

# การโปรแกรมหุ่นยนต์อุตสาหกรรม (Industrial Robot Programming)

ตอนที่ 1



ชำนาญ ปัญญาใส และ  
รพีพงศ์ โชครุ่งอิสรานุกูล

ทีมระบบไซเบอร์-กายภาพ (CPS)  
กลุ่มวิจัยไอโอทีและระบบอัตโนมัติสำหรับงานอุตสาหกรรม (IIARG)  
ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC)

# การโปรแกรมหุ่นยนต์อุตสาหกรรม

## (Industrial Robot Programming) ตอนที่ 1

โดย ชำนาญ ปัญญาใส และ สพิพงค์ ไชครุ้งอิสราณกุล

ทีมระบบไซเบอร์-กายภาพ (CPS), แคนเทค สวทช.

### บทนำ



ในช่วงอุตสาหกรรมยุคที่ 3 หุ่นยนต์อุตสาหกรรมถูกนำมาใช้ในโรงงานอย่างแพร่หลาย เช่น ในอุตสาหกรรมผลิตรถยนต์ และอุตสาหกรรมผลิตเครื่องดื่ม ปัจจุบันหุ่นยนต์เข้ามามีบทบาทอย่างสูงในโรงงานสมัยใหม่ที่ต้องการการผลิตอย่างต่อเนื่อง การควบคุมคุณภาพที่เข้มงวด และความประณีตสูงในช่วงวิกฤติโรคระบาด ความต้องการใช้หุ่นยนต์ยิ่งสูงขึ้นอย่างมาก จากปัจจัยความเสี่ยงของการหยุดสายการผลิตเนื่องจากคนงานเจ็บป่วย หุ่นยนต์และระบบอัตโนมัติจึงเป็นสิ่งสำคัญอย่างยิ่งในการลดความเสี่ยงของการหยุดงานสายการผลิต รวมถึงลดผลกระทบกับห่วงโซ่อุปทาน เมื่อความต้องการใช้หุ่นยนต์ในแต่ละสาขาอุตสาหกรรมมีมากขึ้น ความต้องการวิศวกรทางด้านหุ่นยนต์ย่อมมีความต้องการเป็นเท่าตัว การพัฒนาองค์ความรู้และทักษะในการเขียนโปรแกรมควบคุม จึงมีความสำคัญสำหรับวิศวกรหรือช่างเทคนิคควบคุมการผลิตด้วยหุ่นยนต์

ในบทความนี้จะได้อธิบายถึงหุ่นยนต์ และวิธีการเขียนโปรแกรมควบคุมในเบื้องต้น เพื่อเป็นแนวทางให้ผู้สนใจได้ศึกษาค้นคว้าต่อในเชิงลึกสำหรับการพัฒนาทักษะและสมรรถนะทางด้านหุ่นยนต์สืบต่อไป

## ประเภทหุ่นยนต์อุตสาหกรรม (Industrial Robot)

เราสามารถแบ่งประเภทของหุ่นยนต์ตามการใช้งานได้ 6 ประเภท ดังนี้

**หุ่นยนต์แบบ Cartesian** เป็นหุ่นยนต์ที่แกนทั้ง 3 ของหุ่นยนต์จะเคลื่อนที่เป็นแนวเส้นตรง หุ่นยนต์มีความแข็งแรงและมีประสิทธิภาพ ในการรับน้ำหนักได้มาก หุ่นยนต์ชนิดนี้นิยมใช้ในการหยิบจับเพื่อทำการประกอบ เช่น การประกอบชิ้นส่วนอิเล็กทรอนิกส์ หรือติดตั้งส่วนยานยนต์ เนื่องจากโครงสร้างมีความแข็งแรงตลอดแนวการเคลื่อนที่ ดังนั้น จึงเหมาะกับงานเคลื่อนย้ายชิ้นงานหนัก ๆ หรือ เรียกว่า Pick-and-Place เช่น ใช้โหลดชิ้นงานเข้าเครื่องจักร (Machine loading) ใช้จัดเก็บชิ้นงาน (Stacking) นอกจากนี้ยังสามารถใช้ในงานประกอบ (Assembly)



ที่มา: yamaha-motor<sup>1</sup>

**หุ่นยนต์ทรงกระบอก (Cylindrical Robot)** เป็นหุ่นยนต์ที่มีลักษณะ การเคลื่อนที่คล้าย Cartesian แต่มีความแตกต่างกันอยู่ที่ตัวฐานของหุ่นจะเป็นการหมุนรอบแกน แทนการเคลื่อนที่แบบเคลื่อนเข้าเลื่อนออก โดยทั่วไปใช้ในงานการหยิบ-วางชิ้นงาน (Pick-and-Place) ขนย้ายสิ่งของ งานเชื่อม รวมถึงงานประกอบต่างๆ เน้นการทำงานที่รวดเร็ว เพราะสามารถเคลื่อนที่เข้าออกบริเวณที่เป็นช่องโพรงเล็กๆ ได้สะดวก หุ่นยนต์ประเภทนี้จึงนิยมใช้ในห้องปฏิบัติการ

<sup>1</sup> <https://www.yamaha-motor-im.com/robotics/clean-type/cartesian-robots-xy-xc>



ที่มา: Thermo Scientific<sup>2</sup>

**SCARA** หรือ Selective Compliance Assembly Robot Arm เป็นหุ่นยนต์ที่มีการเคลื่อนที่แบบหมุน 2 จุด โดยจะสามารถเคลื่อนที่ได้รวดเร็วในแนวระนาบ และมีความแม่นยำสูงจึงเหมาะกับงานประกอบชิ้นส่วนทางอิเล็กทรอนิกส์ที่ไม่ต้องการ ความรวดเร็วในการหมุนมากนัก แต่จะไม่เหมาะกับงานประกอบชิ้นส่วนทางกล (Mechanical Part) งานตรวจสอบ (Inspection) และงานบรรจุภัณฑ์ (Packaging)

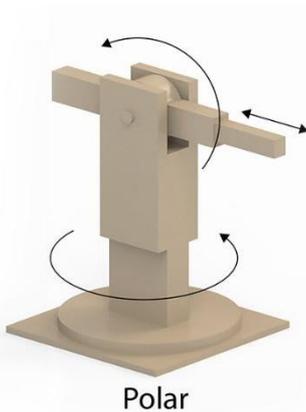


ที่มา: Delta Electronics<sup>3</sup>

**Polar** หรือที่เรียกว่า Spherical Robot คือหุ่นยนต์ที่มีแนวการเคลื่อนที่เป็นการหมุนสองแกน คือที่ฐานและที่ไหล่ของหุ่นยนต์ และส่วนมือจับสามารถยึดหดในแนวเส้นตรงได้ หุ่นยนต์ประเภทนี้สามารถทำงานในการหยิบจับชิ้นงาน รวมถึงงานเชื่อมรูปแบบต่าง ๆ ได้ดี เหมาะกับการใช้ในงานที่มีการเคลื่อนที่ในแนวตั้ง (Vertical) เพียงเล็กน้อย เช่น การโหลดชิ้นงานเข้าออกจากเครื่องปั๊ม (Press) หรืออาจจะใช้งานเชื่อมจุด (Spot Welding)

<sup>2</sup> thermofisher.com/OrbitorRS2

<sup>3</sup> [https://deltathailand.com/imgadmins/products/model\\_photo/20190419192004.jpg](https://deltathailand.com/imgadmins/products/model_photo/20190419192004.jpg)



ที่มา: howtorobot.com<sup>4</sup>



ที่มา: ieee.org<sup>5</sup>

หุ่นยนต์ Delta หรือ หุ่นยนต์ Parallel มีรูปทรงคล้ายแมงมุม มีจุดเด่น ในเรื่อง ความเร็ว ความแม่นยำ และความแข็งแรงทนทาน มีส่วนประกอบระบบเซนเซอร์ และระบบการมองเห็นที่ติดไว้บนแขนหุ่นยนต์ ทำให้การจับและการวางสิ่งของที่กำลังเคลื่อนที่มีความแม่นยำสูง ดังนั้นหุ่นยนต์ประเภทนี้จึงเหมาะกับการหยิบจับวางชิ้นงาน ขนาดเล็กที่เคลื่อนด้วยความเร็วสูง การหยิบจับวางบนสายพานการผลิต การบรรจุภัณฑ์ชิ้นงาน เช่น ชิ้นส่วนอิเล็กทรอนิกส์ ขนม หรือ อาหาร

<sup>4</sup> <https://www.howtorobot.com/expert-insight/industrial-robot-types-and-their-different-uses>

<sup>5</sup> <https://robots.ieee.org/robots/unimate>



ที่มา: Fanuc<sup>6</sup>

**Articulated (Jointed Arm)** ทุกแกนการเคลื่อนที่จะเป็นแบบหมุน (Revolute) รูปแบบการเคลื่อนที่จะคล้ายกับแขนคน ซึ่งจะประกอบด้วยช่วงเอว ท่อนแขนบน ท่อนแขนล่าง ข้อมือ การเคลื่อนที่ทำให้ได้พื้นที่การทำงาน หุ่นยนต์ชนิดนี้สามารถใช้งานได้กว้างขวาง เพราะสามารถเข้าถึง ตำแหน่งต่างๆ ได้ดี เช่น งานเชื่อม Spot Welding, Path Welding, งานยกของ, งานตัด, งานทากาว, งานที่มีการเคลื่อนที่ยากๆ เช่น งานพันสี งาน Sealing เป็นต้น

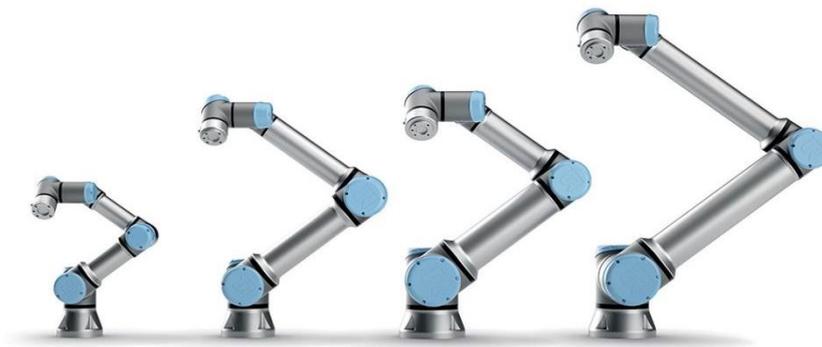


ที่มา: OMRON<sup>7</sup>

หุ่นยนต์อุตสาหกรรมโดยทั่วไปจะมีพื้นที่ในการทำงานจำเพาะ มีการกำหนดพื้นที่ความปลอดภัยมนุษย์ไม่อาจเข้าทำงานใกล้ได้ ในปัจจุบันได้มีการพัฒนาหุ่นยนต์ที่ทำงานร่วมกับมนุษย์เรียกว่า Collaborative Robot หรือ Cobot หรือ โคบอทเป็นหุ่นยนต์ที่สามารถทำงานร่วมกับมนุษย์ได้โดยตรงและปลอดภัยในพื้นที่ที่ทำงานร่วมกัน ได้ถูกนำมาใช้งานในอุตสาหกรรมอย่างแพร่หลายมากขึ้น หุ่นยนต์โคบอทมีหลายประเภท และหลายยี่ห้อในท้องตลาด โดยทั่วไปโคบอทจะถูกใช้สำหรับงานการหยิบและวาง การจัดเรียงพาเลท การตรวจสอบคุณภาพ และการจัดเรียงเครื่องมือ

<sup>6</sup> <https://www.fanuc.eu/il/en/robots/robot-filter-page/delta-robots/dr-3ib-series>

<sup>7</sup> [https://assets.omron.com/m/1f2e67c869262df6/Landscape\\_L-Viper-s850-jpg.jpg](https://assets.omron.com/m/1f2e67c869262df6/Landscape_L-Viper-s850-jpg.jpg)

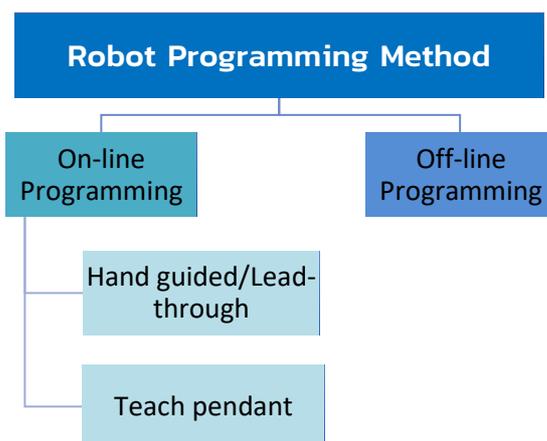


ที่มา: ieee.org<sup>8</sup>

## วิธีการโปรแกรมหุ่นยนต์ (Robot Programming Methods)

### วิธีการโปรแกรมหุ่นยนต์

การเขียนโปรแกรมมีความสำคัญต่อความสำเร็จในการทำงานของระบบหุ่นยนต์ เนื่องจากมีชุดคำสั่งที่จำเป็นสำหรับการประมวลผลหุ่นยนต์ ในการโต้ตอบกับสภาพแวดล้อมเพื่อทำงานเฉพาะ ยกตัวอย่างก่อนที่หุ่นยนต์จะเชื่อมชิ้นงานเข้าด้วยกัน ต้องตั้งโปรแกรมด้วยคำสั่งเฉพาะเพื่อให้สามารถใช้งานได้อย่างถูกต้องแม่นยำ วิธีการเขียนโปรแกรมมีหลายวิธีสำหรับการเขียนโปรแกรมหุ่นยนต์อุตสาหกรรม แต่สามารถแบ่งได้ 2 วิธีหลักคือ วิธีการเขียนโปรแกรมแบบออนไลน์ และการเขียนโปรแกรมแบบออฟไลน์ สำหรับการเขียนโปรแกรมออนไลน์ จะแบ่งย่อยเป็นการเขียนโปรแกรมด้วยวิธี การสอนผ่านหน้าจอ และการสอนนำด้วยมือ



## วิธีการเขียนโปรแกรมออนไลน์ (On-line Programming)

<sup>8</sup> <https://robots.ieee.org/robots/universal/Photos/SD/universal-photo1-full.jpg>

การโปรแกรมแบบออนไลน์ เป็นวิธีการที่วิศวกรหรือโปรแกรมเมอร์ ทำการโปรแกรมควบคุมการทำงานของหุ่นยนต์ ในขณะที่หุ่นยนต์มีการเปิดระบบใช้งานตลอดเวลา โดยอาจเชื่อมต่อสื่อสารผ่านอุปกรณ์หรือแผงควบคุมแบบมือถือ หรือที่เรียกว่าแผงการสอน (teach pendant) หรือ ผ่านเครื่องคอมพิวเตอร์ ดังนั้นถ้าหุ่นยนต์อยู่ในสายการผลิต จำเป็นต้องหยุดสายการผลิตก่อน

### 1. วิธีสอนนำด้วยมือ (Hand guided/Lead-through Method)



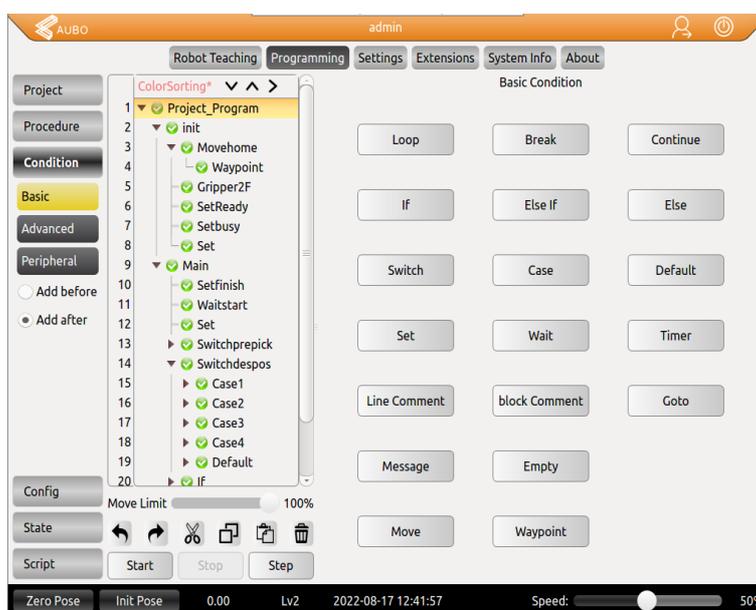
การโปรแกรมโดยวิธีการสอนด้วยมือ (Hand guided) หรือ การสอนนำ (Lead through) ซึ่งโดยทั่วไปแล้วจะเกี่ยวข้องกับการควบคุมเส้นทางการเคลื่อนที่อย่างต่อเนื่อง ผู้ปฏิบัติงานจะเคลื่อนที่แขนของหุ่นยนต์ด้วยมือในเส้นทางที่ต้องการ ขณะที่แขนหุ่นยนต์มีการเคลื่อนที่ ส่วนควบคุมจะจัดเก็บตำแหน่ง ปัจจุบันและทิศทางของจุดศูนย์กลางเครื่องมือเป็นระยะๆ หลังจากที่มีการบันทึกตำแหน่ง การเคลื่อนที่เรียบร้อยแล้วสามารถสั่งให้หุ่นยนต์เล่นย้อนกลับ (play-back) ตามที่ได้สอนนำไว้แล้ว ตัวอย่าง เช่น งานเชื่อมแบบต่อเนื่อง งานพับสี การขัดพื้นผิว เป็นต้น วิธีการโปรแกรมหุ่นยนต์ลักษณะนี้ สามารถทำได้รวดเร็วและสะดวกกว่าการสอนหุ่นยนต์ โดยวิธีการจิ้มหรือพิมพ์คำสั่งงานที่หน้าจอควบคุมแบบมือถือ ไม่จำเป็นต้องมีทักษะการเขียนโปรแกรมคอมพิวเตอร์ ก็สามารถทำการโปรแกรมหุ่นยนต์ได้ แต่ข้อเสียของวิธีการโปรแกรมด้วยวิธีการสอนด้วยมือ คือ ไม่เหมาะกับหุ่นยนต์ที่มีขนาดใหญ่ มีน้ำหนักมาก และงานที่ต้องการความแม่นยำเที่ยงตรงสูง การสอนนำด้วยการแนะนำหุ่นยนต์ด้วยมือไม่อาจกำหนดจุดอย่างแม่นยำได้ จำเป็นต้องใช้การสอนผ่านแผงควบคุมแบบมือถือ มาช่วยกำหนดค่าตำแหน่งที่แม่นยำ

### 2. Teach Pendant Programming การโปรแกรมแบบการสอนหน้าจอ



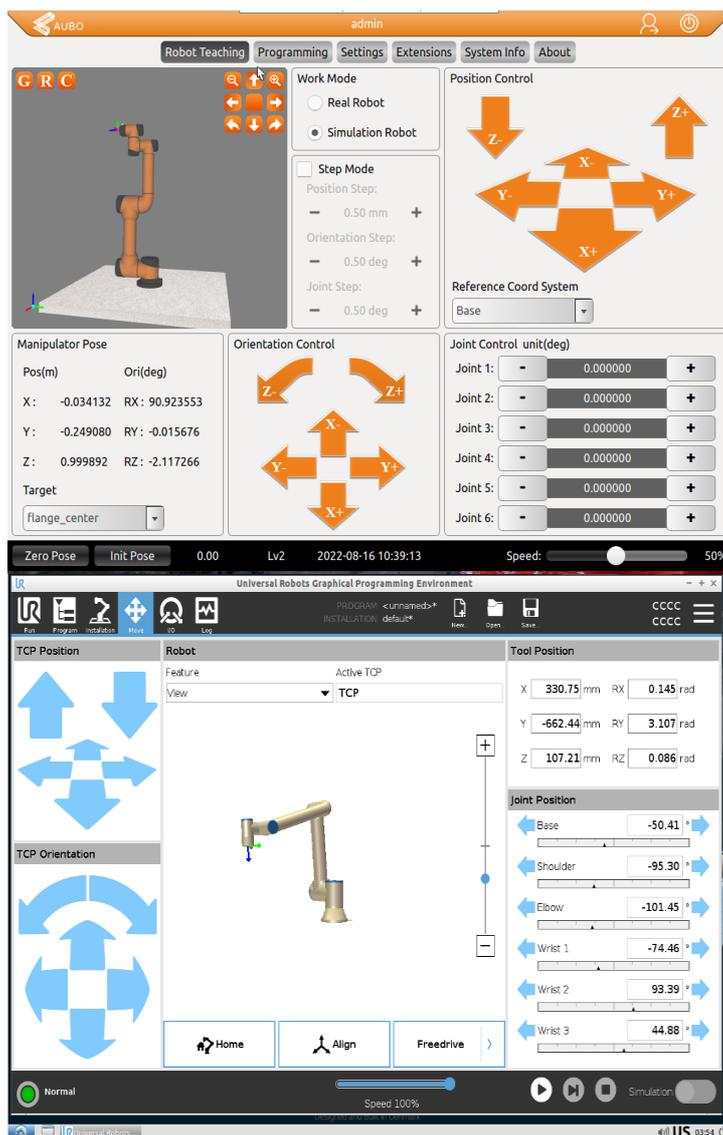
### การเขียนโปรแกรมผ่านหน้าจอบูการณ์ควบคุมแบบมือถือ (Teach pendant)

โปรแกรมเมอร์จะเพิ่มคำสั่งที่ละบรรทัดและกำหนดเงื่อนไขแต่ละคำสั่งเป็นลำดับ ชุดคำสั่งโดยทั่วไปแบ่งเป็น 3 กลุ่ม ได้แก่ กลุ่มคำสั่งพื้นฐาน (Basic) เช่น คำสั่ง Move, Waypoint, Set, Wait กลุ่มคำสั่งขั้นสูง (Advanced) เช่น คำสั่ง Thread, Procedure, Script และ กลุ่มคำสั่งอุปกรณ์ต่อพ่วง (Peripheral) เช่น Gripper, Smart3D



### การเขียนโปรแกรมแบบออนไลน์ผ่านหน้าจอบูการณ์ควบคุมแบบมือถือ

**ตัวอย่าง** หากต้องการให้ตำแหน่งหุ่นยนต์เคลื่อนที่ Move ไปยังตำแหน่งพิกัดตาม Waypoint โดยการจี้ที่ลูกศรที่ช่อง **Position Control** เพื่อเลื่อนแกนหุ่นยนต์ไปตามแนวแกน XYZ และปรับมุมทิศทางของหน้าแปลนจุดศูนย์กลางเครื่องมือ (Tool Center Point:TCP) โดยการจี้ที่ ลูกศรที่ช่อง **Orientation Control** ให้หมุนไปตามทิศทางที่ต้องการ หุ่นยนต์บางตัวที่แผงควบคุม แบบมือถืออาจจะขยับหรือเลื่อนตำแหน่งของแกนหุ่นยนต์โดยใช้ Joystick



การควบคุมการเคลื่อนที่ของแขนหุ่นยนต์โดยการจิ้มลูกศรปรับตำแหน่ง และมุมของจุดศูนย์กลางเครื่องมือ (TCP)

เมื่อผู้ปฏิบัติงานทำการเคลื่อนแขนหุ่นยนต์ให้อยู่ในตำแหน่งพิกัดที่ต้องการ โดยใช้แผงควบคุมแบบมือถือแล้ว ผู้ปฏิบัติงานสามารถบันทึกพิกัดภายในรายการตำแหน่งได้ โดยการกดปุ่มบันทึก หรือยืนยัน เนื่องจากเหตุผลด้านความปลอดภัย หุ่นยนต์สามารถขับเคลื่อนในโหมดการทำงานนี้ได้เฉพาะในแบบช้าๆ เท่านั้น เนื่องจากผู้ปฏิบัติงานทำงานภายในพื้นที่ทำงานของหุ่นยนต์ แผงควบคุมจึงจะต้องมีการติดตั้งสวิตช์ฉุกเฉิน ตัวอย่างลักษณะของงาน สำหรับวิธีการโปรแกรมลักษณะนี้ จะเป็นการทำงานแบบทั่วไปที่ไม่ซับซ้อน เช่น งานการเชื่อมแบบจุด งานหยิบและวาง

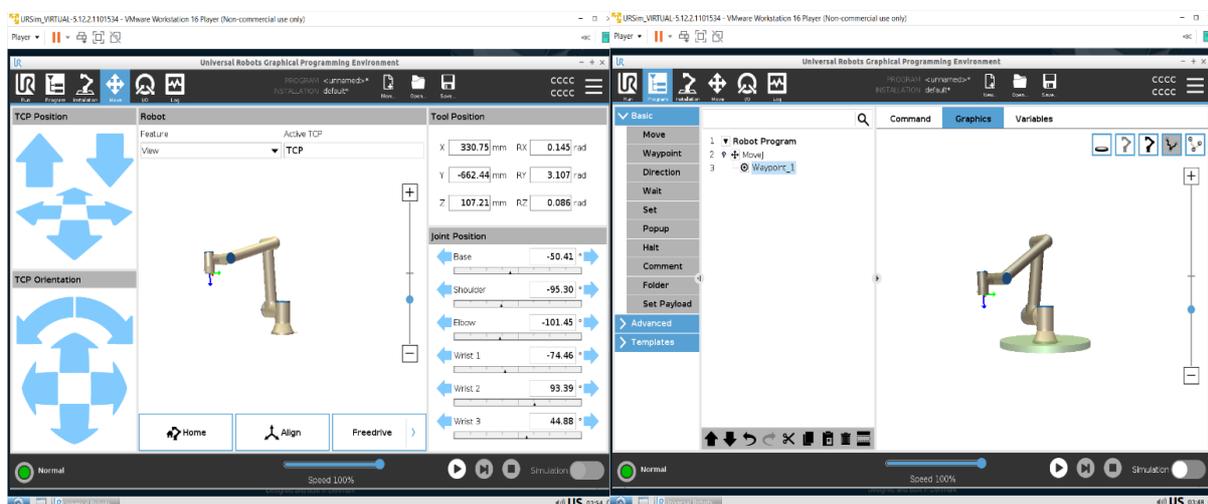
## การโปรแกรมแบบออฟไลน์ Off-line Programming (OLP)

การโปรแกรมแบบออฟไลน์ ผู้เขียนโปรแกรมไม่จำเป็นต้องปฏิบัติงานร่วมกับหุ่นยนต์ทำงาน จึงไม่จำเป็นต้องหยุดสายการผลิต จะหยุดการผลิตเมื่อมีการโปรแกรมงานลงเครื่องควบคุมหุ่นยนต์เท่านั้น ทั้งนี้

ผู้เขียนโปรแกรมจะใช้ซอฟต์แวร์สำหรับการเขียนโปรแกรม และจำลองการทำงานของหุ่นยนต์ (robot simulator) ที่ติดตั้งบนเครื่องคอมพิวเตอร์แทน โดยการเขียนโปรแกรมทดสอบการทำงานของหุ่นยนต์เสมือนว่ากำลังทำงานกับหุ่นยนต์จริง ซึ่งทำให้ผู้เขียนสามารถทดสอบ หรือทดลองเงื่อนไขต่างๆ ได้หลายรูปแบบ และเลือกเงื่อนไขที่เหมาะสมสำหรับการโปรแกรม หุ่นยนต์ เป็นการลดความเสี่ยง และความผิดพลาดที่จะเกิดกับหุ่นยนต์ก่อนไปทดสอบใช้งานจริง

รูปแบบการเขียนโปรแกรมแบบออฟไลน์ สามารถเขียนโค้ดคำสั่งเป็นตัวหนังสือ (text based) หรือแบบกราฟิก (graphical programming) ขึ้นอยู่กับความสามารถของซอฟต์แวร์โปรแกรม ที่ผู้ผลิตโปรแกรมได้สร้างขึ้นมา ซอฟต์แวร์สำหรับการโปรแกรม และจำลองการทำงานของหุ่นยนต์ โดยทั่วไปมักจะผลิตโดยผู้ผลิตหุ่นยนต์เอง เช่น ROBOGUIDE ของบริษัท Fanuc, KUKA.Sim/KUKA.Apptech ของบริษัท KUKA, URSim5 ของบริษัท Universal Robots, RobotStudio ของบริษัท ABB หรือใช้โปรแกรมออฟไลน์ที่ผลิตโดยผู้ผลิตเจ้าอื่น (third-party vendor) เช่น RoboDK, RobotMaster, OCTOPUZ, Deloi เป็นต้น

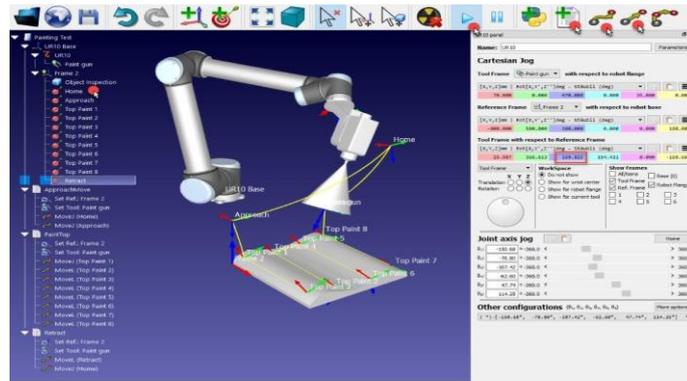
รูปแบบการเขียนโปรแกรมแบบออฟไลน์เหมาะสมที่สุดสำหรับกระบวนการผลิตที่มีแอปพลิเคชันการวางแผนเส้นทางเคลื่อนที่ที่ซับซ้อน โดยส่วนมากจะมามีการทำงานของหุ่นยนต์หลายอย่าง เช่น การตัดด้วยเลเซอร์ การเชื่อม การพ่นสี การขัดผิว ทั้งนี้การหยิบและวางแบบง่าย เช่น การบรรจุหีบห่อ การจัดเรียงพาเลท และการประกอบอาจไม่เหมาะสมสำหรับรูปแบบวิธีการโปรแกรมแบบออฟไลน์



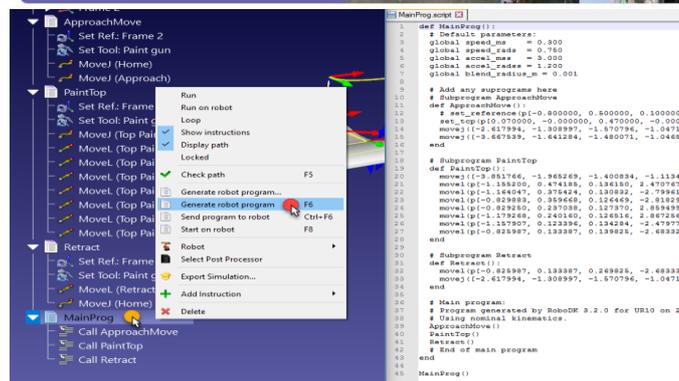
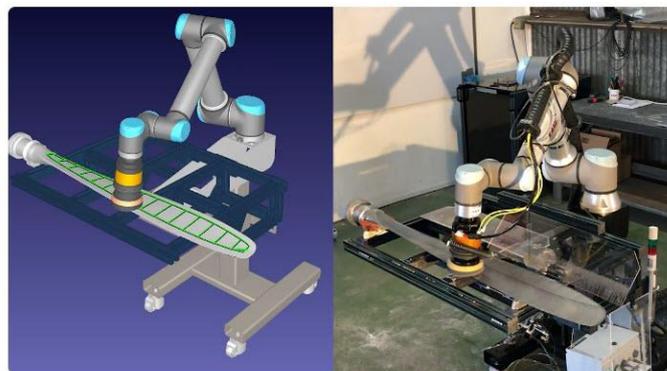
โปรแกรม URSim ของ Universal Robots จำลองหน้าตาเหมือนกับใช้งานบนอุปกรณ์ควบคุมแบบมือถือ

### ตัวอย่างโปรแกรมสำหรับการจำลองการทำงานของหุ่นยนต์

RoboDK (<https://robodk.com/>) เป็นซอฟต์แวร์การเขียนโปรแกรมและจำลองแบบออฟไลน์ สำหรับหุ่นยนต์อุตสาหกรรมซอฟต์แวร์จำลอง สามารถใช้ได้กับงานการผลิตที่หลากหลาย เช่น การกัด การเชื่อม การหยิบและวาง การบรรจุ และการตีฉลาก การจัดวางบนแท่นวางสินค้า การพ่นสี การสอบเทียบหุ่นยนต์ และอื่นๆ



ที่มา: RoboDK<sup>9</sup>

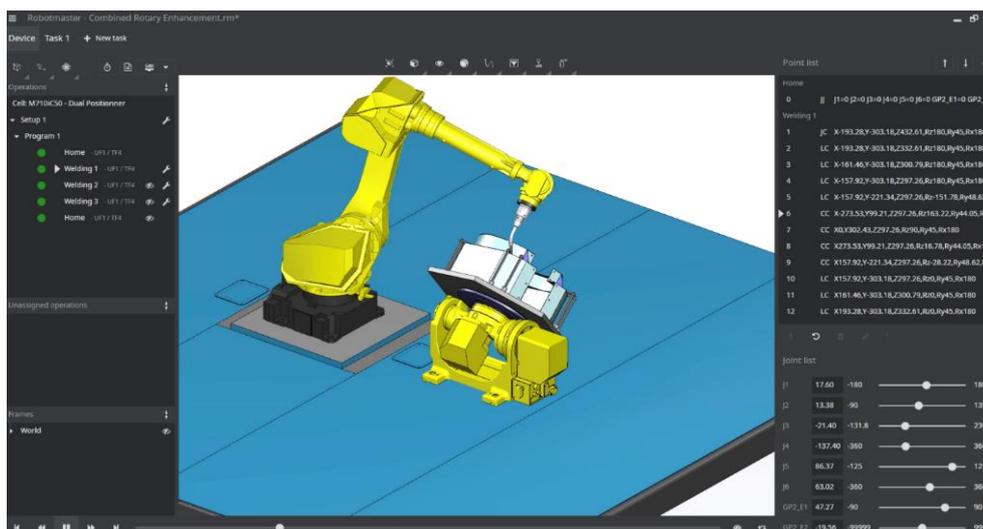


ที่มา: RoboDK<sup>10</sup>

RobotMaster (www.robotmaster.com) เป็นซอฟต์แวร์ CAD/CAM สำหรับหุ่นยนต์อุตสาหกรรมแบบ off-line โดยโปรแกรม สามารถสร้างโปรแกรมเส้นทางการเคลื่อนที่ของหุ่นยนต์ โดยไม่ต้องใช้วิธีสอน (teaching) ลดเวลาในการสร้างโปรแกรม สามารถจำลองการทำงานแบบ 3 มิติได้ และลดข้อผิดพลาดที่เกิดจากการใช้วิธีสอน (teaching)

<sup>9</sup> [https://robodk.com/doc/en/Getting-Started\\_files/img41.png](https://robodk.com/doc/en/Getting-Started_files/img41.png)

<sup>10</sup> [https://robodk.com/doc/en/Getting-Started\\_files/img46.png](https://robodk.com/doc/en/Getting-Started_files/img46.png)



ที่มา: RobotMaster<sup>11</sup>

Gazebo (<https://gazebo.org/>) เป็นโปรแกรมจำลองหุ่นยนต์โอเพนซอร์ส 3 มิติ โดยรวมเอ็นจินทางฟิสิกส์ ODE, การเรนเดอร์ OpenGL และโค้ดโปรแกรมสนับสนุนสำหรับการจำลองเซ็นเซอร์และการควบคุมแอกทูเอเตอร์

Webots (<http://www.cyberbotics.com/>) เป็นโปรแกรมจำลองหุ่นยนต์ 3 มิติ แบบโอเพนซอร์สที่ใช้ในอุตสาหกรรม การศึกษา และการวิจัย

CoppeliaSim (<https://www.coppeliarobotics.com/>) หรือที่รู้จักกันในชื่อ V-REP เป็นหุ่นยนต์จำลองที่ใช้ในอุตสาหกรรม การศึกษา และการวิจัย เดิมได้รับการพัฒนาภายใน R&D ของบริษัทโตชิบา และขณะนี้กำลังได้รับการพัฒนาและบริการโดย Coppelia Robotics AG ประเทศสวิตเซอร์แลนด์

จากข้างต้นที่กล่าวถึงการโปรแกรมหุ่นยนต์อุตสาหกรรมพื้นฐานทั่วไปที่ผลิตและจำหน่ายในท้องตลาด โดยที่ผู้เขียนโปรแกรมมีหน้าที่เพียงเขียนโปรแกรมควบคุมคำสั่งการทำงานของหุ่นยนต์ โดยไม่ได้มีส่วนในการสร้างฮาร์ดแวร์ ภาษาที่ใช้เขียนโปรแกรมสำหรับควบคุมหุ่นยนต์จะจำเพาะหุ่นยนต์แต่ละยี่ห้อ เช่น ABB เขียนด้วยภาษา RAPID, KUKA เขียนด้วยภาษา KRL, FANUC เขียนด้วยภาษา KAREL, UR เขียนด้วยภาษา URScript เป็นต้น

ในกรณีที่ผู้พัฒนาประสงค์ที่จะสร้างหุ่นยนต์ขึ้นมาเอง เป็นต้องเขียนโปรแกรมควบคุมหุ่นยนต์เฉพาะกิจขึ้นมา อาจจะเขียนโปรแกรมด้วยภาษา C/C++, Python, Java, Lua หรือใช้โปรแกรมเช่น MATLAB ในการพัฒนาส่วนควบคุมหุ่นยนต์ หรือใช้ระบบปฏิบัติการหุ่นยนต์ในการพัฒนาหุ่นยนต์ เป็นต้น

## ระบบปฏิบัติการสำหรับหุ่นยนต์

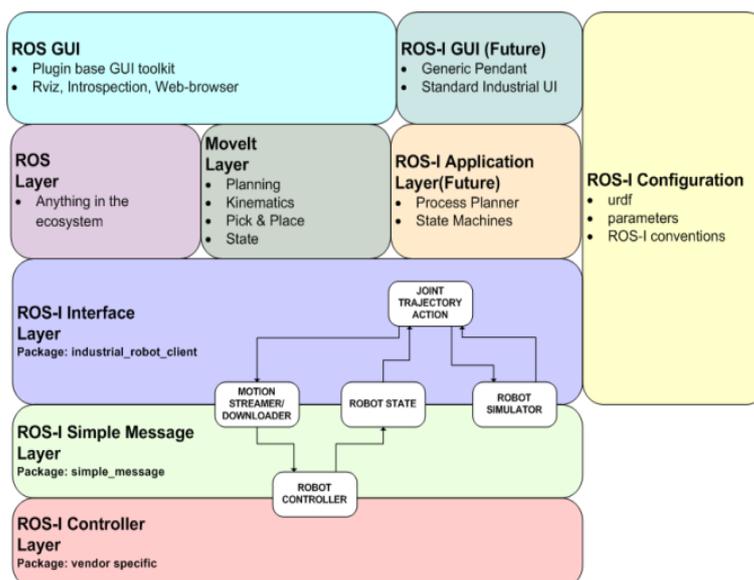
<sup>11</sup> <https://www.robotmaster.com/en/newsroom/offline-robot-programming-software>

เมื่อหุ่นยนต์ถูกใช้งานนอกเหนือจากงานในสายการผลิตอุตสาหกรรม สู่อุตสาหกรรมอื่น ทั้งหุ่นยนต์บริการ ในภาคธุรกิจ หุ่นยนต์การศึกษา การแพทย์ หุ่นยนต์สำรวจ หรือหุ่นยนต์ทางทหาร การพัฒนาแอปพลิเคชันสำหรับหุ่นยนต์ จึงให้ความสนใจในด้านการพัฒนาด้วยแพลตฟอร์มหรือระบบปฏิบัติการสำหรับหุ่นยนต์ (Robot Operating System for robots) เช่น ROS, URBI, Microsoft Robotics Developer Studio

# ROS

ROS (Robot Operating System) (<https://www.ros.org/>)

ระบบปฏิบัติการหุ่นยนต์ (ROS) เป็นแพลตฟอร์มแบบเปิด สำหรับการพัฒนาแอปพลิเคชันสำหรับหุ่นยนต์ทำงานในหลากหลายด้าน นิยมอย่างมากโดยเฉพาะการประยุกต์ใช้งานในกลุ่มหุ่นยนต์ด้านบริการ (service robots) ทั้งนี้ เรายังสามารถประยุกต์ใช้งานสำหรับหุ่นยนต์อุตสาหกรรม โดยมีกลุ่มคณะทำงาน ROS ที่เน้นการใช้งานทางด้านอุตสาหกรรม เรียกว่า ROS-Industrial (<https://rosindustrial.org/>) สำหรับภาพโครงสร้างสถาปัตยกรรม ROS-Industrial แสดงดังรูป



ROS-Industrial High Level Architecture - Rev 0.02.vsd

โครงสร้างสถาปัตยกรรม ROS-Industrial

ที่มา: <https://wiki.ros.org/Industrial>



ส่วนประกอบของ ROS ที่มา: <http://wiki.ros.org/APIs>

ตัวอย่าง ผู้ผลิตหุ่นยนต์ที่สนับสนุนการพัฒนาโปรแกรมด้วย ROS ได้แก่บริษัท Universal Robots, AUBO, Fanuc (<http://wiki.ros.org/Robots>) ตัวอย่างงานวิจัยการใช้งาน ROS ในการพัฒนาแอปพลิเคชันหุ่นยนต์ชื่อ Manipulator Control in Collaborative Assembly, <https://www.makr.org/2021/scorpio>

## ปัญญาประดิษฐ์และการเรียนรู้ของเครื่องจักร (Artificial Intelligence/Machine Learning)

ปัญญาประดิษฐ์ หรือ AI และการเรียนรู้ของเครื่องจักร (machine learning) ได้ถูกนำมาใช้งานกับหุ่นยนต์มายาวนาน ทำให้หุ่นยนต์สามารถประยุกต์ตัวเองและปรับแต่งการทำงาน หรือทำการอัปเดตพารามิเตอร์ใหม่อีกครั้ง ให้สามารถทำงานได้ถูกต้องแม่นยำตามสภาพแวดล้อม การใช้ AI ในการโปรแกรมหุ่นยนต์เพื่อให้หุ่นยนต์สามารถปรับเปลี่ยนตัวเองได้ (reconfigurable) ยังคงอยู่ในขั้นของการวิจัยพัฒนา

## การโปรแกรมด้วย VR

AR (Augmented Reality) ได้ถูกนำมาใช้งานในการฝึกอบรมพัฒนาทักษะของบุคลากรในการปฏิบัติงานกับเครื่องจักรเสมือน โดยอยู่ด้านหน้าเครื่องจริง เป็นวิธีการฝึกฝนที่ผู้เรียนสามารถฝึกฝนได้ตลอดเวลา งานด้าน VR (Virtual Reality) และ AR ได้ถูกนำมาประยุกต์ใช้กับการสอนหุ่นยนต์และการโปรแกรมหุ่นยนต์เสมือนหนึ่งกับการทำงานร่วมกับหุ่นยนต์จริง

## บทสรุป

บทความนี้ได้กล่าวถึงวิธีการโปรแกรมหุ่นยนต์อุตสาหกรรมรูปแบบต่างๆ ทำให้ผู้ที่สนใจเรียนรู้เกี่ยวกับการโปรแกรมหุ่นยนต์เห็นภาพรวมและเข้าใจกระบวนการโปรแกรมหุ่นยนต์ สามารถต่อยอดองค์ความรู้ และทักษะของตนเองในระดับสูงที่ขึ้น อย่างไรก็ตาม การใช้งานหุ่นยนต์ที่มีความซับซ้อนมากขึ้น ซัดความสามารถของหุ่นยนต์ที่สูงขึ้น เทคนิคและวิธีการโปรแกรมหุ่นยนต์ได้มีการพัฒนาเครื่องมือใหม่ๆ เพื่ออำนวยความสะดวกให้การพัฒนาแอปพลิเคชันสำหรับหุ่นยนต์ได้อย่างรวดเร็ว

การเลือกวิธีการโปรแกรมที่ดีที่สุด เป็นสิ่งที่ตอบได้ยาก ขึ้นอยู่กับปัจจัยและสถานการณ์มาประกอบรวมกัน เช่น ลักษณะและความซับซ้อนของงาน ประสบการณ์ของการเขียนโปรแกรมของผู้ปฏิบัติงาน ระยะเวลา ผลกระทบในการหยุดสายการผลิต ฟังก์ชันต่างๆ ที่บริษัทผู้ผลิตหุ่นยนต์ให้มา หรือแม้แต่ความสะดวกของเครื่องมือ หรือซอฟต์แวร์โปรแกรมที่ผู้ผลิตจัดหาให้ เป็นสิ่งที่ผู้เขียนโปรแกรมต้องเลือกใช้ตามความสะดวก และเหมาะสมกับสถานการณ์ การติดตามข้อมูลข่าวสารผ่านสื่อต่างๆ ตลอดจนการฝึกทักษะอย่างต่อเนื่อง จึงสำคัญต่อการพัฒนาความรู้ และทักษะของกำลังคนอุตสาหกรรมยุคใหม่

## เอกสารอ้างอิง

- [1] บุญธรรม ภัทราจารุกุล, หุ่นยนต์อุตสาหกรรม (ปวส.) (รหัสวิชา 30127-2103), ซีเอ็ดยูเคชั่น, 2565
- [2] รายงานการศึกษาเรื่อง “อุตสาหกรรมหุ่นยนต์ของประเทศไทย” โดย ฝ่ายวิจัยนโยบาย สวทช.  
<https://waa.inter.nstda.or.th/prs/pub/Robot-Whitepaper-Cover.pdf>
- [3] อุตสาหกรรมหุ่นยนต์กับการพัฒนาภาคการผลิตของไทย (ตอนที่ 1),  
<https://www.krungsri.com/th/research/research-intelligence/ri-robotic-series1-landscape>
- [4] Fengxin Zhang, Chow Yin Lai, Milan Simic, Songlin Ding, Augmented reality in robot programming, Procedia Computer Science 176 (2020) 1221–1230.
- [5] <http://wiki.ros.org/Industrial>
- [6] Gopika Ajaykumar, Maureen Steele, and Chien-Ming Huang. 2021. A Survey on End-User Robot Programming. ACM Comput. Surv. 54, 8, Article 164 (November 2022), 36 pages.  
<https://doi.org/10.1145/3466819>
- [7] YoonSeok Pyo, HanCheol Cho, RyuWoon Jung, TaeHoon Lim, ROS Robot Programming, ROBOTIS Co.,Ltd., 2017
- [8] รู้จัก 5 ประเภทหุ่นยนต์พื้นฐานในอุตสาหกรรม (Industrial Robot) และการใช้งาน, สถาบันเทคโนโลยีการผลิตสุมิพล, <https://simtec.or.th/blog-5-industrial-robot-12112021>

### ติดต่อ ศูนย์นวัตกรรมการผลิตยั่งยืน (SMC)

112 อุทยานวิทยาศาสตร์ประเทศไทย

ถนนพหลโยธิน ตำบลคลองหนึ่ง

อำเภอคลองหลวง จังหวัดปทุมธานี 12120

เว็บไซต์ : <https://www.nectec.or.th/smc>

Facebook : <https://www.facebook.com/smceeci>