

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



HOÀNG ANH

**ỨNG DỤNG ROS (ROBOT OPERATING SYSTEM)
ĐIỀU KHIỂN ROBOT DỊCH VỤ BÁN CAFE**

LUẬN VĂN THẠC SỸ KỸ THUẬT
(Theo định hướng ứng dụng)

TP. HỒ CHÍ MINH – NĂM 2022

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



HOÀNG ANH

**ỨNG DỤNG ROS (ROBOT OPERATING SYSTEM)
ĐIỀU KHIỂN ROBOT DỊCH VỤ BÁN CAFE**

CHUYÊN NGÀNH: HỆ THỐNG THÔNG TIN

MÃ SỐ: 8.48.01.04

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC:

TS. CHUNG TẤN LÂM

TP. HỒ CHÍ MINH – NĂM 2022

LỜI CAM ĐOAN

Tôi cam đoan rằng luận văn: ***“ỨNG DỤNG ROS (ROBOT OPERATING SYSTEM) ĐIỀU KHIỂN ROBOT DỊCH VỤ BÁN CAFE ”*** là công trình nghiên cứu của chính tôi.

Tôi cam đoan các số liệu, kết quả nêu trong luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Không có sản phẩm/nghiên cứu nào của người khác được sử dụng trong luận văn này mà không được trích dẫn theo đúng quy định.

TP. Hồ Chí Minh, ngày 15 tháng 07 năm 2022

Học viên thực hiện luận văn

Hoàng Anh

LỜI CẢM ƠN

Trong quá trình học tập và thực hiện luận văn, tôi đã nhận được sự quan tâm quý báu và hướng dẫn nhiệt tình của quý Thầy Cô, cùng với sự động viên và ủng hộ của gia đình, bạn bè và đồng nghiệp.

Với lòng kính trọng và biết ơn, tôi xin gửi lời cảm ơn chân thành tới: Ban Giám Đốc, Phòng đào tạo Sau Đại học của Học viện Công Nghệ Bưu Chính Viễn thông cơ sở TP. Hồ Chí Minh và quý Thầy Cô đã tạo mọi điều kiện thuận lợi giúp tôi hoàn thành luận văn.

Tôi xin bày tỏ lòng biết ơn chân thành và sâu sắc nhất tới người thầy kính yêu **Thầy TS. Chung Tấn Lâm** đã hết lòng giúp đỡ, trực tiếp hướng dẫn tận tình, động viên khích lệ, tạo điều kiện cho tôi trong suốt quá trình thực hiện luận văn.

Từ đáy lòng mình tôi xin bày tỏ sự biết ơn vô hạn đến gia đình thân yêu của tôi và xin chân thành cảm ơn bạn bè thân thiết, đồng nghiệp trong cơ quan đã động viên, hỗ trợ tôi trong lúc khó khăn để tôi có thể học tập và hoàn thành luận văn.

Mặc dù đã có nhiều cố gắng, nỗ lực tìm tòi nghiên cứu, nhưng do thời gian có hạn và kinh nghiệm nghiên cứu khoa học còn hạn chế nên không thể tránh khỏi những thiếu sót. Tôi rất mong nhận được sự góp ý thiết thực của quý Thầy Cô cùng bạn bè đồng nghiệp để kiến thức của tôi ngày một hoàn thiện hơn.

Xin chân thành cảm ơn!

TP. Hồ Chí Minh, ngày 15 tháng 07 năm 2022

Học viên thực hiện luận văn

Hoàng Anh

MỤC LỤC

Lời cam đoan	i
Lời cảm ơn	ii
Mục lục	iii
Danh mục các thuật ngữ, chữ viết tắt	v
Danh sách bảng	vi
Danh sách hình vẽ	vii
MỞ ĐẦU	1
CHƯƠNG 1 : NGHIÊN CỨU TỔNG QUAN	4
1.1 Tổng quan về robot dịch vụ	4
1.2 Robot cộng tác	8
1.3 Hệ thống điều hành robot (Robot Operating System)	11
1.4 Phần mềm mô phỏng robot – RobotDK	15
1.5 Tổng kết chương	16
CHƯƠNG 2 : MÔ HÌNH HÓA ROBOT	18
2.1 Xây dựng mô hình robot	18
2.2 Xây dựng mô hình máy pha coffree.....	23
2.3 Xây dựng mô hình cấp ly cafe và bộ giao cafe	24
2.4 Tổng kết chương	26
CHƯƠNG 3 : THIẾT KẾ VÀ MÔ PHỎNG THAO TÁC PHA CHẾ	27
3.1 Thiết kế trình tự pha chế bằng robot	27
3.2 Thao tác lấy ly	28
3.3 Thao tác pha chế cafe.....	31
3.4 Thao tác đưa khách hàng	35
3.5 Giao tiếp API RoboDK dùng C#	38
3.6 Tổng kết chương	40
CHƯƠNG 4 : KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	42
4.1 Kết luận	42

4.2 Hướng phát triển	43
TÀI LIỆU THAM KHẢO	44
PHỤ LỤC	45
PHỤ LỤC P1: Tập lệnh chính của RoboDK	45
PHỤ LỤC P2: Lập trình và mô phỏng trên RoboDK	56

DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT

Viết tắt	Tiếng Anh	Tiếng Việt
Cobot	Collaborative Robot	Robot cộng tác
TCP	Tool Center Point	Tâm điểm dụng cụ
	Reference Frame	Hệ tọa độ tham chiếu
	Tool Frame	Hệ tọa độ dụng cụ
	Target Point	Điểm mục tiêu
CMCN 4.0	Industry 4.0	Cách mạng Công nghiệp lần thứ 4
IoT	Internet of Things	Internet vạn vật
	Smart Actuators	Cơ cấu chấp hành thông minh

DANH SÁCH BẢNG

Bảng 2.1: Thông số Dobot CR10.....	22
Bảng 3.1: Giải thích chương trình con thao tác lấy ly ra khỏi chồng	30
Bảng 3.2: Giải thích chương trình con thao tác pha chế cafe	32
Bảng 3.3: Giải thích chương trình con thao tác đưa cafe cho khách hàng	35

DANH SÁCH HÌNH VẼ

Hình 1.1: Robot BellaBot phục vụ nhà hàng ở một nhà hàng Q.7 TP.HCM.....	6
Hình 1.2: Ella là sản phẩm của Crown Digital	8
Hình 1.3: Sự khác biệt cơ bản về sự an toàn giữa robot công nghiệp và cobot.....	9
Hình 1.4: Cấu trúc cơ bản của khớp đơn cobot.....	10
Hình 1.5: Mẫu thiết kế cơ cấu chấp hành của khớp cobot	10
Hình 1.6: Mẫu thiết kế cơ cấu chấp hành của khớp cobot	11
Hình 1.7: Hệ thống điều khiển tập trung RC8 robot truyền thống (Denso).....	12
Hình 1.8: Hệ thống điều khiển phân tán RC9 cho cobot của hãng Denso.....	13
Hình 1.9: So sánh hệ thống điều khiển tập trung RC8 và phân tán RC9	14
Hình 1.10: Giao diện của phần mềm RoboDK	16
Hình 2.1: Các mô hình 3D Cobot CR10: nền (B0), khớp 1 (B1) , khớp 2 (B2), khớp 3 (B3), khớp 4 (B4), khớp 5 (B5), khớp 6 (B6)	19
Hình 2.2: Kéo thả robot (DOBOT CR10Model) vào Frame Robot	20
Hình 2.3: Dùng bảng điều khiển để test các khớp robot.....	21
Hình 2.4: Kích thước Dobot CR1021	21
Hình 2.5: Xác định và nhập các kích thước động học trên robot builder	22
Hình 2.6: Xác định các giới hạn góc quay của khớp trên robot builder	23
Hình 2.7: Test hoạt động của robot trên robot Panel	24
Hình 2.8: Mô hình cấp ly cafe.....	25
Hình 2.9: Mô hình bộ giao cafe.....	26
Hình 3.1: Mô hình trạm pha cafe dùng robot.....	27
Hình 3.2: Chương trình chính thực hiện thao tác pha chế cafe.....	28
Hình 3.3a: Chương trình con thực hiện thao tác lấy ly	29
Hình 3.3b: Chương trình con thực hiện tiếp cận máy pha chế cafe (vì chuyển hệ tọa độ tham chiếu).....	29
Hình 3.4a: Vị trí tiếp cận chong ly 1	29
Hình 3.4b: Vị trí tiếp cận ly 2	29

Hình 3.4c: Vị trí lấy ly ra khỏi chông	30
Hình 3.5a: Chương trình con thực hiện thao tác pha chế cafe	32
Hình 3.5b: Định nghĩa hệ trục tọa độ cho máy pha cafe (cafe Machine,CM)	32
Hình 3.6a: Tiếp cận máy pha cafe.....	33
Hình 3.6b: Vị trí rót cafe 1	33
Hình 3.6c: Vị trí rót cafe 2	34
Hình 3.6d: Vị trí trung gian.....	34
Hình 3.6e: Vị trí rót kem sữa 1	34
Hình 3.6f: Vị trí rót kem sữa 2.....	34
Hình 3.6g: Vị trí an toàn ra khỏi máy pha cafe.....	34
Hình 3.7: Chương trình con thực hiện thao tác đưa cafe cho khách hàng	35
Hình 3.8a: Vị trí tiếp cận bộ 1	36
Hình 3.8b: Vị trí đặt ly lên bộ	36
Hình 3.8c: Vị trí tiếp cận bộ 2	37
Hình 3.8d: Vị trí tiếp cận 3.....	37
Hình 3.9: Chương trình con thực hiện mang ly trở về vị trí ban đầu.....	37
Hình 3.10: Giao diện Control Panel các tính năng cơ bản RoboDK API.....	38
Hình 3.11: Giao diện điều khiển chuyển động Tool và khớp robot.....	40
Hình 3.12: Giao diện demo RoboDK API dùng C#	40
Hình P1.1: Lệnh Joint Move và truy xuất Target	45
Hình P1.2: Báo lỗi điểm kì dị.....	46
Hình P1.3: Chuyển Linear Move sang Joint Move.....	46
Hình P1.4: Đặt hệ tọa độ tham chiếu hiện hành.....	47
Hình P1.5: Thiết lập hệ tọa độ công cụ.....	47
Hình P1.6: Lệnh Circular Move.....	48
Hình P1.7: Cài đặt giá trị vận tốc	49
Hình P1.8: Lệnh Pause	49
Hình P1.9: Gọi chương trình con	50
Hình P1.10: Thiết lập I/O.....	51

Hình P1.11: Thiết lập Timeout (ms)	51
Hình P1.12: Đặt giá trị làm tròn.....	52
Hình P1.13: Cửa sổ Program events	53
Hình P1.14: Lệnh sự kiện.....	54
Hình P1.15: Lệnh sự kiện Detach object	54
Hình P2.1: Tạo một project mới với station.....	56
Hình P2.2: Thêm lệnh cho chương trình.....	57
Hình P2.3: Thêm hệ tọa độ tham chiếu.....	59
Hình P2.4: Thay đổi vị trí của hệ tọa độ tham chiếu	59

PHẦN MỞ ĐẦU

1. Tính cấp thiết của đề tài

Hiện nay, Cách mạng Công nghiệp lần thứ 4 (CMCN 4.0) đang là xu hướng trên thế giới nó diễn ra trên mọi lĩnh vực ngành nghề và đời sống xã hội. Việt Nam nói chung và Học viện Công nghệ Bru Chính Viễn Thông nói riêng cũng không nằm ngoài xu thế đó. “Học viện Công Nghệ Bru Chính Viễn Thông là đơn vị tiên phong về chuyển đổi số trong hệ thống các trường đại học Việt Nam và hướng đến trở thành đại học số hàng đầu Việt Nam về đào tạo công nghệ số phục vụ chuyển đổi số quốc gia”, trích lời Giám đốc Học viện Công Nghệ Bru Chính Viễn Thông PGS.TS Vũ Văn San.

Hệ thống robot dịch vụ có thể chủ động giám sát và thích ứng nhanh với những thay đổi liên tục trong các dây chuyền sản xuất. Bằng khả năng kết nối nhiều thiết bị, robot ngày càng thích ứng linh hoạt với không chỉ nhiệm vụ của bản thân nó mà còn cộng tác thực hiện nhiệm vụ với các robot cộng tác khác, các hệ thống thiết bị khác và ngay cả với con người trong các nhà máy thông minh. Vì vậy, nghiên cứu phát triển và nghiên cứu ứng dụng robot cộng tác là xu hướng tất yếu hiện nay và trong tương lai.

Nếu như, khi phát triển các ứng dụng di động trên iOS hoặc Android người ta có sẵn một bộ phần mềm phát triển (SDK) để xây dựng các ứng dụng đó, vậy với robot cộng tác thì như thế nào? Phần mềm cho robot đó là sự tích hợp giao tiếp với của các công nghệ như Cơ điện tử, thị giác máy tính và nhiều công nghệ khác... Tóm lại, việc phát triển phần mềm cho robot không phải là nhiệm vụ đơn giản.

Xuất phát từ những lý do trên cùng với sự đồng ý của thầy TS. Chung Tân Lâm tôi chọn đề tài luận văn : “**Ứng dụng ROS (Robot Operating System) điều khiển robot dịch vụ bán cafe**”, luận văn góp phần vào việc giải quyết các vấn đề lựa chọn công cụ cần thiết để phát triển ứng dụng mô phỏng và điều khiển thực tế cho riêng robot dạng cánh tay giúp giảm chi phí xây dựng phần mềm và thời gian

nghiên cứu. Đề tài có sự tham gia hỗ trợ về định hướng ứng dụng của Công ty TNHH Chế tạo máy 3C (Công ty được hỗ trợ bởi Quỹ đổi mới sáng tạo Vingroup VINIF trong dự án mã số VINIF.2020.NCUD.DA059) để kết quả đề tài mang tính thực tiễn, có khả năng ứng dụng sau khi nghiên cứu.

2. Mục đích nghiên cứu

Mục tiêu chính : Xây dựng một hệ điều hành robot tích hợp gồm Nghiên cứu, tìm hiểu phần mềm mô phỏng robot là RoboDK, cách cài đặt, các hệ tọa độ, tập lệnh chuyển động, lập trình mô phỏng và giao tiếp thông qua API để xây dựng nền tảng ban đầu hệ điều hành robot. Ứng dụng vào robot dịch vụ cafe. Từ mục tiêu chính trên, luận văn sẽ dự kiến các kết quả đạt được như sau:

- Tìm hiểu tổng quan về robot dịch vụ.
- Tìm hiểu phần mềm mô phỏng RoboDK.
- Thiết kế một số thao tác robot pha chế cafe và lập trình mô phỏng.
- Lập trình RobotDK API dùng C#.

3. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu

- Đối tượng nghiên cứu chính: phần mềm RoboDK.
- Mô phỏng robot thực hiện dịch vụ pha chế cafe trên máy tính.
- Lập trình giao tiếp với RoboDK API.

Phạm vi nghiên cứu

- Môi trường mô phỏng và lập trình là windows 10, là hệ điều hành phổ biến trong công nghiệp, cũng như trong điều khiển realtime công nghiệp. Môi trường Linux được khảo sát trong nghiên cứu khác.
- Robot dịch vụ được khảo sát là loại robot cộng tác được ứng dụng nhiều trong các dịch vụ hiện nay. Robot cộng tác được chọn là loại thương mại

DOBOT-CR10 được mô hình hóa 3D vào phần mềm RoboDK. Đây là loại robot thực hiện được các thao tác phức tạp và thao tác chính xác có thể lập trình được.

- Luận văn ứng dụng robot cụ thể vào dịch vụ pha chế cafe để chứng minh tính khả thi. Các thao tác của robot là loại Point-To-Point. Lập trình cho robot thực hiện 1 số thao tác cơ bản phục vụ cafe bao gồm: lấy ly cafe, rót cafe vào ly và đưa ly cafe vào vị trí phục vụ khách hàng.
- Thực hiện giao tiếp với RoboDK thông qua API dùng C#. Đề tài không thực hiện lập trình điều khiển real-time cho robot thật do liên quan đến các kỹ thuật điều khiển và phần cứng chuyên dùng thuộc lĩnh vực robot.

4. Phương pháp nghiên cứu

Phương pháp luận: Dựa trên cơ sở phân tích tính năng hệ thống phần mềm cấp cao có sẵn, phân tích hệ thống điều khiển cấp thấp, thực hiện mô phỏng robot và giao tiếp thông qua các API giữa robot và phần mềm.

Phương pháp đánh giá bằng mô phỏng trên máy tính: Xây dựng mô hình robot, mô phỏng điều khiển robot đã đề xuất, kết nối được với môi trường mô phỏng robot dùng C# để từ đó sẵn sàng điều khiển với các cơ cấu chấp hành theo thời gian thực với truyền thông EtherCAT.

CHƯƠNG 1: NGHIÊN CỨU TỔNG QUAN

1.1 Tổng quan về robot dịch vụ

1.1.1 Robot dịch vụ

Robot được nghiên cứu và phát triển ngày càng rộng với nhiều lĩnh vực khác nhau. Trong đó robot cộng tác được định hướng nghiên cứu để mở rộng thêm các ứng dụng, đặc biệt trong lĩnh vực dịch vụ.

Robot dịch vụ có hai loại: loại điều khiển realtime và loại điều khiển non-realtime. Robot điều khiển non-realtime bao gồm các robot di động hoặc các cánh tay thao tác theo một quỹ đạo cố định. Robot điều khiển realtime đó là các robot có các quỹ đạo phức tạp hoặc quỹ đạo chính xác cần phải bám theo (tracking) để thực hiện các thao tác khác nhau của con người với đặc trưng chuyển động Point-to-Point. Robot dịch vụ như vậy được đưa vào sử dụng với các công việc có tính lặp lại và hoạt động bất chước thao tác của con người. Đề tài tập trung vào lớp realtime robot.

Robot trong lĩnh vực dịch vụ như trên có thể ứng dụng trong nhiều ngành, trong đó lĩnh vực ẩm thực, nhà hàng, ăn uống, dịch vụ (F&B - Food and Beverage Service). Đây là một minh họa tốt nhất cho các hoạt động dịch vụ, và robot phục vụ cafe được chọn làm mô hình ứng dụng trong đề tài này.

1.1.2 Lợi ích khi sử dụng robot dịch vụ

Một số lợi ích tiêu biểu khi sử dụng robot phục vụ thay thế cho con người trong ngành dịch vụ như sau:

Robot thực hiện công việc phục vụ mang đến sự tiện lợi hơn cho các cơ sở kinh doanh. Năng suất hoạt động và làm việc của robot so với con người cao hơn nhiều, điều này giúp việc kinh doanh giảm bởi được chi phí thuê nhân lực và chi phí quản lý. Một robot có thể hoạt động nhiều giờ liên tục trong ngày với các loại robot được thiết kế đặc biệt có độ bền của sản phẩm cao.

Đối với những công việc phục vụ đòi hỏi tính chính xác cao và có sự nguy hiểm, đưa robot vào ứng dụng robot sẽ đảm bảo hiệu quả hơn một cách rõ rệt.

Hiện nay, robot được ứng dụng trí tuệ nhân tạo AI, điều này khiến robot có thể thực hiện công việc thông minh hơn và xử lý nhận dạng đa dạng hơn với các tình huống trong công việc phục vụ của ngành dịch vụ.

1.1.3 Một số robot phục vụ phổ biến hiện nay

Khi nói đến robot được thiết kế và lập trình với mục đích phục vụ thì phổ biến nhất hiện nay phải kể đến robot dịch vụ y tế và dịch vụ nhà hàng khách sạn.

(1) Robot dịch vụ y tế

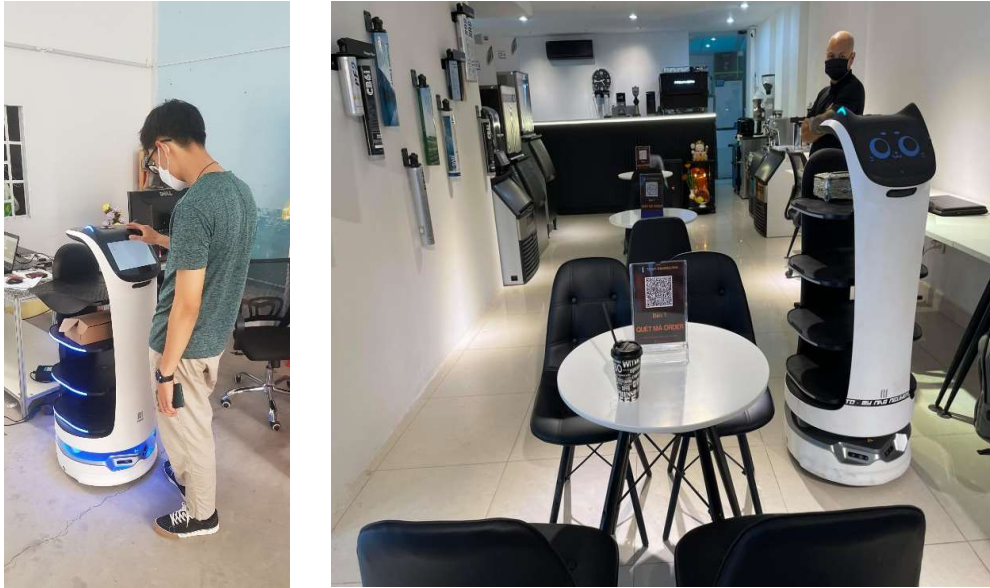
Có nhiều loại robot khác nhau phục vụ y tế hiện nay được đưa vào ứng dụng. Tính đến thời điểm này, đã có hơn 12 loại robot khác nhau ứng dụng trong lĩnh vực này như:

- Robot lễ tân bệnh viện.
- Robot điều dưỡng.
- Robot tư vấn sức khỏe từ xa.
- Robot cứu thương.
- Robot phục vụ trong bệnh viện.
- Robot phun xịt thuốc khử trùng.
- Robot vệ sinh bệnh viện.
- Robot trị xạ.
- Robot hỗ trợ phục hồi chức năng.
- Robot phẫu thuật.
- Robot giao hàng ngoài trời.
- Robot thực phẩm.

Các loại robot này được đưa vào áp dụng thay thế cho sức người và đang làm việc hiệu quả theo chương trình lập trình sẵn và là xu hướng hiện nay. Các robot phục vụ bệnh viện không chỉ giúp các bác sĩ và nhân viên y tế thực hiện đối với các

nhiệm vụ phức tạp và nó còn thực hiện công việc một cách chính xác hơn, nhờ vậy khối lượng công việc của con người trong lĩnh vực này được giảm bớt và tạo hiệu quả hoạt động cao hơn cho các cơ sở y tế.

(2) Robot di động phục vụ nhà hàng



Hình 1.1: Robot BellaBot phục vụ nhà hàng ở một nhà hàng Q.7 TP.HCM

Robot di động được sử dụng rất phổ biến trong lĩnh vực dịch vụ do chuyển động linh hoạt trên bánh xe, đặc biệt là ở các cửa hàng ăn uống, nhà hàng, khách sạn. Một robot phục vụ nhà hàng của hãng Pudu là BellaBot sử dụng trí tuệ nhân tạo, kết hợp với thiết bị định vị và định hướng với nhiệm vụ đặt món, mang món ăn cho khách. Sau khi nhận được thông tin đặt món trên hệ thống thông minh, nhân viên bếp sẽ tiến hành chế biến và đưa các món cho Bella phục vụ. Robot Bella sẽ tìm đến đúng số bàn để mang phần ăn cho khách. Sử dụng robot phục vụ trong các nhà hàng đang là xu hướng hiện nay, đặc biệt với tình hình đại dịch Covid đang phức tạp như hiện nay thì sử dụng robot được xem là giải pháp thiết thực. Không chỉ vậy, nếu nhân viên bình thường khi phục vụ tại nhà hàng, mỗi ngày có thể phục vụ được khoảng 200 khách thì Bella có thể phục vụ từ 300 cho đến 400 khách.

Với sự phát triển của công nghệ định hướng và định vị mà các robot này được huấn luyện lập bản đồ nhà hàng, có thiết kế để có thể di chuyển một cách dễ dàng và êm ái, tránh được các vật cản và một số loại còn có thể giao tiếp cơ bản với khách hàng.

Để mang đến sự thân thiện với khách hàng khi vào quán, các robot được thiết kế và lập trình hiển thị biểu cảm gương mặt như nháy mắt, chớp mắt, cười, ... biểu cảm đa dạng.

Robot phục vụ Bella tại quán cafe có khả năng di chuyển thông minh bằng camera và cảm biến, việc di chuyển này để đảm bảo đồ uống chúng mang không bị chông chênh hoặc đổ ra ngoài. Robot Bella được lập trình theo những đường đi theo giải thuật, sau khi mang cafe đến bàn của khách xong thì nó sẽ tự động di chuyển về vị trí ban đầu của mình.

Hiện nay trên thị trường có nhiều loại robot phục vụ khác nhau trong nhà hàng, chúng khác nhau trong thiết kế, lập trình và khả năng làm việc ổn định.

(3) Robot dịch vụ pha chế cafe

Robot không chỉ phục vụ mang phần ăn trong nhà hàng mà còn được sử dụng nhiều trong pha chế thực phẩm. Nó chính là giải pháp thay thế cho nhân viên pha chế bằng sức người rất hiệu quả. Hiện nay tay máy robot phục vụ trong quán cafe được nhiều nơi tiên phong trên thế giới đưa vào sử dụng ngày càng nhiều. Robot pha chế cafe được đặt trong một không gian nhỏ gọn tại các khu vực giải trí, nhà hàng, khách sạn, sân bay, bệnh viện, trường học ... mà không cần nhân viên phục vụ. Với những tính năng này nhằm hướng tới các đô thị thông minh trong xu hướng chuyển đổi số các hoạt động của cuộc sống trong tương lai.



Hình 1.2: Ella là sản phẩm của Crown Digital

Crown Digital là một công ty khởi nghiệp dựa vào IoT với mục tiêu giải quyết những khó khăn trong ngành F&B. Ella là sản phẩm đầu tay của Crown Digital [2]: một robot tự động được thiết kế để mô phỏng công việc của một người phục vụ cafe (hình 1.2).

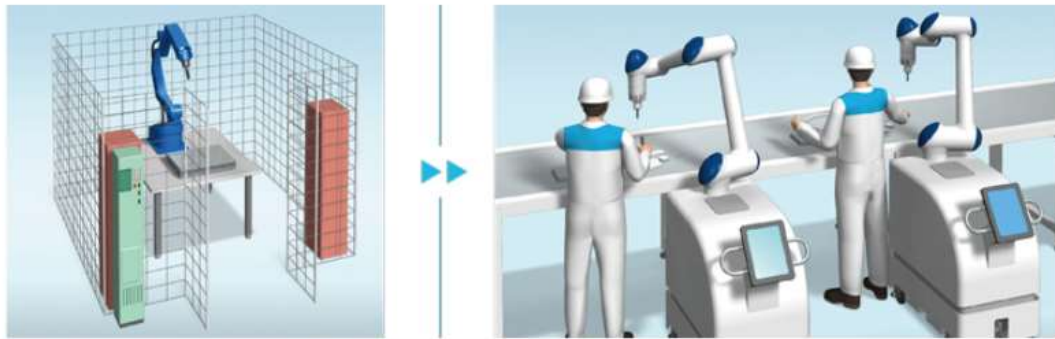
Ella có dạng một cánh tay tự động, do công ty robot Techman Robot sản xuất [3]. Ella nằm bên trong một ki-ốt trong suốt, rộng khoảng 5m². Robot hoạt động suốt ngày đêm và có thể phục vụ khoảng 200 tách cafe mỗi giờ - nhanh gấp 4 lần so với một nhân viên pha cafe thông thường.

Các nguyên liệu pha chế - bao gồm hạt cafe và sữa tươi (được bổ sung thêm sau khi robot phục vụ khoảng 360 cốc cafe). Việc này sẽ được thực hiện bởi các nhân viên chịu trách nhiệm giám sát Ella thông qua một ứng dụng theo dõi. Trung tâm chỉ huy nội bộ của Crown Digital cho phép Ella phát hiện và giải quyết từ xa sự cố tràn nước hoặc các lỗi thiết bị khác. Robot này giải quyết được vấn đề tốc độ, tính nhất quán và khả năng đặt hàng dễ dàng.

1.2 Robot cộng tác

Robot dịch vụ là một ứng dụng phổ biến và đầy đủ nhất của robot cộng tác. Robot cộng tác (collaborative robot - cobot) là một loại robot được thiết kế để làm

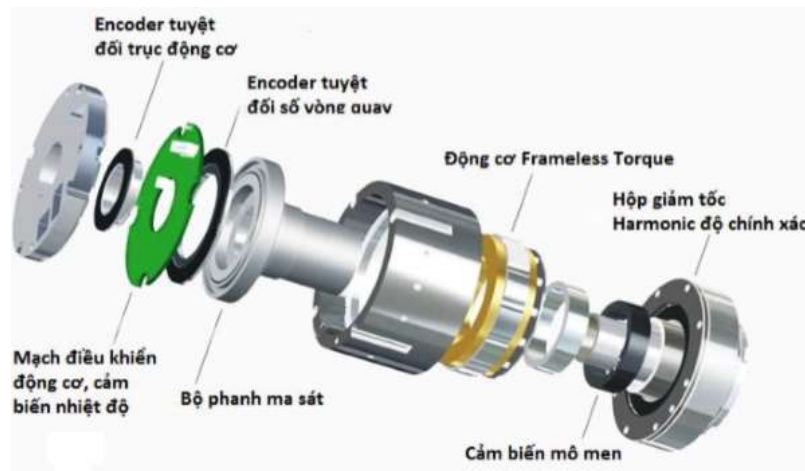
việc chung với con người trong một không gian chung. Một trong những điểm khác biệt giữa cobot so với robot công nghiệp chính là robot công nghiệp thường hoạt động độc lập, cần có thiết bị bảo vệ và rào chắn để tránh gây nguy hiểm đến con người; còn cobot được đánh giá cao bởi tính an toàn, chính vì thế đây được coi là trợ thủ đáng tin cậy khi làm việc cùng con người trong các hoạt động sản xuất công nghiệp và dịch vụ. Cobot có yêu cầu kỹ thuật và thiết kế cao và khác biệt hơn so với robot truyền thống: Các bộ phận cấu thành cần phải được thiết kế đặc biệt tinh gọn (compact), đảm bảo độ an toàn và tin cậy cao, tương tác linh hoạt và đáp ứng nhanh với người sử dụng và làm việc cùng, dễ điều khiển với chức năng dạy học (teach mode), và phải hiệu quả về mặt chi phí.



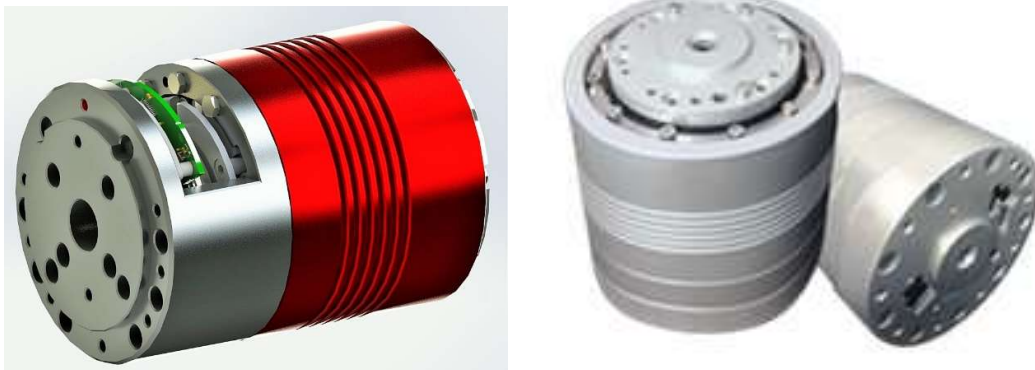
Hình 1.3: Sự khác biệt cơ bản về sự an toàn giữa robot công nghiệp và cobot

Đối với các cobot, tất cả các bộ phận và phần tử cơ-điện cần phải được thiết kế và lắp đặt tinh gọn ngay trong mỗi khâu và khớp quay của robot (robot joint). Bên cạnh đó, động cơ sử dụng cho cobot cũng phải có kích thước nhỏ gọn, nhưng phải đáp ứng đủ công suất và mô men xoắn. Vì vậy, các động cơ thông thường không được sử dụng, mà cần phải sử dụng động cơ mô-men xoắn không khung sườn (frameless torque motor). Ngoài ra, vì cobot làm việc cùng với người, cho nên yếu tố an toàn trong sử dụng và vận hành rất quan trọng. Các thiết kế và chế tạo liên quan tới cobot cần phải đảm bảo về độ an toàn và tin cậy cao. Do đó, các phần tử và thiết bị thông minh (smart devices, smart sensors), cảm biến lực (force sensors), và cảm biến hình ảnh (image sensors, camera) thường được sử dụng để giúp cobot nhận biết được sự hiện diện của người sử dụng hay làm việc cùng, và có khả năng

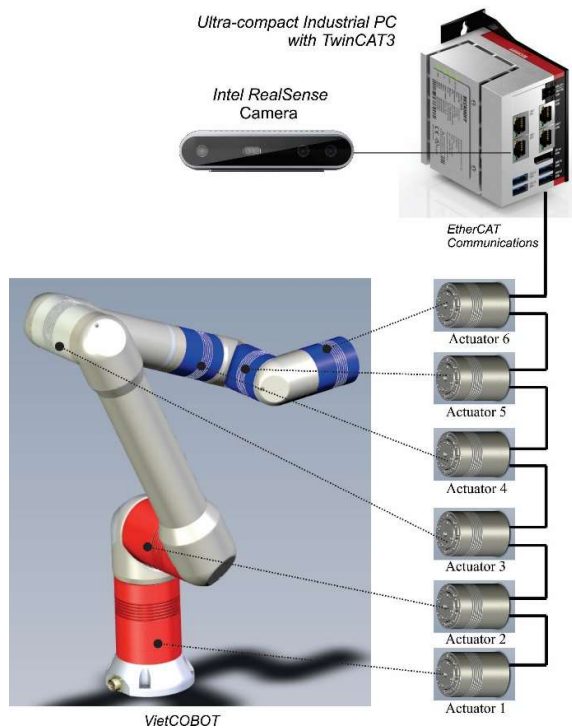
đáp ứng tức thì, tránh gây nguy hại tới người sử dụng và làm việc cùng. Cobot thường hỗ trợ làm các công việc với tải trọng không lớn (từ 10 - 15 kg), với độ chính xác cũng như khả năng linh hoạt và độ khéo cao (high dexterity), và vì vậy số bậc tự do thường từ 6 trở lên, giúp cho có thể mở rộng không gian làm việc, cũng như cho phép tiếp cận nhiều hướng và góc khác nhau, tránh va chạm với người sử dụng và vận hành. Ngoài ra, cobot cần phải được trang bị chức năng điều khiển bằng dạy học (teach mode), với độ chính xác và linh hoạt cao, dễ học và sử dụng đối với người không có kinh nghiệm về lập trình.



Hình 1.4: Cấu trúc cơ bản của khớp đơn cobot



Hình 1.5: Mẫu thiết kế cơ cấu chấp hành của khớp cobot



Hình 1.6: Mẫu thiết kế cơ cấu chấp hành của khớp cobot

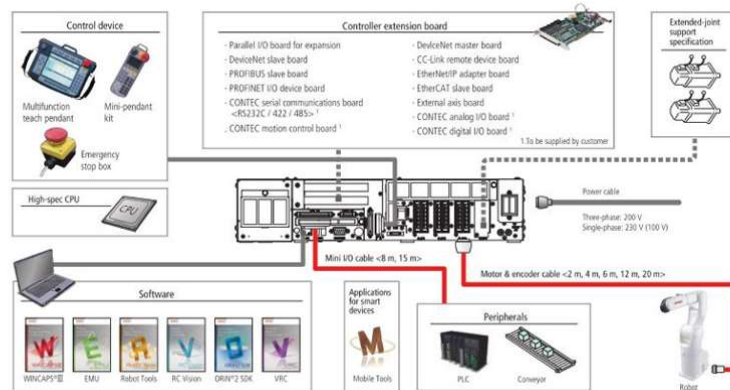
Robot cộng tác có ưu điểm là cấu trúc nhỏ gọn, độ tin cậy cao, tiêu thụ điện năng thấp, dễ xử lý và giá thành rẻ. Những tính năng này làm cho robot cộng tác rất hiệu quả để áp dụng rộng rãi cho nhiều ngành công nghiệp và cuộc sống. Ngày nay, robot cộng tác không chỉ cách mạng hóa ngành công nghiệp sản xuất, chúng còn được sử dụng kết hợp với các công nghệ hiện tại để đổi mới các ngành dịch vụ nhằm tăng giá trị thực và cho cuộc sống của chúng ta

Trong tương lai robot cộng tác sẽ là xu thế mới trong các công ty ứng dụng công nghệ 4.0 vào trong sản xuất công nghiệp và dịch vụ trong cuộc sống và giải trí. Với các tính năng an toàn và dễ dàng lắp đặt cũng như lập trình. Xu thế sử dụng robot cộng tác sẽ ngày càng phổ biến trong công nghiệp và trong cuộc sống trong tương lai.

1.3 Hệ thống điều hành robot (Robot Operating System)

1.2.1 Hệ thống điều khiển tập trung robot truyền thống

Trong hệ thống điều khiển tập trung, các driver điều khiển động cơ được thiết kế tập trung tại tủ điện điều khiển, do đó có sự kết nối cáp điện khá lớn từ robot về tủ điều khiển. Hệ thống điều khiển này có thể điều khiển các hệ thống robot lớn nhờ các hệ điện tử công suất cao có đủ không gian bố trí trong tủ điện. Trước khi có các nghiên cứu về robot công tác, hệ thống điều khiển tập trung được sử dụng phổ biến trong công nghiệp với tên thường gọi là robot công nghiệp và được ứng dụng trong công nghiệp xe hơi, công nghiệp hàn, công nghiệp sơn, đóng gói hàng nặng trong kho, Một ví dụ minh họa của hệ thống điều khiển robot kiểu tập trung RC8 của hãng Denso cho trên hình 1.6. Trong hệ thống này, các servo driver được gắn ngay trên bộ điều khiển với các khe cắm PCI của máy tính công nghiệp được chế tạo chuyên dùng cho hãng Denso [1].



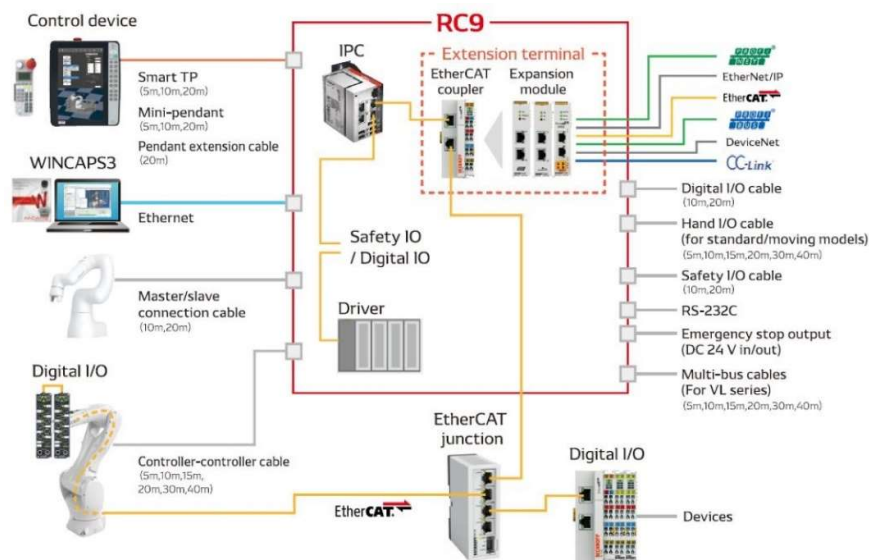
Hình 1.7: Hệ thống điều khiển tập trung RC8 cho robot truyền thống (Denso)

1.2.2 Hệ thống điều khiển phân tán cải tiến cho robot công tác

Với sự bảo hòa của việc ứng dụng robot trong công nghiệp, các nhà hoạch định chiến lược robot mong muốn mở rộng khả năng robot trong nhiều lĩnh vực khác, nhất là trong lĩnh vực dịch vụ làm việc chung với con người. Hệ thống điều khiển cũng được cải tiến thay đổi tương ứng thành hệ thống điều khiển phân tán chuyên dùng cho robot công tác. Trong hệ thống điều khiển phân tán các servo

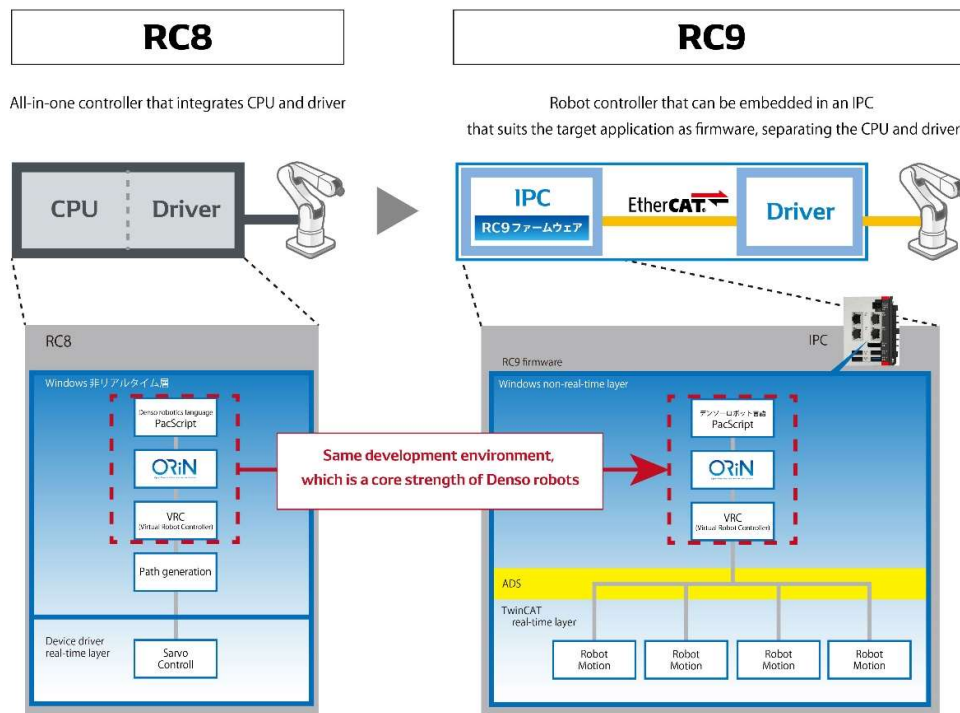
driver được thiết kế tích hợp ngay trên các động cơ khớp của robot tạo thành mô khối tích hợp mật độ cao được gọi là Joint actuator (cơ cấu chấp hành khớp) xem hình 1.5. Do đó bộ điều khiển phân tán không có các servo driver mà chúng đã được phân tán trên các khớp quay của robot. Việc kết nối điện giữa robot và tủ điện cũng gọn hiệu quả hơn nhiều vì chỉ còn có sự kết nối truyền thông (không phải là các kết nối điện công suất cao) giữa tủ điện và robot. Một kết nối truyền thông sử dụng phổ biến trong trường hợp này là EtherCAT. Việc so sánh 2 bộ điều khiển được trình bày ở phần dưới.

Hệ thống điều khiển phân tán dựa trên nền tảng truyền thông realtime EtherCAT thông qua cổng LAN Gigabit. Các module chuyên dùng EtherCAT slave như Servo motor driver, I/O mô-đun, được giao tiếp có dây với Master EtherCAT với tốc độ truyền realtime. Hệ thống điều khiển cobot được cải thiện đáng kể về tốc độ truyền và kết cấu phần cứng nhỏ gọn kiểu mô-đun hóa. Trên hình là minh họa bộ điều khiển cobot phân tán cải tiến cho cobot của hãng Denso [7]. Hệ thống sử dụng nền tảng realtime win10 trên IPC của hãng Beckhoff. Với nền tảng này thì khả năng mở rộng của bộ RC9 rất linh hoạt và có thể tích hợp theo nhu cầu thực tế của ứng dụng.



Hình 1.8: Hệ thống điều khiển phân tán RC9 cho cobot của hãng Denso

So sánh hệ thống điều khiển robot tập trung RC8 và hệ điều khiển robot phân tán RC9 cho trên hình 1.8. Đối với RC8, bộ xử lý chính CPU tích hợp trực tiếp với các driver động cơ servo của robot, gọi là All-in-One controller. Phân kết nối của bộ điều khiển với các động cơ servo của robot rất cồng kềnh tuy rằng nó có thể gánh các servo công suất cao tùy ý, tuy nhiên điều đó không có ý nghĩa nhiều đối với các ứng dụng dịch vụ. Đối với RC9, các driver động cơ được thiết kế phân tán đi kèm theo các động cơ servo nên kết cấu nhỏ gọn và bộ điều khiển chính có thể dùng là một máy tính công nghiệp chuyên dùng (IPC) của hãng Beckhoff với TwinCAT realtime driver được phân bố trên hệ điều hành được tối ưu cho mục đích điều khiển realtime. Nền tảng phần mềm cấp cao của Denso vẫn giữ nguyên trong lớp non-real-time của IPC. Với cách bố trí như vậy, hệ thống điều khiển robot rõ ràng và hiệu quả hơn rất nhiều so với truyền thống. Và đề tài này cũng nhằm mục đích xây dựng một hệ thống tương tự với các phần cứng sẵn có



Hình 1.9: So sánh hệ thống điều khiển tập trung RC8 và phân tán RC9

1.4 Phần mềm mô phỏng robot - RoboDK

RoboDK là một phần mềm mô phỏng và lập trình ngoại tuyến cho robot công nghiệp [3]. Phần mềm mô phỏng có thể được sử dụng cho nhiều dự án sản xuất bao gồm lĩnh vực công nghiệp như phay, hàn, gấp-đặt, đóng gói và dán nhãn, xếp kho hàng, sơn, lắp ráp, và lĩnh vực dịch vụ như nhà hàng, siêu thị, ẩm thực,

Chương trình điều khiển robot cổ thể được lập trình cấp cao thao tác trực tiếp trên robot mô phỏng, sau đó được xuất ra chương trình tương thích với các robot thật hiện có trên thị trường. Môi trường mô phỏng có thể được giao tiếp thông qua API có thể được viết bằng C#, C ++, Python, Matlab.

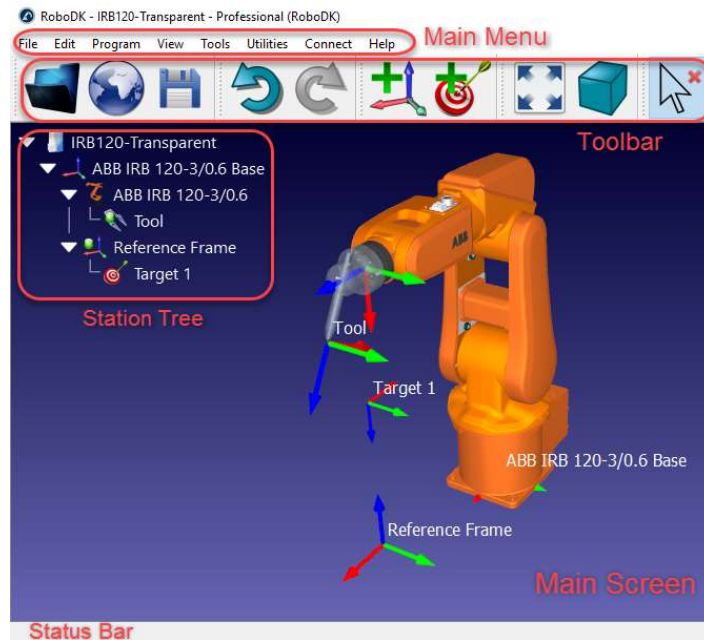
Thư viện robot khá lớn với thư viện các robot công nghiệp và các đồ gá từ hơn 40 nhà sản xuất robot trên thế giới.

Bộ xử lý xuất chương trình RoboDK hỗ trợ nhiều bộ điều khiển robot thương mại trên thị trường, bao gồm:

- ABB RAPID (mod/prg)
- Fanuc LS (LS/TP)
- KUKA KRC/IIWA (SRC/java)
- Motoman Inform (JBI)
- Universal Robots (URP/script), ...

1.4.1 Giao diện RoboDK

Giao diện chính của RoboDK bao gồm màn hình chính (Main Screen), quản lý các đối tượng robot dạng cây (Station Tree), các tác vụ được chọn từ thanh thực đơn (Main Menu) hoặc từ thanh công cụ (Toolbar) và thông tin được hiển thị trên thanh trạng thái (Status Bar), xem hình 1.9.



Hình 1.10: Giao diện của phần mềm RoboDK

1.4.2 Tập lệnh chính của robotDK

Xin tham khảo *Phụ lục P1*

1.4.3 Lập trình và mô phỏng trên RoboDK

Xin tham khảo *Phụ lục P2*

1.5 Kết luận chương

Chương này tổng quan các vấn đề liên quan đến đề tài bao gồm tổng quan về robot dịch vụ phổ biến nhất như robot di động phục vụ trong nhà hàng và robot dịch vụ thao tác pha chế cafe. Phân loại các loại bộ điều khiển robot bao gồm bộ điều khiển robot tập trung và bộ điều khiển phân tán để làm tiền đề cho việc định hướng nội dung và các giới hạn nghiên cứu phù hợp cho đề tài. Tìm hiểu về phần mềm RoboDK chuyên dùng cho mô phỏng các loại robot hiện có trên thị trường cũng như các loại robot tự nghiên cứu thiết kế đáp ứng việc ứng dụng nhanh robot vào thực tế công nghiệp và cuộc sống. Hơn nữa, việc kết hợp phần mềm mô phỏng với phần cứng điều khiển robot realtime là nền tảng để xây dựng một hệ điều hành robot có vai trò quan trọng trong việc nghiên cứu phát triển robot. Đề tài ứng dụng

vào việc sử dụng hiệu quả phần mềm RoboDK trong việc thiết kế robot thực hiện phục vụ các thao tác pha chế cafe để minh họa cho việc ứng dụng hiệu quả.

CHƯƠNG 2: MÔ HÌNH HÓA ROBOT PHA CHẾ

2.1 Xây dựng mô hình robot

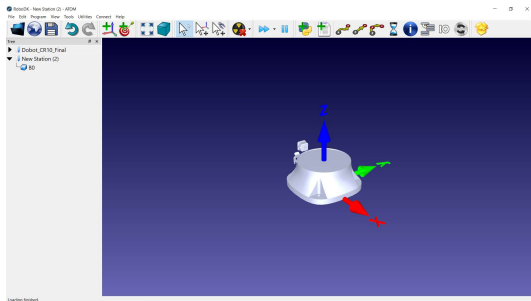
Robot khảo sát để ứng dụng mô phỏng thao tác pha chế trong chương này là Dobot-CR10 (hãng DOBOT) [8]. Robot này có 6 bậc tự do bao gồm 6 khớp được đánh số là J1, J2, J3, J4, J5, J6. Việc phân tích và xây dựng biểu thức động học của robot dạng cánh tay 6 bậc tự do này được trình bày trong [3]. Trong đề tài này sử dụng công cụ có sẵn trên RoboDK để minh họa tính linh hoạt và đa năng của phần mềm này.

2.1.1 Xây dựng các đối tượng 3D của robot

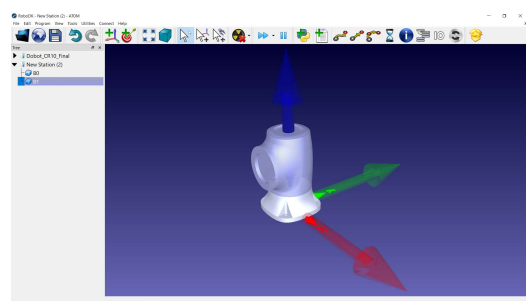
Trước tiên, ta cần thu thập thông tin về robot:

1. Mô hình 3D (như tập tin STEP hay IGES) bao gồm: nền (Hình 2.1a), khớp 1 (Hình 2.1b), khớp 2 (Hình 2.1c), khớp 3 (Hình 2.1d), khớp 4 (Hình 2.1e), khớp 5 (Hình 2.1f), khớp 6 (Hình 2.1g), và mô hình đầy đủ (Hình 2.2)
2. Datasheet của robot hay sổ tay kỹ thuật robot (robot manual) để có khoảng cách của của khớp (Hình 2.3)

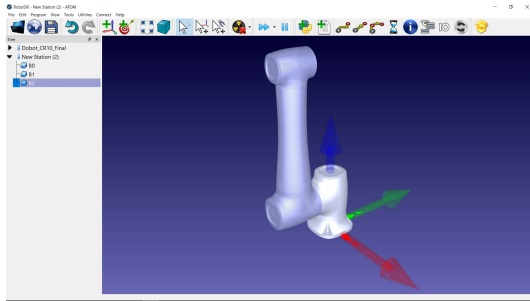
Khi có tập tin 3D vào RoboDK, ta **Kéo thả** robot vào RoboDK hay chọn select **file** và sau đó **open**. Thực hiện chia nhỏ một object robot thành các thành phần khác nhau để có thể xử lý từng phần riêng biệt như hình 2.1.



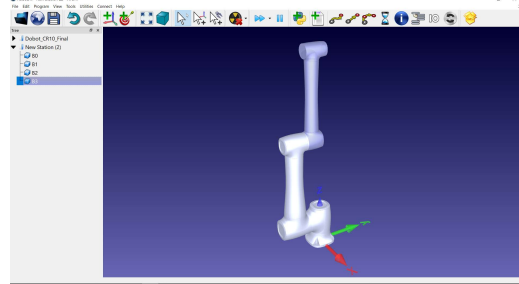
Hình 2.1a: Mô hình 3D nền (B0)



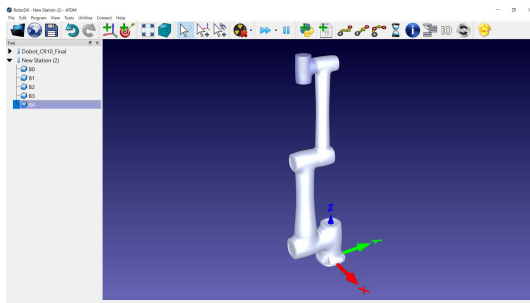
Hình 2.1b: Mô hình 3D khớp 1 (B1)



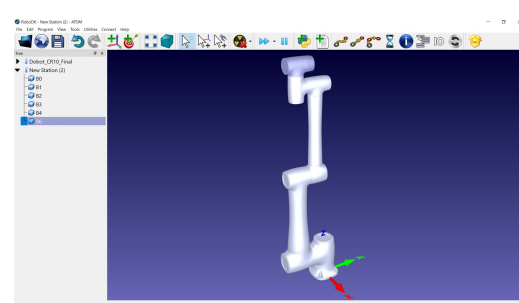
Hình 2.1c: Mô hình khớp 2 (B2)



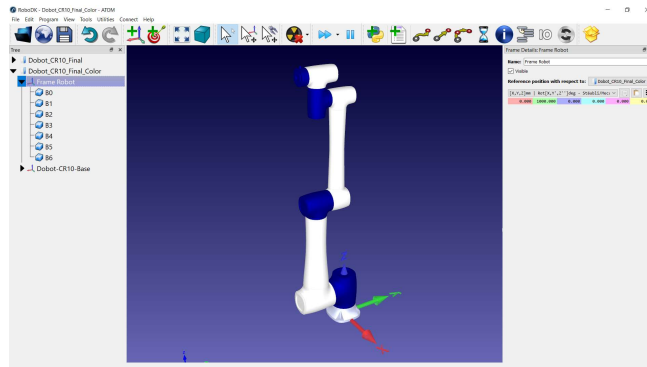
Hình 2.1d: Mô hình 3D khớp 3 (B3)



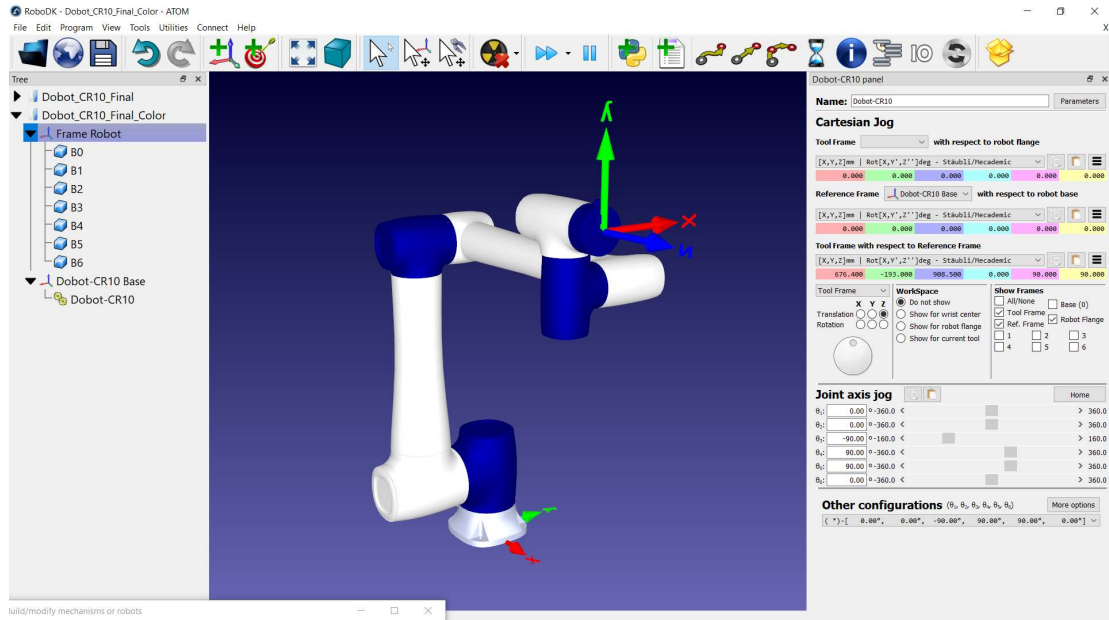
Hình 2.1e: Mô hình khớp 4 (B4)



Hình 2.1f: Mô hình 3D khớp 5 (B5)



Hình 2.1g: Mô hình 3D khớp 6 (B6)



Hình 2.2: Mô hình đầy đủ robot dịch vụ

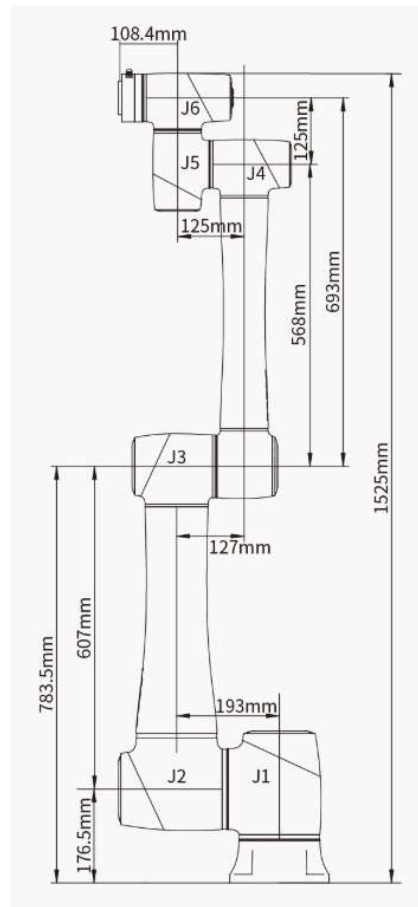
2.1.2 Cài đặt các thông số động học và giới hạn chuyển động khớp

Với các mô hình 3D tĩnh được ghép lại, ta có thể thực hiện các bước để định nghĩa mô hình động học cho robot bằng cách sau: Chọn **Utilities** → **Model Mechanism or Robot**. Dưới mục **Robot Type**, chọn **6 Axis Collaborative Robot**.

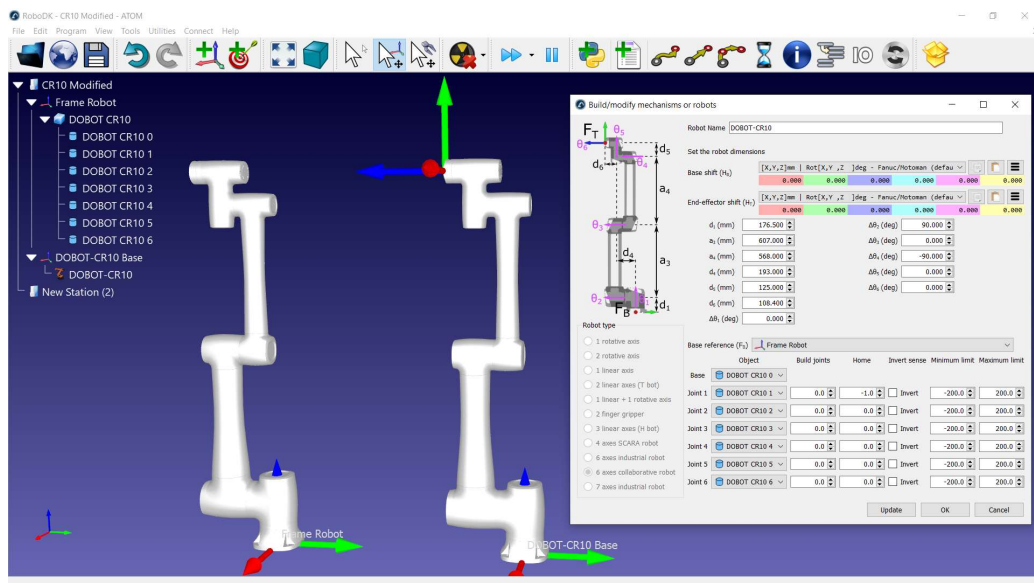
Nếu các thành phần của robot theo đúng thứ tự, thì tất cả các mô hình 3D được gộp lại theo đúng thứ tự đó. Khi đó, ta có thể liên kết bằng tay từng object tạo thành một robot đúng theo thứ tự trong Robot Builder.

Điền thông tin kích thước của robot bằng cách mở **data sheet** của Dobot-CR10 (Hình 2.3). Trên bản vẽ 3D của data sheet, ta sẽ tìm được tất cả giá trị kích thước cần tìm (Bảng 2.1). Điền các giá trị đó vào RoboDK, và khi kết quả thỏa mãn chuyển động động học của robot, chọn Update (Hình 2.4).

Dùng bảng điều khiển robot để quay từng khớp để bảo đảm số liệu động học nhập chính xác (Hình 2.4). Xác định các thông số góc quay giới hạn của từng khớp (Hình 2.5).



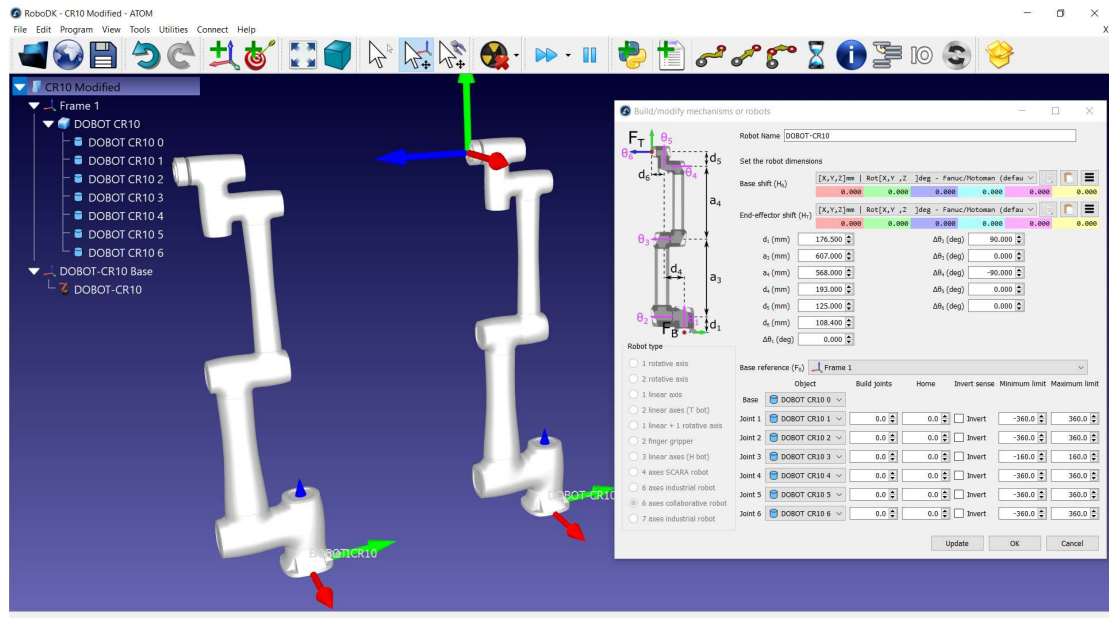
Hình 2.3: Kích thước Dobot CR10



Hình 2.4: Xác định và nhập các kích thước động học trên robot builder

Bảng 2.1: Thông số Dobot CR10

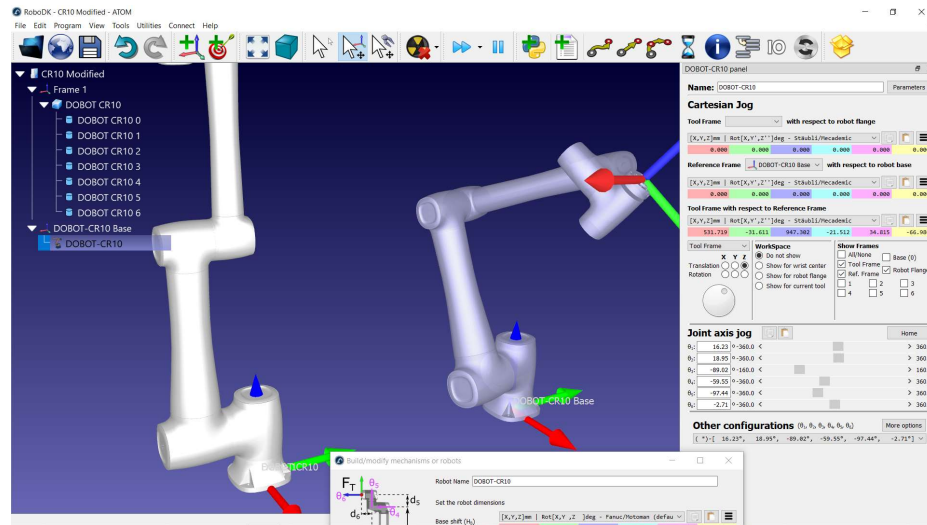
Tên cobot		DOBOT CR10
Khối lượng (Weight)		40kg
Tải tối đa (Maximum Payload)		10kg
Tầm vuron (Max Reach)		1525mm
Maximum Speed of TCP		4m/s
Giới hạn khớp (Joint Ranges)	J1	360°
	J2	360°
	J3	160°
	J4	360°
	J5	360°
	J6	360°



Hình 2.5: Xác định các giới hạn góc quay của khớp trên robot builder

2.1.3 Thử nghiệm mô hình robot

Sau khi nhập các tham số động học vào RoboDK trong Robot Builder. Ta có thể test thử nghiệm sự chuyển động của các khớp robot cũng như thiết lập vị trí HOME (Hình 2.7).



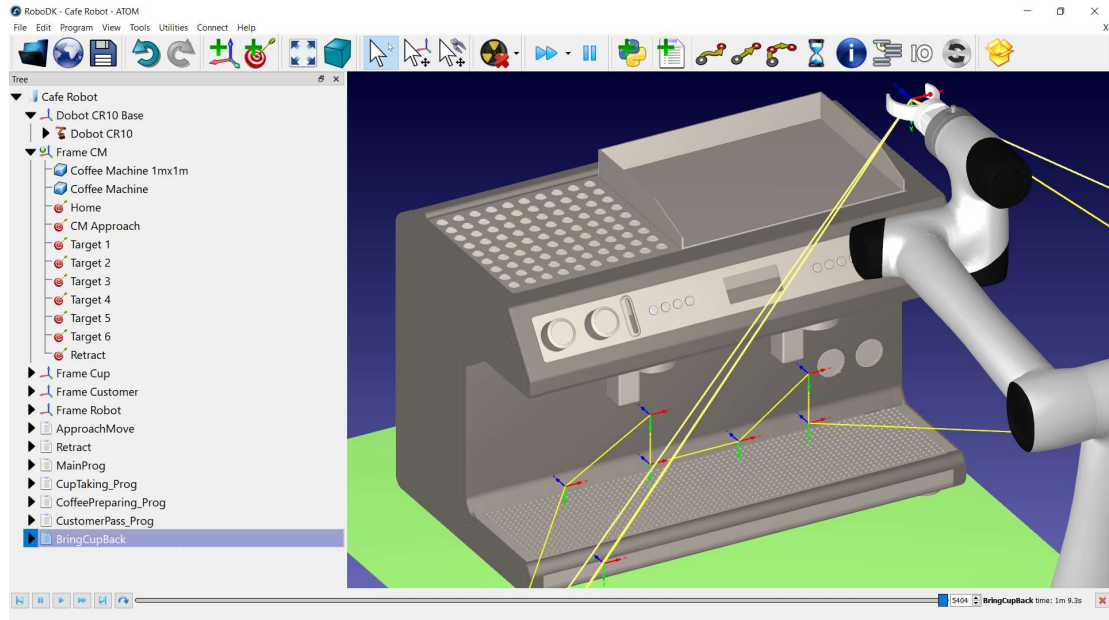
Hình 2.6: Test hoạt động của robot trên robot Panel

2.2 Xây dựng mô hình máy pha cafe

Mô hình máy pha cafe bao gồm:

1. Hệ tọa độ tham chiếu (Reference Frame): Frame CM
2. Các đối tượng (objects):
 - **Cafe Machine 1mx1m**: đối tượng khu vực đặt máy cafe
 - **Cafe Machine**: máy rót cafe
3. Các mục tiêu (Targets) sau:
 - **Home**: vị trí Home của robot
 - **CM Approach**: vị trí an toàn tiếp cận máy cafe
 - **Target 1, Target 2, Target 3, Target 4, Target 5, Target 6**: vị trí trình tự rót cafe vào ly
 - **Retract**: vị trí an toàn ra khỏi máy pha cafe

Mô hình máy pha chế cafe được cho trên Hình 2.7



Hình 2.7: Mô hình máy pha chế cafe

2.3 Xây dựng mô hình cấp ly cafe và bộ giao cafe

(1) Mô hình cấp ly cafe: bao gồm

1. Hệ tọa độ tham chiếu (Reference Frame): **Frame Cup**

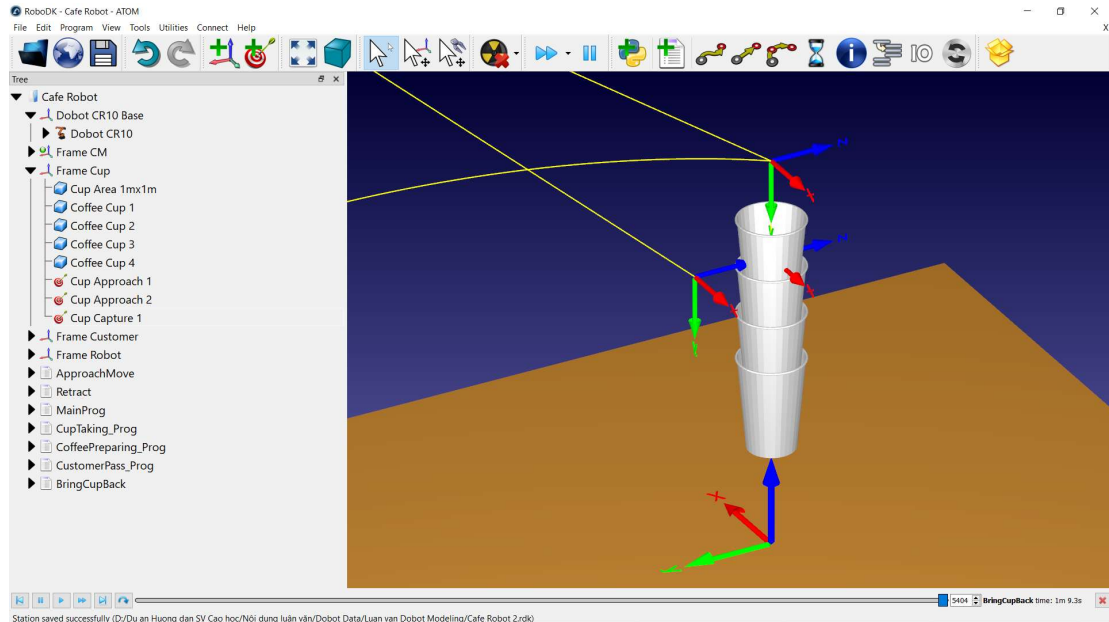
2. Các đối tượng (objects):

- **Cup Area 1mx1m:** đối tượng khu vực cung cấp ly cafe
- **Cafe Cup 1, cafe Cup 2, cafe Cup 3, cafe Cup 1:** ly cafe

3. Các mục tiêu (Targets) sau:

- **Cup Approach 1:** Vị trí tiếp cận ly 1
- **Cup Approach 2:** Vị trí tiếp cận ly 2
- **Cup Capture 1:** Vị trí lấy ly ra khỏi chồng ly

Mô hình cấp ly cafe được cho trên Hình 2.8.



Hình 2.8: Mô hình cấp ly cafe

(2) Mô hình bộ giao cafe cho khách hàng: bao gồm

1. Hệ tọa độ tham chiếu (Reference Frame): **Frame Customer**

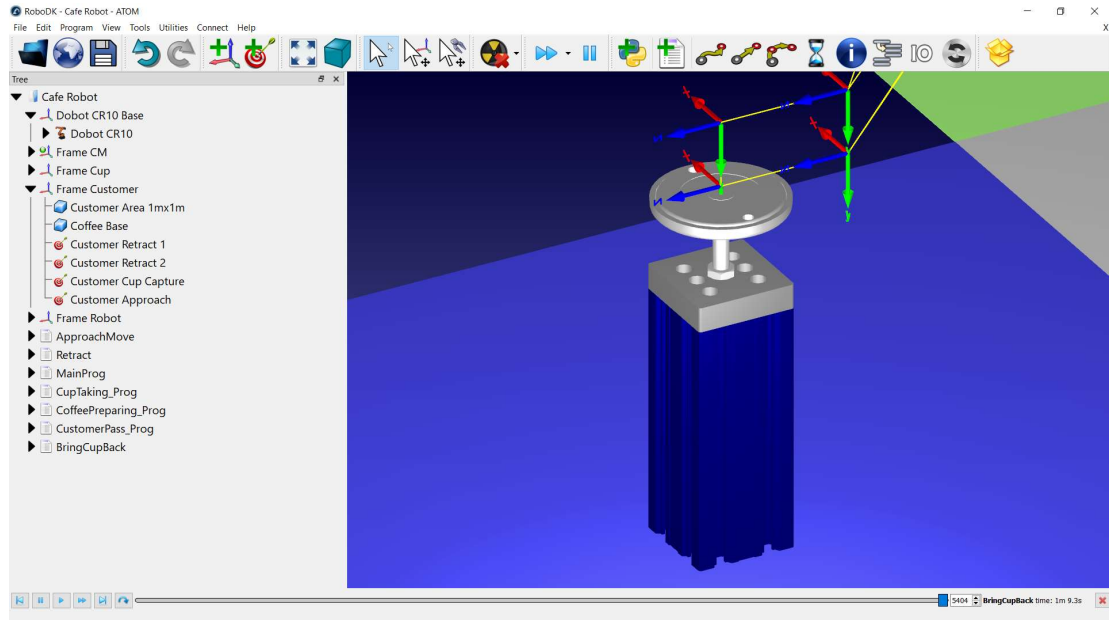
2. Các đối tượng (objects):

- **Customer Area 1mx1m:** đối tượng khu vực đặt ly cafe
- **Cafe Base:** bộ ly cafe

3. Các mục tiêu (Targets) sau:

- **Customer Retract 1:** vị trí tiếp cận ly 1
- **Customer Retract 2:** vị trí tiếp cận ly 2
- **Customer Cup Capture:** vị trí gấp ly
- **Customer Approach:** vị trí an toàn lấy ly ra

Mô hình bộ giao cafe được cho trên Hình 2.9.



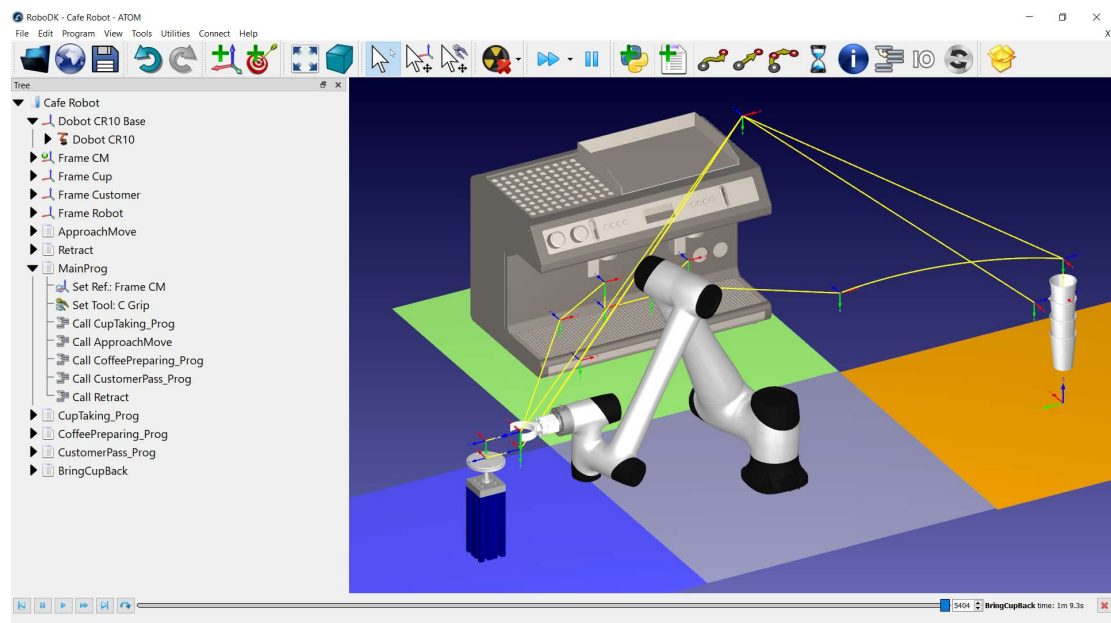
Hình 2.9: Mô hình bộ giao cafe

2.4 Tổng kết chương

Chương này trình bày trình tự xây dựng một mô hình trạm robot dịch vụ. Robot dịch vụ dạng cánh tay 6 bậc tự do được xây dựng hoàn chỉnh từ các khối 3D rời rạc. Với tính năng xây dựng cấu trúc động học của phần mềm RoboDK (tính năng *Model Mechanism or Robot*). Mẫu thử nghiệm Dobot-CR10 được gắn các khớp quay vào các trục phù hợp tạo các chuyển động hợp lý cho cánh tay robot. Ngoài ra, các mô hình khác cũng được dựng như mô hình cấp ly cafe, mô hình máy pha cafe, mô hình bộ giao cafe phục vụ cho việc mô phỏng. Với mô hình trạm robot dịch vụ này, chúng ta có thể thực hiện việc lập trình cho các chuyển động thao tác chế biến ở chương sau.

CHƯƠNG 3: THIẾT KẾ VÀ MÔ PHỎNG THAO TÁC PHA CHẾ

Trong chương 2, các mô hình 3D của các đối tượng được xây dựng phục vụ cho việc lập trình mô phỏng một trạm phục vụ cafe. Trong đó, mô hình robot dịch vụ là một mô hình được thiết kế có chuyển động động học 6 bậc tự do như các robot công tác phổ biến. Trong chương này, các mô hình đó sẽ được tích hợp như trong hình 3.1.

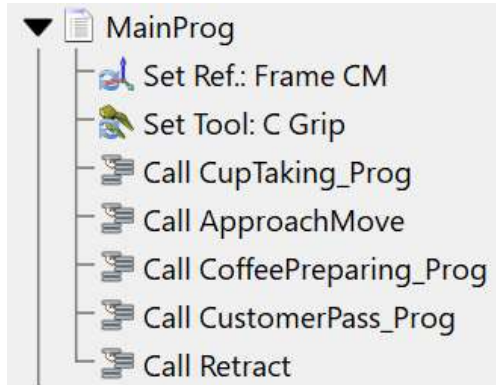


Hình 3.1: Mô hình trạm pha cafe dùng robot

3.1 Thiết kế trình tự pha chế bằng robot

Hoạt động của một robot dịch vụ cafe được thiết kế là một điều khiển trình tự gồm nhiều bước đi qua các vị trí được xác định trong không gian làm việc của robot, và được chia thành 4 chương trình con chính và 1 chương trình con phụ trợ bao gồm: chương trình con lấy ly (CupTaking_Prog); Chương trình con tiếp cận máy pha cafe (ApproachMove); chương trình con pha chế cafe (CafePreparing_Prog); chương trình con đưa ly vào bộ cho khách hàng

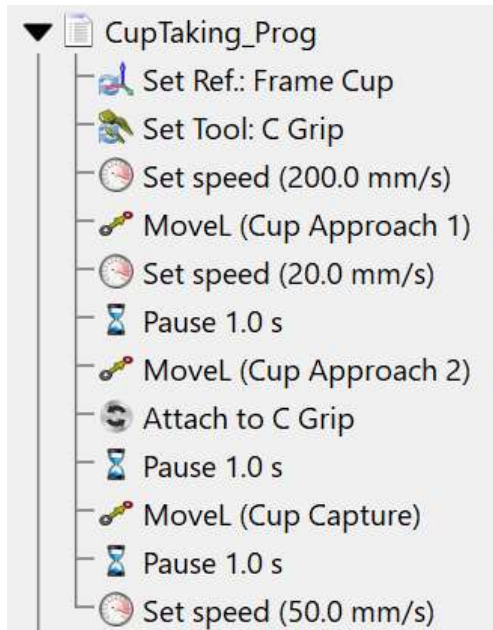
(CustomerPass_Prog). Chương trình chính gọi các chương trình con trên được cho trên hình 3.2.



Hình 3.2: Chương trình chính thực hiện thao tác pha chế cafe

3.2 Thao tác lấy ly

Chồng ly được giả định úp ngược, ly có dạng hình côn (phía dưới nhỏ hơn phía trên) và có một khoảng cách nhất định giữa các ly. Thao tác lấy ly được đơn giản hóa bao gồm thao tác đưa tay kẹp vào vị trí phía dưới ly và di chuyển thẳng phía trên để lấy ly ra khỏi chồng ly. Để thực hiện thao tác, cần có 3 target mà robot cần đi qua: vị trí tiếp cận chồng ly (Cup Approach 1), vị trí tiếp cận ly (Cup Approach 1) và vị trí thoát ra khỏi chồng ly (Cup Capture). Chương trình con thực hiện lấy ly được cho trong Hình 3.3a và chương trình con tiếp cận máy pha chế cho trên hình 3.3b. Giải thích chương trình lấy ly được cho trên Bảng 3.1.

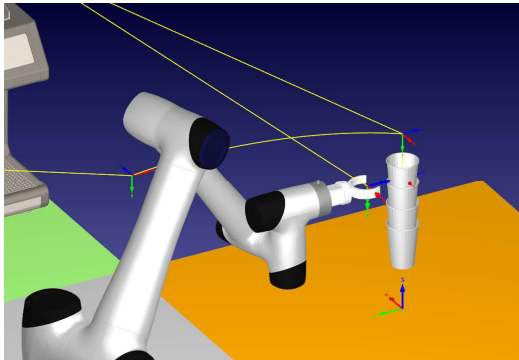


(a)

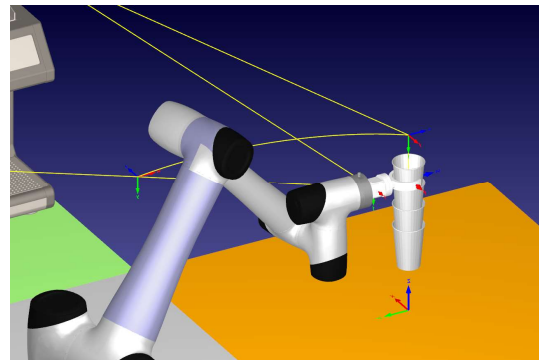


(b)

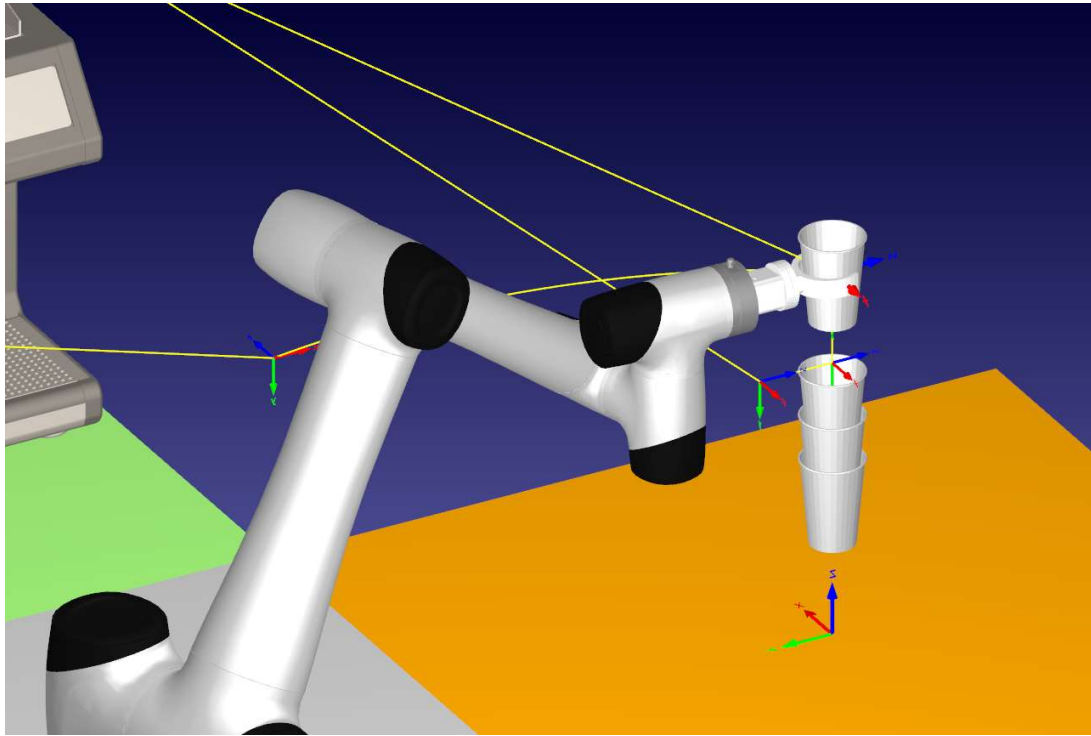
Hình 3.3: (a) Chương trình con thực hiện thao tác lấy ly, (b) Chương trình con thực hiện tiếp cận máy pha chế cafe (vì chuyển hệ tọa độ tham chiếu)



Hình 3.4a: Vị trí tiếp cận chông ly 1






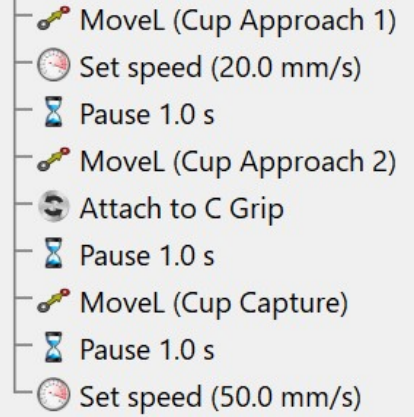
Hình 3.4b: Vị trí tiếp cận ly 2



Hình 3.4c: Vị trí lấy ly ra khỏi chồng

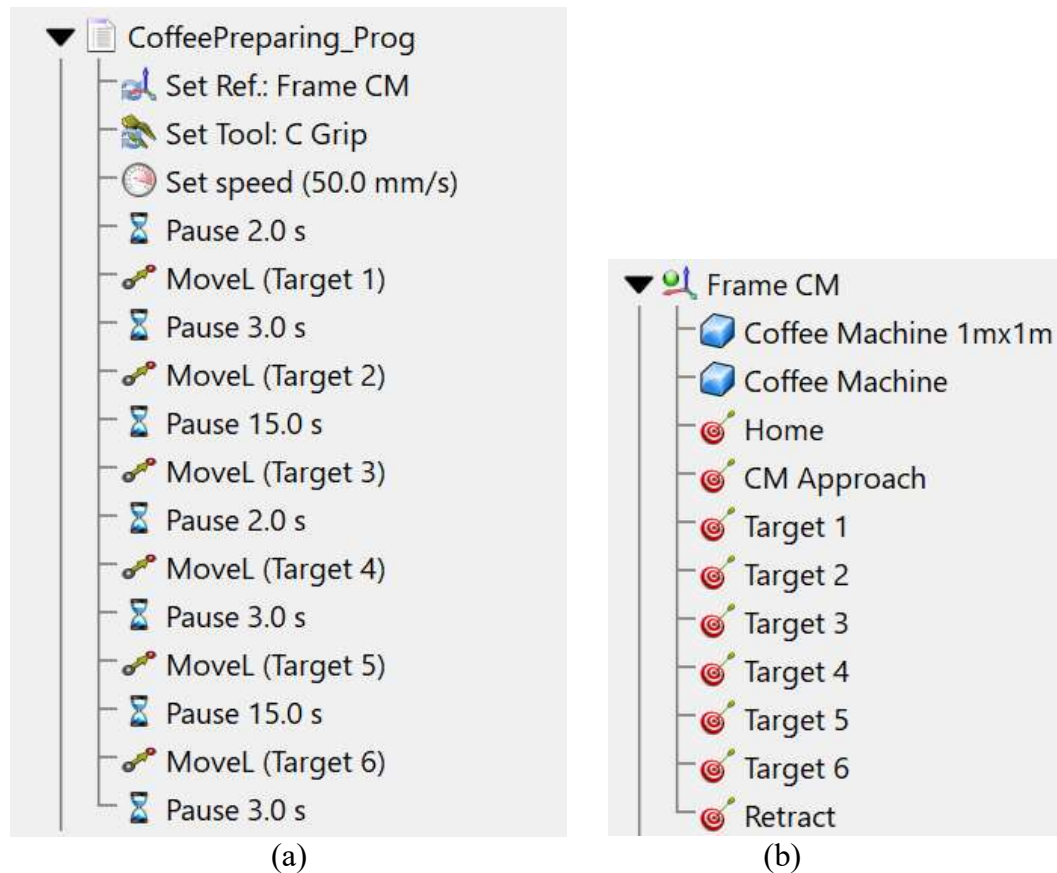
Bảng 3.1: Giải thích chương trình con thao tác lấy ly ra khỏi chồng

1	 Set Ref.: Frame Cup
<p>Sử dụng hệ tọa độ tham chiếu hiện hành có tên “Frame Cup”. Đây là hệ tọa độ gắn với kệ cấp ly.</p>	
2	 Set Tool: C Grip
<p>Sử dụng hệ tọa độ công cụ có tên “C Grip” gắn ở đầu cuối robot. Các lệnh MOVL sẽ di chuyển hệ tọa độ Tool này trùng khớp với tọa độ Target tương ứng</p>	
3	 Set speed (200.0 mm/s)

<p>Cài đặt vận tốc và gia tốc cho Tool gắn trên đầu robot. Vận tốc được cài đặt nhanh hơn bình thường là 200mm/s.</p>	
4	
<p>Lệnh dịch chuyển Tool đi qua một tập hợp các tọa độ Target đã được xác định trước tạo thành một thao tác lấy một cái ly bao gồm: Vị trí tiếp cận chông ly (Hình 3.4a); Vị trí tiếp cận ly 2 (Hình 3.4b); Vị trí lấy ly ra khỏi chông (Hình 3.3c).</p>	
<p> </p>	
<p> </p>	



3.3 Thao tác pha chế cafe















Thao tác pha chế cafe được đơn giản hóa bao gồm thao tác châm cafe và thao tác châm kem sữa. Để thực hiện thao tác này, cần có 6 target mà robot cần đi qua. Ngoài ra, còn có 2 vị trí đặc biệt là vị trí tiếp cận máy pha cafe (CM Approach) và vị trí thoát ra khỏi máy pha cafe an toàn (Retract). Chương trình con thực hiện pha chế được cho trong Hình 3.5a và hệ tọa độ máy pha cafe cho trên hình 3.5b. Giải thích chương trình cho trên Bảng 3.2

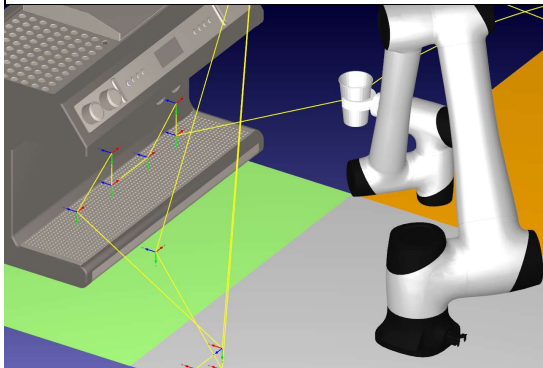


Hình 3.5: (a) Chương trình con thực hiện thao tác pha chế cafe, (b) Định nghĩa hệ trục tọa độ cho máy pha cafe (Cafe Machine, CM)

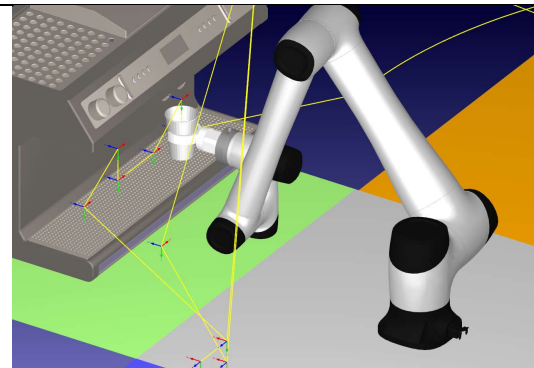
Bảng 3.2: Giải thích chương trình con thao tác pha chế cafe

1	 Set Ref.: Frame CM
<p>Sử dụng hệ tọa độ tham chiếu hiện hành có tên “Frame CM”. Đây là hệ tọa độ gắn với máy pha chế cafe. Các lệnh MoveL theo sau trong đoạn chương trình là đối với hệ tọa độ này.</p>	
2	 Set Tool: C Grip
<p>Sử dụng hệ tọa độ công cụ có tên “C Grip” gắn ở đầu cuối robot.</p>	

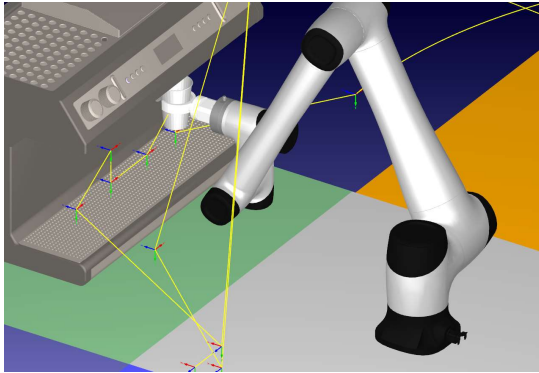
3	
<p>Cài đặt vận tốc và gia tốc cho Tool gắn trên đầu robot. Vận tốc của robot thay đổi sang tốc độ chậm là 50mm/s vì robot đang mang ly có cafe bên trong.</p>	
4	<ul style="list-style-type: none">  Pause 2.0 s  MoveL (Target 1)  Pause 3.0 s  MoveL (Target 2)  Pause 15.0 s  MoveL (Target 3)  Pause 2.0 s  MoveL (Target 4)  Pause 3.0 s  MoveL (Target 5)  Pause 15.0 s  MoveL (Target 6)  Pause 3.0 s
<p>Lệnh dịch chuyển Tool đi qua một tập hợp các tọa độ Target đã được xác định trước tạo thành một thao tác pha chế cafe bao gồm: Vị trí tiếp cận máy pha cafe (Hình 3.6a); Vị trí rót cafe 1 (Hình 3.6b); Vị trí rót cafe 2 (Hình 3.6c); Vị trí trung gian (Hình 3.6d); Vị trí rót kem sữa 1 (Hình 3.6e); Vị trí rót kem sữa 2 (Hình 3.6f); Vị trí an toàn ra khỏi máy pha cafe (Hình 3.6g).</p>	



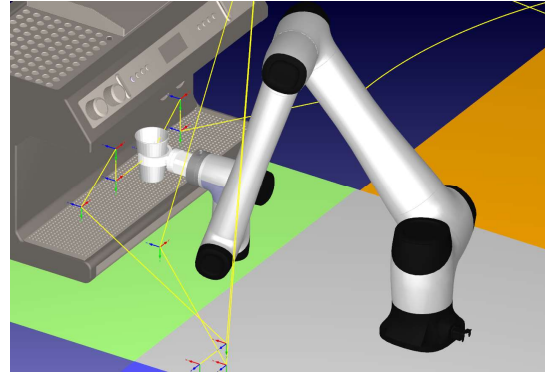
Hình 3.6a: Tiếp cận máy pha cafe



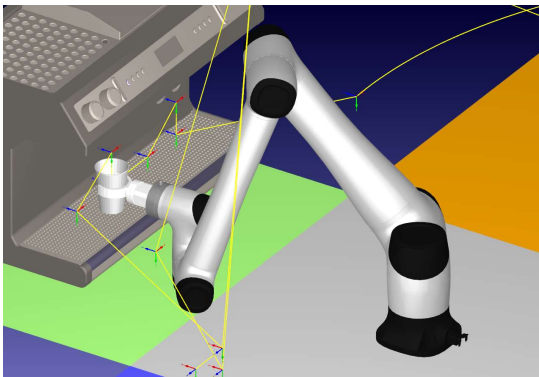
Hình 3.6b: Vị trí rót cafe 1



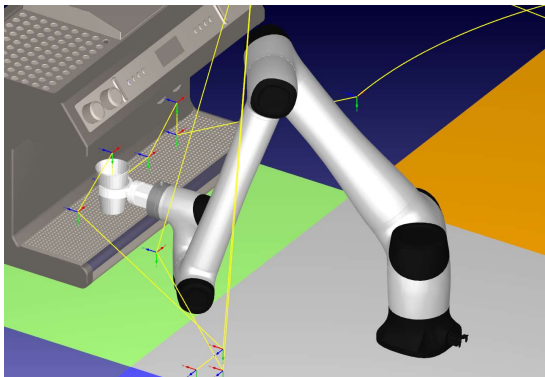
Hình 3.6c: Vị trí rót cafe 2



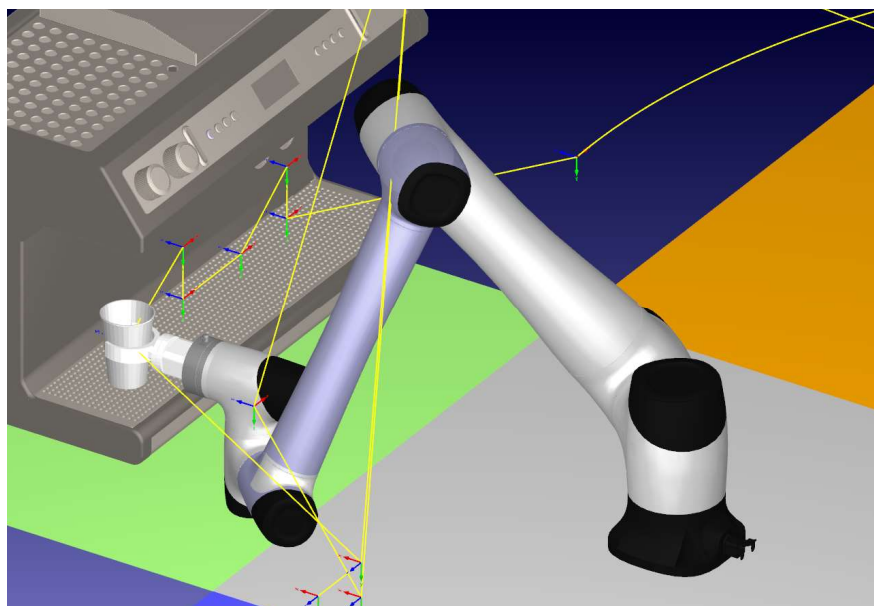
Hình 3.6d: Vị trí trung gian



Hình 3.6e: Vị trí rót kem sữa 1



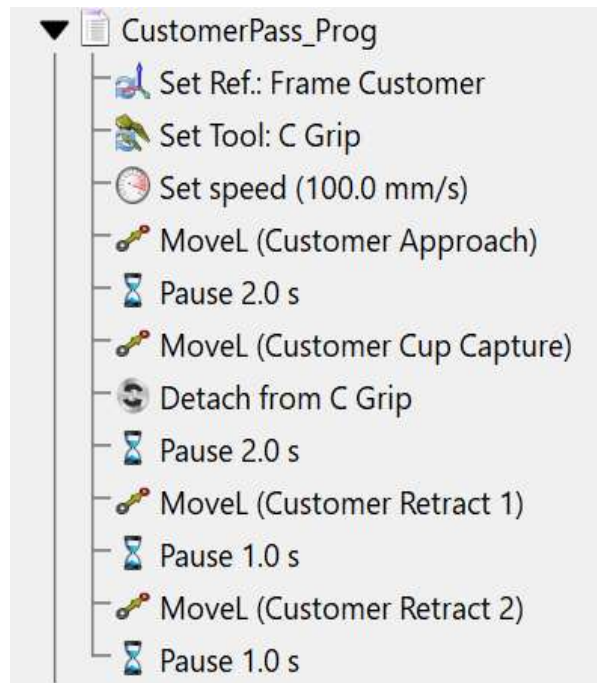
Hình 3.6f: Vị trí rót kem sữa 2



Hình 3.6g: Vị trí an toàn ra khỏi máy pha cafe



3.4 Thao tác đưa khách hàng











Thao tác đưa cafe cho khách hàng được đơn giản hóa bao gồm thao tác đưa ly đến bộ giao cafe và thao tác đặt ly cafe xuống bộ cho khách hàng. Để thực hiện thao tác này, cần có 4 target mà robot cần đi qua. Chương trình con thực hiện thao tác đưa khách hàng được cho trong Hình 3.7 . Giải thích chương trình cho trên Bảng 3.3

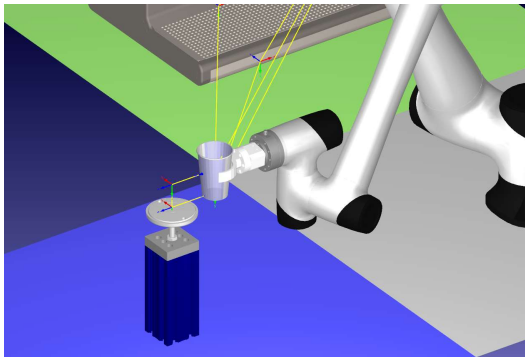


Hình 3.7: Chương trình con thực hiện thao tác đưa cafe cho khách hàng

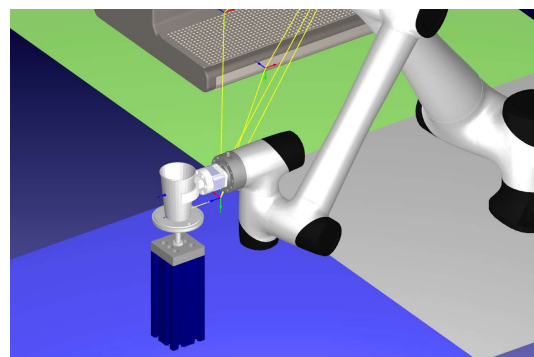
Bảng 3.3: Giải thích chương trình con thao tác đưa cafe cho khách hàng

1	 Set Ref.: Frame Customer
Sử dụng hệ tọa độ tham chiếu hiện hành có tên “Frame Customer”. Đây là hệ tọa độ gắn với bộ đặt ly cho khách hàng.	
2	 Set Tool: C Grip
Sử dụng hệ tọa độ công cụ có tên “C Grip” gắn ở đầu cuối robot.	

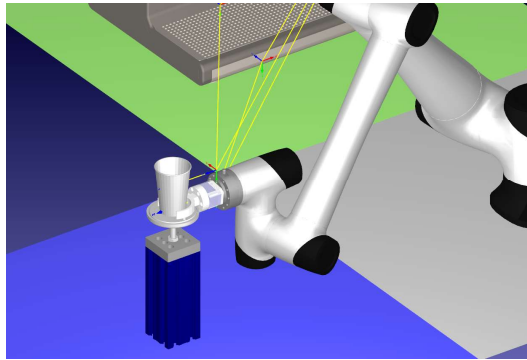
3	 Set speed (100.0 mm/s)
<p>Cài đặt vận tốc và gia tốc cho Tool gắn trên đầu robot. Vận tốc của robot thay đổi sang tốc độ trung bình là 100mm/s vì đang mang ly có cafe bên trong và khoảng dịch chuyển nhỏ.</p>	
4	<ul style="list-style-type: none">  MoveL (Customer Approach)  Pause 2.0 s  MoveL (Customer Cup Capture)  Detach from C Grip  Pause 2.0 s  MoveL (Customer Retract 1)  Pause 1.0 s  MoveL (Customer Retract 2)  Pause 1.0 s
<p>Lệnh dịch chuyển Tool đi qua một tập hợp các tọa độ Target đã được xác định trước tạo thành một thao tác mang ly cafe lên bệ bao gồm: Vị trí tiếp cận bệ 1(Hình 3.8a); Vị trí đặt ly lên bệ (Hình 3.8b); Vị trí tiếp cận bệ 2 (Hình 3.6c); Vị trí tiếp cận bệ 3 (Hình 3.6d)</p>	



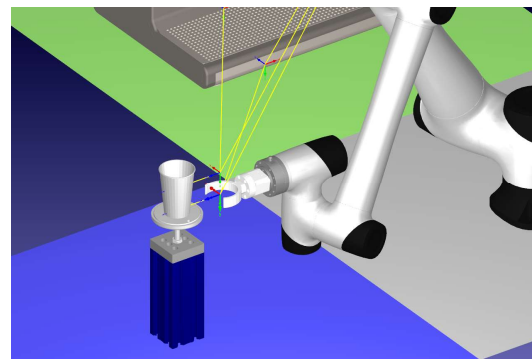
Hình 3.8a: Vị trí tiếp cận bệ 1



Hình 3.8b: Vị trí đặt ly lên bệ

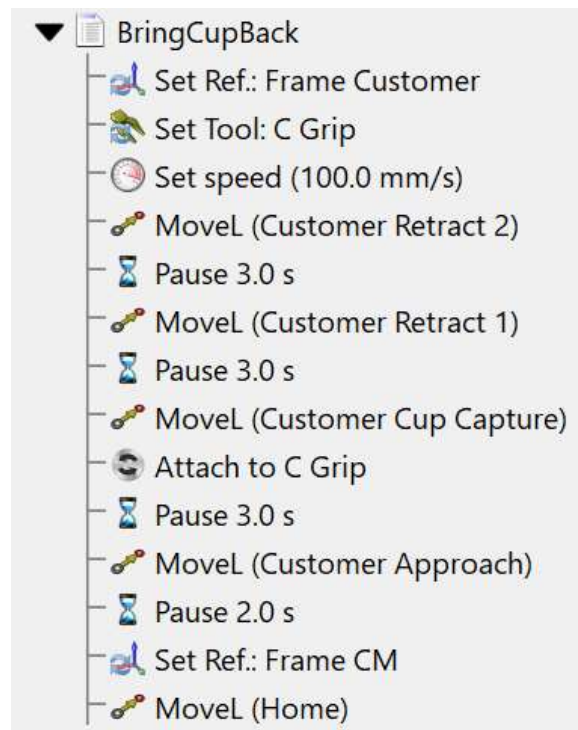


Hình 3.8c: Vị trí tiếp cận bộ 2



Hình 3.8d: Vị trí tiếp cận bộ 3

Ngoài ra, để thuận tiện cho việc đặt lại vị trí ban đầu để tiếp tục cho chu trình mới, một chương trình con BringCupBack. Chương trình con được lập trình dựa trên các Target đã được xác định trong các thao tác trước đó. Chương trình con được trình bày trong hình 3.9.



Hình 3.9: Chương trình con thực hiện mang ly trở về vị trí ban đầu

3.5 Giao tiếp API RoboDK dùng C#

RoboDK API dùng C# là một tập tin nguồn RoboDK.cs có chứa RoboDK class, RoboDK.Item class, và một số công cụ khác về robotics như tính toán ma trận RoboDK.Mat với các phép biến đổi không gian 3D.

Khai báo đối tượng RoboDK để sử dụng trong chương trình C#

```
// RDK holds the main object to interact with RoboDK.
```

```
// The RoboDK application starts when a RoboDK object is created.
```

```
private IRoboDK _rdk;
```

3.5.1 Tính năng “Simulation”

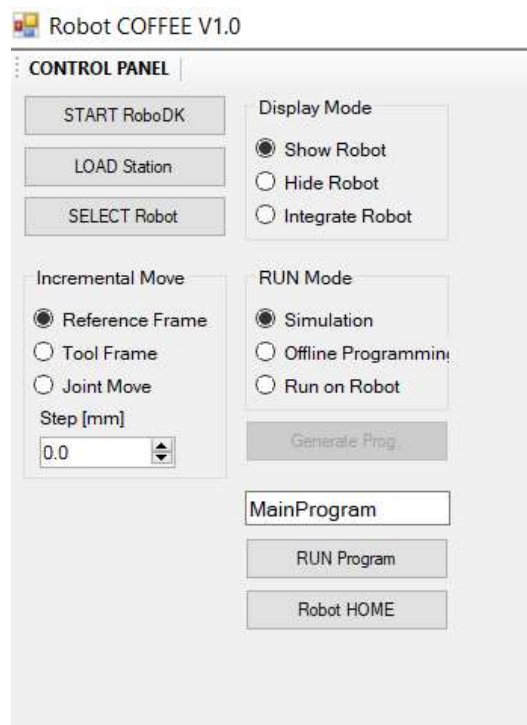
Chương trình C# được mô phỏng khi dùng các lệnh như MoveJ, MoveL hay setDO).

```
//Quan trọng: dùng các chương trình RoboDK trước đó (nếu chọn programming mode)
```

```
_rdk.Disconnect();
```

```
// Chuyển sang simulation mode:
```

```
_rdk.SetRunMode();
```



Hình 3.10: Giao diện Control Panel các tính năng cơ bản RoboDK API

3.5.2 Tính năng “Offline Programming”

```
// retrieve the newly added item
var item = _rdk.AddFile(selectFile.FileName);
if (item.Valid())
{
    notifybar.Text = $"Loaded: {item.Name()}";
    // attempt to retrieve the ROBOT variable (a robot available in the station)
    SelectRobot();
}
else
{
    notifybar.Text = $"Could not load: {filename}";
}
}
```

3.5.3 Tính năng “Run on Robot”

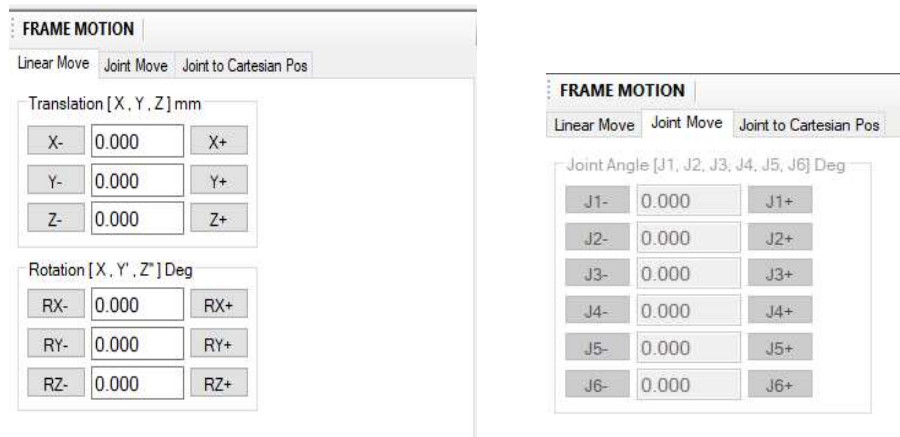
Chương trình C# có thể được thực thi trực tiếp trên robot sử dụng cùng mã lệnh C# được dùng cho mô phỏng. Lập trình online cho phép chạy một chương trình trên một bộ điều khiển robot xác định dùng các Robot Drivers. Các bước sau kiểm tra tính năng này như sau:

1. Chọn **Run on Robot**
2. Chọn bất kỳ lệnh nào làm chuyển động robot

```
//Important: stop any previous program generation
_rdk.Disconnect();

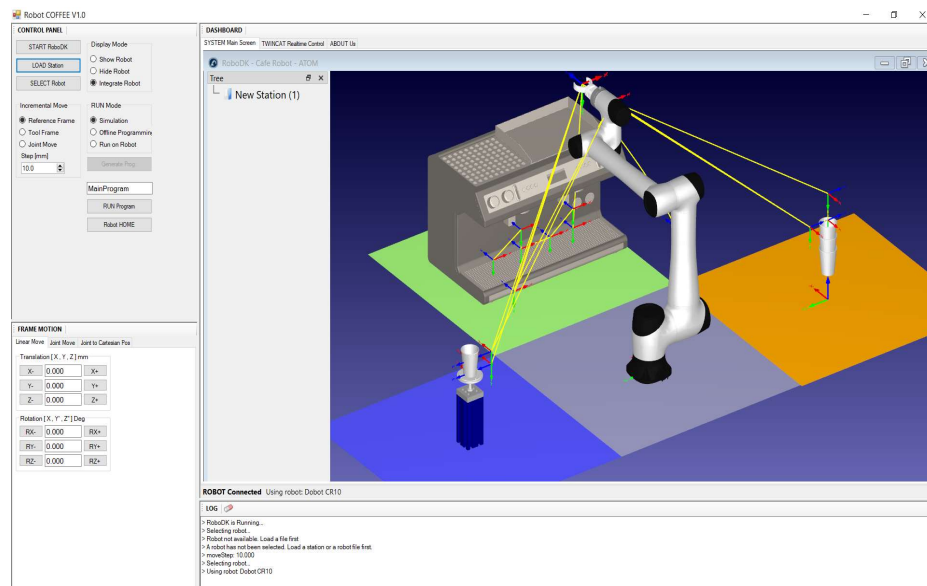
// Connect to real robot
if (_robot.Connect())
{
    // Set to Run on Robot robot mode:
    _rdk.SetRunMode(RunMode.RunRobot);
}
}
```

3.5.4 Tính năng chuyển động



Hình 3.11: Giao diện điều khiển chuyển động Tool và khớp robot

Các tính năng API được tích hợp trên một phần mềm dùng C# được cho trên Hình 3.12.



Hình 3.12: Giao diện demo RoboDK API dùng C#

3.6 Tổng kết chương

Chương này trình bày việc lập trình cho robot dịch vụ thực hiện thao tác pha chế cafe. Các chương trình con thực hiện các bao gồm: (1) thao tác lấy ly, (2) thao tác tiếp cận máy pha cafe; (3) thao tác pha chế cafe; (4) thao tác đưa ly vào bệ cho khách hàng. Xây dựng một chương trình trung gian RoboDK API được sử dụng để

giao tiếp với chương trình mô phỏng và đồng thời sẽ được giao tiếp tiếp với phần cứng thật theo truyền thông realtime EtherCAT (được đề xuất trong trong phần hướng phát triển).

CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1 Kết luận

Đề tài bao gồm 4 chương thực hiện một mục tiêu chính, đó là, xây dựng một hệ điều hành robot tích hợp: bao gồm một phần mềm mô phỏng RoboDK và một phần mềm trung gian giao tiếp RoboDK API dùng C# để sẵn sàng cho các tính năng điều khiển realtime cho robot thực. Đề tài chọn ứng dụng robot dịch vụ cafe để minh họa cho quá trình nghiên cứu của mình. Các công việc thực hiện như sau:

- (1) Tìm hiểu robot cộng tác ứng dụng dịch vụ và phần mềm mô phỏng RoboDK
- (2) Mô hình hóa 3D robot cộng tác Dobot-CR10 và các thành phần tương ứng như mô hình máy pha cafe, mô hình cấp ly và mô hình bộ giao cafe.
- (3) Lập trình mô phỏng cho robot thực hiện pha chế một ly cafe : thao tác lấy ly cafe, thao tác pha chế cafe, thao tác đưa khách hàng.
- (4) Lập trình giao tiếp RoboDK API truy xuất các tính năng của RoboDK như load station, thực hiện các lệnh cơ bản, đọc các vị trí hiện hành của Tool, điều khiển vị trí tool đến vị trí mới.
- (5) Hệ thống được kiểm chứng bằng cách cho chương trình robot gọi chạy từ chương trình trung gian.

4.2 Hướng phát triển

Như đã trình bày trong mục 1.3, việc khảo sát bộ điều khiển của robot cộng tác với nền tảng realtime EtherCAT, đề tài thực hiện việc giao tiếp RoboDK API. Đây là bước quan trọng nhất của phần mềm trung gian để tiếp tục thực hiện bước điều khiển cấp thấp với robot thật để hiện thực hệ điều hành robot chuyên dùng cho nghiên cứu và phát triển thuật toán sau này. Đây là giải pháp cho bài toán điều khiển cấp thấp nhất, đó là cấp điều khiển các cơ cấu chấp hành thông minh (smart actuators). Các hướng phát triển của đề tài như sau:

- Kết nối hệ thống thiết kế quỹ đạo cho robot RoboDK vào hệ thống thu nhận dữ liệu để phòng kỹ thuật có thể thiết kế và truyền chương trình robot xuống khu vực hiện trường khi sử dụng robot đó cho một dịch vụ khác

- Lập trình tính năng điều khiển cấp thấp dùng truyền thông EtherCAT để điều khiển robot thật, sử dụng hiệu quả dữ liệu vị trí trên phần mềm mô phỏng RoboDK.

- .Lập trình tính năng khảo sát động lực học cho robot cộng tác phục vụ cho việc nghiên cứu tính năng quan trọng là phát hiện va chạm trên robot cộng tác.

- Lập trình giao tiếp với cảm biến và các cơ cấu chấp hành khác để hoàn thiện phần mềm Hệ điều hành robot.

- Lập trình hệ điều hành robot giao tiếp với bộ điều khiển camera là một tính năng không thể thiếu đối với robot cộng tác


DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Ramkumar Gandhinathan, Lentin Joseph, ROS Robotics Projects, Second Edition, 2019
- [2] Giuseppe Carbone, Marco Ceccarelli and Doina Pisla, *New Trends in Medical and Service Robotics - Advances in Theory and Practice*, Springer Nature Switzerland, 1st Edition, 2019.
- [3] Chung Tan Lam and Truong Trong Toai, “Development of a Collaborative Robot – VietCobot,” *Journal of Science and Technology on Information and Communications*,” CS.01 2021
- [4] Robot dịch vụ, [Online]. <https://maysanxuattudong.com/robot-phuc-vu/>
- [5] Robot Ella ngành dịch vụ F&B, cafe, [Online]. <https://www.crowndigital.io/>
- [6] Robot cộng tác Techman, [Online]. <https://www.tm-robot.com/en/>
- [7] Hệ thống điều khiển robot Denso, [online]. <https://www.densorobotics.com/>
- [8] Robot cộng tác Dobot, [Online]. <https://www.dobot.cc/>
- [9] Phần mềm RoboDK, [Online]. <https://robodk.com/doc/en/Getting-Started.html>

PHỤ LỤC

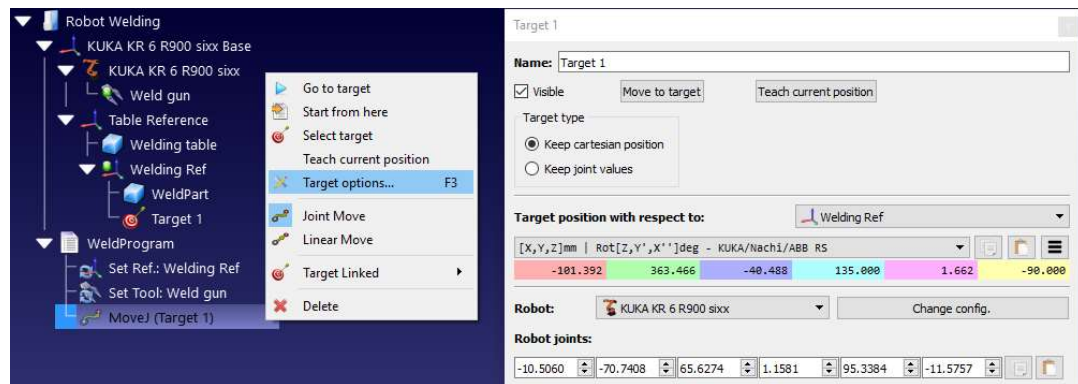
PHỤ LỤC P1: Tập lệnh chính của RoboDK

P1.1 Lệnh Joint Move

Chọn Program →  **Move Joint Instruction** để thêm một lệnh chuyển động dựa trên chuyển động của khớp quay robot.


Trừ khi một Target được chọn trước lệnh này sẽ tạo một Target mới liên kết với lệnh. Nếu Target di chuyển, sự chuyển động của nó cũng được hiệu chỉnh theo.

Nếu là lệnh này được thêm vào một Program đầu tiên thì có thêm 2 lệnh được thêm vào trước lệnh này: đó là lệnh chọn hệ tọa độ tham chiếu (Reference Frame) và lệnh chọn hệ tọa độ dụng cụ (Tool Frame). Điều đó bảo đảm khi chương trình thực hiện lệnh chuyển động này thì robot có cùng hệ tọa độ tham chiếu và hệ tọa độ dụng cụ.



Hình P1.1: Lệnh Joint Move và truy xuất Target

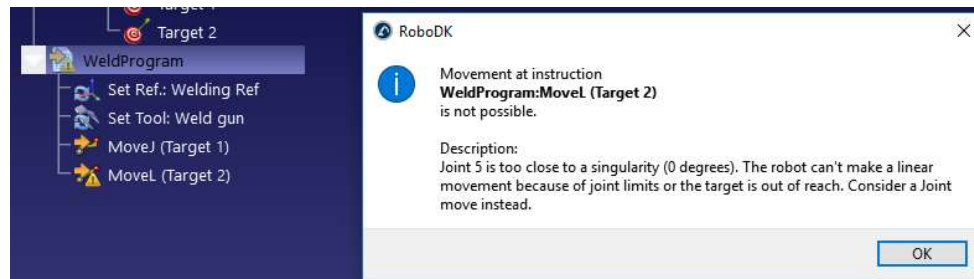
P1.2 Lệnh Linear Move

Chọn Program →  **Move Linear Instruction** để thêm một lệnh chuyển động thẳng cho dụng cụ Tool.

Chuyển động khớp và chuyển động thẳng giữa 2 Target có cùng hành vi chuyển động chỉ khác không gian khảo sát nên có thể chuyển đổi qua lại với nhau.

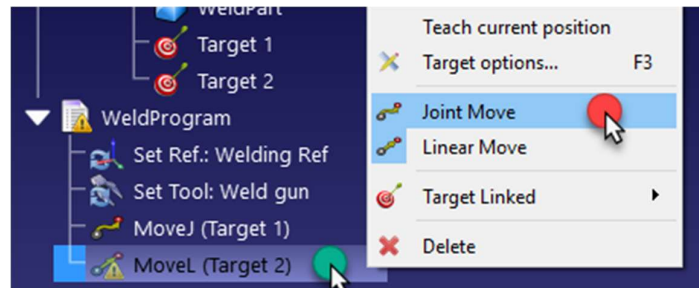
Lưu ý: ta cần giữ chuyển động đầu tiên của chương trình là một chuyển động khớp dùng Joint Target. Việc này sẽ thiết lập được cấu hình đúng như mong muốn ngay từ chuyển động đầu tiên và đảm bảo robot thật di chuyển giống với mô phỏng.

Trái với chuyển động khớp, chuyển động thẳng là có tính đến các điểm kỳ dị của robot và giới hạn trục quay. Ví dụ, robot 6 trục không thể vượt qua một điểm kỳ dị bằng một chuyển động thẳng. Hình bên dưới là một ví dụ về báo lỗi điểm kỳ dị.




Hình P1.2: Báo lỗi điểm kỳ dị

Nếu chuyển động thẳng không thực sự cần thiết, ta bấm chuột phải vào lệnh MoveL và chọn sang Joint Move. Cách khác là chọn Target, TCP (Tool Center Point) hay là vị trí trong hệ tọa độ tham chiếu, phải được hiệu chỉnh để tránh điểm kỳ dị đó.



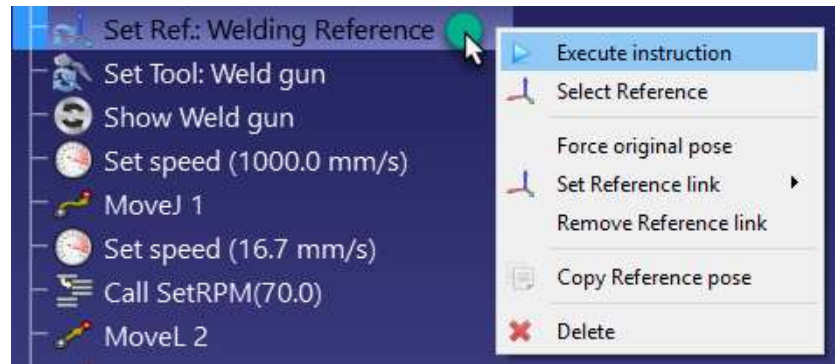
Hình P1.3: Chuyển Linear Move sang Joint Move.

P1.3 Lệnh Set Reference Frame

Chọn Program →  *Set Reference Frame Instruction* để thiết lập hiện hành một hệ tọa độ tham chiếu. Tác vụ này cập nhật hệ tọa độ hiện hành trên bộ điều khiển cho các lệnh chuyển động theo sau và thay đổi hệ tọa độ tham chiếu đang dùng (Active) của robot trong RoboDK cho mục đích mô phỏng. Việc chọn này sẽ


tạo ra các lệnh chuyển động đến các Target xác định (Target trong hệ Decart) đối với tập hợp hệ tọa độ tham chiếu trước đó.

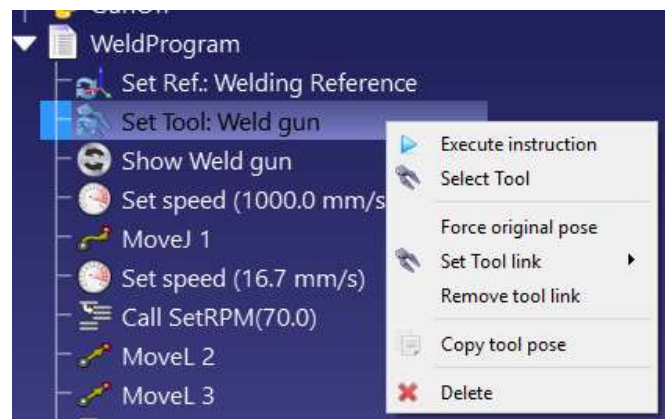
Hệ tọa độ tham chiếu không có tên cố định, nó còn được gọi là Work Object (robot ABB), UFRAME (robot Fanuc), FRAME (đối với robot Motoman) hoặc \$BASE (đối với robot KUKA).



Hình P1.4: Đặt hệ tọa độ tham chiếu hiện hành


P1.4 Lệnh Set Tool Frame

Chọn Program →  *Set Tool Frame Instruction* để thiết lập hiện hành một hệ tọa độ công cụ (TCP - Tool Center Point). Lệnh này sẽ cập nhật lại hệ tọa độ công cụ hiện hành trong chương trình cho các lệnh chuyển động theo sau và thay đổi hệ tọa độ công cụ đang dùng (Active) của robot trong RoboDK cho mục đích mô phỏng. Có nghĩa là các lệnh chuyển động đến các Target sẽ được thi hành dùng hệ tọa độ dụng cụ mới được cập nhật.



Hình P1.5: Thiết lập hệ tọa độ công cụ.

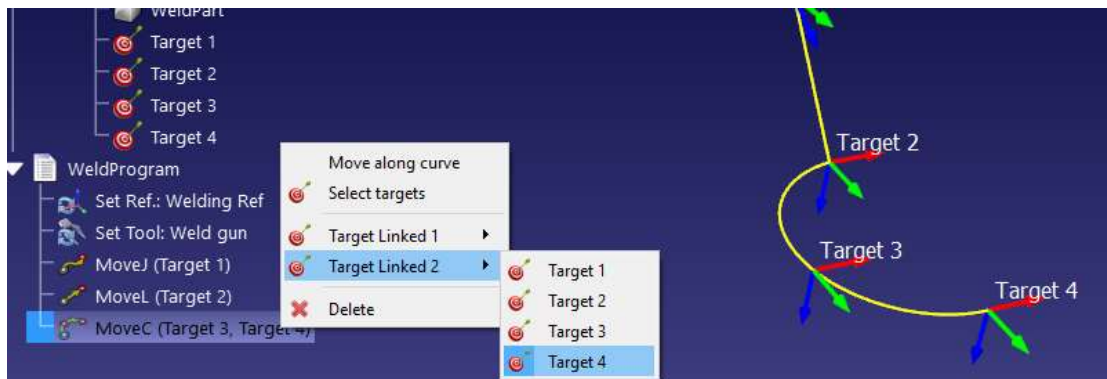
P1.5 Lệnh Circular Move

Chọn Program →  **Move Circular Instruction** để thêm một chuyển động dạng cung tròn cho công cụ Tool.

Ngoài hai Target được chọn trước khi thêm lệnh, lệnh này sẽ không tạo ra Target mới. Ta cần thêm hai Target riêng biệt nữa và liên kết chúng từ lệnh di chuyển cung tròn, như trong hình ảnh bên dưới.


Đường cong là một cung tạo từ điểm robot đang dừng, đi qua điểm đầu tiên điểm cong đầu tiên (Target Linked 1) và kết thúc tại điểm cuối (Target Link 2).

Chú ý: không thể thực hiện 1 vòng tròn chỉ với một lệnh cung tròn. Một vòng tròn phải được chia thành 2 chuyển động cung tròn tách biệt.



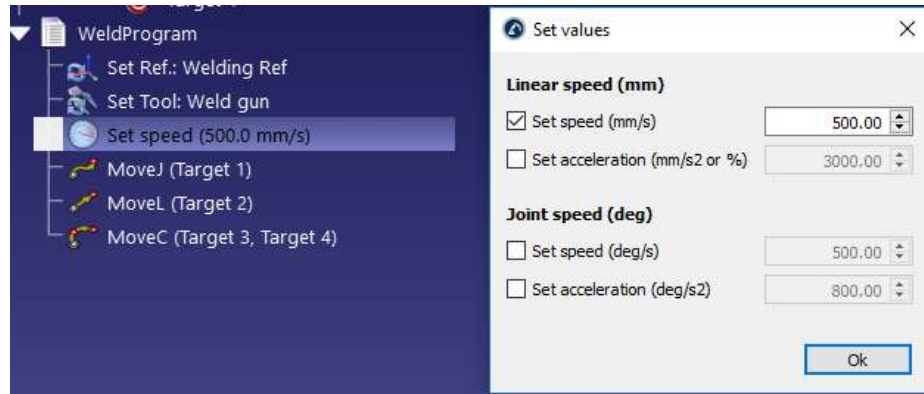
Hình P1.6: Lệnh Circular Move

P1.6 Lệnh Set Speed

Chọn Program →  **Set Speed Instruction** để thêm lệnh thay đổi vận tốc và/hoặc gia tốc. Ta có thể chỉ định vận tốc và gia tốc trong không gian khớp và trong không gian Cartesian.


Kích hoạt áp đặt vận tốc và gia tốc xác định trong chương trình. Vận tốc của robot thay đổi khi lệnh được thi hành. Vận tốc của robot cũng có thể thay đổi trên menu parameters: double click vào robot, chọn **paramters**.

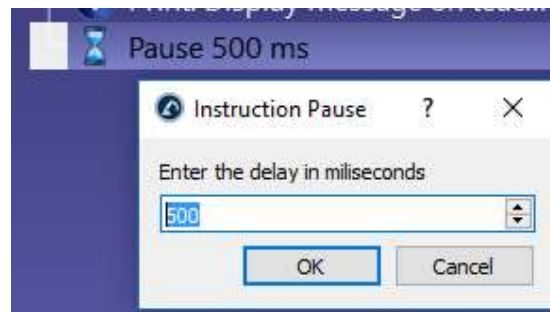
Không phải tất cả các bộ điều khiển robot đều hỗ trợ thiết lập gia tốc chính xác. Thiết lập tốc độ phù hợp là rất quan trọng để tính toán chính xác thời gian chương trình (circle time).



Hình P1.7: Cài đặt giá trị vận tốc

P1.7 Lệnh Pause


Chọn Program →  *Pause Instruction* để thêm một lệnh tạm dừng việc thực thi chương trình trong một khoảng thời gian xác định đến khi người vận hành muốn tiếp tục chương trình.



Hình P1.8: Lệnh Pause

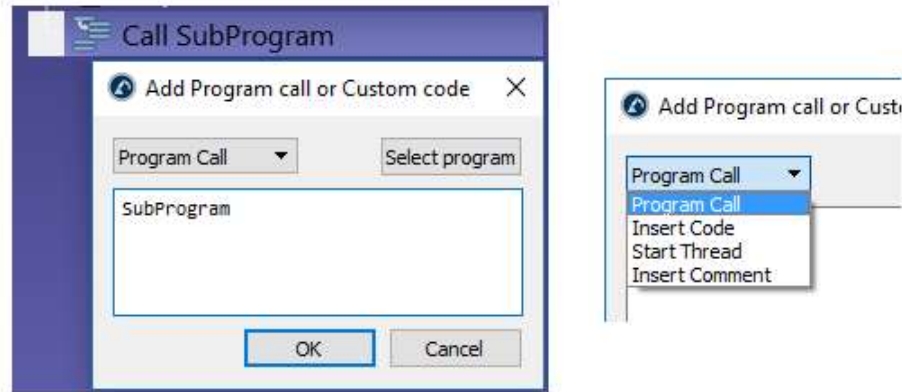
Lưu ý: nhập giá trị -1 để tạm dừng chương trình đến khi người vận hành muốn tiếp tục chương trình. Khi đó, lệnh được đặt tên tự động là *Stop*. Trong mô phỏng, 5 giây tạm dừng sẽ mất 1 giây để mô phỏng đối với tỷ lệ mô phỏng mặc định là 5.

P1.8 Lệnh Program Call

Chọn Program →  *Program Call Instruction* để gọi chương trình con từ chương trình hiện hành.

Mặc định là gọi một chương trình con xác định. Tuy nhiên, ta có thể dùng tính năng **Insert Code** để nhập lệnh tại vị trí của lệnh này thay cho việc gọi chương trình con. Nó có thể thuận tiện đối với một ứng dụng cụ thể và một bộ điều khiển cụ thể.

Ta có thể nhập nhiều dòng để tự động thiết lập nhiều lệnh gọi chương trình trên 1 hàng.



Hình P1.9: Gợi chương trình con

Chuyển từ **Program Call** sang **Start Thread** để kích hoạt việc gọi một chương trình con mà không ngắt chương trình chính. Trong trường hợp này, bộ điều khiển sẽ bắt đầu một Thread mới. Tùy chọn này chỉ dùng được cho một số bộ điều khiển nào đó và chỉ làm việc đối với một số tác vụ đặc thù.

P1.9 Lệnh Set/Wait IO

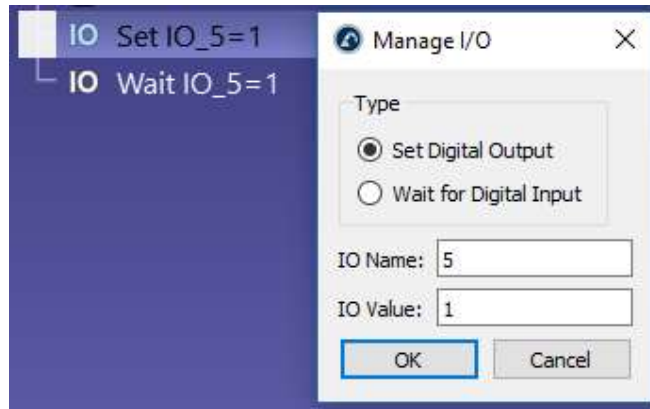
Chọn Program →  **Set or Wait I/O Instruction** để kích hoạt các cổng ra Digital Output (DO).

Mặc định, lệnh này tùy chọn là **Set Digital Output**. Lệnh này cũng có thể chờ cho cổng Digital Input.

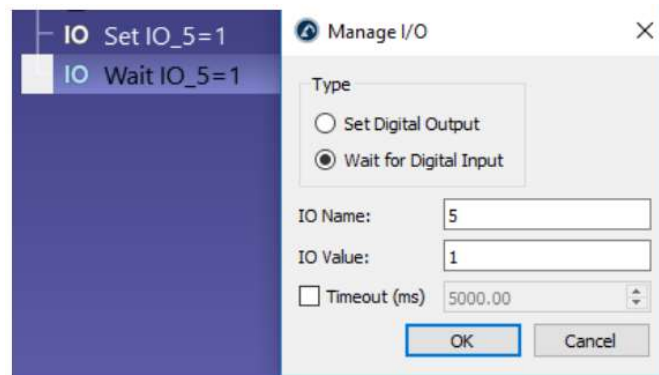
IO Name có thể là một con số hoặc giá trị text nếu nó là tên của một biến (var). **IO Value** có thể là một con số (0 cho False và 1 cho True) hoặc một giá trị text nếu nó là tên của một trạng thái (state).

Tùy chọn **Wait for Digital Input** để dừng thi hành chương trình chính chờ đến khi một cổng digital input thay đổi trạng thái của nó. Hơn nữa, một số bộ điều khiển robot có tính năng *timeout delay* để đưa ra một cảnh báo lỗi nếu thời gian chờ vượt

quá giá trị timeout đã xác định. Check vào tùy chọn **Timeout (ms)** để kích hoạt tính năng này.



Hình P1.10: Thiết lập I/O




Hình P1.11: Thiết lập Timeout (ms)

Lưu ý: khi thực hành, chẳng hạn nếu robot có gắn phần cứng chuyên dùng như bộ kẹp (gripper) hay trục phay (spindle), để kích hoạt các phần cứng này dùng cổng Digital Output, sau đó chờ cổng Digital Input (DI) để chuyển trạng thái.

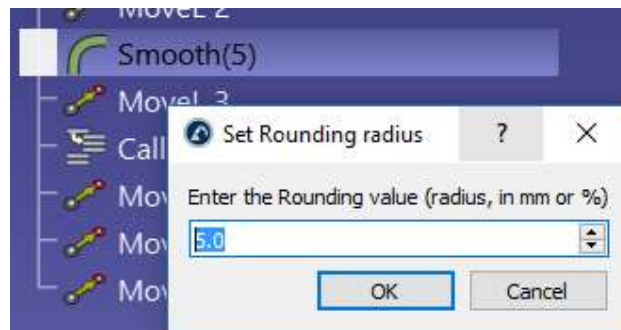
Sự thay đổi trạng thái mô phỏng của các DI và DO sẽ tạo mới các biến **station**. Để kiểm tra trạng thái của các biến này, nhấp chuột phải vào **station** và chọn **Station Parameters**. Ta cũng có thể đọc hoặc hiệu chỉnh các biến này thông qua API.

P1.10 Lệnh Set Rounding

Chọn Program →  **Set Rounding Instruction** để thay đổi độ chính xác làm tròn. Độ chính xác làm tròn được sử dụng để làm mịn các cạnh giữa các chuyển

động liên tiếp nhau. Thay đổi này có hiệu lực từ thời điểm lệnh này được thực hiện trong một chương trình (giống như với tất cả các lệnh khác), nên thường lệnh này được đặt ở đầu chương trình.

Nếu không có lệnh làm tròn, robot sẽ có tốc độ 0 khi kết thúc mỗi chuyển động (trừ khi chuyển động tiếp theo tiếp tuyến với chuyển động trước). Việc này sẽ gây ra gia tốc lớn và thay đổi vận tốc nhanh để đảm bảo độ chính xác tốt nhất cho mỗi chuyển động.

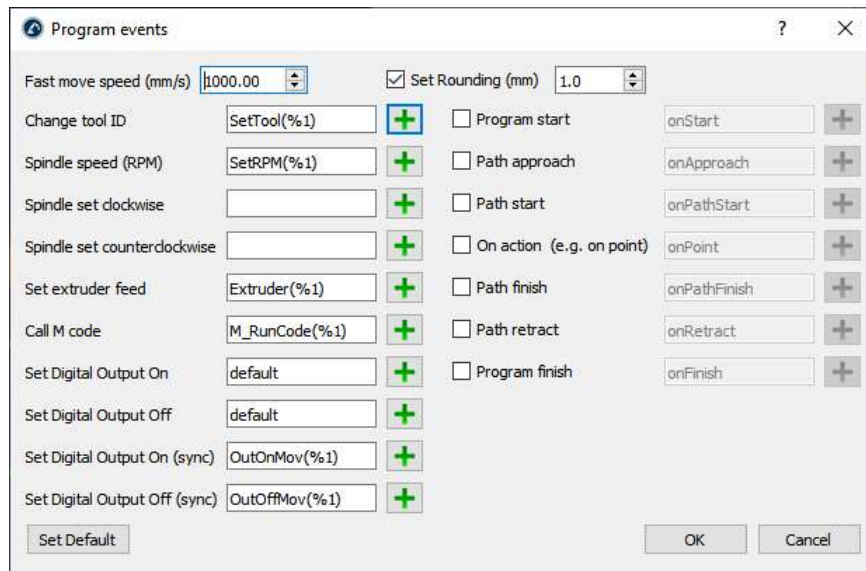


Hình P1.12: Đặt giá trị làm tròn

Lưu ý: giá trị làm tròn cao sẽ đảm bảo tốc độ không đổi trên đường đi của robot đổi lại việc mất độ chính xác trên các cạnh của đường đi đó. Tùy thuộc vào từng ứng dụng, người ta thường tìm ra sự thỏa hiệp tốt giữa độ chính xác và vận tốc làm tròn.

Một số bộ điều khiển yêu cầu đặt giá trị này dạng phần trăm, ví dụ bộ điều khiển Fanuc, nếu muốn dùng lệnh CNT5, ta nên nhập giá trị 5.


Ta cũng có thể ghi *rounding parameter* trong cửa sổ *Program Events* nếu ta đang lập trình để robot gia công, in 3D hoặc bám đường cong bám điểm.



Hình P1.13: Cửa sổ Program events

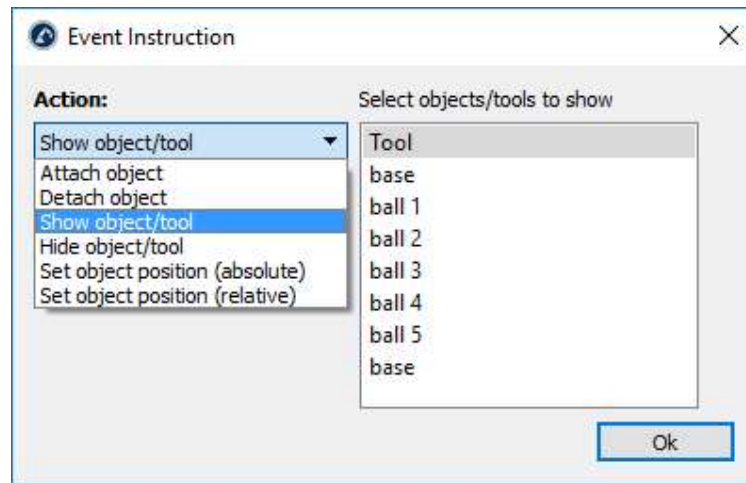
Kiểm tra *path accuracy* của RoboDK cho phép ta hiểu rõ hơn về ảnh hưởng của các giá trị làm tròn khác nhau.

P1.11 Lệnh Simulation Event

Chọn Program →  **Simulation Event Instruction** để kích hoạt một sự kiện trong mô phỏng. Các sự kiện không ảnh hưởng đến việc sinh mã chương trình và chỉ dùng để kích hoạt một sự kiện cụ thể trong mô phỏng.

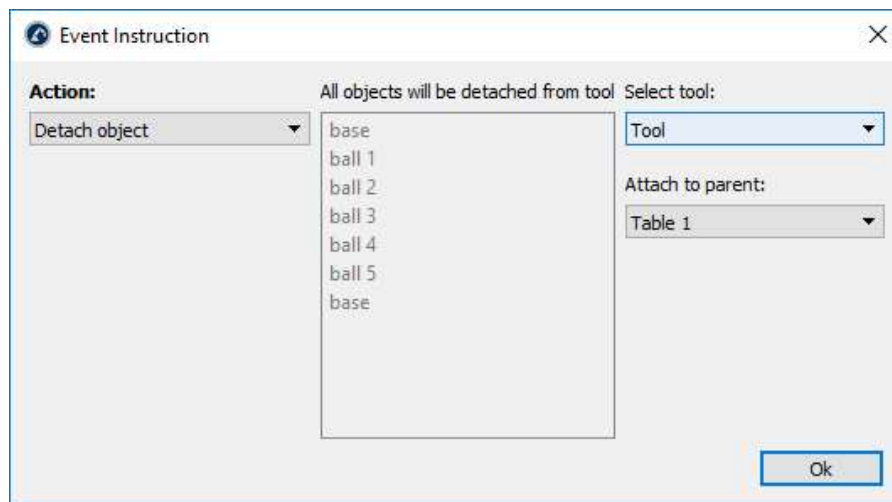
Các sự kiện cho phép:

- Gắn hoặc tách các object vào tool
- Hiện thị hoặc ẩn các object hoặc Tool
- Thay đổi vị trí các object và hệ tọa độ tham chiếu.



Hình P1.14: Lệnh sự kiện

Ví dụ nếu robot di chuyển đến một vị trí xác định để gấp một *object*, ta có thể thiết lập một sự kiện *Attact object* để di chuyển object đó gắn vào robot. Sau đó, sau khi robot di chuyển và bỏ *object* đó đi, ta có thể thiết lập sự kiện *Detach object* để rời khỏi *object* mà robot đã gấp.



Hình P1.15: Lệnh sự kiện Detach object

Lưu ý: Các sự kiện cũng có thể được mô phỏng bằng macro (cần có kinh nghiệm lập trình). Ví dụ dùng API, có thể làm cho các object xuất hiện ngẫu nhiên tại các vị trí xác định để mô phỏng tác vụ gấp đặt.

Khi gắn *object* vào *tool*, *object* gần nhất sẽ được gắn nếu nó không xa hơn một khoảng cách cho trước. Khoảng cách này theo mặc định là 200 mm và có thể thay đổi trong: Tools → Option → ***Maximum distance to attach an object to a robot tool***. Ngoài ra, theo mặc định, khoảng cách này được kiểm tra từ vị trí TCP đến *object* tham chiếu. Hoặc cách khác là có thể sử dụng khoảng cách giữa TCP và dạng hình học của đối tượng bằng cách chọn ***Check shortest distance between TCP and the object shape***

PHỤ LỤC P2: Lập trình và mô phỏng trên RoboDK

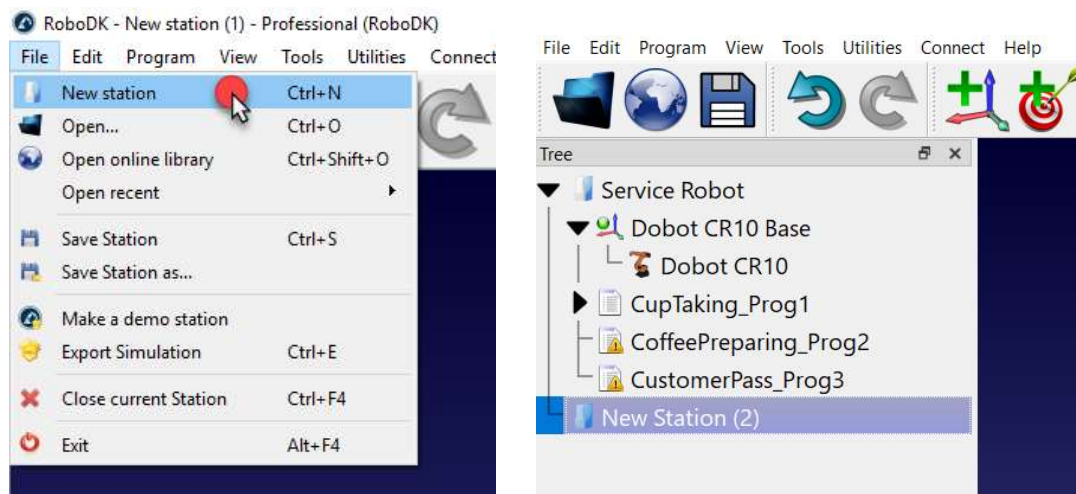
Lập trình offline là lập trình robot không phải tại hiện trường sản xuất. Lập trình offline hạn chế thời gian chết so với khi lập trình tại xưởng sản xuất.

Mô phỏng và lập trình offline cho phép khảo sát nhiều tình huống của một phân xưởng trang bị robot trước khi lắp đặt. Lỗi khi thiết kế phân xưởng có thể được dự đoán trước.

P2.1 Tạo Project mới

Tất cả robot, đối tượng (objects) và dụng cụ (tools) trong project được lưu dạng **RoboDK station** (gọi là tập tin RDK). Một **RoboDK station** chứa mọi thiết lập liên quan đến robot, dụng cụ, hệ các tọa độ tham chiếu, các targets, các đối tượng và các tham số khác. RoboDK station được lưu dưới dạng một tập tin (với đuôi mở rộng RDK) Không cần lưu riêng các tập tin robot, công cụ và các đối tượng vì chúng đã được lưu trong một tập tin RDK

Chọn **File** →  **New Station (Ctrl+N)** để bắt đầu một **project** mới




Hình P2.1: Tạo một project mới với station

P2.2 Tạo chương trình mô phỏng

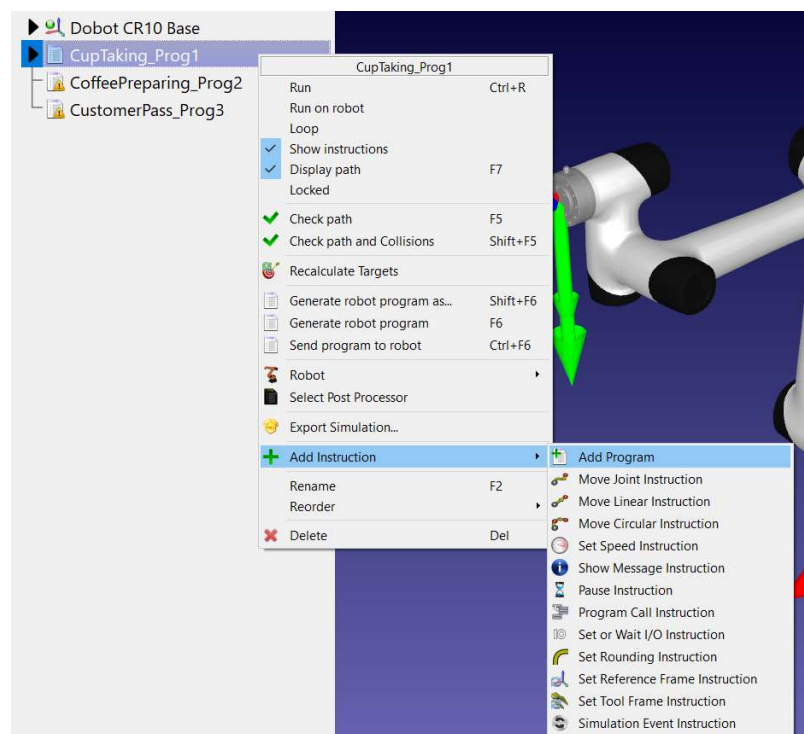
Việc mô phỏng có thể được thực hiện bằng cách thêm một chuỗi lệnh trong một chương trình. Mỗi lệnh đại diện cho mã cụ thể cho một bộ điều khiển cụ thể.

RoboDK cung cấp giao diện người dùng đồ họa (GUI) để xây dựng các chương trình robot dễ dàng hơn không cần phải viết từng mã lệnh trực tiếp.

Để tạo một chương trình mới trên RoboDK:

1. Chọn Program →  **Add Program** hoặc chọn trên thanh công cụ.
2. Chọn Tools → **Rename item...** (F2) để đổi tên chương trình.

Việc này sẽ tạo một chương trình mới và cho phép thêm các lệnh mới bằng cách nhấp chuột phải vào chương trình hoặc chọn từ menu **Program**.



Hình P2.2: Thêm lệnh cho chương trình

P2.3 Thêm hệ tọa độ tham chiếu

Hệ tọa độ tham chiếu (Reference frame) cho phép đặt các đối tượng tại một vị trí đối với một robot hay đối với các đối tượng khác trong không gian 3D (bao gồm giá trị vị trí và hướng).

Để thêm một hệ tọa độ tham chiếu:

1. Chọn Program →  **Add Reference Frame** hay có thể chọn trên toolbar

2. Nhập đúp chuột vào reference frame (trên cây station hay trên hình 3D của màn hình chính) để nhập vào tọa độ (X,Y,Z cho vị trí và các góc Euler cho hướng). Có thể dùng roller của chuột để nhập nhanh các giá trị này khi rê chuột đến các ô nhập tọa độ tương ứng.

Màu sắc thường dùng mặc định như sau:

- X coordinate → Red
- Y coordinate → Green
- Z coordinate → Blue
- Góc quay 1st Euler → Cyan
- Góc quay 2nd Euler → Magenta
- Góc quay 3rd Euler → Yellow

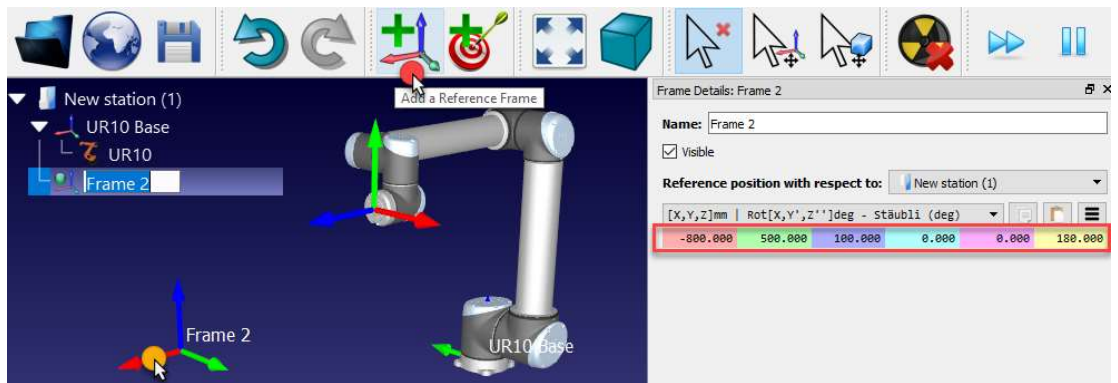
Tip: Chọn Tools → Options → Display → Display XYZ axis letters để hiển thị ký hiệu XYZ trên các trục.

3. Chọn **View** → **Make reference frames bigger (+)** để tăng kích thước hệ trục tọa độ tham chiếu

4. Chọn **View** → **Make reference frames smaller (-)** để giảm kích thước hệ trục tọa độ tham chiếu

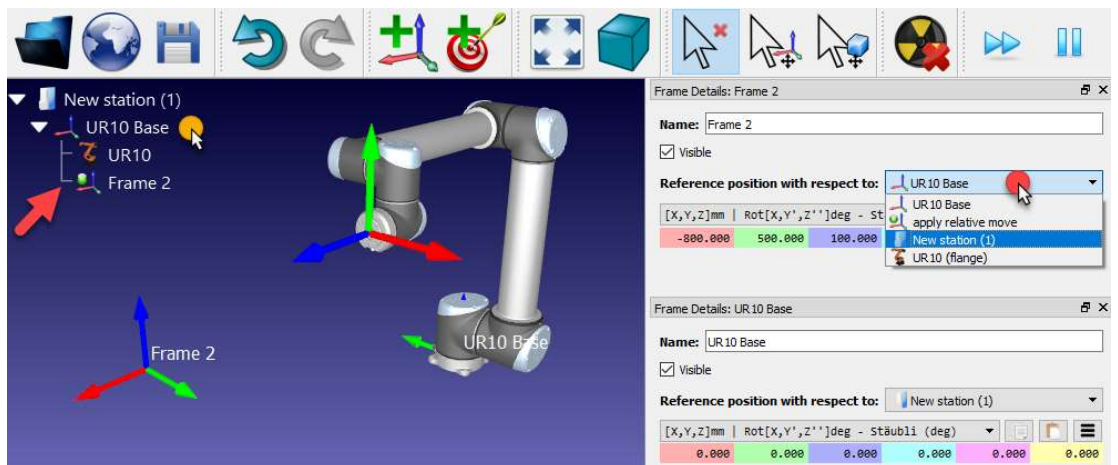
5. Chọn **View** → **Show/Hide text on screen (/)** để hiển thị hay dấu text trên màn hình

6. Thay đổi tên các hệ trục tọa độ tham chiếu hay các đối tượng trên cây station bằng cách chọn **F2**




Hình P2.3: Thêm hệ tọa độ tham chiếu

Nếu dùng nhiều một *reference frame*, ta có thể kéo thả chúng vào trong Station tree để tạo sự phụ thuộc như khi lắp đặt trực tiếp. Ví dụ, *reference Frame 2* có thể được đặt thuộc *robot base reference*. Trong trường hợp này, nếu robot di chuyển thì Frame 2 cũng di chuyển theo. Cho dù sự phụ thuộc khác nhau ta vẫn có thể nhập vào hay thu được tọa độ của bất kỳ reference nào đối với các reference khác. Hầu hết robot đều cần tọa độ của reference frame đối với robot base frame.



Hình P2.4: Thay đổi vị trí của hệ tọa độ tham chiếu

Reference frames cũng có thể di chuyển bằng cách nhấn giữ phím **Alt**, hay chọn nút  trên toolbar. Sau đó kéo *reference frame* bằng chuột trên màn hình. Khi *reference frame* di chuyển, thì tọa độ của nó của được cập nhật theo.