 1104-1107). IEEE.
- Li, X., Shang, W., & Cong, S. (2020). Model-Based Reinforcement Learning for Robot Control. *2020 5th International Conference on Advanced Robotics and Mechatronics (ICARM)* (pp. 300-305). IEEE.
- Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.
- Lins, I., Moura, M., Silva, M., Drogue, E., Veleda, D., Araujo, M., & Jacinto, C. (2010). Sea surface temperature prediction via support vector machines combined with particle swarm optimization. *Proceedings of the 10th International Probabilistic Safety Assessment & Management Conference*.
- Luigi, F. (2020). *What is like inside a Convolutional Neural Network?* Retrieved June 21, 2022, from <https://medium.com/@luigi.fiori.lf0303/what-is-like-inside-a-convolutional-neural-network-a5de21e27883>
- Mateo, G. C. (2020). *Q-Learning*. Retrieved June 21, 2022, from <https://www.eecs.tufts.edu/~mguama01/post/q-learning/>
- Mistakes., J. M. (2022). *R+ Motion 1.0*. Retrieved July 13, 2022, from <https://emanual.robotis.com/docs/en/software/rplus1/motion/>
- Moerland, T. M., Broekens, J., & Jonker, C. M. (2020). Model-based reinforcement learning: A

- survey. *arXiv preprint arXiv:2006.16712*.
- Neha, K., & Sridhar, V. (2021). Video classification using transfer learning techniques for human-machine interface. *Applications of Machine Learning 2021* (pp. 197-208). SPIE.
- Nwadiugwu, M. C. (2020). Neural Networks, Artificial Intelligence and the Computational Brain. *arXiv preprint arXiv:2101.08635*.
- Otterlo, M. v., & Wiering, M. (2012). Reinforcement learning and markov decision processes. In *Reinforcement learning* (pp. 3-42). Springer.
- Padilla, R., Netto, S. L., & Da Silva, E. A. (2020). A survey on performance metrics for object-detection algorithms. *2020 international conference on systems, signals and image processing (IWSSIP)* (pp. 237-242). IEEE.
- Peng, B., Li, X., Gao, J., Liu, J., Wong, K.-F., & Su, S.-Y. (2018). Deep dyna-q: Integrating planning for task-completion dialogue policy learning. *arXiv preprint arXiv:1801.06176*.
- Phan, T. H., & Yamamoto, K. (2020). Resolving class imbalance in object detection with weighted cross entropy losses. *arXiv preprint arXiv:2006.01413*.
- Prakhar, G. (2019). *Object Detection : Simplified*. Retrieved June 21, 2022, from <https://towardsdatascience.com/object-detection-simplified-e07aa3830954>
- Rehiara, A. B. (2011). Kinematics of adept three robot arm. *Robot Arms*, 2, 21-38.
- Rivera, G. P., Eichmann, C., Scherzinger, S., Puck, L., Roennau, A., & Dillmann, R. (2019). Flexible, Personal Service Robot for ALS Patients. *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)* (pp. 1595-1600). IEEE.
- Rokbani, N., Neji, B., Slim, M., Mirjalili, S., & Ghandour, R. (2022). A Multi-Objective Modified PSO for Inverse Kinematics of a 5-DOF Robotic Arm. *Applied Sciences*, 12(14), 7091.
- Roman, V. (2020). *Convolutional Neural Networks*. Retrieved April 18, 2022, from <https://towardsdatascience.com/convolutional-neural-networks-357b9b2d75bd>
- Rootstrap. (2022). *How AI is helping to improve Robotic Arms*. Retrieved April 15, 2023, from <https://www.rootstrap.com/blog/how-ai-is-helping-to-improve-robotic-arms>
- Sagi, S. (2020). *Finite Markov Decision Processes*. Retrieved June 21, 2022, from <https://towardsdatascience.com/introduction-to-reinforcement-learning-rl-part-3-finite-markov-decision-processes-51e1f8d3ddb7>
- Sekkat, H., Tigani, S., Saadane, R., & Chehri, A. (2021). Vision-Based Robotic Arm Control

- Algorithm Using Deep Reinforcement Learning for Autonomous Objects Grasping. *Applied Sciences*, 11(17), 7917.
- Shweta, B. (2018). *5 Things You Need to Know about Reinforcement Learning*. Retrieved June 21, 2022, from <https://www.kdnuggets.com/2018/03/5-things-reinforcement-learning.html>
- Sonja, S., & Derek, B. (2013). *Optimization Test Problems*. Retrieved June 21, 2022, from <https://www.sfu.ca/~ssurjano/spheref.html>
- Sumit, S. (2018). *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*. Retrieved June 21, 2022, from <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).
- Tan, U., Rabaste, O., Adnet, C., & Ovarlez, J.-P. (2019). On the eclipsing phenomenon with phase codes. *2019 International Radar Conference (RADAR)* (pp. 1-5). IEEE.
- Thomas, S. (2018). *Diving deeper into Reinforcement Learning with Q-Learning*. Retrieved June 21, 2022, from <https://www.freecodecamp.org/news/diving-deeper-into-reinforcement-learning-with-q-learning-c18d0db58efe/>
- Van Rossum, G., & Drake, F. L. (2009). Python 3 Reference Manual. *Scotts Valley, CA: CreateSpace*.
- Vapnik, V. (1999). *The nature of statistical learning theory*. Springer science & business media.
- Wai, Y. J., Yussof, Z. M., & Salim, S. I. M. (2019). A scalable FPGA based accelerator for Tiny-YOLO-v2 using OpenCL. *Int. J. Embed. Syst*, 8, 206.
- Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine learning*, 8(3), 279-292.
- Wilimitis, D. (2018). *The Kernel Trick in Support Vector Classification*. Retrieved January 12, 2023, from <https://towardsdatascience.com/the-kernel-trick-c98cdbcaeb3f>
- Williams, C., & Rasmussen, C. (1995). Gaussian processes for regression. *Advances in neural information processing systems*, 8.
- Yang, X.-S. (2011). Optimization algorithms. In *Computational optimization, methods and algorithms* (pp. 13-31). Springer.
- Zhang, Z., & Zheng, C. (2022). Simulation of Robotic Arm Grasping Control Based on Proximal

- Policy Optimization Algorithm. *Journal of Physics: Conference Series*, 2203(1), 012065.
- กอบเกียรติ สระอุบล. (2565). *เรียนรู้ AI : Deep Learning ด้วย Python* (พิมพ์ครั้งที่ 1). สำนักพิมพ์
อินเตอร์มีเดีย.
- ธนวุฒิ ประกอบผล. (2552). โครงข่ายประสาทเทียม. *วารสารมจร.วิชาการ*, 12(24), 73-87.
- พงศ์แสน พิทักษ์วัชร. (2557). *พื้นฐานของหุ่นยนต์ : กลศาสตร์ของหุ่นยนต์แบบอนุกรม* (พิมพ์ครั้งที่
1). สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย.



107404525

BUU-IThesis 64910146 thesis / recv: 20062566 19:32:29 / seq: 122

ภาคผนวก

ภาคผนวก ก

ตารางผนวกที่ ก.1 ข้อมูลจำเพาะของกล้อง 3 มิติ OAK-D

รายละเอียด	ข้อมูลจำเพาะ	
Color camera	Image sensor	IMX378
	Max framerate	60 fps
	H. 265 Framerate	30 fps
	Resolution	12 MP (4056 x 3040 px)
	Field of view	81 DFOV ^o -68.8 HFOV ^o
	Lens size	1/2.3 inch
	Autofocus	8 cm-∞
	F-number	2.0
Stereo cameras	Image sensor	OV9282
	Max framerate	120 fps
	Pixel size	3 um x 3um
	Resolution	1280 x 800 pixels
	Field of view	81 DFOV ^o - 71.8 HFOV ^o
	Lens size	1/2.3 inch
	Focus (fixed)	19.6 cm-∞
	F-number	2.2
Myriad X Visual processing unit	Compute capacity	4 Trillion Ops/sec
	Vector processors	16 SHAVEs
	Vision accelerators	20+
	Memory bandwidth	450 GB/sec
	Also features	2x Neural compute engine (1.4 TOPS)



107404525

BUU_1Thesis_64910146_thesis / recv: 20062566 19:32:29 / seq: 122

ตารางผนวกที่ ก.2 ข้อมูลจำเพาะของ Braccio Robotic Arm

รายละเอียด	ข้อมูลจำเพาะ
Maximum operating distance	80 cm
Maximum Height	52 cm
Base Width	14 cm
Gripper Width	9 cm
Cable Length	40 cm
Load Capacity	Maximum weight at 32 cm operating distance
	Maximum weight at the minimal Braccio configuration
	150 g
	400 g
Weight	0.792 kg



107404525

BUU_1Thesis_64910146_thesis / recv: 20062566_19:32:29 / seq: 122

รางวัลหรือทุนการศึกษา

(2023) Mueangprasert, M., Chermprayong, P. and Boonlong, K., Robot Arm Movement Control by Model-based Reinforcement Learning using Machine Learning Regression Techniques and Particle Swarm Optimization, In 2023 Third International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP), pp. 83-86, IEEE.

(2566) เมธา เมืองประเสริฐ, พิศักดิ์ เจริมประยงค์ และกิตติพงษ์ บุญโล่ง, การเรียนรู้แบบเสริมกำลัง โดยใช้โมเดลสำหรับควบคุมการเคลื่อนที่ในแนวระนาบของแขนหุ่นยนต์, วิศวกรรมสารเกษมบัณฑิต, 13(1), 12-34.

(2565) ได้รับทุนการศึกษาจากคณะวิศวกรรมศาสตร์ ตามโครงการส่งเสริมและยกระดับคุณภาพบัณฑิตศึกษาเพื่อพัฒนาผู้ช่วยนักวิจัยที่ยั่งยืน คณะวิศวกรรมศาสตร์ มหาวิทยาลัยบูรพา ปีการศึกษา ๒๕๖๕

(2564) เกียรตินิยมอันดับ 1 เหรียญทอง คณะวิศวกรรมศาสตร์ สาขาวิศวกรรมเครื่องกล, มหาวิทยาลัยบูรพา

(2563) เหรียญรางวัลเรียนดีประจำปี พ.ศ.2563 กองทุนเพื่อการศึกษาและวิจัยทางด้านวิศวกรรมศาสตร์ ในพระราชูปถัมภ์ สมเด็จพระบรมโอรสาธิราชฯ สยามมกุฎราชกุมาร วิศวกรรมสถานแห่งประเทศไทยในพระราชูปถัมภ์