

**TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH**  
**KHOA ĐIỆN – ĐIỆN TỬ**  
**BỘ MÔN ĐIỀU KHIỂN TỰ ĐỘNG**

**VŨ HUY HỢP**

**LUẬN VĂN TỐT NGHIỆP**  
**PHÂN LOẠI HÀNG HÓA TRÊN BĂNG TẢI**  
**DÙNG ROBOT SCARA**

**KỸ SƯ NGÀNH KỸ THUẬT ĐIỀU KHIỂN & TỰ ĐỘNG HÓA**

**TP. HỒ CHÍ MINH, 2019**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH**  
**KHOA ĐIỆN – ĐIỆN TỬ**  
**BỘ MÔN ĐIỀU KHIỂN TỰ ĐỘNG**

**VŨ HUY HỢP - 1511190**

**LUẬN VĂN TỐT NGHIỆP**  
**PHÂN LOẠI HÀNG HÓA TRÊN BĂNG TẢI**  
**DÙNG ROBOT SCARA**  
**(CLASSIFYING OBJECTS ON CONVEYOR**  
**BY SCARA ROBOT)**

**KỸ SƯ NGÀNH KỸ THUẬT ĐIỀU KHIỂN & TỰ ĐỘNG HÓA**

**GIẢNG VIÊN HƯỚNG DẪN**

**KS. Lê Quang Thuận**

**TP. HỒ CHÍ MINH, 2019**

TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ  
MINH  
KHOA ĐIỆN – ĐIỆN TỬ  
BỘ MÔN: ĐIỀU KHIỂN TỰ ĐỘNG

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  
Độc lập - Tự do - Hạnh phúc

TP. HCM, ngày....tháng.....năm.....

**NHẬN XÉT LUẬN VĂN TỐT NGHIỆP  
CỦA CÁN BỘ HƯỚNG DẪN**

**Tên luận văn:**

**PHÂN LOẠI HÀNG HÓA TRÊN BẢNG TẢI DỪNG ROBOT SCARA**

**Nhóm Sinh viên thực hiện:**

VŨ HUY HỢP

**Cán bộ hướng dẫn:**

1511190

KS. LÊ QUANG THUẦN

**Đánh giá Luận văn**

1. Về cuốn báo cáo:

Số trang	_____	Số chương	_____
Số bảng số liệu	_____	Số hình vẽ	_____
Số tài liệu tham khảo	_____	Sản phẩm	_____

Một số nhận xét về hình thức cuốn báo cáo:

2. Về nội dung luận văn:

3. Về tính ứng dụng:

4. Về thái độ làm việc của sinh viên:

**Đánh giá chung:** Luận văn đạt/không đạt yêu cầu của một luận văn tốt nghiệp kỹ sư, xếp loại  
Giỏi/ Khá/ Trung bình

**Điểm từng sinh viên:**

**Vũ Huy Hợp:...../10**

Cán bộ hướng dẫn  
(Ký tên và ghi rõ họ tên)

TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ  
MINH  
KHOA ĐIỆN – ĐIỆN TỬ  
BỘ MÔN: ĐIỀU KHIỂN TỰ ĐỘNG

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  
Độc lập - Tự do - Hạnh phúc

TP. HCM, ngày....tháng.....năm.....

**NHẬN XÉT LUẬN VĂN TỐT NGHIỆP  
CỦA CÁN BỘ PHẢN BIỆN**

**Tên luận văn:**

**PHÂN LOẠI HÀNG HÓA TRÊN BẢNG TẢI DỪNG ROBOT SCARA**

**Nhóm Sinh viên thực hiện:**

VŨ HUY HỢP

**Cán bộ phản biện:**

1511190

**Đánh giá Luận văn**

1. Về cuốn báo cáo:

Số trang	_____	Số chương	_____
Số bảng số liệu	_____	Số hình vẽ	_____
Số tài liệu tham khảo	_____	Sản phẩm	_____

Một số nhận xét về hình thức cuốn báo cáo:

2. Về nội dung luận văn:

3. Về tính ứng dụng:

4. Về thái độ làm việc của sinh viên:

**Đánh giá chung:** Luận văn đạt/không đạt yêu cầu của một luận văn tốt nghiệp kỹ sư, xếp loại  
Giỏi/ Khá/ Trung bình

**Điểm từng sinh viên:**

**Vũ Huy Hợp:...../10**

Người nhận xét  
(Ký tên và ghi rõ họ tên)

TP. HCM, ngày....tháng.....năm.....

ĐỀ CƯƠNG CHI TIẾT

**TÊN LUẬN VĂN: PHÂN LOẠI HÀNG HÓA TRÊN BĂNG TẢI DÙNG ROBOT  
SCARA**

**Cán bộ hướng dẫn: KS. LÊ QUANG THUẦN**

**Thời gian thực hiện:** Từ ngày.....đến ngày.....

**Sinh viên thực hiện:**

**Vũ Huy Hợp - 1511190**

**Nội dung đề tài:**

- Xây dựng mô hình cánh tay robot SCARA 4 bậc tự do.
- Thiết kế mạch điều khiển và giao tiếp với máy tính.
- Xác định vị trí của vật thể di chuyển trên băng tải, gắp vật và phân loại theo vị trí.
- Xây dựng giao diện điều khiển trên máy tính.

**Kế hoạch thực hiện:**

Vũ Huy Hợp	Lên mục tiêu và nhiệm vụ luận văn	22 tháng 1
Vũ Huy Hợp	Mô phỏng tính toán động học của cánh tay robot trên matlab	05 tháng 3
Vũ Huy Hợp	Xây dựng xong mô hình cánh tay robot 4 bậc tự do	12 tháng 3
Vũ Huy Hợp	Lập trình khiển động học thuận, động học học ngược của mô hình	19 tháng 3
Vũ Huy Hợp	Xây dựng giao diện điều khiển máy tính	26 tháng 3
Vũ Huy Hợp	Kế hoạch các thuật toán xử lý ảnh xác định hướng và vị trí của vật	02 tháng 4
Vũ Huy Hợp	Xây dựng thuật toán gắp vật di chuyển trên băng tải	23 tháng 4
Vũ Huy Hợp	Tích hợp hệ thống	1 tháng 5
Vũ Huy Hợp	Kiểm tra và sửa chữa	10 tháng 5
Vũ Huy Hợp	Viết báo cáo	21 tháng 5
Vũ Huy Hợp	Xây dựng slide PowerPoint	1 tháng 6

**Xác nhận của Cán bộ hướng dẫn**  
(Ký tên và ghi rõ họ tên)

TP. HCM, ngày....tháng .....năm.....  
**Sinh viên**  
(Ký tên và ghi rõ họ tên)



## **DANH SÁCH HỘI ĐỒNG BẢO VỆ LUẬN VĂN**

Hội đồng chấm luận văn tốt nghiệp, thành lập theo Quyết định số .....  
ngày ..... của Hiệu trưởng Trường Đại học Bách khoa TP.HCM.

1. .... – Chủ tịch.
2. .... – Thư ký.
3. .... – Ủy viên.
4. .... – Ủy viên.
5. .... – Ủy viên.

## LỜI CẢM ƠN

Lời đầu tiên xin gửi lời cảm ơn chân thành đến thầy LÊ QUANG THUẦN đã tận tình chỉ dẫn, giúp đỡ và định hướng trong suốt thời gian làm đồ án và luận văn, tạo điều kiện thuận lợi để hoàn thành đề tài luận văn này.

Bên cạnh đó xin cảm ơn quý thầy cô trong bộ môn Điều khiển và Tự động hóa đã trang bị cho em những kiến thức nền tảng, bổ ích trong khoảng thời gian học tập tại trường.

Tôi xin gửi lời biết ơn sâu sắc đến cha mẹ, bạn bè và gia đình đã luôn quan tâm ủng hộ tôi trong suốt thời gian qua. Đó chính là nguồn động lực vô cùng lớn giúp tôi vượt qua những khó khăn và thử thách.

Cuối cùng, xin gửi lời cảm ơn đến các thành viên trong câu lạc bộ Pay It Forward đã có những sự hỗ trợ nhiệt tình trong quá trình thực hiện đề tài để đề tài luận văn có thể hoàn thành tốt.

Tp. Hồ Chí Minh, ngày 28 tháng 05 năm 2019

## MỤC LỤC

TÓM TẮT LUẬN VĂN.....	1
Chương 1. TỔNG QUAN VỀ ĐỀ TÀI.....	2
1.1. Đặt vấn đề.....	2
1.2. Mục tiêu của đề tài.....	2
1.3. Đối tượng và phạm vi nghiên cứu.....	3
1.4. Các bước thực hiện đề tài.....	3
Chương 2. TỔNG QUAN VỀ ROBOT VÀ CƠ SỞ LÝ THUYẾT.....	4
2.1. Giới thiệu sơ lược Robot.....	4
2.1.1. Lịch sử phát triển robot.....	4
2.1.2. Các định nghĩa về robot.....	5
2.1.3. Phân loại robot tay máy.....	6
2.1.3.1. Robot công nghiệp nổi tiếp.....	7
2.1.3.2. Robot song song.....	9
2.2. Động học thuận và ngược robot nổi tiếp.....	10
2.2.1. Biểu diễn DENVIT - HARTENBERG của bài toán động học Robot.....	11
2.2.2. Tính toán động học thuận cho robot SCARA của đề tài.....	14
2.2.3. Bài toán động học ngược robot SCARA.....	16
2.3. Động học vận tốc và ma trận Jacobian.....	17
2.3.1. Ma trận đối xứng ghènh.....	17
2.3.1.1. Định nghĩa.....	17
2.3.1.2. Một số tính chất của ma trận đối xứng ghènh và ma trận quay.....	17
2.3.2. Vận tốc góc đầu công tác.....	18
2.3.3. Vận tốc dài đầu công tác.....	19
2.3.4. Ma trận Jacobian.....	20

2.3.5. Điểm kỳ dị của tay máy.....	21
2.4. Mô phỏng tính toán trên Matlab thư viện robotic toolbox.....	22
2.4.1. Lập trình mô phỏng tính toán với Matlab.....	22
2.5. Đo kích thước vật thể qua kích thước vật tham chiếu bằng xử lý ảnh.....	25
2.6. Thuật toán nhận dạng vật thể.....	27
2.6.1. So sánh giữa 1 số thuật toán nhận diện vật thể: SIFT, SURF, HOG.....	27
2.6.2. Cơ sở lý thuyết thuật toán SURF.....	27
2.7. Thuật toán theo dõi.....	29
2.7.1. So sánh một số thuật toán theo dõi:.....	30
2.7.2. Cơ sở lý thuyết thuật toán KCF:.....	30
2.8. Thuật toán PCA.....	32
<b>Chương 3. XÂY DỰNG MÔ HÌNH.....</b>	<b>35</b>
3.1. Sơ đồ khối hệ thống.....	35
3.2. Xây dựng mô hình tay máy SCARA.....	37
3.2.1. Xây dựng mô hình trên máy tính.....	37
3.2.2. Hình ảnh thực tế của mô hình.....	39
3.3. Thiết kế mạch điều khiển.....	40
3.3.1. Kit điều khiển STM32F407 discovery.....	40
3.3.2. Driver điều khiển động cơ bước DRV8825.....	41
3.3.3. Thiết kế mạch điều khiển mô hình.....	42
3.3.4. Thiết kế mạch điều khiển bằng tải.....	45
3.3.5. Động cơ điều khiển.....	48
<b>Chương 4. GIAO DIỆN MÁY TÍNH VÀ GIẢI THUẬT ĐIỀU KHIỂN.....</b>	<b>50</b>
4.1. Thiết kế giao diện điều khiển trên máy tính bằng Qt Creator.....	50
4.1.1. Hình ảnh giao diện đã thiết kế.....	51

4.2. Giải thuật điều khiển Robot.....	53
Chương 5. KẾT QUẢ THỰC NGHIỆM.....	57
5.1. Kiểm tra độ chính xác tay máy.....	57
5.2. Thử nghiệm hoạt động của hệ thống phân loại hàng hóa.....	61
5.3. Đánh giá.....	65
Chương 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	66
6.1. Kết luận.....	66
6.2. Hướng phát triển.....	66
TÀI LIỆU THAM KHẢO.....	68

## DANH MỤC HÌNH ẢNH

Hình 2.1 Kết cấu robot.....	6
Hình 2.2 Robot tọa độ vuông góc đề các.....	7
Hình 2.3 Robot tọa độ trụ.....	7
Hình 2.4 Robot tọa độ cầu.....	8
Hình 2.5 Robot SCARA.....	8
Hình 2.6 Robot khủy tay.....	9
Hình 2.7 Robot nối tiếp với cơ cấu động học kín.....	10
Hình 2.8 Robot song song 6 bậc tự do.....	10
Hình 2.9 Mô hình hóa khâu và khớp theo D-H.....	12
Hình 2.10 Xây dựng hệ trục tọa độ cho robot SCARA.....	14
Hình 2.11 Giải động học ngược theo phương pháp hình học.....	16
Hình 2.12 Kết quả mô phỏng với thư viện robotic toolbox.....	24
Hình 2.13 Mô phỏng không gian làm việc của robot trên mặt phẳng.....	24
Hình 2.14 Mô phỏng không gian làm việc của robot trong không gian.....	25
Hình 2.15 Ảnh của vật qua thấu kính camera.....	25
Hình 2.16 Xác định vị trí của vật thể trong không gian.....	26
Hình 2.17 Phương pháp xấp xỉ box filter.....	28
Hình 2.18 Hướng đại diện của keypoint.....	28
Hình 2.19 Tìm kiếm keypoint phù hợp.....	29
Hình 2.20 Thuật toán theo dõi đối tượng.....	31
Hình 2.21 Ý tưởng chính của thuật toán PCA.....	33
Hình 2.22 PCA dưới góc nhìn thống kê.....	34
Hình 2.23 Các bước thực hiện thuật toán PCA.....	34

Hình 3.1 Sơ đồ khối hệ thống.....	35
Hình 3.2 Bảng tải.....	36
Hình 3.3 Camera logiect C270.....	36
Hình 3.4 Hình ảnh 3D của mô hình.....	37
Hình 3.5 ảnh 3D của các chi tiết.....	38
Hình 3.6 Ghép nối các chi tiết.....	38
Hình 3.7 Hình ảnh toàn bộ hệ thống.....	39
Hình 3.8 Sơ đồ kết nối phần cứng.....	40
Hình 3.9 Board Discovery STM32F4.....	40
Hình 3.10 Kết nối DRV8825 với động cơ và vi điều khiển.....	41
Hình 3.11 Module PTN78020WAD.....	42
Hình 3.12 Khối nguồn.....	42
Hình 3.13 Các khối tín hiệu.....	43
Hình 3.14 Khối MCU.....	43
Hình 3.15 Hình ảnh PCB của mạch.....	44
Hình 3.16 Hình ảnh 3D của mạch.....	44
Hình 3.17 Sơ đồ khối mạch điều khiển băng tải.....	45
Hình 3.18 Khối nguồn mạch điều khiển băng tải.....	45
Hình 3.19 Khối vi điều khiển stm8s.....	46
Hình 3.20 Một số khối tín hiệu mạch điều khiển băng tải.....	46
Hình 3.21 Hình ảnh sau khi layout của mạch.....	47
Hình 3.22 Hình ảnh thực tế của mạch điều khiển băng tải.....	47
Hình 3.23 Động cơ bước.....	48
Hình 3.24 Động cơ servo MG90S .....	49

Hình 4.1 Giao diện Qt creator.....	50
Hình 4.2 Hình ảnh bảng điều khiển bằng tay.....	51
Hình 4.3 bảng điều khiển tự động.....	52
Hình 4.4 Lưu đồ giải thuật chương trình.....	53
Hình 4.5 Lưu đồ giải thuật xử lý ảnh.....	55
Hình 5.1 Kiểm tra sai số theo trục x.....	57
Hình 5.2 Kiểm tra độ chính xác theo trục y.....	58
Hình 5.3 Kiểm tra độ chính xác trục Z.....	59
Hình 5.4 Xác định vị trí của vật bằng camera.....	60
Hình 5.5 Robot xếp bao xi-măng trong thực tế.....	61
Hình 5.6 Tay gắp sử dụng trong đề tài.....	62
Hình 5.7 Vật thể chọn để phân loại.....	62
Hình 5.8 Calib camera.....	63
Hình 5.9 Chọn vị trí xếp vật.....	63
Hình 5.10 Hệ thống bắt đầu hoạt động.....	64
Hình 5.11 Kết quả thu được.....	64



## DANH MỤC BẢNG

Bảng 2.1 Bảng ma trận D-H robot SCARA.....	14
Bảng 2.2 So sánh giữa các thuật toán nhận dạng.....	27
Bảng 2.3 So sánh giữa một số thuật toán theo dõi.....	30
Bảng 3.1 Thông số kỹ thuật động cơ bước.....	48
Bảng 5.1 Thông số của cánh tay robot.....	57
Bảng 5.2 Kết quả thí nghiệm điều khiển vị trí.....	59

## DANH MỤC TỪ VIẾT TẮT

SCARA	<b>S</b> elective <b>C</b> ompliance <b>A</b> rticulated <b>R</b> obot <b>A</b> rm	Cánh tay robot có khớp nối có chọn lọc
SURF	<b>S</b> peed <b>U</b> p <b>R</b> obust <b>F</b> eature	Thuật toán nhận diện vật thể có tốc độ xử lý nhanh
KCF	<b>K</b> ernelized <b>C</b> orrelation <b>F</b> ilters	Thuật toán theo dõi dựa trên bộ lọc tương quan hạt nhân
PCA	<b>P</b> rincipal <b>C</b> omponents <b>A</b> nalysis	Thuật toán phân tích chiều dữ liệu

## TÓM TẮT LUẬN VĂN

Đề tài “Phân loại hàng hóa trên băng tải bằng cánh tay robot Scara”.

Hệ thống gồm 4 thiết bị chính là: Robot Scara, băng tải, camera logitec C270, và máy tính laptop.

Máy tính laptop đóng vai trò là trung tâm điều khiển các thiết bị với nhau. Hệ thống có khả năng phân loại các vật thể khác nhau di động trên băng tải, ngoài ra thì hệ thống còn có khả năng xác định đúng hướng của vật cần gấp và sắp xếp theo đúng vị trí. Hệ thống có thể làm việc với điều kiện tốc độ băng tải thay đổi và có độ chính xác cao khi tốc độ băng tải trong khoảng từ 0 - 3 cm/s.

Thuật toán xử lý kết hợp giữa nhận diện vật thể (detection) và theo dõi vật thể (tracking) để có thể xử lý chính xác khi băng tải có tốc độ cao.

Giao diện phần mềm thân thiện, dễ sử dụng, được viết dựa trên framework QT ngôn ngữ C++, hỗ trợ đa nền tảng. Thông qua giao diện phần mềm người sử dụng có thể điều khiển, quan sát, nắm bắt tổng thể trạng thái của hệ thống.

## **Chương 1. TỔNG QUAN VỀ ĐỀ TÀI**

### **1.1. Đặt vấn đề**

Cùng với sự phát triển mạnh mẽ của khoa học kỹ thuật trên thế giới cũng như ở nước ta hiện nay, việc ứng dụng các công nghệ mới, máy móc vào các dây chuyền sản xuất là rất quan trọng nhằm cải thiện năng suất lao động và chất lượng của sản phẩm, là một trong những tiêu chí đánh giá sự phát triển kinh tế của mỗi quốc gia.

Xây dựng một hệ thống có khả năng ứng dụng cao trong thực tế sản xuất, thay thế phần lớn hoạt động con người trong sản xuất nâng cao năng suất và chất lượng của sản phẩm là một lĩnh vực có ứng dụng vô cùng lớn và tiềm năng.

Hiện nay trong nước các hệ thống tự động hóa xuất hiện ngày càng nhiều, phần lớn chỉ tự động hóa các khâu rời rạc và vẫn còn cần sự can thiệp lớn từ con người trong một dây chuyền sản xuất. Chỉ có một số ít các dây chuyền sản xuất có thể tự động hóa xuyên suốt từ đầu đến cuối khi đóng gói sản phẩm.

Hệ thống phân loại, sắp xếp sản phẩm trên băng chuyền là một trong những hệ thống rất phổ biến có tính ứng dụng cao trong các nhà máy và xí nghiệp. Vì vậy với mục đích bổ sung kiến thức và kỹ năng chuyên môn, và có thể ứng dụng đóng góp cho sự phát triển của nền công nghiệp hóa đất nước em đã quyết định chọn đề tài: “Phân loại sản phẩm trên băng tải bằng cách tay Robot Scara”.

Nhận định đây là một hệ thống tự động có khả năng ứng dụng rộng rãi trong nhiều quy trình sản xuất, đặc biệt là quy trình sản xuất hàng hóa tiêu dùng, công nghiệp thực phẩm. Đồng thời đây cũng là hệ thống có thể làm việc nhanh, hiệu quả, giảm thiểu chi phí sản xuất. Hệ thống được kết hợp dựa trên nhiều lĩnh vực khác nhau như: cơ khí, thiết kế, lập trình điều khiển tự động, xử lý ảnh và lập trình giao diện người dùng.

### **1.2. Mục tiêu của đề tài**

- Hiểu cơ bản cách xây dựng các hệ trục tọa độ cho từng khớp của một cánh tay robot, biết cách tính toán động học thuận, động học ngược của cánh tay robot.
- Kiểm chứng được những kiến thức đã được học về lập trình vi xử lý, điều khiển động cơ step, xử lý ảnh, thiết kế giao diện...

- Xây dựng được một mô hình hệ thống phân loại sản phẩm có khả năng ứng dụng được trong thực tế.

### **1.3. Đối tượng và phạm vi nghiên cứu**

Luận văn tập trung vào việc xây dựng một hệ thống phân loại hàng hóa tự động trên băng tải. Trong quá trình thực hiện để đáp ứng được yêu cầu của đề tài ta cần tìm hiểu các lĩnh vực khác nhau nhằm đáp ứng xây dựng thành công mô hình. Phạm vi nghiên cứu của luận văn gồm nhiều mảng kiến thức:

- Tìm hiểu cách mô hình hóa xây dựng các hệ trục tọa độ của một cánh tay robot, giải các bài toán liên quan động học thuận, động học ngược và điểm kỳ dị của robot.
- Thiết kế và lập trình mạch điều khiển động cơ step.
- Nghiên cứu các giải thuật xử lý ảnh phù hợp để phân loại sản phẩm.
- Thiết kế giao diện điều khiển trên máy tính.

### **1.4. Các bước thực hiện đề tài**

- Mô phỏng các tính toán động học thuận, động học ngược của robot trên phần mềm Matlab với thư viện robotic toolbox.

- Xây dựng mô hình cánh tay Robot Scara 4 bậc tự do từ vật liệu chủ yếu là nhựa PLA, phần lớn các chi tiết được gia công bằng máy in 3D. Robot gồm 3 động cơ bước dùng để điều khiển vị trí điểm cuối, 1 động cơ mini servo điều khiển hướng của đầu gấp, 1 động cơ mini servo điều khiển đầu gấp.

- Lập trình điều khiển các khớp xoay của robot bằng kit STM32F407 discovery theo lệnh nhận được từ máy tính.

- Nhận diện xác định vị trí, hướng của vật trong không gian làm việc của robot bằng thuật toán SURF, KCF và PCA.

- Kết hợp thuật toán nhận diện và theo dõi vật thể gấp vật di chuyển trên băng chuyền và sắp xếp vật theo đúng vị trí.

- Thiết kế giao diện phần mềm trên máy tính quan sát và điều khiển hệ thống.

## **Chương 2. TỔNG QUAN VỀ ROBOT VÀ CƠ SỞ LÝ THUYẾT**

### **2.1. Giới thiệu sơ lược Robot**

#### **2.1.1. Lịch sử phát triển robot**

Thuật ngữ ROBOT xuất hiện vào năm 1920 trong một tác phẩm của nhà văn Tiệp Khắc có tên Karel Capek.

Thuật ngữ Industrial Robot (IR) - xuất hiện đầu tiên ở Mỹ do công ty AMF (American Machine and Foundry company) quảng cáo mô tả một thiết bị mang dáng dấp và có một số thao tác sản xuất thiết bị có tên gọi Versatran.

Quá trình phát triển của IR có thể tóm tắt như sau:

- Từ những năm 50 ở Mỹ xuất hiện viện nghiên cứu đầu tiên.
- Đầu những năm 60 xuất hiện sản phẩm đầu tiên tên Versatran của công ty AMF.
- Ở Anh người ta bắt đầu nghiên cứu và chế tạo IR theo bản quyền của Mỹ từ năm 1967.
- Ở các nước Tây Âu khác như: Đức, Ý, Pháp, Thụy điển, từ những năm 70.
- Châu á có Nhật bắt đầu nghiên cứu ứng dụng IR từ năm 1968.
- Nhất là vào những năm 1990 số lượng robot công nghiệp đã gia tăng với nhiều tính năng vượt bậc.
- Đến nay trên thế giới có khoảng trên 200 công ty sản xuất robot công nghiệp, trong đó Mỹ và Nhật chiếm đa số.

Theo chủng loại, mức độ điều khiển, và khả năng nhận biết thông tin của robot ta có thể phân loại IR thành các thế hệ sau:

- Thế hệ 1: thế hệ có kiểu điều khiển theo chu trình dạng chương trình cứng không có khả năng nhận biết thông tin.
- Thế hệ 2: thế hệ có kiểu điều khiển theo chu trình dạng chương trình mềm bước đầu đã có khả năng nhận biết thông tin.

- Thế hệ 3: thế hệ có kiểu điều khiển dạng tinh khôn, có khả năng nhận biết thông tin và bước đầu đã có một số chức năng lý trí của con người.

Đối với tay máy công nghiệp đã có hơn 250 loại, trong đó có hơn 40% là loại tay máy có kiểu điều khiển đơn giản thuộc thế hệ thứ nhất.

Sự xuất hiện của robot và sự gia tăng vai trò của chúng trong sản xuất và xã hội loài người làm xuất hiện một ngành khoa học mới là ngành robot học (Robotics). Trên thế giới ở nhiều nước đã xuất hiện những viện nghiên cứu riêng về robot.

### **2.1.2. Các định nghĩa về robot**

Có khá nhiều định nghĩa về Robot cho đến nay. Năm 1986, McKerrow định nghĩa Robot là một loại máy móc cơ khí có thể lập trình để thực hiện một số công việc nào đó.

Schulussel (1985) lại định nghĩa Robot là một tay máy đa chức năng, khả năng được thiết kế để vận chuyển nguyên vật liệu, phối, chi tiết gia công.

Ben Zion S.(1999) Robot là một cơ cấu thường ở dạng hở, các khâu được nối tiếp nhau và di chuyển tương đối với nhau nhằm mục đích gắp và di chuyển đối tượng theo một số bậc tự do nhất định.

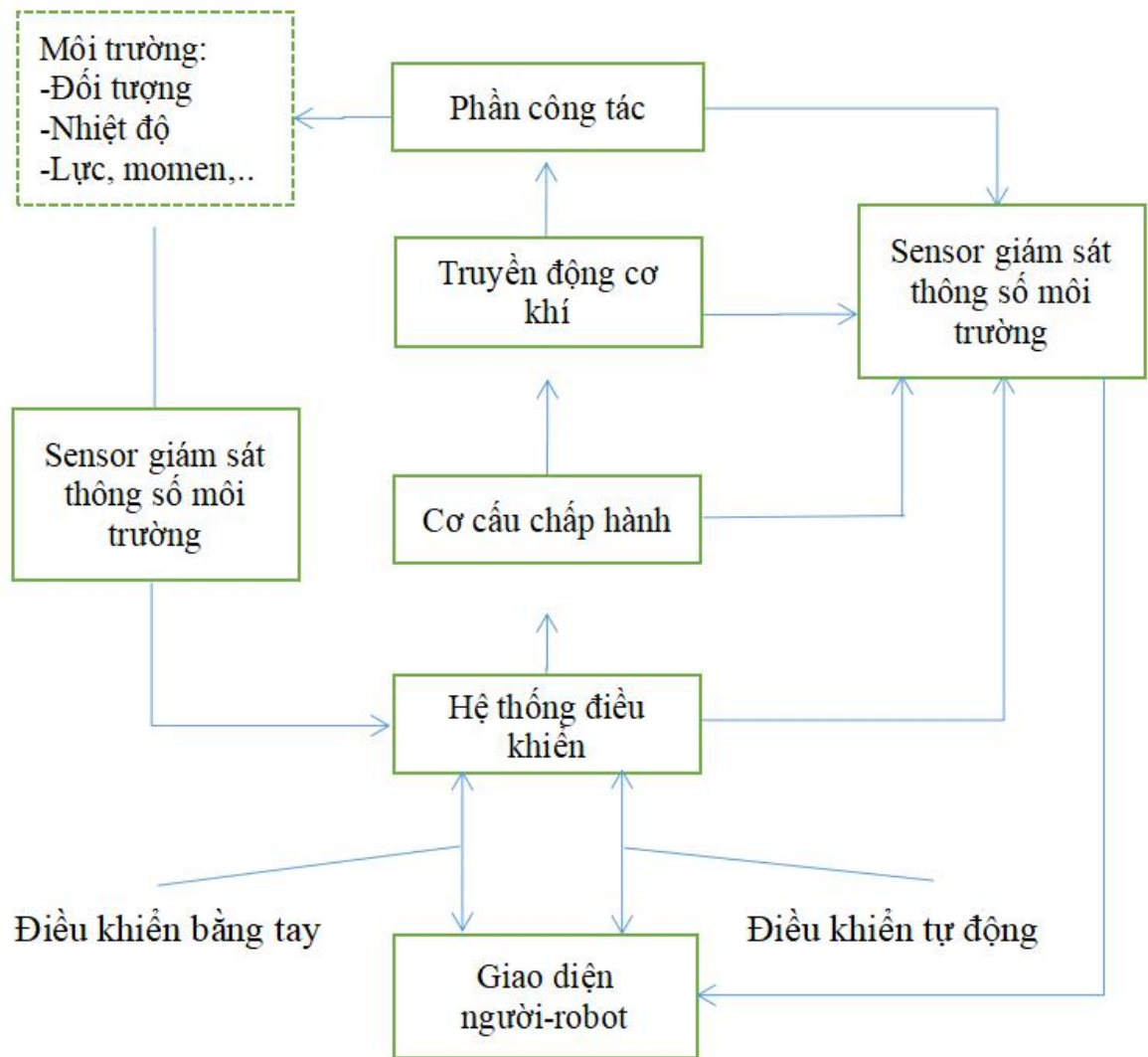
Robot là một đối tượng máy móc có thể lập trình điều khiển, có chức năng nhiệm vụ, có thể lập trình, có thể được điều khiển tự động hoặc điều khiển bằng tay,...

Định nghĩa robot còn được Mikell P.Groover, một nhà nghiên cứu hàng đầu trong lĩnh vực robot, mở rộng hơn như sau: “Robot công nghiệp là những máy, thiết bị tổng hợp hoạt động theo chương trình có những đặc điểm nhất định như con người”.

Định nghĩa của M.P.Groover về robot không dừng lại ở tay máy mà mở rộng ra nhiều đối tượng khác có những đặc tính tương tự như con người như là suy nghĩ, có khả năng đưa ra quyết định và có thể nhìn thấy hoặc cảm nhận vật hay đối tượng mà nó phải thao tác hoặc xử lý. Theo Artobolevski I.I, Vorobiv M.V và các nhà nghiên cứu thuộc trường phái khối SEV trước đây thì phải biểu rằng:

“Robot công nghiệp là những máy hoạt động tự động được điều khiển theo chương trình để thực hiện việc thay đổi vị trí của những đối tượng thao tác khác nhau với mục đích tự động hóa các quá trình sản xuất”.

Kết cấu cơ bản của robot:



Hình 2.1 Kết cấu robot

### 2.1.3. Phân loại robot tay máy

Trong công nghiệp người ta sử dụng những đặc điểm khác nhau cơ bản nhất của robot để giúp cho việc nhận xét đánh giá được dễ dàng đối với từng loại robot từ đó ứng dụng cho phù hợp với từng mục đích và nhiệm vụ của mỗi loại công việc. Có nhiều phương pháp để phân loại robot tay máy, tùy theo kết cấu, ứng dụng của mỗi loại robot, tuy nhiên thông thường thì có 4 yếu tố chính để phân loại robot tay máy:

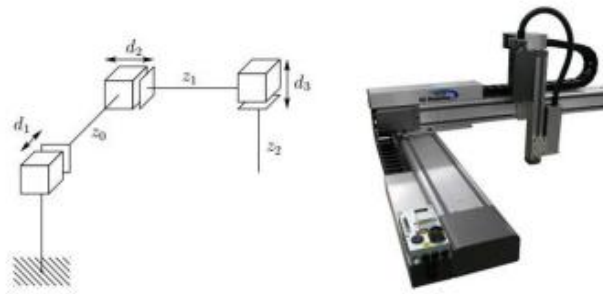


theo dạng hình học của không gian hoạt động, theo thể hệ robot, theo bộ điều khiển, theo nguồn dân động.

### 2.1.3.1. Robot công nghiệp nối tiếp

- Robot tọa độ vuông góc (cartesian robot):

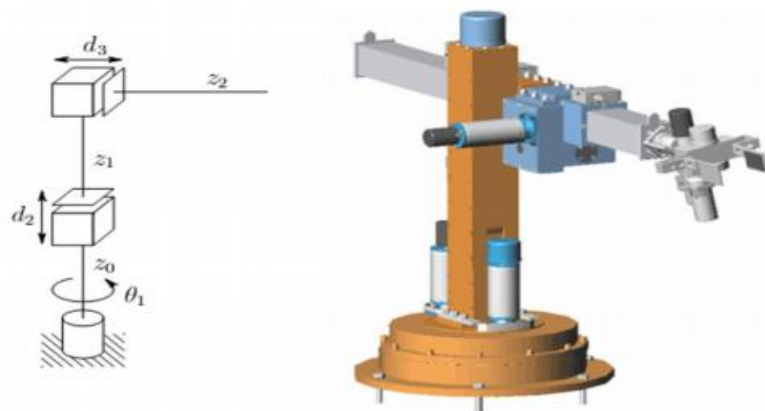
Robot loại này có ba bậc chuyển động cơ bản gồm ba chuyển động tịnh tiến theo ba trục vuông góc.



Hình 2.2 Robot tọa độ vuông góc đề các

- Robot tọa độ trụ (cylindrical robot):

Tay máy kiểu tọa độ trụ: khớp đầu tiên dùng khớp xoay thay cho khớp trượt. Vùng làm việc có dạng hình trụ rỗng. Độ cứng vững cơ học của tay máy trụ tốt, thích hợp với tải nặng, nhưng độ chính xác giảm trong mặt phẳng nằm ngang khi tầm làm việc tăng.



Hình 2.3 Robot tọa độ trụ

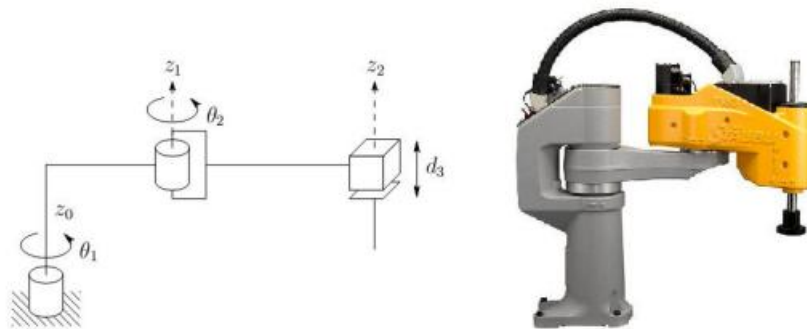
- Robot tọa độ cầu (spherical robot):

Khớp thứ hai được thay bằng khớp quay. Vùng làm việc của nó là khối cầu rỗng, độ cứng vững của loại tay máy này thấp hơn 2 loại trên và độ chính xác phụ thuộc vào tầm với. Loại này có thể nhặt được cả các vật thể ở dưới nền.



Hình 2.4 Robot tọa độ cầu

- Robot SCARA (Selective Compliance Assembly Robot Arm): có ba khớp quay với trục song song RRP thường dùng cho công việc lắp ráp board mạch điện tử



Hình 2.5 Robot SCARA

- Robot khuỷu tay (elbow, articulated, anthropomorphic) mô phỏng tay người: đây là loại phổ biến nhất, có 3 khớp quay RRR, hai trục quay của hai khớp cuối song song với nhau và thẳng góc trục quay khớp thứ nhất.

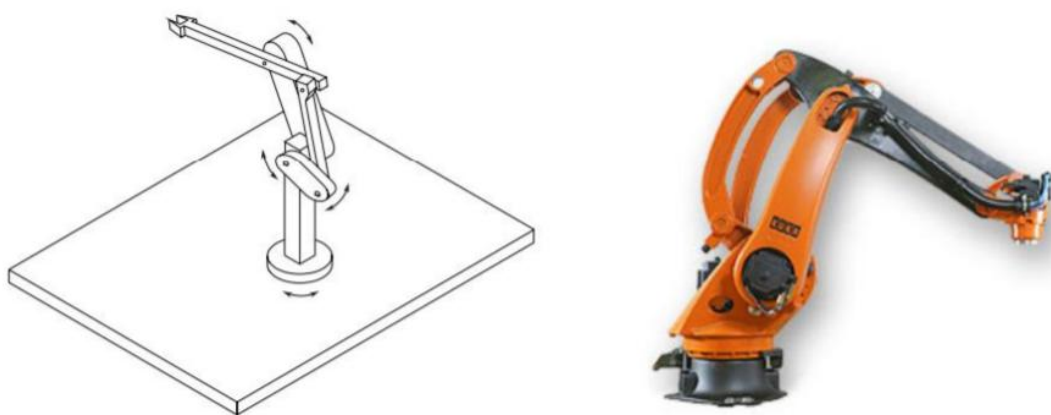


Hình 2.6 Robot khủy tay

Robot dùng để thực hiện các thao tác thay thế con người, do đó phần quan trọng của robot là bàn tay (end effector) để gắn các dụng cụ vào hay để gấp các các vật, bàn tay nối với cẳng tay qua khớp cổ tay, khớp này có từ hai đến ba bậc do để tạo sự khéo léo, linh hoạt, khớp cổ tay, khớp này có từ hai đến ba bậc tự do để tạo sự khéo léo, linh hoạt, khớp cổ tay là kết hợp các khớp quay với các trục quay thẳng góc nhau và gặp nhau ở tâm cổ tay, như vậy robot sẽ có năm hoặc sáu bậc tự do. Tên gọi góc quay của khớp cổ tay thường sử dụng là Roll, Pitch, Raw các góc này được đặt tên theo tên góc của máy bay trên không, tàu trên mặt nước hay đầu người,...

#### 2.1.3.2. Robot song song

Robot nối tiếp có độ cứng kém do cơ cấu động học hở, khả năng mang tải của robot chỉ khoảng vài kg, muốn tăng độ cứng của robot ta cần sử dụng cơ cấu động học kín hình bình hành hoặc robot song song (Hình 2.6). Robot này gồm hai đế một đế cố định và một di chuyển, nối với nhau bằng các khâu, độ cứng tăng lên nhưng kích thước vùng làm việc giảm xuống. Một loại robot song song thông dụng là robot Stewart-Gough, có sáu thanh kiểu khớp trượt chủ động nối với hai đế bằng khớp cầu thụ động (Hình 2.7)



Hình 2.7 Robot nối tiếp với cơ cấu động học kín



Hình 2.8 Robot song song 6 bậc tự do

## 2.2. Động học thuận và ngược robot nối tiếp

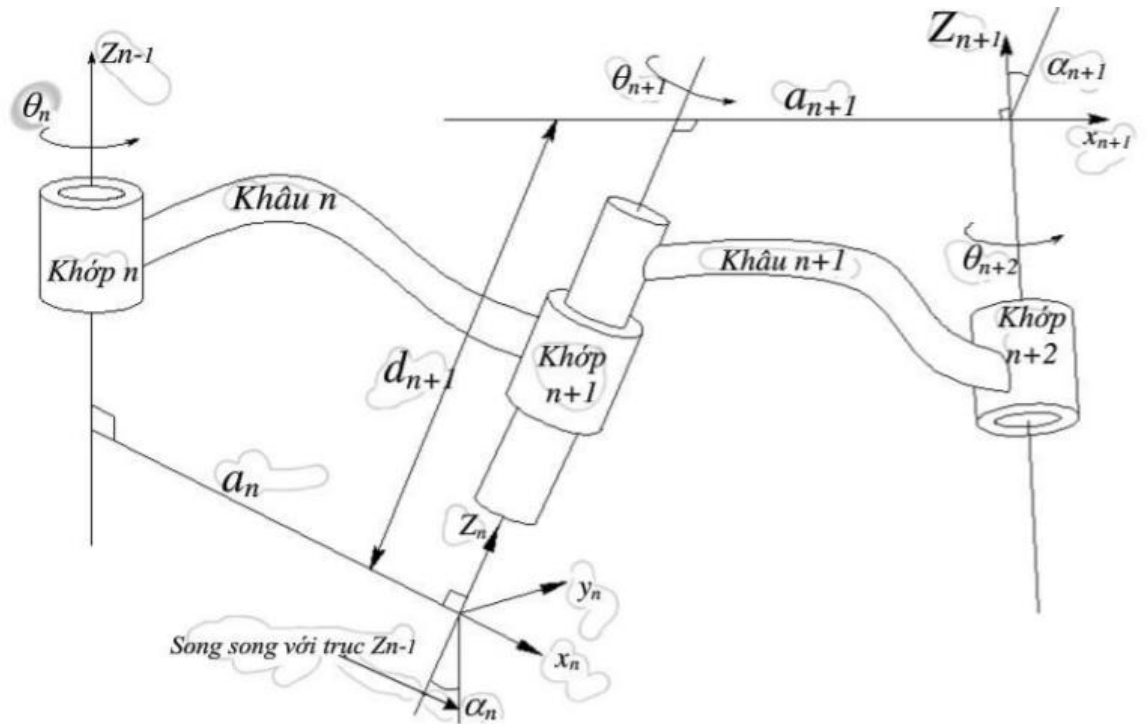
Robot nối tiếp là chuỗi động học hở gồm nhiều khâu nối với nhau bởi khớp quay hoặc trượt. Để robot liên kết với hệ tọa độ trục chuẩn tham chiếu cố định, đầu công tác nằm ở đầu cuối của chuỗi. Vị trí tương đối giữa hai khâu kế nhau phụ thuộc góc quay  $\theta$  hay độ dịch chuyển  $d$  gọi chung là biến khớp  $q$ . Động học thuận robot có mục đích là tìm vị trí và hướng đầu công tác so với hệ tọa độ cố định khi các khớp chuyển động còn động học ngược là tìm các biến khớp khi có trước vị trí và hướng đầu công tác.

### 2.2.1. Biểu diễn DENAVIT - HARTENBERG của bài toán động học Robot

Vào năm 1955, Denavit và Hartenberg đăng tải một bài báo của tạp chí Cơ học ứng dụng ASME (Journal of Applied Mechanics). Từ các phương pháp giải trong bài báo này, việc ứng dụng phương pháp giải quyết bài toán được đưa vào biểu diễn và mô hình hóa robot và đề toàn ra phương trình di chuyển. Phương pháp này trở thành phương pháp tiêu biểu để biểu diễn robot và mô hình hóa các chuyển động của nó. Mô hình hóa Denavit - Hertenberg (Viết tắt phương pháp D-H) là cách biểu diễn đơn giản mô hình các khâu và khớp của robot và có thể sử dụng cho bất cứ cấu hình robot nào, kể cả bài toán phức tạp hay đơn giản. Và chúng có thể dùng phương pháp này để biểu diễn cho bất kỳ hệ trục tọa độ nào như: Descartes, trụ, cầu, Euler, RPY, ... Thêm vào đó, nó có thể sử dụng để biểu diễn cho tất cả robot mà chúng ta có thể gặp trong công nghiệp và robot dịch vụ, robot biến hình. Mặc dù, chúng ta có thể giải các bài toán đơn giản bằng phương pháp ma trận thuần nhất hay hình học nhưng với việc biểu diễn D-H chúng ta còn có thể sử dụng kết quả của nó cho các bài toán như jacobi, phân tích lực, độ cứng ...

Giả sử robot là một chuỗi các khâu và các khớp nối tiếp. Các khớp này có thể là khớp trượt (tịnh tiến) hay khớp trụ (quay), và chúng có thể sắp xếp theo bất kỳ thứ tự nào và có thể nằm trong bất kỳ mặt phẳng nào. Các khâu cũng có chiều dài bất kỳ kể cả bằng 0, có thể xoắn hoặc cong. Vì vậy bất kỳ một tập hợp các khâu, các khớp đều có thể tạo thành cấu hình một robot và chúng ta có thể giải quyết các vấn đề đối với nó thông qua việc mô hình hóa.

Để làm được việc này, cần phải gán một hệ trục tham chiếu tới mỗi khớp và sau đó xác định sự chuyển vị từ khớp này đến khớp tiếp theo. Nếu kết hợp tất cả các chuyển vị từ bộ đến khi khớp cuối cùng chúng ta sẽ có ma trận chuyển vị tổng cộng. Ở phần dưới, ta sẽ xác định giải thuật tổng quát dựa vào biểu diễn D-H để gán các hệ trục tham chiếu lên mỗi khớp. Sau đó xác định chuyển vị giữa hai hệ trục kế nhau. Cuối cùng sẽ có được ma trận chuyển vị của robot.



Hình 2.9 Mô hình hóa khâu và khớp theo D-H

Giả sử robot được thiết kế với một số lượng khâu, các khớp tùy ý và có thể bố trí bất kỳ. Hình 2.8 biểu diễn ba khâu kế tiếp nhau với hai khớp. Các khớp này có thể là khớp tịnh tiến hoặc khớp quay hoặc có thể là cả hai. Trong thực tế, việc thiết kế robot thường là 5 khớp để dễ dàng điều khiển nhưng trong hình vẽ thường biểu diễn các khớp này có hai bậc tự do. Mỗi khớp trong mô hình hóa có thể quay hoặc tịnh tiến, giả sử ta đánh số  $n$  cho khớp thứ nhất trong hình 2.8,  $n+1$  cho khớp thứ hai và  $n+2$  cho khớp thứ ba. Như vậy có thể hiểu rằng: trước và sau ba khớp này có thể có rất nhiều khớp khác tác động và gắn lên hệ thống robot. Mỗi khâu cũng sẽ được đánh số, khâu  $n$  sẽ nằm giữ khớp  $n$  và khớp  $n+1$ , khâu  $n+1$  sẽ nằm giữ khớp  $n+1$  và khớp  $n+2$ .

Để mô hình hóa robot với sự biểu diễn D-H, việc đầu tiên cần làm là gắn một hệ trục tham chiếu địa phương lên mỗi và mọi khớp. Vì vậy, mỗi khớp chúng ta sẽ phải gắn trục  $z$  và trục  $x$ . Chúng ta không nhất thiết phải gắn trục  $y$  vì chúng ta có thể xác định trục  $y$  từ trục  $z$  và  $x$  theo quy tắc bàn tay phải. Ngoài ra vì biểu diễn D-H không cần thiết sử dụng trục  $y$ . Quá trình gắn hệ trục tham chiếu:

- Bước 1: Xác định và gắn nhãn cho trục  $z_0, \dots, z_{n-1}$  của các khớp.

- Bước 2: Thành lập hệ trục tham chiếu (cơ sở). Đặt điểm gốc nằm trên trục  $z_0$ . Chọn các trục  $x_0, y_0$  theo quy tắc bàn tay phải. Làm tương tự cho các khớp  $i = 1, \dots, n-1$  theo các bước 3 4 5.

- Bước 3: Xác định gốc  $o_i$  là các điểm vuông góc chung của  $z_i$  và  $z_{i-1}$ . Nếu  $z_i$  cắt  $z_{i-1}$  thì điểm giao nhau đó chính là điểm gốc  $o_i$ . Nếu  $z_i$  và  $z_{i-1}$  song song thì  $o_i$  có thể chọn là điểm bất kì trên trục  $z_i$ .

- Bước 4: Trục  $x_i$  được xác định dọc theo đường vuông góc chung của  $z_i$  và  $z_{i-1}$  qua điểm  $o_i$ , hoặc theo hướng vuông góc với mặt phẳng  $z_i - z_{i-1}$  nếu  $z_i$  và  $z_{i-1}$  cắt nhau.

- Bước 5: Trục  $y_i$  xác định theo quy tắc bàn tay phải.

- Bước 6: Xác định hệ trục  $o_n x_n y_n z_n$ . Giả sử khớp  $n$ -th là khớp xoay, đặt  $z_n = a$  theo hướng của trục  $z_{n-1}$ . Chọn điểm gốc  $o_n$  trên trục  $z_n$  sao cho thuận tiện. Đặt  $y_n = s$  trùng với hướng đóng mở của đầu gấp và đặt  $x_n = n$ .

- Bước 7: Tạo bảng bao gồm các thông số  $a_i, d_i, \alpha_i, \theta_i$ .

$a_i$  = khoảng cách theo trục  $x_i$  từ  $o_i$  tới giao điểm trục  $x_i$  và  $z_{i-1}$ .

$d_i$  = khoảng cách theo trục  $z_{i-1}$  từ  $o_{i-1}$  tới giao điểm trục  $x_i$  và  $z_{i-1}$ .  $d_i$  là biến nếu khớp  $i$  là khớp trượt.

$\alpha_i$  = góc giữa  $z_i$  và  $z_{i-1}$ .

$\theta_i$  = góc giữa  $x_{i-1}$  và  $x_i$ .  $\theta_i$  là biến nếu khớp  $i$  là khớp xoay.

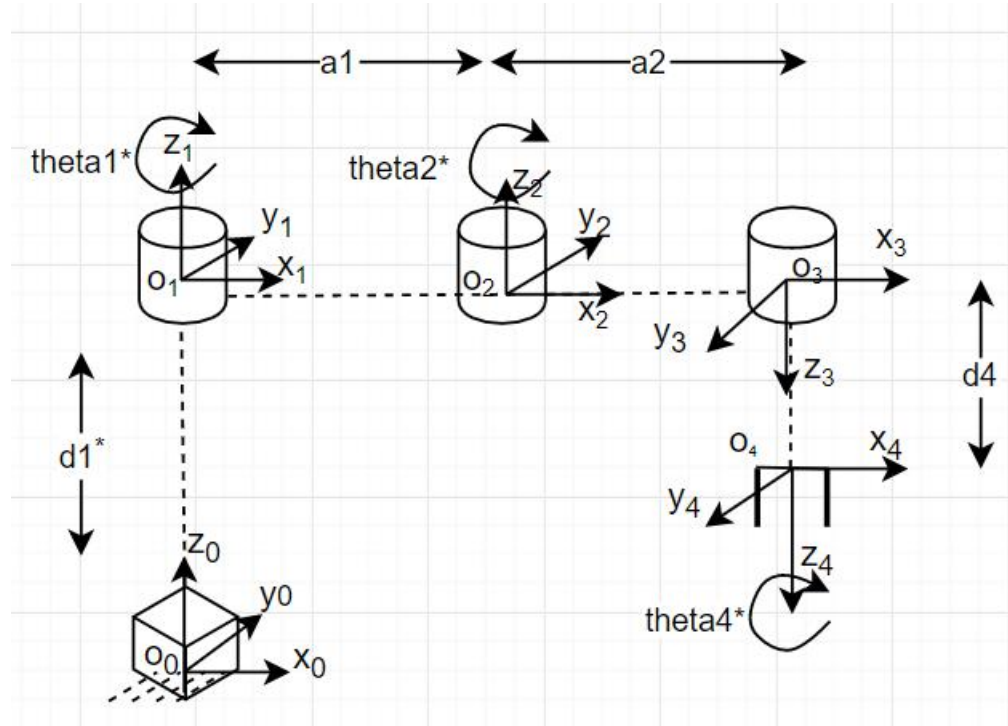
- Bước 8: Tính ma trận  $A_i$  theo công thức:

$$A_i = \begin{bmatrix} C_{\theta_i} & -C_{\alpha_i} * S_{\theta_i} & S_{\alpha_i} * S_{\theta_i} & a * C_{\theta_i} \\ S_{\theta_i} & C_{\alpha_i} * C_{\theta_i} & -S_{\alpha_i} * C_{\theta_i} & a * S_{\theta_i} \\ 0 & S_{\alpha_i} & C_{\alpha_i} & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

- Bước 9: Tính ma trận biến đổi thuần nhất  $T_n^0 = A_1 \dots A_n$ . Từ ma trận biến đổi thuần nhất có thể tính được vị trí và hướng của điểm cuối theo góc tọa độ.

### 2.2.2. Tính toán động học thuận cho robot SCARA của đề tài

Trước khi tính toán động học cho robot SCARA ta cần gán các hệ trục tọa độ cho từng khớp theo quy tắc đã đưa ra ở trên.



Hình 2.10 Xây dựng hệ trục tọa độ cho robot SCARA

- Từ hệ trục tọa độ đã xây dựng ta có được bảng ma trận D-H:

Link	$d_i$	$a_i$	$\alpha_i$	$\theta_i$
1	$d_1^*$	0	0	0
2	0	$a_1$	180	$\theta_1^*$
3	0	$a_2$	0	$\theta_2^*$
4	$d_4$	0	0	$\theta_3^*$

Bảng 2.1 Bảng ma trận D-H robot SCARA

- Tính toán ma trận  $A_i$  cho từng khâu:

$$A_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1^* \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



$$A_2 = \begin{bmatrix} c_1 & -s_1 & 0 & a_1 c_1 \\ s_1 & c_1 & 0 & a_1 s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} c_2 & s_2 & 0 & a_2 c_2 \\ s_2 & -c_2 & 0 & a_2 s_2 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} c_3 & -s_3 & 0 & 0 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Tính ma trận biến đổi thuần nhất:

$$T_0^4 = A_1 * A_2 * A_3 * A_4$$

$$T_0^4 = \begin{bmatrix} c_3 c_{12} + s_3 s_{12} & c_3 s_{12} - s_3 c_{12} & 0 & a_1 c_1 + a_2 c_1 c_2 - a_2 s_1 s_2 \\ c_3 s_{12} - s_3 c_{12} & -c_3 c_{12} - s_3 s_{12} & 0 & a_1 s_1 + a_2 c_1 s_2 + a_2 s_1 c_2 \\ 0 & 0 & -1 & d_1 - d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

Từ ma trận biến đổi thuần nhất  $T_0^4$  ta có thể tính được vị trí điểm cuối của đầu gắp chính là:

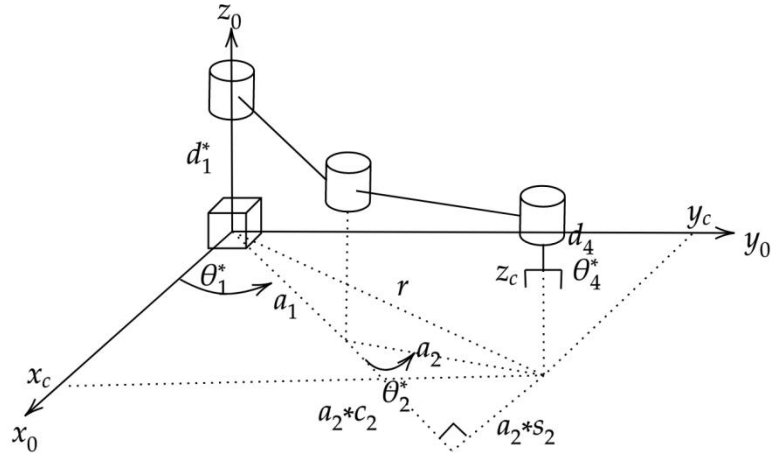
$$O_4(a_1 c_1 + a_2 c_1 c_2 - a_2 s_1 s_2; a_1 s_1 + a_2 c_1 s_2 + a_2 s_1 c_2; d_1 - d_4)$$

Và ma trận hướng của điểm cuối:

$$\begin{bmatrix} n & s & a \end{bmatrix} = \begin{bmatrix} c_3 c_{12} + s_3 s_{12} & c_3 s_{12} - s_3 c_{12} & 0 \\ c_3 s_{12} - s_3 c_{12} & -c_3 c_{12} - s_3 s_{12} & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (2.3)$$

### 2.2.3. Bài toán động học ngược robot SCARA

Có 2 phương pháp chính để giải bài toán động học ngược của tay máy đó là phương pháp đại số và hình học. Trong bài toán này em chọn phương pháp hình học để giải vì lý do tay máy SCARA có cấu hình đơn giản.



Hình 2.11 Giải động học ngược theo phương pháp hình học

Giả sử điểm mà tay máy muốn tới có tọa độ:  $(x_c, y_c, z_c)$  và  $\alpha$  là hướng đầu gấp

Theo định lý pythagore:  $r^2 = x_c^2 + y_c^2$

Áp dụng công thức Cos tính được:  $\cos \theta_2 = \frac{r^2 - a_1^2 - a_2^2}{2 * a_1 a_2}$ ,  $\sin \theta_2 = \pm \sqrt{1 - \cos^2 \theta_2}$

(2.4)

Ta sẽ tìm được 2 góc  $\theta_2$  thỏa mãn trường hợp vậy nên robot sẽ có 2 cấu hình thỏa mãn bài toán động học ngược.

$$\theta_2 = a \tan 2(\cos \theta_2, \pm \sqrt{1 - \cos^2 \theta_2})$$

$$\theta_1 = a \tan 2(x_c, y_c) - a \tan 2(a_1 + a_2 \cos \theta_2, a_2 \sin \theta_2)$$

$$\theta_3 = \theta_1 + \theta_2 - \alpha$$

$$d_1 = z_c + d_4$$

Vậy là theo phương pháp hình học ta đã tính được các xoay  $\theta_1, \theta_2, \theta_3, d_1$  để robot tới được điểm  $(x_c, y_c, z_c)$  và hướng điểm cuối là  $\alpha$  mong muốn.

## 2.3. Động học vận tốc và ma trận Jacobian

Trong phần trước thì ta đã tìm được các phương trình động học thuận động học ngược biểu diễn mối liên hệ giữa vị trí các khớp đối với vị trí điểm cuối và hướng. Tiếp theo ta cần phải xây dựng được mối liên hệ giữa vận tốc dài và vận tốc góc của đầu gấp với vận tốc từng khớp xoay.

Để xây dựng được các mối liên hệ giữa vận tốc các khớp xoay của robot và vận tốc đầu gấp ta cần tính được ma trận jacobian. Trong giải tích vector ma trận jacobian là ma trận chứa các đạo hàm riêng bậc nhất của các hàm giữa hai không gian vector. Jacobian là một trong những đại lượng quan trọng nhất để giải bài toán phân tích và điều khiển chuyển động của robot. Nó xuất hiện trong hầu hết các mặt của bài toán cánh tay robot ví dụ như: dựng quỹ đạo đường đi cho tay máy, xác định điểm kì dị của tay máy, giải bài toán động lực học tay máy...

### 2.3.1. Ma trận đối xứng ghènh

#### 2.3.1.1. Định nghĩa

Ma trận vuông  $S_{3 \times 3}$  là ma trận đối xứng ghènh khi:

$$S + S^T = 0 \quad \text{và} \quad s_{ij} + s_{ji} = 0 \quad \text{với } i, j = 1, 2, 3$$

$$S = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} = S(\omega)$$

Ta có thể coi như S là biểu diễn dạng ma trận của vector  $\omega = [\omega_1 \quad \omega_2 \quad \omega_3]^T$

#### 2.3.1.2. Một số tính chất của ma trận đối xứng ghènh và ma trận quay

Cho a, b là các vector ba chiều, R là ma trận quay 3x3 góc  $\theta, \alpha, \beta$  là các số vô hướng, có thể chứng minh các tính chất sau:

- $S(\alpha a + \beta b) = \alpha S(a) + \beta S(b)$
- $R(a \times b) = Ra \times Rb$
- $RS(a)R^T b = R(a \times R^T b) = Ra \times (RR^T b) = (Ra) \times b = S(Ra)b$

$$- RS(a)R^T = S(Ra)$$

Nếu  $R_{k,\theta}$  là ma trận quay góc  $\theta$  quanh vector đơn vị  $k$  thì:

$$\frac{dR_{k,\theta}}{d\theta} = S(k)R_{k,\theta}$$

Lấy đạo hàm theo thời gian:

$$\frac{dR_{k,\theta}}{dt} = \frac{d\theta}{dt} S(k)R_{k,\theta} = S(\dot{\theta}k)R_{k,\theta} = S(\omega)R_{k,\theta}$$

(2.5)

### 2.3.2. Vận tốc góc đầu công tác

Cho robot  $n$  bậc tự do, hướng đầu công tác biểu thị bởi các tích ma trận quay

$$R_0^n = R_0^1 R_1^2 \dots R_{n-1}^n$$

$$\frac{dR_0^n}{dt} = \frac{dR_0^1}{dt} R_1^2 \dots R_{n-1}^n + R_0^1 \frac{dR_1^2}{dt} R_2^3 \dots R_{n-1}^n + \dots + R_0^1 R_1^2 \dots R_{n-2}^{n-1} \frac{dR_{n-1}^n}{dt}$$

$$\frac{dR_0^n}{dt} = \frac{dR_0^1}{dt} R_1^n + R_0^1 \frac{dR_1^2}{dt} R_2^n + \dots + R_0^{n-1} \frac{dR_{n-1}^n}{dt}$$

Gọi  $\omega_{i-1}^i$  là vận tốc góc khâu  $i$  so với khâu  $i-1$ ,  $\omega_0^n$  là vận tốc góc đầu công tác so với hệ trục gốc theo phương trình (2.5):

$$\frac{dR_0^n}{dt} = S(\omega_0^n)R_0^n$$

$$\frac{dR_{i-1}^i}{dt} = S(\omega_{i-1}^i)R_{i-1}^i$$

Từ đó ta có thể suy ra:

$$\omega_0^n = \omega_0^1 + R_0^1 \omega_1^2 + \dots + R_0^{n-1} \omega_{n-1}^n \quad (2.6)$$

Vận tốc góc đầu công tác so với hệ trục gốc là tổng vận tốc góc các khâu so với hệ tọa độ gốc. Nếu khớp tịnh tiến thì vận tốc góc của khớp bằng 0, do đó vận tốc góc đầu công tác so với hệ trục gốc là tổng vận tốc góc các khớp quay so với hệ tọa độ gốc. Đặt  $\eta_i = 1$  nếu khớp thứ  $i$  là khớp xoay, còn  $\eta_i = 0$  nếu khớp  $i$  là khớp trượt,

$q_i = \theta_i$  là biến khớp,  $z_i^i = (0 \ 0 \ 1)^T$  vector đơn vị trục quay của khớp thứ I trong hệ trục i,  $z_0^i = R_0^i z_i^i$  tương ứng là cột thứ 3 trong ma trận xoay  $R_0^i$

$$\omega_0^n = \eta_1 \dot{q}_1 z_0 + \eta_2 \dot{q}_2 R_0^1 z_1^1 + \eta_3 \dot{q}_3 R_0^2 z_2^2 + \dots + \eta_n \dot{q}_n R_0^{n-1} z_{n-1}^{n-1}$$

$$\omega_n = \begin{bmatrix} \eta_1 z_0 & \eta_2 z_0^1 & \eta_3 z_0^2 \dots & \eta_n z_0^{n-1} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dots \\ \dot{q}_n \end{bmatrix} = J_\omega \dot{q} \quad (2.7)$$

$J_\omega$  là ma trận jacobian vận tốc góc có kích thước  $3 \times n$ ,  $J_\omega = z_0^{i-1}$  nếu i là khớp quay,  $J_\omega = 0$  nếu i là khớp tịnh tiến.

### 2.3.3. Vận tốc dài đầu công tác

Vận tốc dài đầu công tác là đạo hàm vector vị trí theo thời gian

$$p_x = f_1(q_1, q_2, q_3, \dots, q_n)$$

$$p_y = f_2(q_1, q_2, q_3, \dots, q_n)$$

$$p_z = f_3(q_1, q_2, q_3, \dots, q_n)$$

$$\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} \frac{dp_x}{dt} \\ \frac{dp_y}{dt} \\ \frac{dp_z}{dt} \end{bmatrix} = \begin{bmatrix} \frac{df_1}{dq_1} & \frac{df_1}{dq_2} & \dots & \frac{df_1}{dq_n} \\ \frac{df_2}{dq_1} & \frac{df_2}{dq_2} & \dots & \frac{df_2}{dq_n} \\ \frac{df_3}{dq_1} & \frac{df_3}{dq_2} & \dots & \frac{df_3}{dq_n} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dots \\ \dot{q}_n \end{bmatrix} = J_v \dot{q} \quad (2.8)$$

$J_v$  là ma trận jacobian vận tốc dài có kích thước  $3 \times n$  cột

Cách xác định  $J_v$ :

- Trường hợp khớp i là khớp tịnh tiến:

$$J_{vi} = z_0^{i-1} \text{ với } z_0^{i-1} \text{ là hình chiếu của trục } z_{i-1} \text{ lên hệ tọa độ gốc.}$$

- Trường hợp khớp i quay quanh trục  $z_{i-1}$ :

$$J_{vi} = z_0^{i-1} \times (p_0^n - p_0^{i-1})$$

$p_0^n$  là tọa độ đầu công tác so với hệ tọa độ gốc

$p_0^{i-1}$  là tọa độ gốc hệ trục i-1 trong hệ tọa độ gốc

$p_0^n - p_0^{i-1}$  vector nối gốc hệ trục với đầu công tác, biểu thị trong hệ tọa độ gốc

#### 2.3.4. Ma trận Jacobian

Ma trận jacobian là một ma trận có số cột là số bậc tự do và số hàng là 6. Trong đó 3 hàng đầu là  $J_v$  và 3 hàng sau là  $J_\omega$ .

$$J = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix} = [J_1 \quad J_2 \quad \dots \quad J_n]$$

Nếu khớp i là khớp quay:

$$J_i = \begin{bmatrix} z_0^{i-1} \times (p_0^n - p_0^{i-1}) \\ z_0^{i-1} \end{bmatrix} \quad (2.9)$$

Nếu khớp i là khớp tịnh tiến:

$$J_i = \begin{bmatrix} z_0^{i-1} \\ 0 \end{bmatrix} \quad (2.10)$$

Với  $p_0^{i-1}$  là vị trí gốc tọa độ i-1 so với hệ trục gốc, cột thứ 4 của ma trận  $H_0^{i-1}$

$z_0^{i-1}$  là vector trục  $z_{i-1}$  so với hệ trục gốc, là cột thứ 3 của ma trận quay  $R_0^{i-1}$

Sự liên hệ của đạo hàm biến khớp và vận tốc dài, vận tốc đầu công tác:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = J \dot{q} \quad (2.11)$$

Áp dụng tính toán ma trận jacobian cho robot SCARA của đề tài hình 2.9:

$$J = \begin{bmatrix} z_0^0 & z_0^1 \times (p_0^4 - p_0^1) & z_0^2 \times (p_0^4 - p_0^2) & z_0^3 \times (p_0^4 - p_0^3) \\ 0 & z_0^1 & z_0^2 & z_0^3 \end{bmatrix}$$

Do  $z_0^3$  song song với  $(p_0^4 - p_0^3)$  vật nên:  $z_0^3 \times (p_0^4 - p_0^3) = 0$

Thế phương trình 2.2 đã tính trong phần động học thuận ta được:

$$p_0^1 = \begin{bmatrix} 0 \\ 0 \\ d_1 \end{bmatrix} \quad p_0^2 = \begin{bmatrix} a_1 c_1 \\ a_1 s_1 \\ 0 \end{bmatrix}$$

$$p_0^4 = \begin{bmatrix} a_1 c_1 + a_2 c_{12} \\ a_1 s_1 + a_2 s_{12} \\ d_1 - d_4 \end{bmatrix} \quad p_0^3 = \begin{bmatrix} a_1 c_1 + a_2 c_{12} \\ a_1 s_1 + a_2 s_{12} \\ 0 \end{bmatrix}$$

$$z_0 = z_1 = z_2 = [0 \ 0 \ 1], \quad z_3 = z_4 = [0 \ 0 \ -1]$$

$$J = \begin{bmatrix} 0 & -a_1 s_1 - a_2 s_{12} & -a_2 s_{12} & 0 \\ 0 & a_1 c_1 + a_2 c_{12} & a_2 c_{12} & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & -1 \end{bmatrix}$$

### 2.3.5. Điểm kỳ dị của tay máy

Điểm kỳ dị của tay máy là điểm mà tại đó cấu hình của tay máy bị mất bậc tự do dẫn tới giải bài toán động học ngược thường xảy ra vô nghiệm hoặc vô số nghiệm làm cho hệ thống bị mất điều khiển. Hầu hết các cánh tay robot đều có điểm kỳ dị vậy nên ta cần tìm ra điểm kỳ dị đó để tránh cho tay máy mắc phải.

Để xác định điểm kỳ dị của tay máy ta cần giải phương trình  $\det(J) = 0$  cụ thể trong bài toán cánh tay robot SCARA của luận văn ta tìm điểm kỳ dị như sau:

$$\det(J) = 0$$

$$a_2 c_{12} (-a_1 s_1 - a_2 s_{12}) + a_2 s_{12} (a_1 c_1 + a_2 c_{12}) = 0$$

$$a_1 a_2 (-c_{12} s_1 + s_{12} c_1) = 0$$

$$s_2 = 0$$

Suy ra điểm kỳ dị của robot scara là tại điểm:  $\theta_2 = 0^\circ$  hoặc  $\theta_2 = 180^\circ$

## 2.4. Mô phỏng tính toán trên Matlab thư viện robotic toolbox

Phần mềm Matlab là phần mềm cung cấp môi trường tính toán và lập trình, do MathWorks thiết kế. Matlab được Cleve phát minh vào cuối thập niên 1970, ban đầu được viết bằng ngôn ngữ Fortran, cho đến năm 1980, nó chỉ là phần mềm được sử dụng nội bộ trong trường đại học Stanford. Năm 1983, Jack Little một lập trình viên từng học tại trường MIT và Stanford, đã viết lại Matlab bằng ngôn ngữ C, dựng thêm các thư viện phục vụ cho thiết kế hệ thống điều khiển, hệ thống mô phỏng. Jack xây dựng Matlab thành ngôn ngữ lập trình trên cơ sở ma trận (matrix-based programming language).

Phiên bản đầu tiên là Matlab1.0 ra đời năm 1984 viết bằng C cho MS-DOS PC được phát hành đầu tiên tại hội thảo thiết kế và điều khiển IEEE, Las Vegas, Nevada. Ban đầu, Matlab được phát triển để hỗ trợ sinh viên sử dụng 2 thư viện LINPACK và EISPACK dùng cho đại số tuyến tính. Năm 1990, Simulink được gói chung với Matlab. Sau đó là các công cụ hỗ trợ vẽ 2D, 3D, bộ Debugger và GUI.

Thư viện robotic toolbox xây dựng trên Matlab cho đến nay đã có tới mười phiên bản và được phát triển trên hai mươi năm bởi Peter Corke. Thư viện robotic toolbox cũng cấp các hàm và các lớp để biểu diễn hướng và cấu hình của robot trong mặt phẳng 2D và 3D. Bên cạnh đó thì thư viện cũng cung cấp các hàm cho việc tính toán chuyển đổi các kiểu dữ liệu, vector, ma trận biến đổi thuận nhất. Một trong những ứng dụng nổi bật nhất đó nó có thể mô phỏng và tính toán động học, động lực học của tay máy nổi tiếp như một đối tượng trong Matlab. Có nhiều những ví dụ được xây dựng sẵn của các loại tay máy nổi tiếng trên thế giới tích hợp sẵn trong thư viện như là: Robot của Kinova, Universal Robotics, Puma 560, cánh tay Stanford.

### 2.4.1. Lập trình mô phỏng tính toán với Matlab

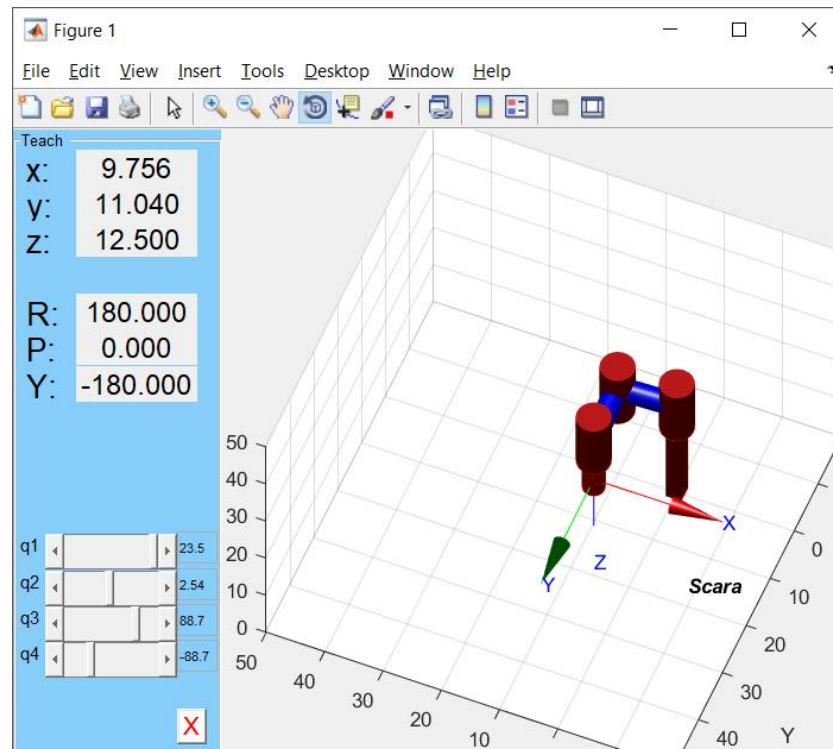
```
a2 = 10.6;  
d4 = 11;  
d1_min = 11;  
d1_max = 23.5;  
syms d1 q1 q2 q3;  
L1 = Link([0 0 0 0 1]);  
L2 = Link([0 0 a1 0 0]);
```



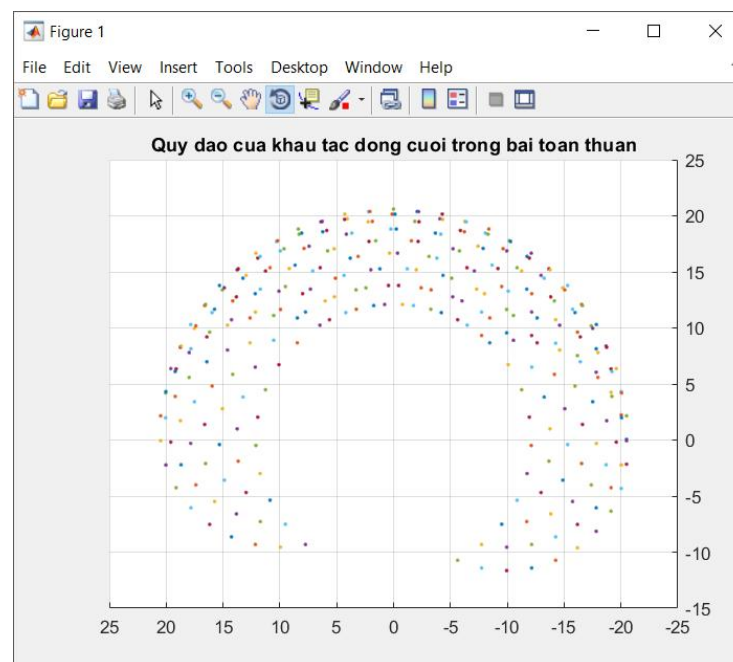
```

L3 = Link([0 0 a2 pi 0]);
L4 = Link([0 d4 0 0 0]);
L1.qlim = [11 23.5];
Scara = SerialLink([L1 L2 L3 L4]);
Scara.name = 'Scara';
%forward kinetic
%calculate joint Ai
A1 = Ai_Mat([0 d1 0 0]);
A2 = Ai_Mat([q1 0 a1 0]);
A3 = Ai_Mat([q2 0 a2 180]);
A4 = Ai_Mat([q3 d4 0 0]);
%calculate homogeneous transformations
T = A1*A2*A3*A4;
subs(T,{d1,q1,q2,q3},{10,7,3,0})
%inverse kinetic
syms px py pz anpha;
c2 = (px^2+py^2-a1^2-a2^2)/(2*a1*a2);
s2_1 = sqrt(1-c2^2);
s2_2 = -sqrt(1-c2^2);
theta2 = atan2(s2_1,c2);
thetal =
atan2(py,px)-atan2(a2*sin(theta2),a1+a2*cos(theta2));
theta4 = thetal + theta2 - anpha;
d1_inv= pz+d4;
%Check result
joint=[d1_inv thetal theta2 theta4];
subs(joint,{px,py,pz,anpha},{10,7,3,0})*360/(2*pi)
hold on;
grid on;
view([-37.5 30])
axis([-20 50 -20 50 0 50])
xlabel('x');
xlabel('y');
xlabel('z');
Scara.teach([11 0 0 0]);

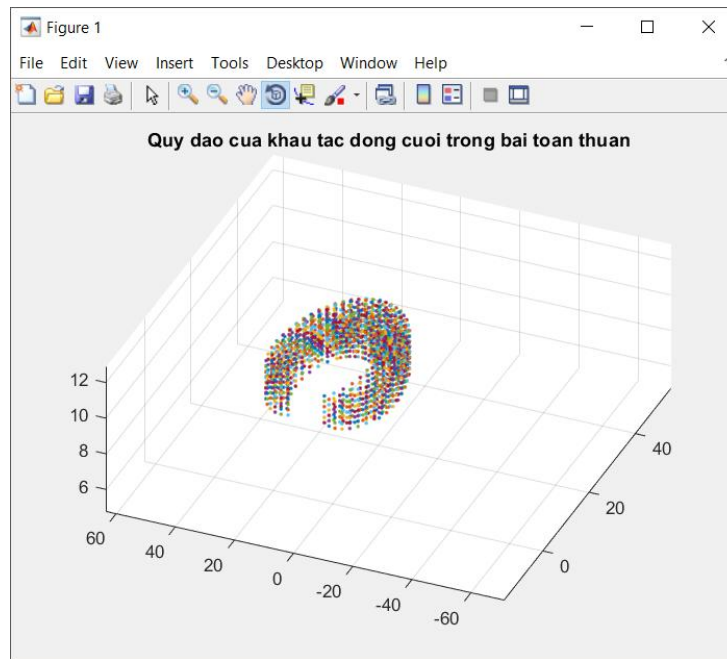
```



Hình 2.12 Kết quả mô phỏng với thư viện robotic toolbox



Hình 2.13 Mô phỏng không gian làm việc của robot trên mặt phẳng

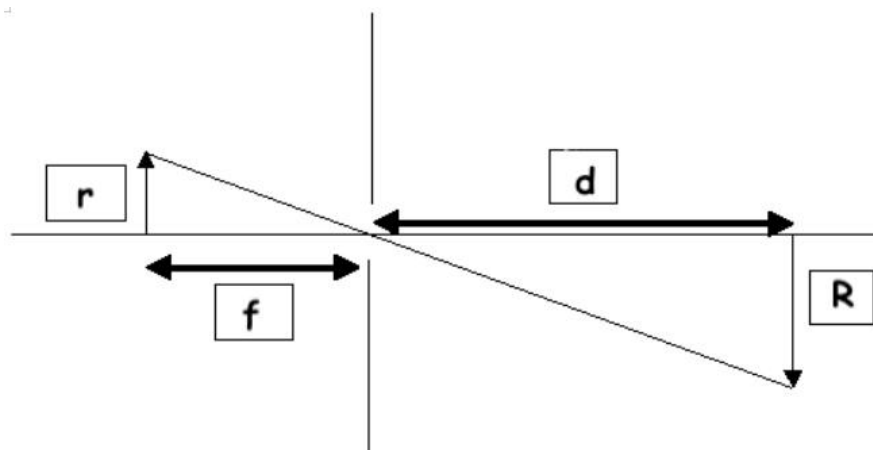


Hình 2.14 Mô phỏng không gian làm việc của robot trong không gian

## 2.5. Đo kích thước vật thể qua kích thước vật tham chiếu bằng xử lý ảnh

Sau khi giải được bài toán điều khiển robot theo động học thuận và động học ngược thì để robot có thể biết được vị trí của vật thể cần gấp trong không gian làm việc của mình ta cần xác định khoảng cách của vật thể tới gốc tọa độ của tay máy.

Ảnh của một vật qua camera:



Hình 2.15 Ảnh của vật qua thấu kính camera

Theo lý thuyết về tam giác đồng dạng ta có thể suy ra được:

$$\frac{r}{R} = \frac{f}{d}$$

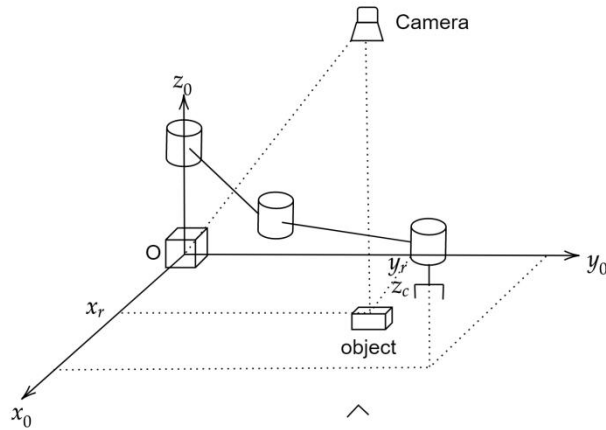
Vậy nên nếu như khoảng cách từ vật thể cần đo tới camera (d) và tiêu cự của camera (f) là không đổi ta có thể suy ra:

$$\frac{r_1}{R_1} = \frac{r_2}{R_2} = \frac{d}{f} \quad (2.12)$$

Trong đó:

- $r_1$  và  $R_1$  là kích thước số điểm ảnh và kích thước thực tế của vật tham chiếu.
- $r_2$  và  $R_2$  là kích thước số điểm ảnh và kích thước thực tế của vật thể cần đo.

Áp dụng lý thuyết vào bài toán của luận văn:



Hình 2.16 Xác định vị trí của vật thể trong không gian

Do Camera và gốc tọa độ của tay máy O là cố định nên ta có thể biết được tọa độ của gốc O trong khung hình của camera:  $x_{oi}, y_{oi}$ . Và tọa độ của vật thể cần đo trong khung hình camera là  $x_{ri}, y_{ri}$ .

Đặt tỉ lệ số điểm ảnh trên kích thước thực tế cen-ti-met là  $\frac{r_1}{R_1} = \text{pixelpercm}$  (calib được bằng vật tham chiếu biết trước kích thước).

Theo phương trình 2.12 ta tính được tọa độ của vật thể trong thực tế:

$$x_r = (x_{ri} - x_{oi}) / \text{pixelpercm}$$

$$y_r = (y_{ri} - y_{oi}) / \text{pixelpercm}$$

## 2.6. Thuật toán nhận dạng vật thể

### 2.6.1. So sánh giữa 1 số thuật toán nhận diện vật thể: SIFT, SURF, HOG

	SIFT	SURF	HOG
Tốc độ	Bình thường	nhanh	Rất nhanh
Độ chính xác	Cao	Cao	Cao
Hướng thay đổi	Có	Có	Không
Kích thước thay đổi	Có	Có	Có
Góc nhìn thay đổi	Tốt	Bình thường	Thấp
Ánh sáng thay đổi	Tốt	Bình thường	Tốt

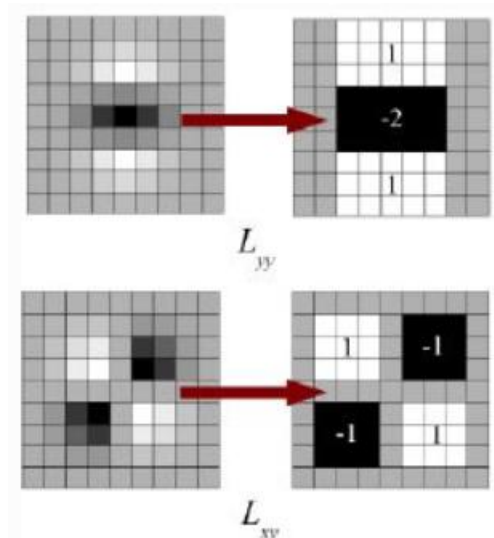
Bảng 2.2 So sánh giữa các thuật toán nhận dạng

Với nhiều đặc tính ưu việt của thuật toán SURF như là có thể nhận dạng vật thể với độ chính xác cao, các góc xoay khác nhau và tốc độ xử lý nhanh thì SURF là phù hợp với yêu cầu của luận văn để nhận diện các vật thể khác nhau chạy trên băng chuyền.

### 2.6.2. Cơ sở lý thuyết thuật toán SURF

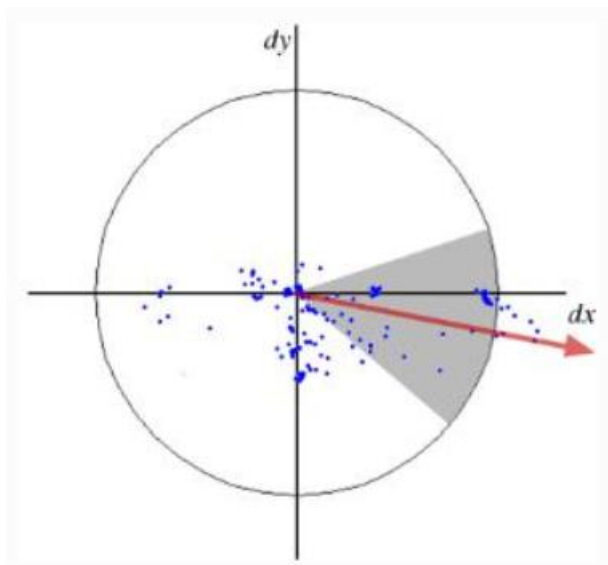
SURF là viết tắt của speed up robust features được giới thiệu vào năm 2006 bởi Herbert Bay, Tinne Tuytelaars, Luc Van Gool. Thuật toán SURF cũng bao gồm những đặc tính như SIFT nhưng với tốc độ xử lý cao hơn đáng kể.

Với SIFT sử dụng xấp xỉ LoG (Laplacian of Gaussian) với DoG (Difference of Gaussian) để tìm kiếm tỉ lệ trong không gian. Còn trong SURF đã cải tiến hơn xấp xỉ LoG với box filter (hình 2.13). Phương pháp xấp xỉ với box filter có ưu điểm làm tăng tốc độ tính toán trong các phép tích chập. Thêm nữa thì SURF xác định ma trận Hessian cho cả hai yếu tố vị trí và tỉ lệ.



Hình 2.17 Phương pháp xấp xỉ box filter

Đối với xác định hướng, SURF sử dụng đáp ứng gợn sóng (Wavelet responses) theo cả 2 chiều ngang và chiều dọc cho các điểm lân cận keypoint. Hướng đại diện cho keypoint được xác định bằng cách tính tổng các đáp ứng bên trong cửa sổ trượt của góc  $60^\circ$ . Trong nhiều ứng dụng ta không cần nhận diện vật thể bị xoay thì có thể bỏ qua phần xác định hướng của keypoint để tăng tốc độ tính toán.



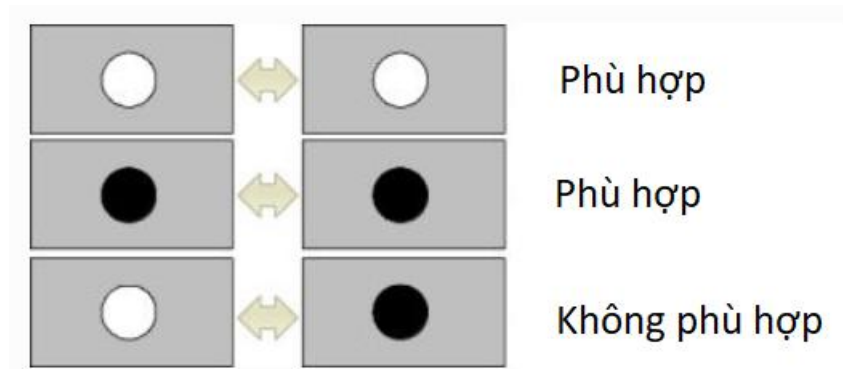
Hình 2.18 Hướng đại diện của keypoint

Để mô tả đặc trưng của vật thể SURF cũng sử dụng đáp ứng gợn sóng (Wavelet responses) theo chiều ngang và dọc. Xét những vùng lân cận xung quanh keypoint có kích thước bằng  $20s \times 20s$  với  $s$  là kích thước của keypoint. Chia vùng đang xét thành  $4 \times 4$  vùng nhỏ. Tại mỗi vùng nhỏ đó tính ta tính đáp ứng gợn sóng tìm ra vector mô tả có dạng:

$v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$  là vector đại diện của đặc trưng SURF 64 chiều.

Khi giảm số chiều của vector đặc trưng tốc độ tính toán sẽ tăng nhưng sẽ làm giảm khả năng nhận diện.

Nếu ứng dụng của chúng ta cần tới khả năng phát hiện có độ chính xác cao thì SURF có thể mở rộng tính toán vector đặc trưng lên tới 128 chiều. Tổng của  $d_x$  và  $|d_x|$  được tính trong 2 trường hợp  $d_x < 0$  và  $d_x \geq 0$  tương tự với  $d_y$  và  $|d_y|$ . Trong bước tìm kiếm những keypoint phù hợp ta chỉ so sánh những nếu chúng có cùng độ tương phản (hình 2.15)



Hình 2.19 Tìm kiếm keypoint phù hợp

## 2.7. Thuật toán theo dõi

Một vấn đề đặt ra của luận văn đó là phải xác định được vật chạy trên băng tải, nếu như ta chỉ sử dụng thuật toán nhận diện vật SURF thì khi chạy trên băng tải thuật toán SURF sẽ rất khó bắt được vật liên tục vì lý do khi vật di chuyển ảnh của vật sẽ không còn rõ nét. Để có thể nhận diện được vị trí của vật một cách liên tục ta cần phải kết hợp thêm một thuật toán theo dõi vật (tracking).

### 2.7.1. So sánh một số thuật toán theo dõi:

	Bossting	MIL	KCF	Median Flow
Ưu điểm	Hoạt động tốt	Hoạt động tốt nếu bị che lấp một phần	Độ chính xác và tốc độ tốt hơn MIL, phát hiện sự cố tốt hơn bossting	Phát hiện rất tốt khi đối tượng ra khỏi video
Nhược điểm	Mất dấu khi đối tượng rời khỏi khung hình	Không phát hiện được khi đối tượng rời khỏi khung hình, không tự phục hồi nếu bị che lấp hoàn toàn	Không tự phục hồi khi đối tượng hoàn toàn bị che lấp	FPS rất thấp do phải tính toán nhiều

Bảng 2.3 So sánh giữa một số thuật toán theo dõi

### 2.7.2. Cơ sở lý thuyết thuật toán KCF:

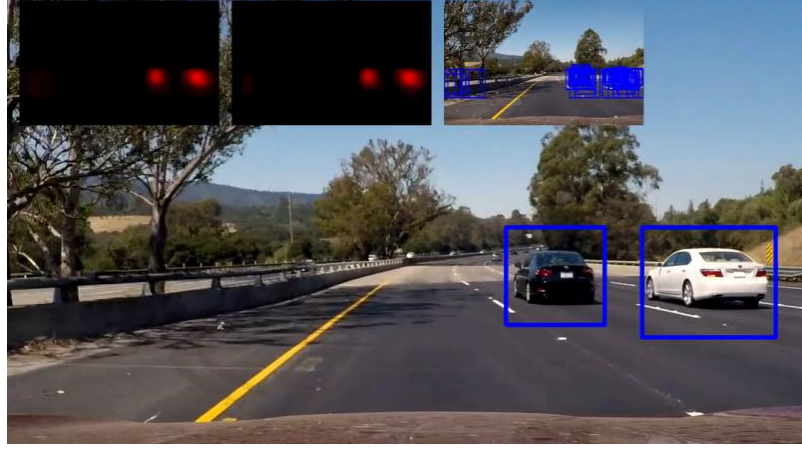
KCF là viết tắt của từ tracking with correlation filters có thể dịch là theo dõi đối tượng bằng bộ lọc tương quan. Một đối tượng được theo dõi thường được xác định ban đầu bởi một hình chữ nhật bao quanh nó. Thuật toán sẽ làm nhiệm vụ theo dõi đối tượng trong video bằng cách cập nhật các chỉ số của khung chữ nhật bao quanh nó.

Ý tưởng của thuật toán KCF đó là đánh giá bộ lọc tương quan nhằm tối ưu quá trình lọc với hình ảnh đầu vào tạo ra được đáp ứng mong muốn. Đáp ứng này thường có hình Gaussian tập chung ở giữa và giảm dần theo khoảng cách.

Bộ lọc tương quan sẽ được học để biết trước đáp ứng đối với đối tượng cần theo dõi. Sau đó thì ảnh của video được cập nhật bộ lọc tương quan sẽ tính đáp ứng để đánh giá vị trí dịch chuyển của đối tượng. Ngay sau đó vị trí mới của đối tượng lại được bỏ vào bộ lọc để học đáp ứng, quá trình này diễn ra liên tục trong các khung hình của video. Do đó thuật toán KCF có thể theo dõi đối tượng di chuyển trong video.



Một điểm đặc biệt của KCF đó là khả năng tính toán hiệu quả. Do mọi thứ được chuyển sang miền fourier để tính toán nên tốc độ tính toán của thuật toán KCF rất nhanh có thể đạt được tới vài trăm khung hình trên giây.



Hình 2.20 Thuật toán theo dõi đối tượng

#### - Bộ lọc tương quan tuyến tính

Tối ưu bộ lọc  $\omega$  bằng phương pháp bình phương cực tiểu:

$$\min(\|X\omega - y\|^2 + \lambda\|\omega\|^2) \quad (2.13)$$

Trong đó  $X$  là một ma trận xoay vòng của ảnh đối tượng. Các hàng của ma trận  $X$  chứa các khả năng đối tượng dịch chuyển qua lại,  $y$  là đáp ứng đầu ra mong muốn,  $\lambda$  là hệ số để hiệu chỉnh.

Sau khi giả phương trình 2.13:

$$\omega = (X^T X + \lambda I)^{-1} X^T y \quad (2.14)$$

Do  $X$  là ma trận xoay vòng,  $\omega$  trong phương trình 2.14 có thể được tính trong miền fourier như sau:

$$\hat{w} = \frac{\hat{x} \odot \hat{y}}{\hat{x}^* \odot \hat{x} + \lambda} \quad (2.15)$$

Trong đó  $\odot$  nhân các phần tử của ma trận với nhau,  $\hat{\cdot}$  là ảnh trong miền fourier,  $*$  là liên hợp phức.

$$\hat{r} = \hat{z}^* \odot \hat{w} \quad (2.16)$$

#### - Bộ lọc tương quan hạt nhân

Bằng các sử dụng thủ thuật hạt nhân (kernel-trick), ví dụ như sắp xếp dữ liệu đầu vào bằng một hàm phi tuyến.  $x \rightarrow \varphi(x)$

$$\omega = \sum_i \alpha_i \varphi(x_i)$$

$$\min_{\alpha} (\|K\alpha - y\|^2 + \lambda \alpha^T K \alpha)$$

K là ma trận hạt nhân với  $k_{ij} = \varphi(x_i)^T \varphi(x_j)$

$$\alpha = (K + \lambda I)^{-1} y$$

$$\hat{\alpha} = \frac{\hat{y}}{\hat{k} + \lambda} \quad (2.17)$$

Đảo với đáp ứng nhanh

$$\hat{r} = \hat{k}^{xz} \odot \hat{\alpha} \quad (2.18)$$

Hạt nhân Gaussian có dạng  $k(x, x') = \exp(-\frac{1}{\sigma^2} \|x - x'\|^2)$

Do đó:

$$k^{xx'} = \exp\left(-\frac{1}{\sigma^2} (\|x\|^2 + \|x'\|^2 - 2F^{-1}(\hat{x}^* \odot \hat{x}'))\right) \quad (2.19)$$

$$k^{xx'} = F^{-1}(\hat{x}^* \odot \hat{x}') \quad (2.20)$$

Bằng cách thay phương trình 2.20 vào 2.17 và 2.18 ta có thể tính được giá trị của bộ lọc tương quan trong phương trình 2.13 và 2.14.

## 2.8. Thuật toán PCA

Phép phân tích thành phần chính (Principal Components Analysis - PCA) là một thuật toán thống kê sử dụng phép biến đổi trực giao để biến đổi một tập hợp dữ liệu từ một không gian nhiều chiều sang một không gian mới ít chiều hơn (2 hoặc 3 chiều) nhằm tối ưu hóa việc thể hiện sự biến thiên của dữ liệu.

Một số ưu điểm của thuật toán PCA:

- Giảm số chiều của không gian chứa dữ liệu khi nó có số chiều lớn, không thể thể hiện trong không gian 2 hay 3 chiều.

- Xây dựng những trục tọa độ mới, thay vì giữ lại các trục của không gian cũ, nhưng lại có khả năng biểu diễn dữ liệu tốt tương đương, đảm bảo độ biến thiên của dữ liệu trên mỗi chiều mới.

- Tạo điều kiện để các liên kết tiềm ẩn của dữ liệu có thể được khám phá trong không gian mới, mà nếu đặt trong không gian cũ thì khó phát hiện vì những liên kết này không thể hiện rõ.

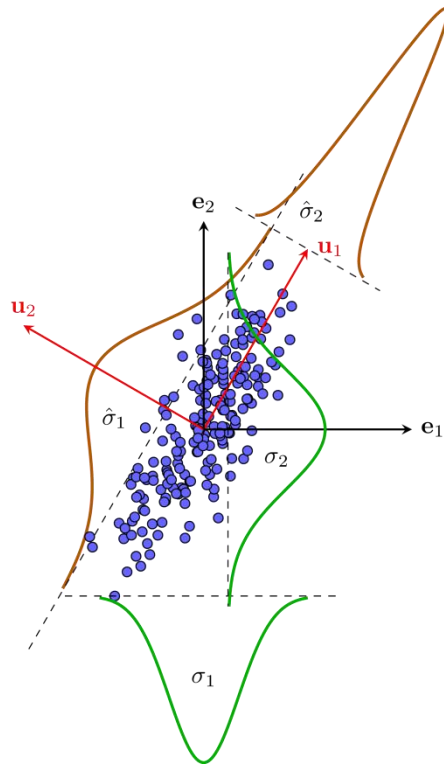
- Đảm bảo các trục tọa độ trong không gian mới luôn trực giao đôi một với nhau, mặc dù trong không gian ban đầu các trục có thể không trực giao.

Giả sử hệ cơ sở trục chuẩn mới là  $\mathbf{U}$  và chúng ta muốn giữ lại  $\mathbf{K}$  tọa độ trong hệ cơ sở mới này. Không mất tính tổng quát, giả sử đó là  $\mathbf{K}$  thành phần đầu tiên:

$$\begin{array}{c}
 \begin{array}{|c|} \hline N \\ \hline D \quad \mathbf{X} \\ \hline \end{array} \\
 \text{Original data}
 \end{array}
 =
 \begin{array}{|c|c|} \hline K & D-K \\ \hline D \quad \mathbf{U}_K & \bar{\mathbf{U}}_K \\ \hline \end{array}
 \times
 \begin{array}{|c|} \hline N \\ \hline K \quad \mathbf{Z} \\ D-K \quad \mathbf{Y} \\ \hline \end{array} \\
 \text{An orthogonal matrix} & \text{Coordinates in new basis}
 \end{array}
 =
 \begin{array}{|c|} \hline K \\ \hline D \quad \mathbf{U}_K \\ \hline \end{array}
 \times
 \begin{array}{|c|} \hline N \\ \hline K \quad \mathbf{Z} \quad D \\ \hline \end{array}
 +
 \begin{array}{|c|} \hline D-K \\ \hline \bar{\mathbf{U}}_K \\ \hline \end{array}
 \times
 \begin{array}{|c|} \hline \mathbf{Y} \\ \hline \end{array}$$

Hình 2.21 Ý tưởng chính của thuật toán PCA

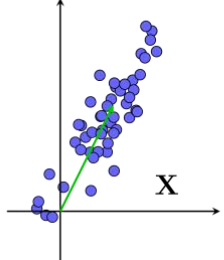
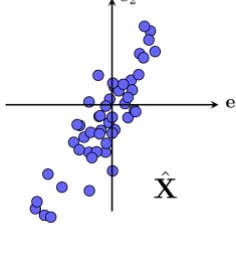
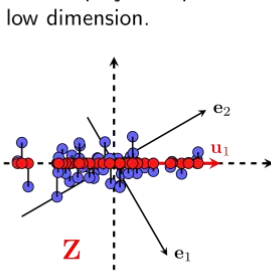
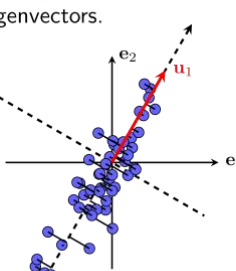
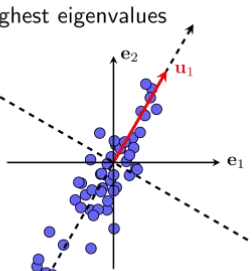
Ta cần phải đi tìm một hệ trục chuẩn mới sao cho trong hệ này, các thành phần quan trọng nằm trong  $\mathbf{K}$  thành phần đầu tiên. Mục đích của PCA là đi tìm ma trận trực giao  $\mathbf{U}$  sao cho phần lớn thông tin được giữ lại ở phần màu xanh  $\mathbf{U}_K \mathbf{Z}$  và phần màu đỏ  $\mathbf{U}_K \mathbf{Y}$  sẽ được lược bỏ và thay bằng một ma trận không phụ thuộc vào từng điểm dữ liệu.



Hình 2.22 PCA dưới góc nhìn thống kê

PCA có thể được coi là phương pháp đi tìm một hệ cơ sở trực chuẩn đóng vai trò một phép xoay, sao cho trong hệ cơ sở mới này, phương sai theo một số chiều nào đó là rất nhỏ, và ta có thể bỏ qua.

### PCA procedure

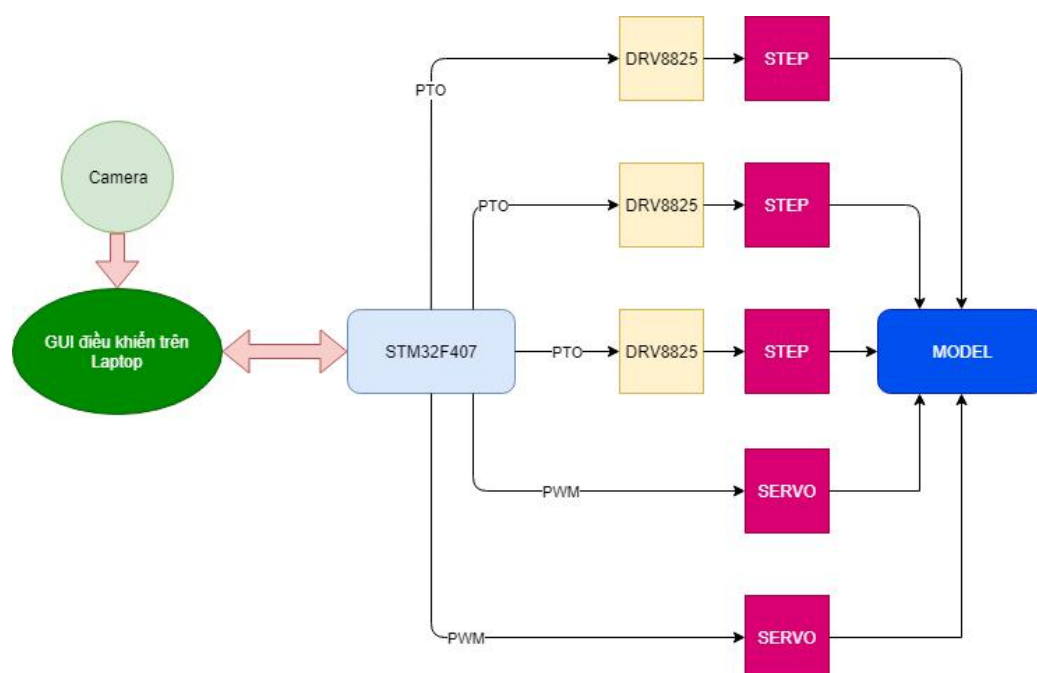
<p>1. Find mean vector</p>  <p><math>\mathbf{X}</math></p>	<p>2. Subtract mean</p>  <p><math>\hat{\mathbf{X}}</math></p>	<p>3. Compute covariance matrix:  <math>\mathbf{S} = \frac{1}{N} \hat{\mathbf{X}} \hat{\mathbf{X}}^T</math></p> <p>4. Computer eigenvalues and eigenvectors of <math>\mathbf{S}</math>:  <math>(\lambda_1, \mathbf{u}_1), \dots, (\lambda_D, \mathbf{u}_D)</math>  Remember the orthonormality of <math>\mathbf{u}_i</math>.</p>
<p>7. Obtain projected points in low dimension.</p>  <p><math>\mathbf{Z}</math></p>	<p>6. Project data to selected eigenvectors.</p> 	<p>5. Pick <math>K</math> eigenvectors w. highest eigenvalues</p> 

Hình 2.23 Các bước thực hiện thuật toán PCA

## Chương 3. XÂY DỰNG MÔ HÌNH

### 3.1. Sơ đồ khối hệ thống

Khối điều khiển chính của hệ thống là máy tính. Máy tính sẽ đọc hình ảnh từ camera về để xử lý. Nếu sau khi xử lý ảnh xác định được vị trí của vật thể trong không gian làm việc của tay máy, chương trình sẽ gửi tín hiệu điều khiển xuống mạch điều khiển qua cổng truyền thông nối tiếp UART. Sau khi nhận được lệnh mạch điều khiển sẽ xuất xung điều khiển các khớp để thực hiện câu lệnh tương ứng và hồi tiếp trạng thái số xung đã xuất được về GUI.



Hình 3.1 Sơ đồ khối hệ thống

Mô hình được xây dựng dựa trên một mô hình mẫu về cánh tay robot SCARA mã nguồn mở và chỉnh sửa lại các chi tiết để phù hợp với yêu cầu đặt ra của đề tài. Các chi tiết của cánh tay robot được thiết kế trên phần mềm SolidWorks và in 3D bằng vật liệu nhựa PLA.

Tổng thể mô hình gồm 1 cánh tay robot, băng tải, camera và máy tính.

- Cánh tay robot scara 4 bậc tự do trong đó 3 khớp đầu sử dụng động cơ bước step để điều khiển. Khớp cuối cùng điều khiển hướng của đầu gắp sử dụng 1 servo mini.

- Băng tải có kích thước 6x50 cm được truyền động bằng động cơ 24VDC. Để điều khiển động cơ tác giả sử dụng một module cầu H 1298n để tăng giảm tốc độ băng tải



Hình 3.2 Băng tải

- Camera logitech C270 được sử dụng để kết nối với máy tính, quan sát trạng thái của hệ thống và sử dụng các thuật toán về xử lý ảnh để xác định vị trí của vật trên băng truyền. Camera có độ phân giải 720p/30s có khả năng tự chỉnh độ sáng phù hợp cho ảnh rõ nét.

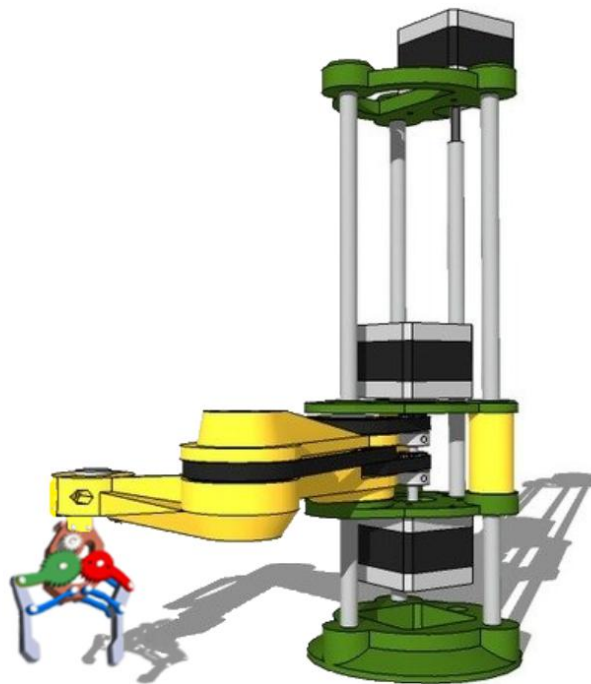


Hình 3.3 Camera logiect C270

## 3.2. Xây dựng mô hình tay máy SCARA

### 3.2.1. Xây dựng mô hình trên máy tính

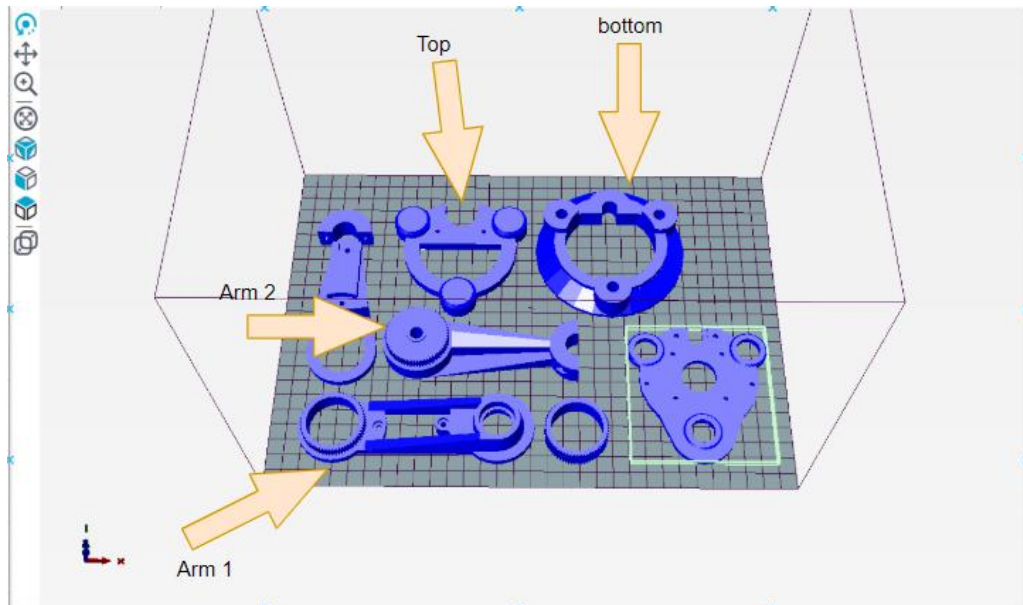
Phần mềm SolidWorks là phần mềm chuyên về thiết kế 3D do hãng Dassault System phát hành dành cho những công ty vừa và nhỏ, đáp ứng được hầu hết các nhu cầu về cơ khí hiện nay. SolidWorks được biết đến với phiên bản 1998 và được du nhập vào nước ta từ phiên bản 2003 cho đến nay thì đã có phiên bản 2019. Phần mềm này đã phát triển đồ sộ về các thư viện cơ khí, không những vậy thì nó còn là một công cụ thiết yếu cho các ngành như, đường ống, kiến trúc, trang trí nội thất, mỹ thuật, ... Với giao diện dễ sử dụng, phù hợp cho thiết kế các bản vẽ cơ khí 2D, 3D SolidWorks là sự lựa chọn hàng đầu cho các hãng sản xuất cũng như các kỹ sư cơ khí nhằm vẽ và mô phỏng hệ thống.



Hình 3.4 Hình ảnh 3D của mô hình

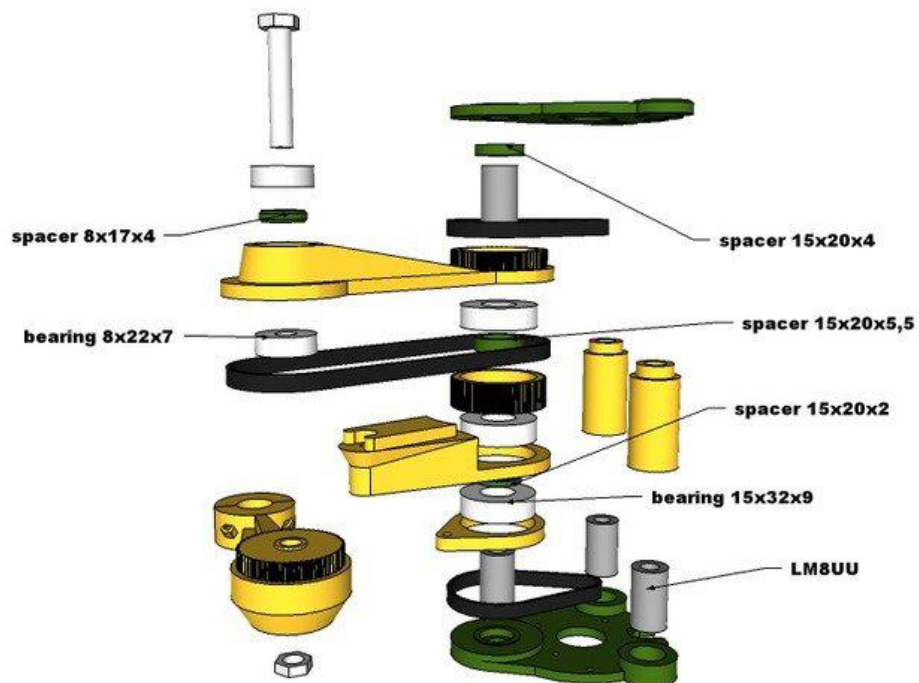
- Cánh tay có độ dài khớp vai 10 cm, khớp khuỷu tay 10 cm, chiều cao 35 cm.
- Khớp vai nối với trục động cơ bằng dây đai 160 mm 80 răng.
- Khớp khuỷu tay nối với trục động cơ bằng 2 dây đai 320 mm 160 răng và 160 mm 80 răng.

- Khớp trượt được truyền động bằng vítme đường kính ren 8mm chiều dài 30 cm và các bạc đạn trượt đường kính 8mm.



Hình 3.5 ảnh 3D của các chi tiết

Các chi tiết 3D sau khi được in ra sẽ lắp ráp với các chi tiết cơ khí để tạo thành một cánh tay robot hoàn chỉnh.

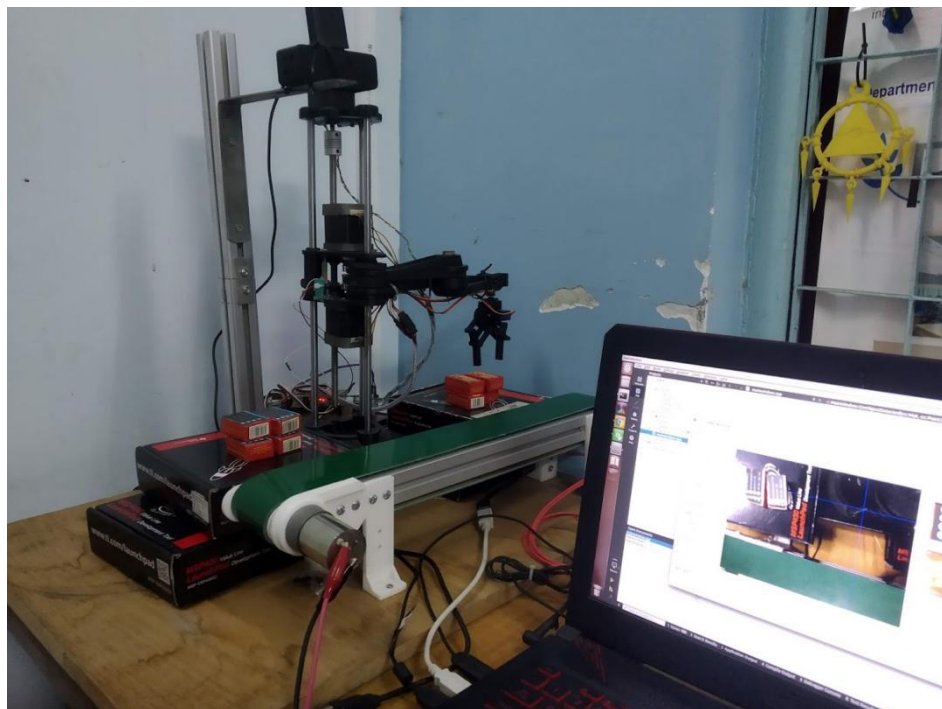
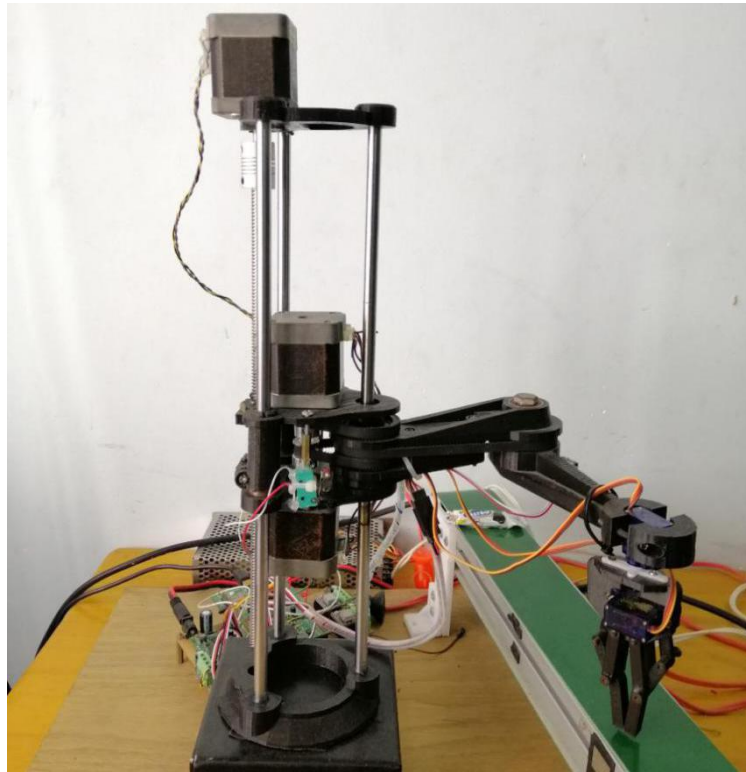


Hình 3.6 Ghép nối các chi tiết



### 3.2.2. Hình ảnh thực tế của mô hình

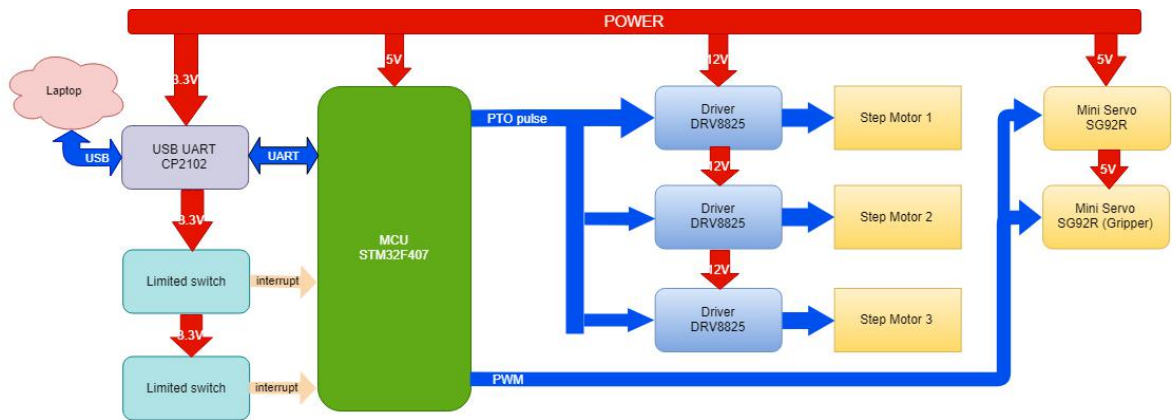
Băng tải được đặt vị trí cố định và song song với trục y của cánh tay robot. Camera được vuông góc với mặt phẳng có góc nhìn từ trên xuống, vị trí của camera cũng không thay đổi trong quá trình làm việc của hệ thống.



Hình 3.7 Hình ảnh toàn bộ hệ thống

### 3.3. Thiết kế mạch điều khiển

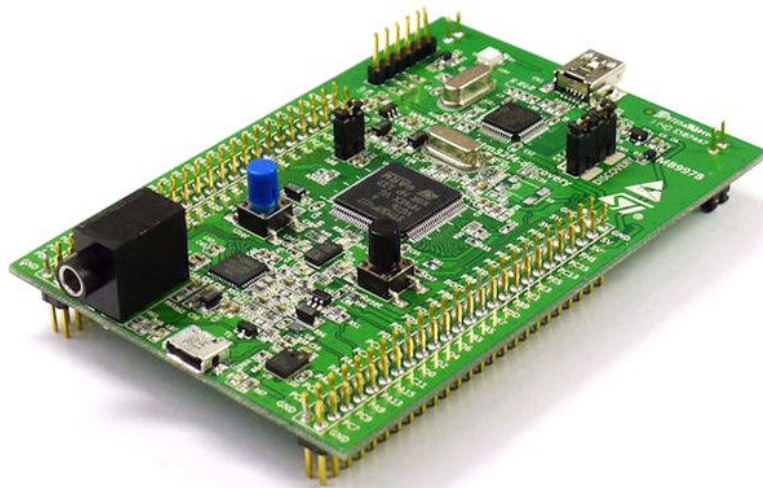
Sơ đồ kết nối phần cứng:



Hình 3.8 Sơ đồ kết nối phần cứng

#### 3.3.1. Kit điều khiển STM32F407 discovery

Kit STM32F407 sử dụng chip điều khiển chính là STM32F407VGT6 lõi ARM cortex M4 32-bit hiệu suất cao. Chip STM32F407VGT6 hoạt động ở tần số lên đến 168MHz và có bộ hỗ trợ tính toán dấu chấm động. Kích thước bộ nhớ chương trình 1MB, SRAM 192KB, hỗ trợ các chuẩn giao tiếp CAN, I2C, SPI, UART, USART, USB. Ngoài ra thì chip còn hỗ trợ 3 bộ ADC 12 bit, 12 timer 16 bit trong đó có hai bộ phát xung PWM điều khiển động cơ.



Hình 3.9 Board Discovery STM32F4

Kit STM32F407 được ứng dụng rộng rãi trong nhiều lĩnh vực như: điều khiển và lái động cơ, trong các thiết bị y tế, trong công nghiệp (PLC, biến tần), trong máy in,...

Với khả năng xử lý nhanh, tốc độ xung clock lớn, hỗ trợ DMA cũng như môi trường lập trình thân thiện tác giả đã quyết định chọn Kit STM32F407 làm vi điều khiển chính cho mô hình cánh tay robot.

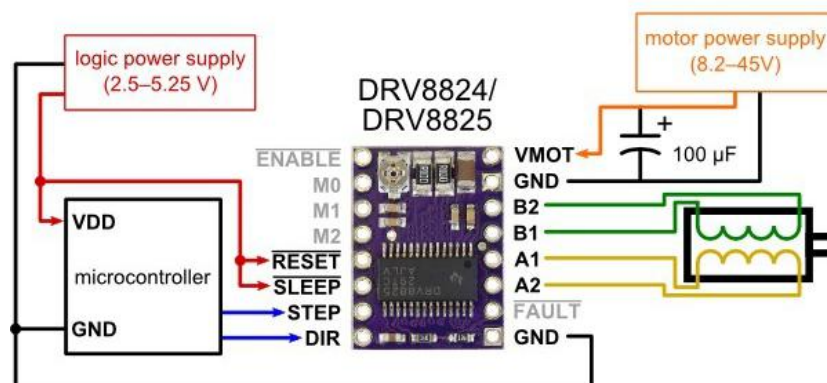
### 3.3.2. Driver điều khiển động cơ bước DRV8825

Driver DRV8825 là một driver đang được sử dụng phổ biến hiện nay. Với giá thành thấp nhưng có nhiều tính năng như một driver chuyên nghiệp như: điều chỉnh dòng giới hạn, vi bước (1/32 bước), bảo vệ quá dòng, bảo vệ quá nhiệt,...

Điện áp hoạt động rộng (từ 8.2V đến 45V) có thể đạt được dòng đỉnh gần 1.8A trên mỗi pha mà không cần tản nhiệt. Driver có cấu hình chân giống với nhiều module khác, vì vậy nó có thể dùng thay thế cho những ứng dụng khác nhau.

Thông số kỹ thuật:

- Điện áp cung cấp: 8.2 - 45 VDC.
- Dòng trung bình: 1.5A, dòng đỉnh lên tới 2.5A.
- Nhiều độ phân giải: full step, half step, 1/4 step, 1/8 step, 1/16 step, 1/32 step.
- Điện áp điều khiển: 3.3V và 5V.
- Tự động tắt khi quá dòng, quá nhiệt.
- Bảo vệ ngắn mạch, bảo vệ quá tải.
- Mạch 4 lớp, 2 lớp phủ đồng giúp tản nhiệt tốt.

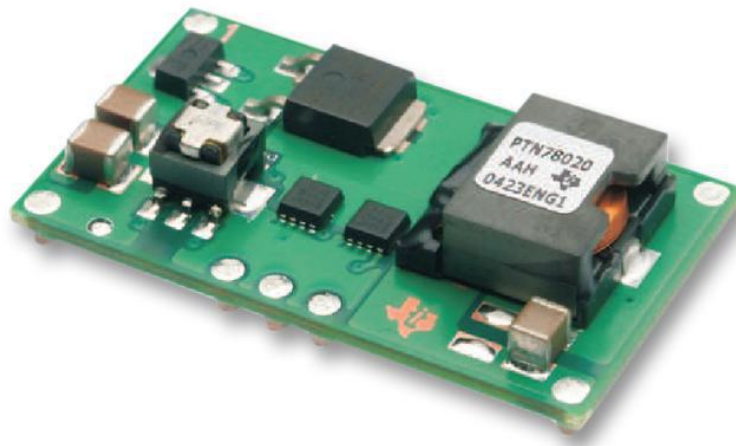


Hình 3.10 Kết nối DRV8825 với động cơ và vi điều khiển

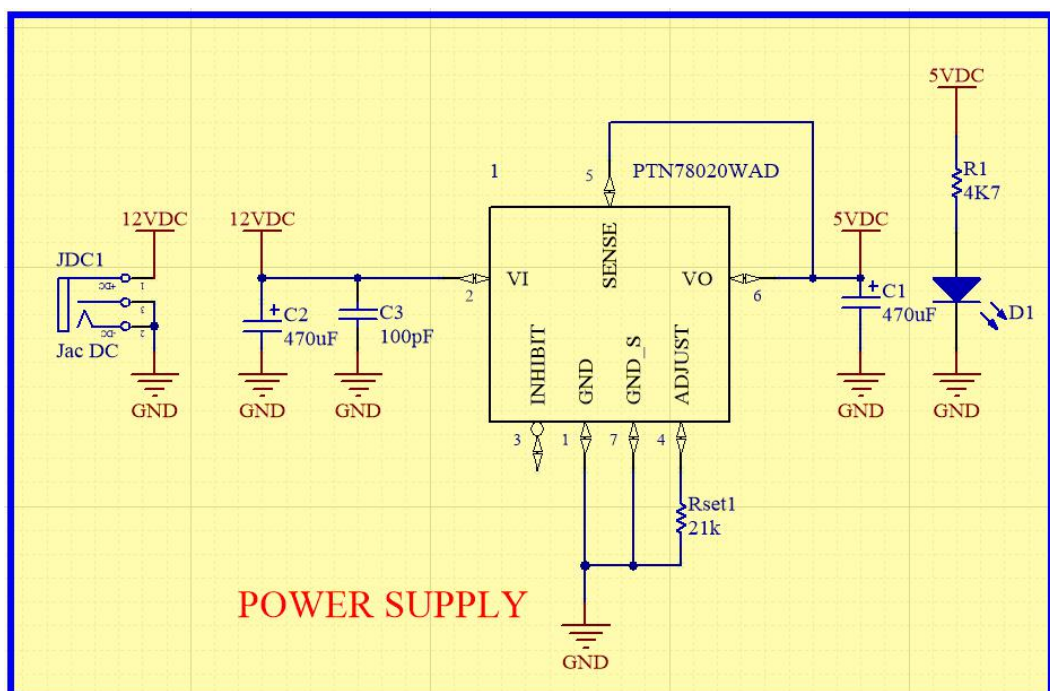
### 3.3.3. Thiết kế mạch điều khiển mô hình

Mạch điều khiển sẽ có nhiệm vụ điều khiển 3 động cơ bước stepper, 2 servo là giao tiếp với máy tính qua UART, xử lý các tín hiệu từ công tắc hành trình.

Khối nguồn sử dụng module chính là PTN78020WAD là một module nguồn xung của TI sản xuất. Với khả năng tạo dòng điện lên tới 6A, tầm điện áp ngõ vào rộng từ 7V - 36V. Điện áp ngõ ra có thể điều chỉnh được trong khoảng 2.5V - 12.6V, hiệu suất cao (lên tới 96%).



Hình 3.11 Module PTN78020WAD

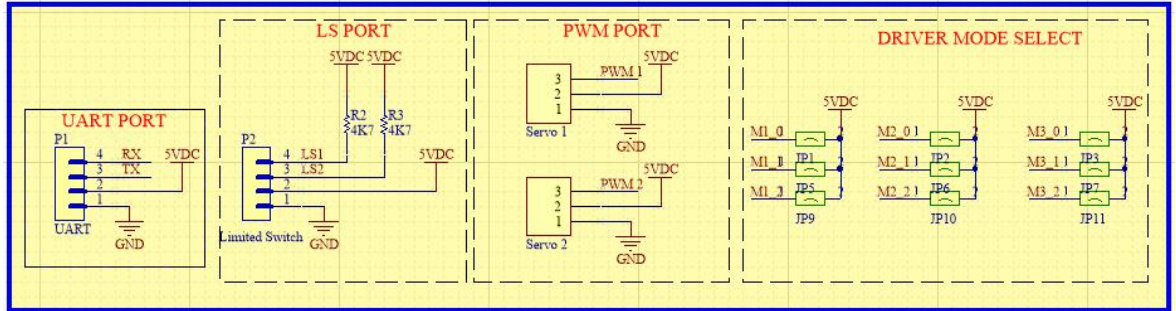


Hình 3.12 Khối nguồn



Các khối PWM dùng để xuất tín hiệu điều khiển servo, UART giao tiếp với máy tính thông qua module CP2102.

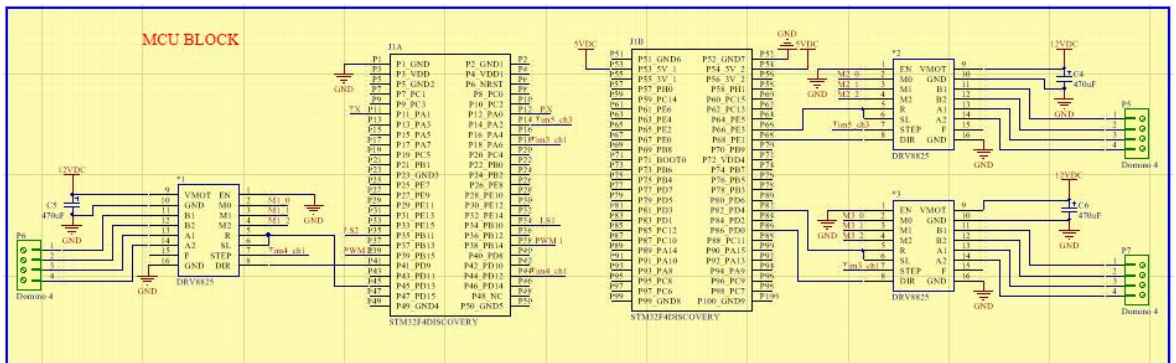
Để cấu hình các chế độ vi bước của động cơ step tác giả sử dụng một khối DRIVER MODE SELECT gồm các jumper để chọn chế độ cho từng động cơ step.



Hình 3.13 Các khối tín hiệu

LS PORT là khối để kết nối với công tắc hành trình, 2 chân này sẽ được kéo lên mức 1. Nếu công tắc hành trình bị tác động thì tín hiệu sẽ chuyển về mức 0. Đọc trạng thái của 2 chân LS1 và LS2 sẽ cho ta biết được tay máy đã về home hay chưa.

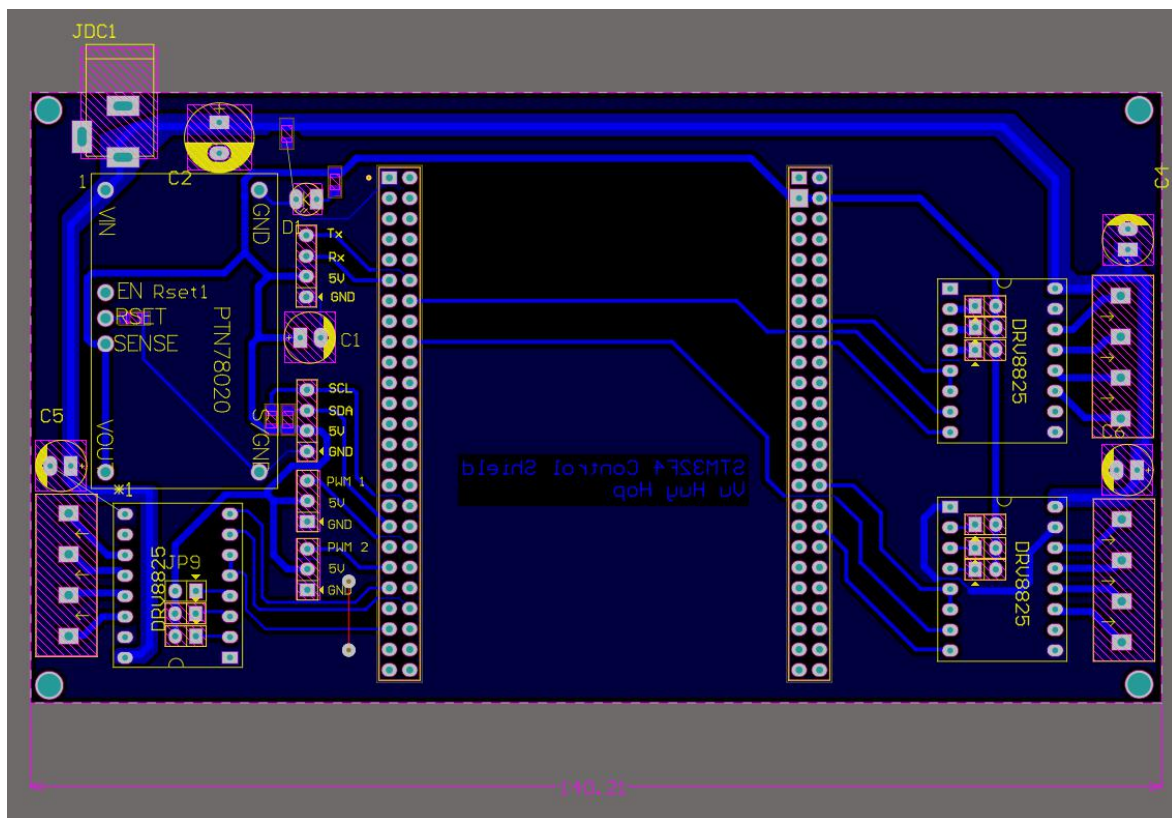
Khối Vi điều khiển chính gồm Kit STM32F407 kết nối với driver 8825 và động cơ stepper. Sử dụng 3 bộ timer mới 3,4,5 để điều khiển 3 động cơ step, 1 bộ timer để điều khiển 2 servo.



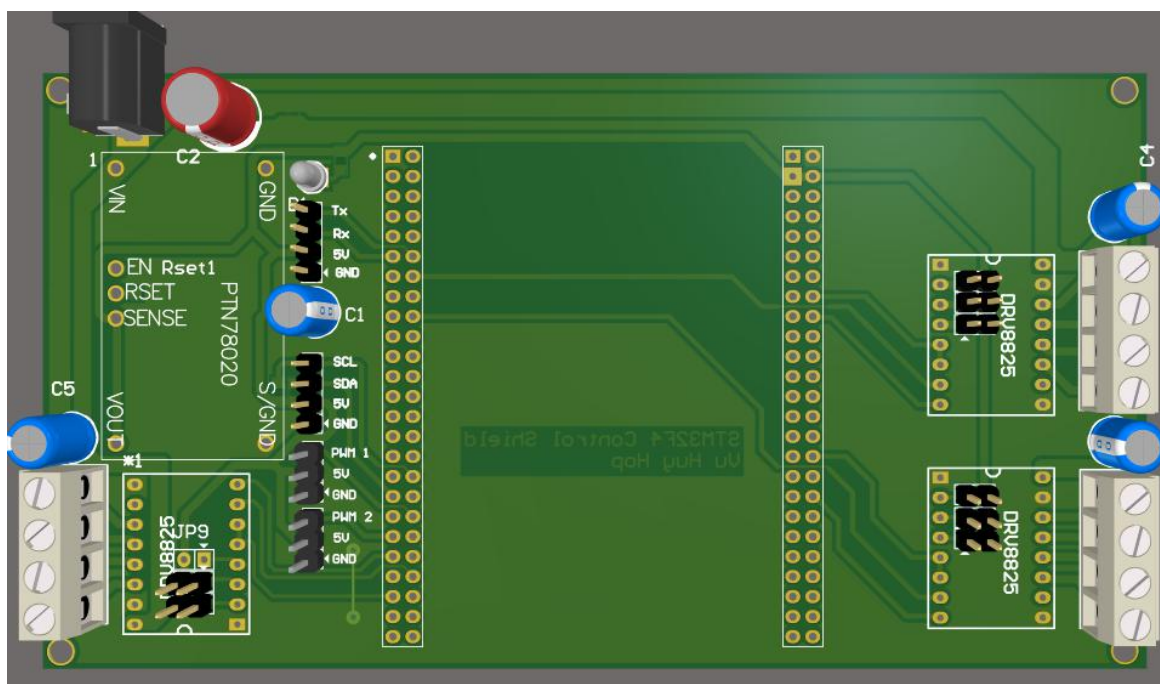
Hình 3.14 Khối MCU

Sau khi thực vẽ xong sơ đồ nguyên lý ta chuyển qua bước layout mạch, khối vi điều khiển sẽ được đặt ở chính giữa và xung quanh là cách driver điều khiển động cơ.

Đường mạch 12V chịu dòng lớn nên có kích thước lớn hơn các đường dây tín hiệu.



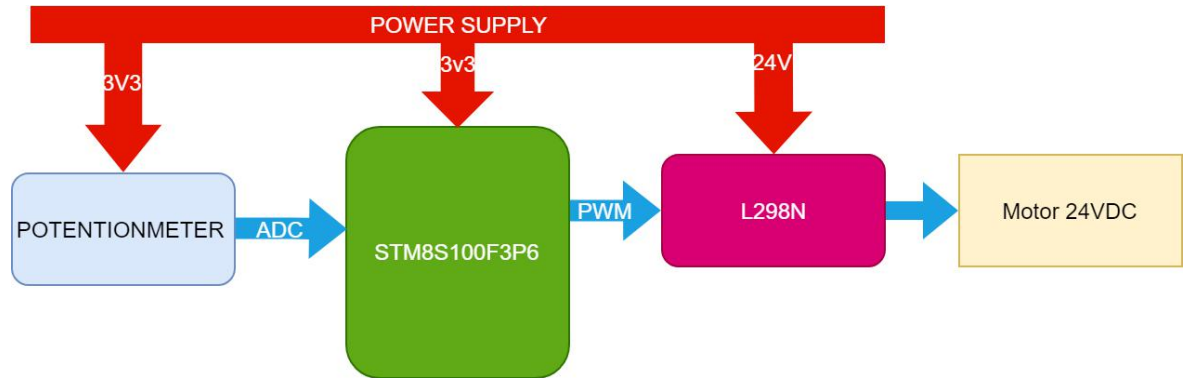
Hình 3.15 Hình ảnh PCB của mạch



Hình 3.16 Hình ảnh 3D của mạch

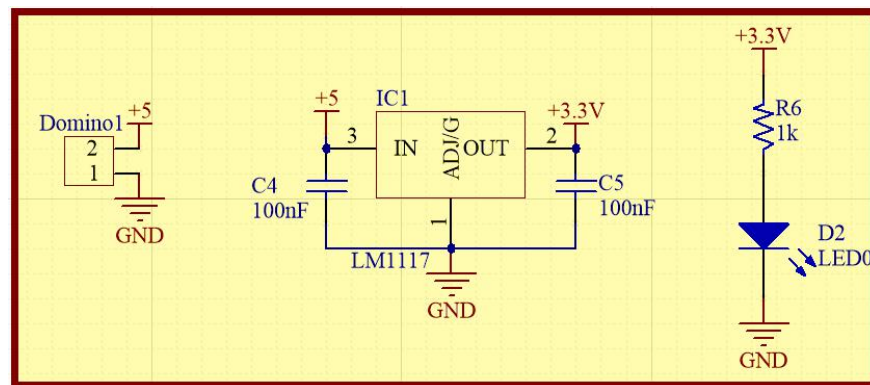
### 3.3.4. Thiết kế mạch điều khiển bằng tải

Mạch điều khiển bằng tải sử dụng chip Stm8s với ứng dụng đọc giá trị từ biến trở và xuất tín hiệu PWM điều khiển mạch cầu H L298n.



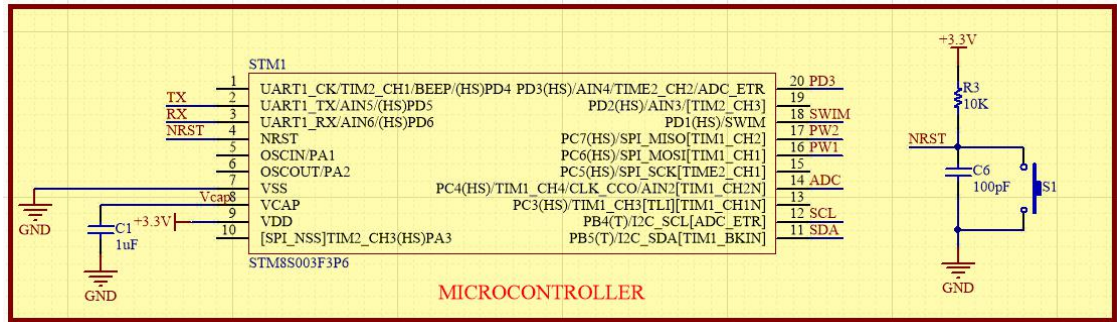
Hình 3.17 Sơ đồ khối mạch điều khiển bằng tải

Khối nguồn sử dụng chip LM1117 hạ áp từ 5v xuống 3v3 cấp cho vi điều khiển.



Hình 3.18 Khối nguồn mạch điều khiển bằng tải

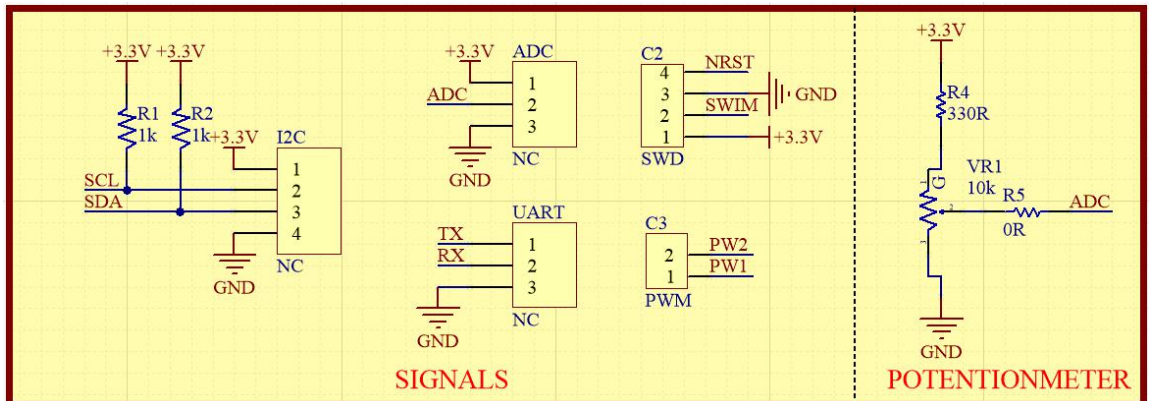
Vi điều khiển sử dụng điều khiển mạch là chip Stm8s100f3p6 chạy được ở tần số xung clock tối đa 16MHz. Chip stm8s100f3p6 có thể tầm điện áp hoạt động rộng (3V-5.5V). Hiện nay thì chip stm8s100f3p6 được sử dụng rất nhiều vì giá thành rẻ và phù hợp cho nhiều ứng dụng đơn giản.



Hình 3.19 Khối vi điều khiển stm8s

Khối SIGNALS bao gồm các chân giao tiếp I2C, UART và 1 số chân nạp code của vi điều khiển. Các khối có ký hiệu NC ở bên dưới là các khối mở rộng dùng để phát triển thêm chứ không dùng trong đề tài này (hình 3.19).

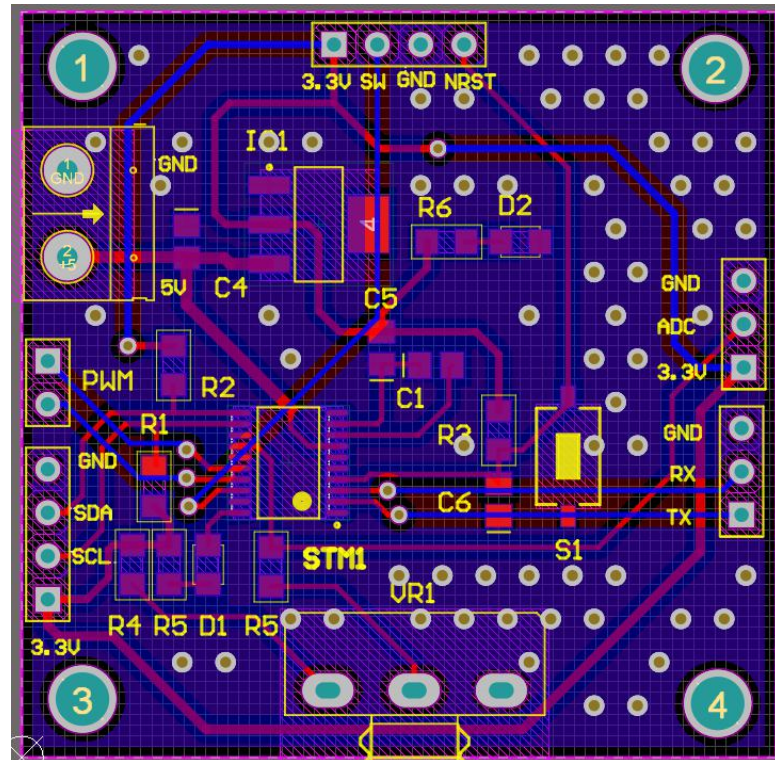
Khối *POTENTIONMETER* chứa 1 biến trở 10K. Ba chân của biến trở sẽ được nối theo thứ tự: 1 chân lên nguồn VCC, 1 chân xuống đất GND, 1 chân nối với chân ADC của vi điều khiển. Khi đọc các giá trị trên chân ADC ta sẽ xác định được biến trở có đang xoay hay không.



Hình 3.20 Một số khối tín hiệu mạch điều khiển bằng tải

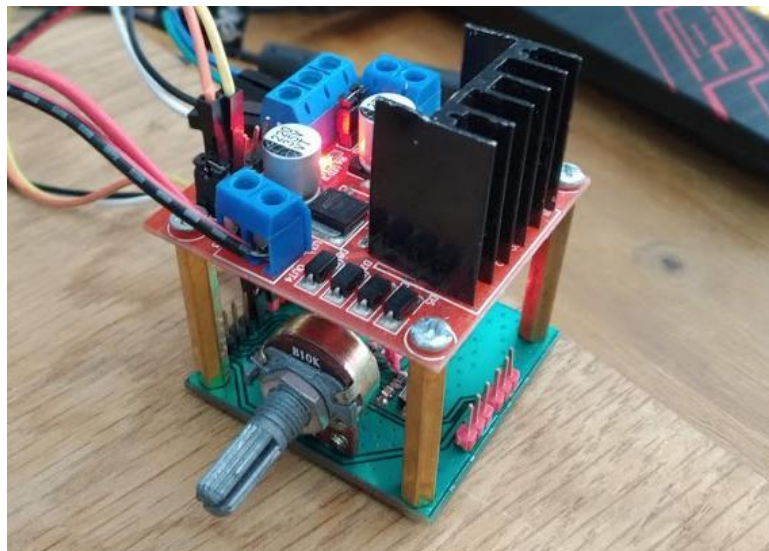


Sau khi hoàn thành sơ đồ nguyên lý chuyển qua layout mạch, mạch sẽ có kích thước nhỏ gọn và vừa với kích thước mạch cầu H L298N.



Hình 3.21 Hình ảnh sau khi layout của mạch

Mạch có 2 lớp và được đặt gia công tại công ty TNHH Kha Thành. Hình ảnh thực tế của mạch sau khi hàn linh kiện.

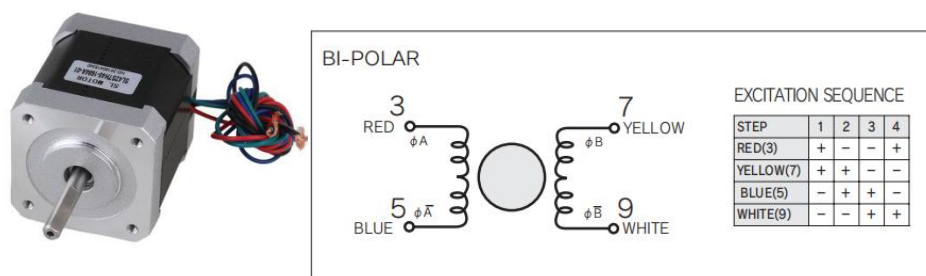


Hình 3.22 Hình ảnh thực tế của mạch điều khiển bằng tải

### 3.3.5. Động cơ điều khiển

#### - Động cơ bước

Động cơ KH42M2 là loại động cơ bước kiểu hỗn hợp có 6 dây ra và có thể cấu hình bằng cả 2 cách là đơn cực và lưỡng cực. Trong đề tài này em sử dụng cấu hình động cơ theo kiểu lưỡng cực dùng 4 dây đỏ, xanh, vàng và trắng để điều khiển.



Hình 3.23 Động cơ bước

Thông số kỹ thuật động cơ bước:

M O D E L	KH42KM2			
	SINGLE SHAFT		-901	-951
	DOUBLE SHAFT		-911	-961
Cấu hình của driver	—	Đơn cực		Lưỡng cực
Số lượng cuộn dây	—	2		2
Góc điều khiển	deg/step	1.8		1.8
Điện áp	V	3.72		2.76
Dòng điện	A/PHASE	1.2		1.2
Trở kháng cuộn dây	$\Omega$ /PHASE	3.1		2.3
Cách điện	mH/PHASE	3.1		4.0
Lực giữ	mN m	340		403
	oz in	48		57
Lực tự giữ	mN m	19.6		19.6
	oz in	2.8		2.8
Quán tính ROTOR	g cm <sup>2</sup>	85		85
	oz in <sup>2</sup>	0.46		0.46
Trọng lượng	g	360		360
	lb	0.79		0.79
Điện trở cách điện	—	500VDC	100M $\Omega$ min.	
Độ cách điện	—	500VAC	50HZ	1min.
Dải nhiệt độ hoạt động			0 to 50	
Cho phép nhiệt độ tăng	K		70	

Bảng 3.1 Thông số kỹ thuật động cơ bước

### - Động cơ mini servo MG90S

Động cơ servo MG90S là bản nâng cấp của servo 9G, sử dụng răng kim loại có độ bền và moment xoắn mạnh hơn. Động cơ được tích hợp sẵn driver điều khiển bên trong nên có thể dễ dàng điều khiển xoay đúng góc với độ rộng xung PWM từ 1ms-2ms. Động cơ MG90S thì đang được ứng dụng rộng rãi phù hợp cho những dự án, sản phẩm như: máy bay mô hình, máy bay RC, robot...

Thông số kỹ thuật:

- + Điện áp hoạt động: 4.8V
- + Tốc độ: 0.1 sec/60 độ (4.8V)
- + Lực kéo: 1.8kg/cm cho 4.8V
- + Nhiệt độ hoạt động trong: 0°C-55°C
- + Dead band: 1us
- + Kích thước: 23x12.2x27mm



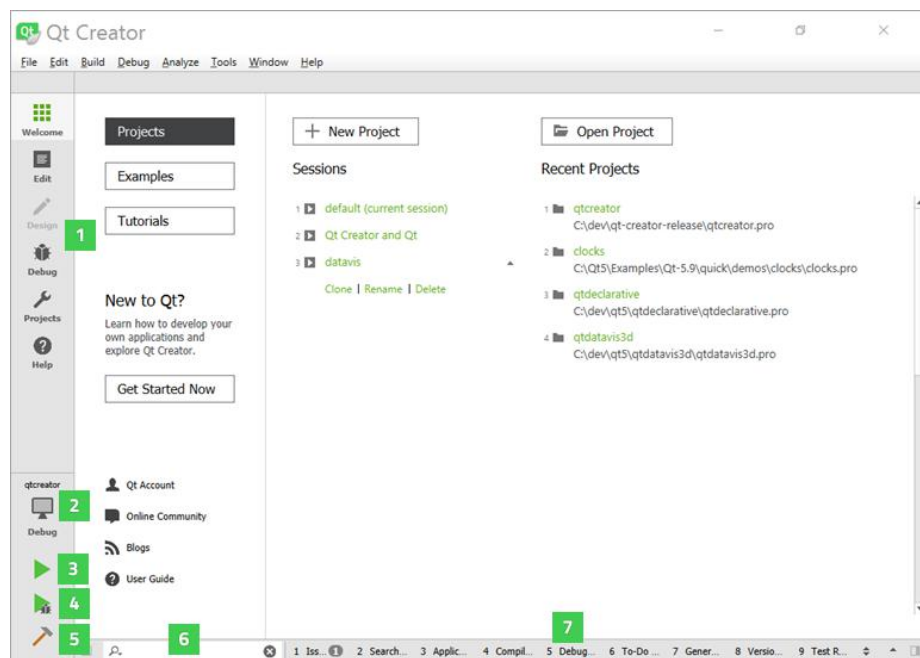
Hình 3.24 Động cơ servo MG90S

## Chương 4. GIAO DIỆN MÁY TÍNH VÀ GIẢI THUẬT ĐIỀU KHIỂN

### 4.1. Thiết kế giao diện điều khiển trên máy tính bằng Qt Creator

Qt Creator là một tiện ích đa nền tảng làm việc thông qua môi trường phát triển tích hợp (viết tắt là IDE) hỗ trợ các nhà phát triển tạo ứng dụng cho nền tảng máy tính và thiết bị di động.

Mặc dù có nhiều tham số riêng nhưng Qt Creator mang đến một giao diện rõ ràng, đơn giản, cho phép người dùng thiết lập một dự án mới bằng cách sử dụng phương pháp từng bước tiện dụng, đặc biệt đối với người chưa có kinh nghiệm vì họ sẽ được trợ giúp trong toàn bộ quá trình.



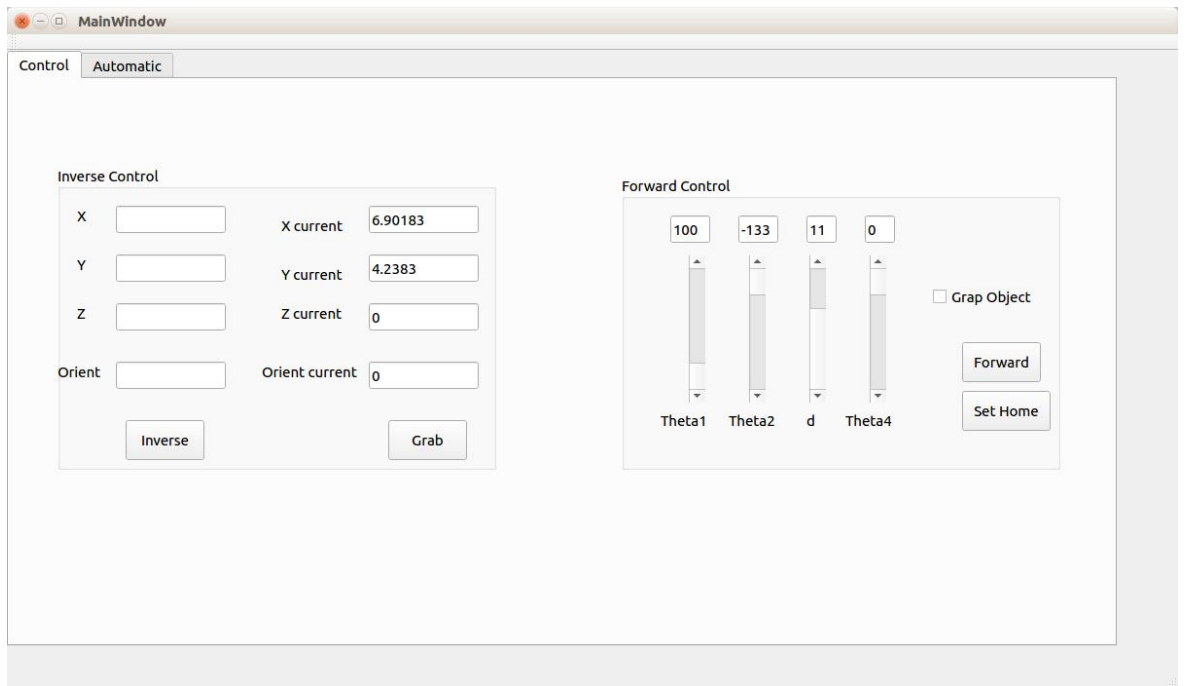
Hình 4.1 Giao diện Qt creator

Một số tính năng mà phần mềm điều khiển đã xây dựng được:

- Giao tiếp được với mạch điều khiển thông qua cổng com
- Điều khiển động học thuận, động học ngược của cánh tay
- Dò điểm home, calib camera
- Truyền ảnh từ camera về GUI
- Chạy các thuật toán xử lý ảnh trên thư viện Opencv

#### 4.1.1. Hình ảnh giao diện đã thiết kế

- Bảng điều khiển bằng tay:



Hình 4.2 Hình ảnh bảng điều khiển bằng tay

Trong bảng điều khiển này gồm 2 khối điều khiển chính là *Inverse control* và *Forward control*.

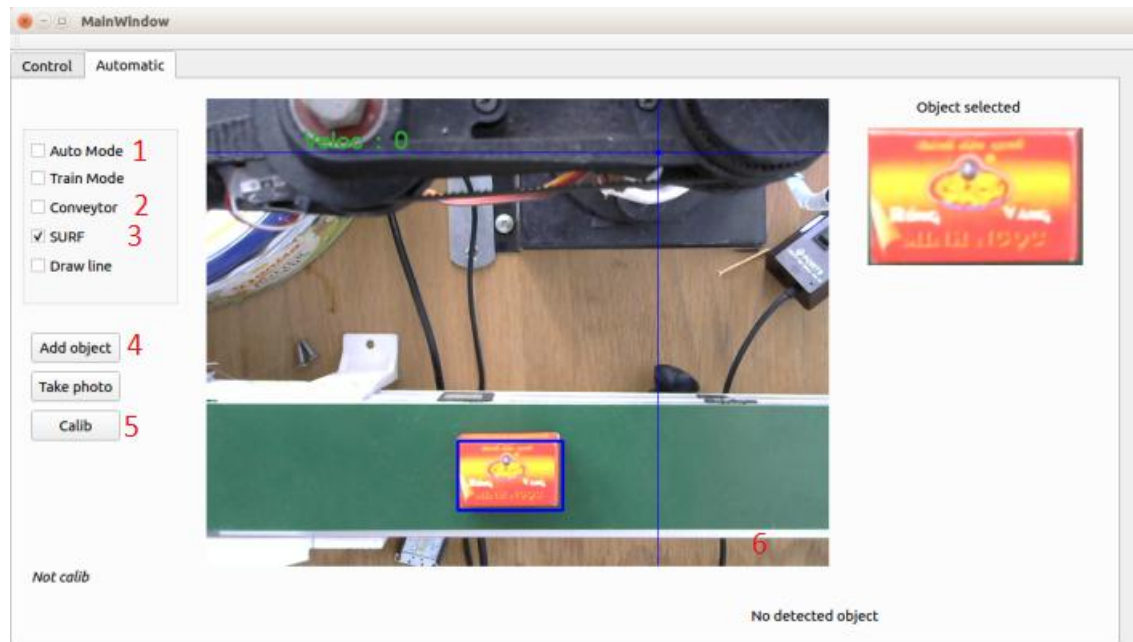
- Đối với khối *Inverse control* ta chỉ cần nhập vào tọa độ x,y,z mà tay máy muốn tới. Chương trình sẽ giải các phương trình động học ngược để tính ra các khớp cần xoay và gửi lệnh xuống mạch điều khiển. Mạch điều khiển nhận được lệnh sẽ xuất các xung PTO điều khiển động cơ tới vị trí mong muốn.

- Ngược lại đối với khối *Forward control* ta sẽ điều khiển từng khớp của robot và chương trình sẽ giải các phương trình động học thuận để tính ra tọa độ hiện tại của tay máy hiển thị lên GUI các giá trị: *x current*, *y current*, *z current*, *Orient current*.

- Ngoài ra trong bảng điều khiển này còn có 2 nút đặc biệt đó là *Grab* để gấp vật và *Set Home* để dò điểm home của robot.



- Bảng điều khiển tự động:



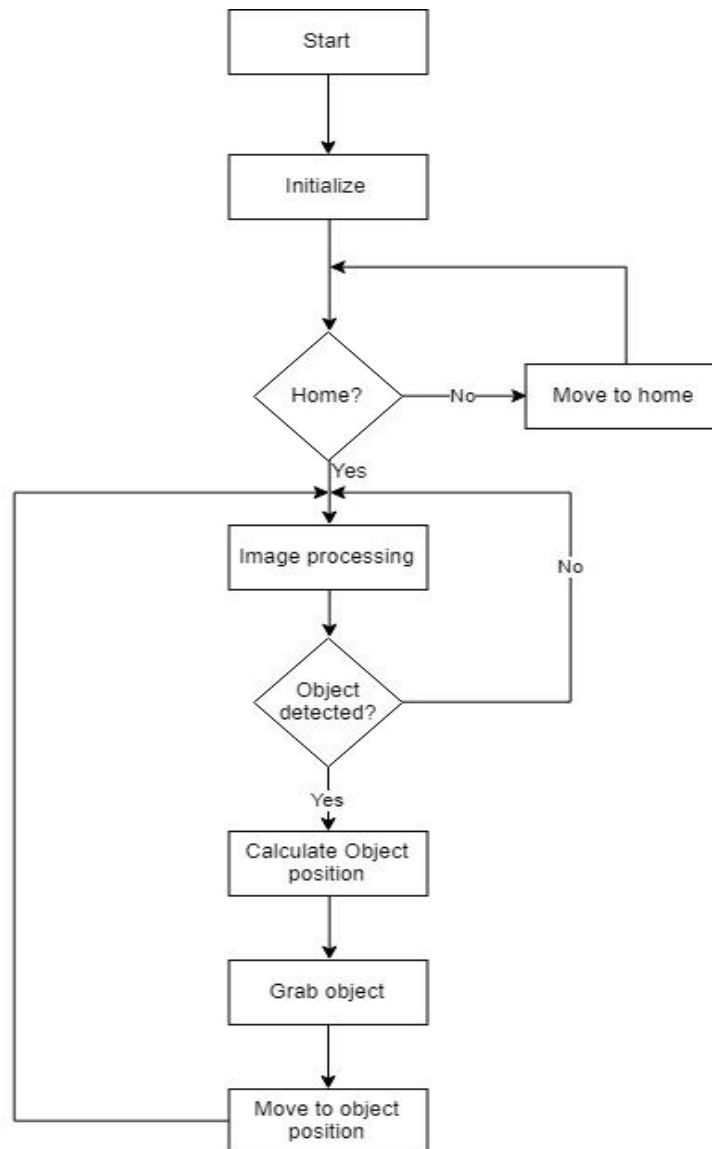
Hình 4.3 bảng điều khiển tự động

Bảng điều khiển tự động sẽ có các tính năng như:

- Calib camera để xác định được vị trí thực tế của vật so với gốc tọa độ của tay máy. (5)
- Lựa chọn giữa chế độ điều khiển bằng tay và auto. (1)
- Chọn chế độ gấp vật đứng yên hoặc vật di chuyển. (2)
- Chọn thuật toán nhận dạng. (3)
- Chọn vật thể cần gấp và vị trí đặt vật sau khi đã gấp được. (4)
- Hiển thị ảnh trực tiếp truyền về từ camera. (6)

## 4.2. Giải thuật điều khiển Robot

- Giải thuật điều khiển robot



Hình 4.4 Lưu đồ giải thuật chương trình

Ban đầu khởi tạo các giá trị của chương trình, sau đó chương trình sẽ kiểm tra xem tay máy đã về vị trí home chưa bằng cách đọc giá trị của cảm biến hành trình. Nếu chưa về home thì sẽ điều khiển tay máy về vị trí home để tiếp tục chương trình.

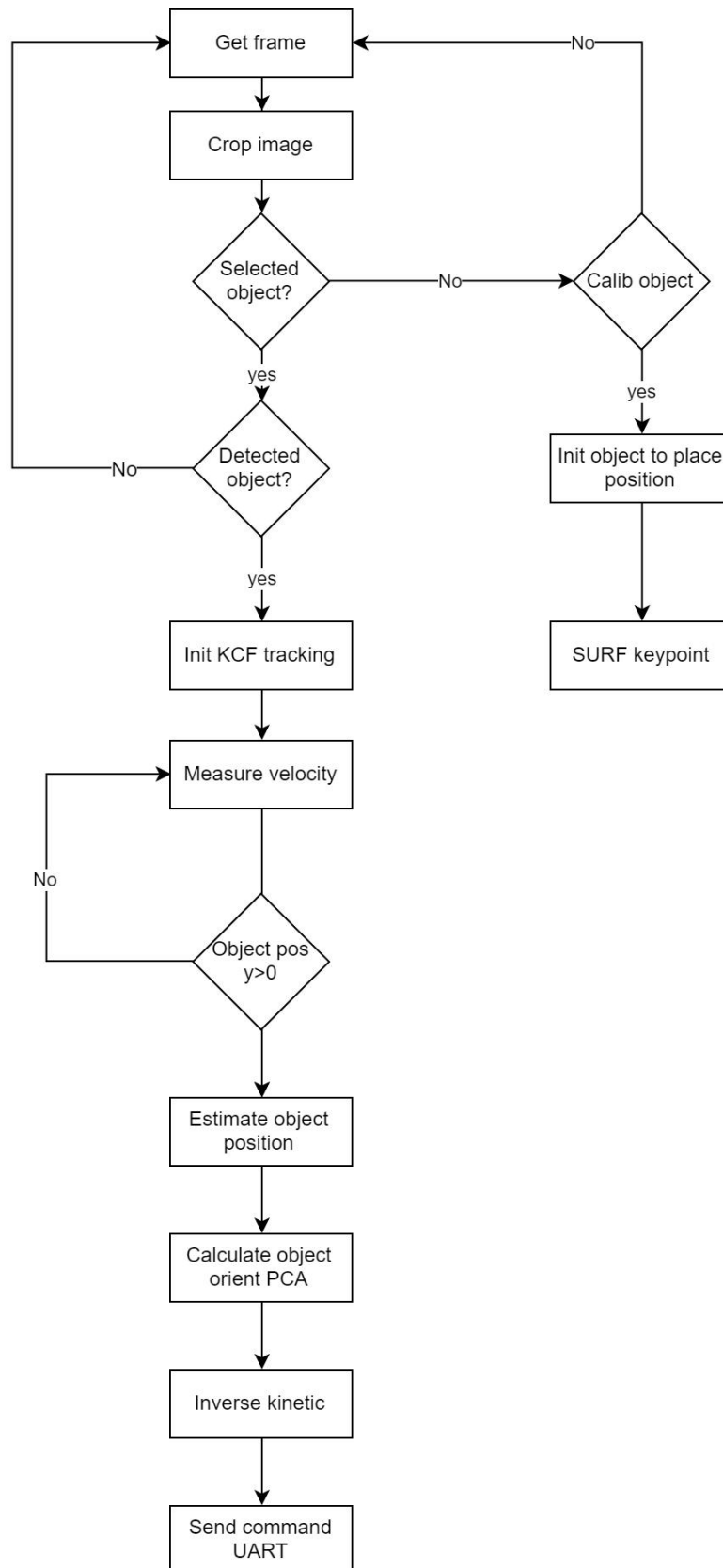
Sau đó chương trình sẽ chạy các hàm xử lý ảnh để xác định vật thể cần gấp có trên băng tải hay không? Nếu xác định được vật thể cần gấp, tay máy sẽ tính toán vị trí của vật trong không gian làm việc so với hệ trục tọa độ gốc.

Từ vị trí của vật thể ta sẽ giải các phương trình động học ngược của tay máy là các phương trình đã được tính toán ở mục 2.2.3 sẽ thu được vị trí các góc cần xoay rồi gửi xuống cho mạch điều khiển. Board điều khiển nhận được dữ liệu sẽ xuất các xung PTO để điều khiển các khớp của tay máy tới đúng vị trí và gửi lại dữ liệu lên chương trình điều khiển.

Sau khi đã gắp được vật để bỏ đúng vị trí và theo thứ tự, chương trình sẽ quay trở lại bước nhận diện vật thể và cứ thế tiếp tục chương trình.



- Giải thuật xử lý ảnh:



Hình 4.5 Lưu đồ giải thuật xử lý ảnh

Trong phần xử lý ảnh thì mỗi khung hình sẽ được đọc vào để xử lý theo các bước:

- Tính các đặc trưng keypoint SURF của bức ảnh vật thể đã chọn ban đầu.
- Đối chiếu với các keypoint của vật thể đã chọn với các keypoint trong khung hình. Nếu phát hiện được vật thể cần gấp đang trong khung hình, ta sẽ đi tìm một hình chữ nhật có kích thước nhỏ nhất và bao quanh vật thể.
- Lấy hình chữ nhật bao quanh đó khởi tạo vị trí cho bộ theo dõi KCF. Khi đã theo dõi vật thể cần gấp ta sẽ phải đo tốc độ của vật thể bằng cách tính quãng đường đi được trong khoảng thời gian 0.3s rồi chia ra vận tốc ( $v_{object}$ ).
- Tiếp theo ta sẽ ước lượng thời gian để tay máy di chuyển từ vị trí hiện tại tới vị trí của vật cần gấp. Khoảng thời gian di chuyển từ điểm hiện tại của tay máy tới vật  $t_{robot\_move}$  được tính bằng cách giải phương trình động học ngược rồi chia cho vận tốc từng khớp. Sau khi có được  $t_{robot\_move}$  và  $v_{object}$  ta sẽ xác định được vị trí mà tay máy cần tới:  $y_{new} = t_{robot\_move} * v_{object}$  do băng tải chỉ chạy theo 1 chiều (trục y) nên giá trị x,z của vật không thay đổi.

## Chương 5. KẾT QUẢ THỰC NGHIỆM

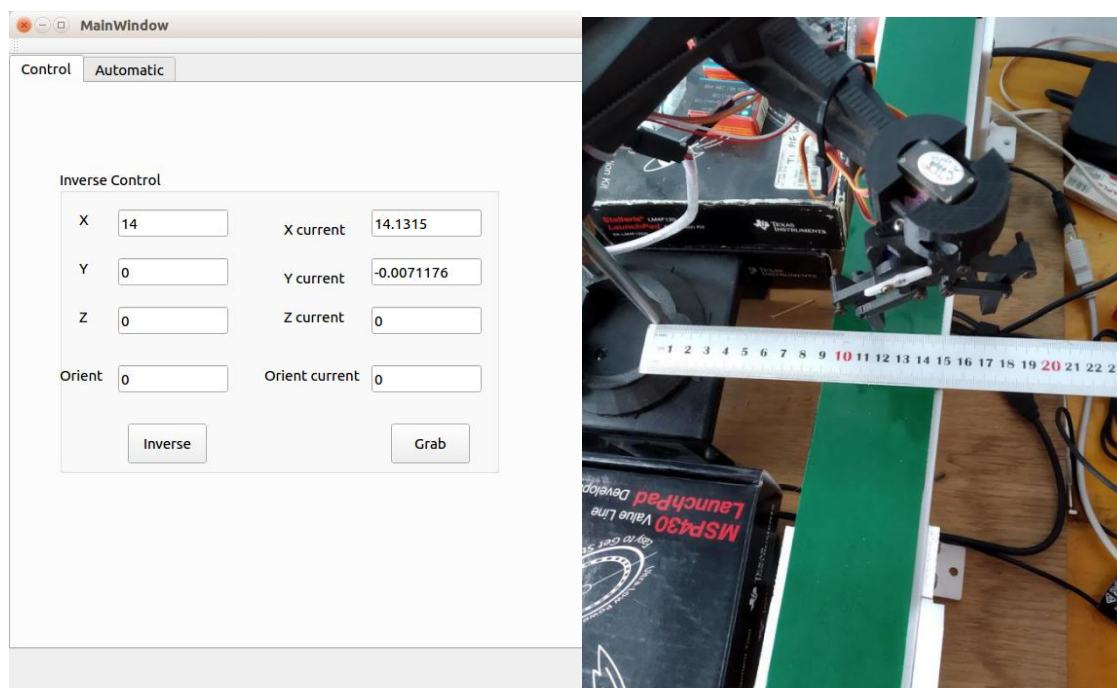
### 5.1. Kiểm tra độ chính xác tay máy

Bảng thông số không gian làm việc của robot:

Thông số	Giá trị
Tọa độ điểm Home	$(x,y,z)=(6.9,4.24,0.0)$ cm
Khớp 1	$[-100^{\circ},100^{\circ}]$
Khớp 2	$[-133^{\circ},133^{\circ}]$
Khớp 3	$[0^{\circ},180^{\circ}]$
Trục z	$[0,5]$ cm

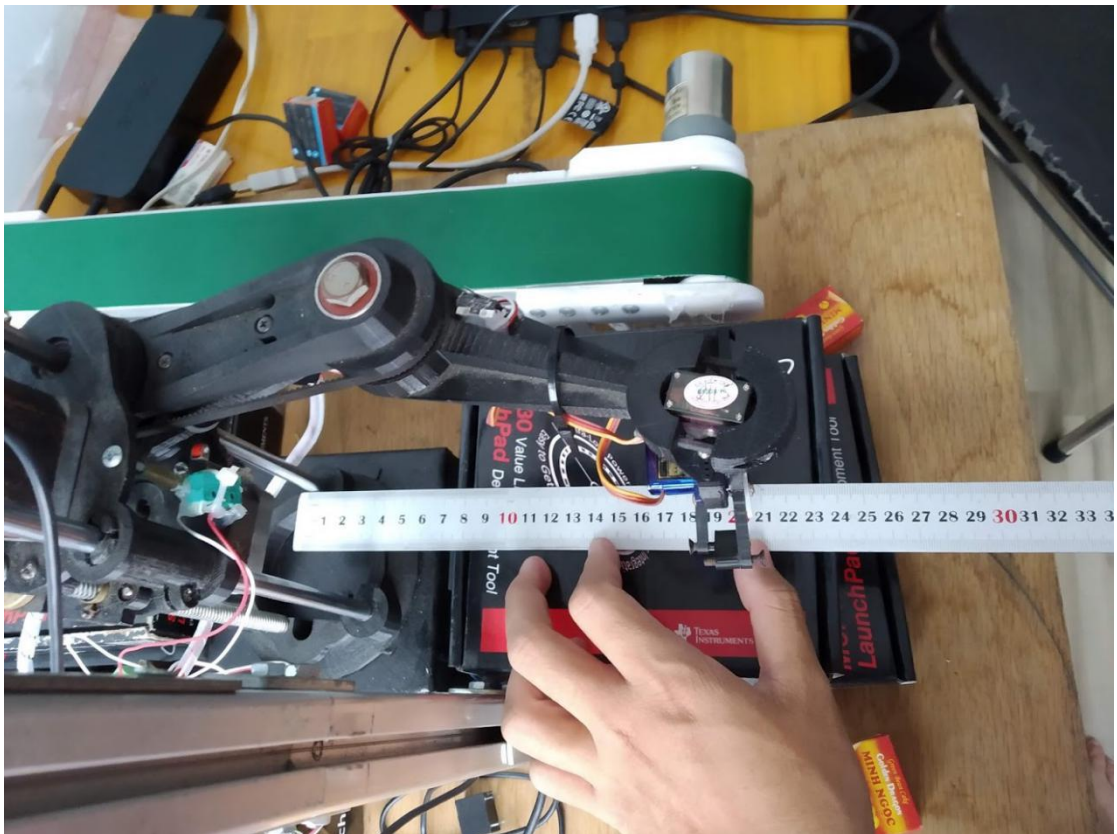
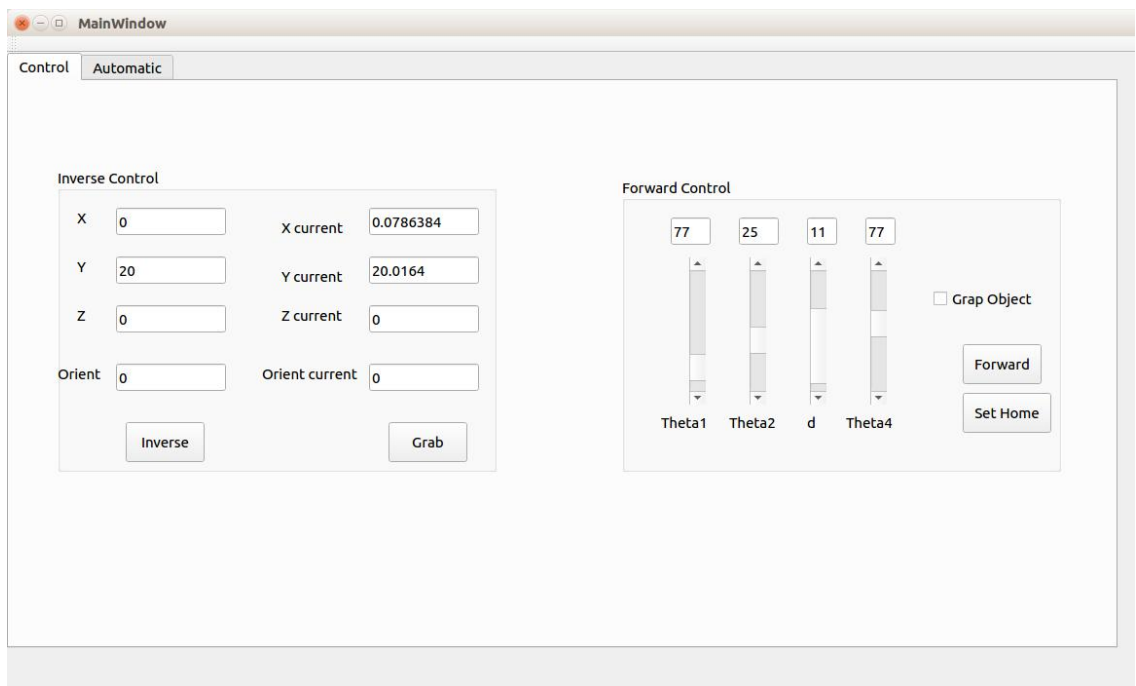
Bảng 5.1 Thông số của cánh tay robot

Điều khiển tay máy tới vị trí  $x,y,z = (14,0,0)$



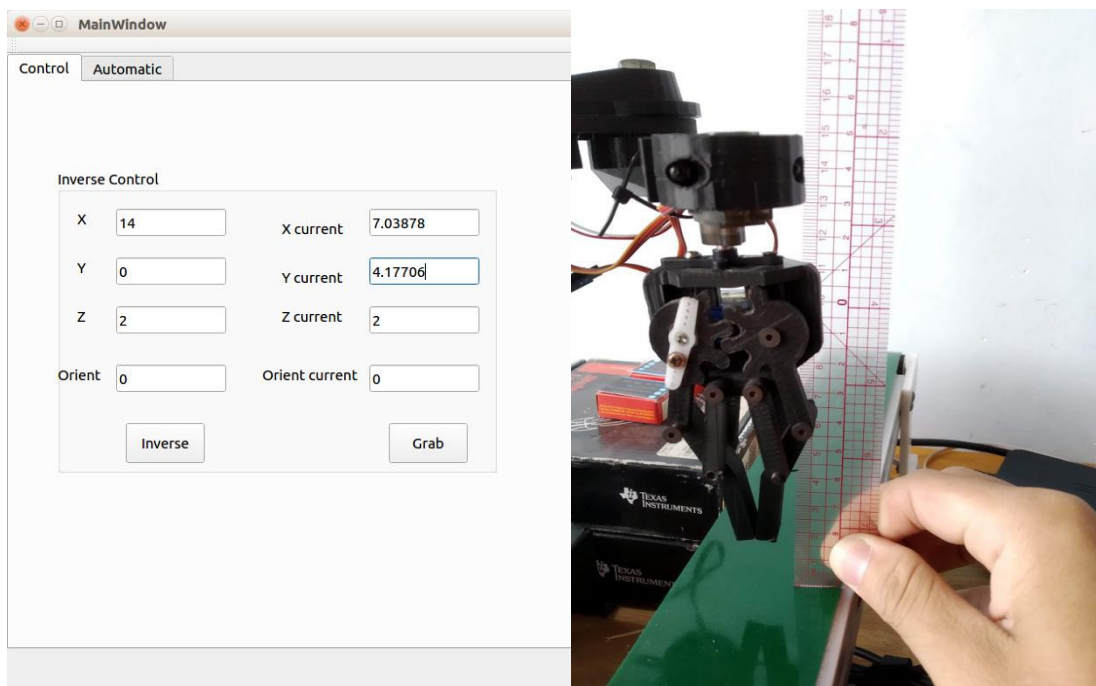
Hình 5.1 Kiểm tra sai số theo trục x

Điều khiển tay máy tới vị trí  $x,y,z=(0,20,0)$



Hình 5.2 Kiểm tra độ chính xác theo trục y

Điều khiển tay máy tới vị trí  $x,y,z = (14,0,2)$



Hình 5.3 Kiểm tra độ chính xác trục Z

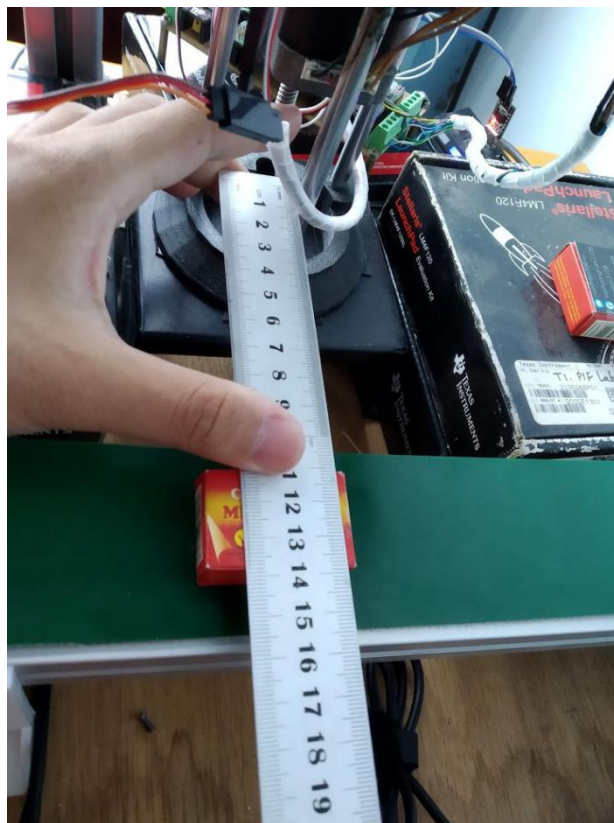
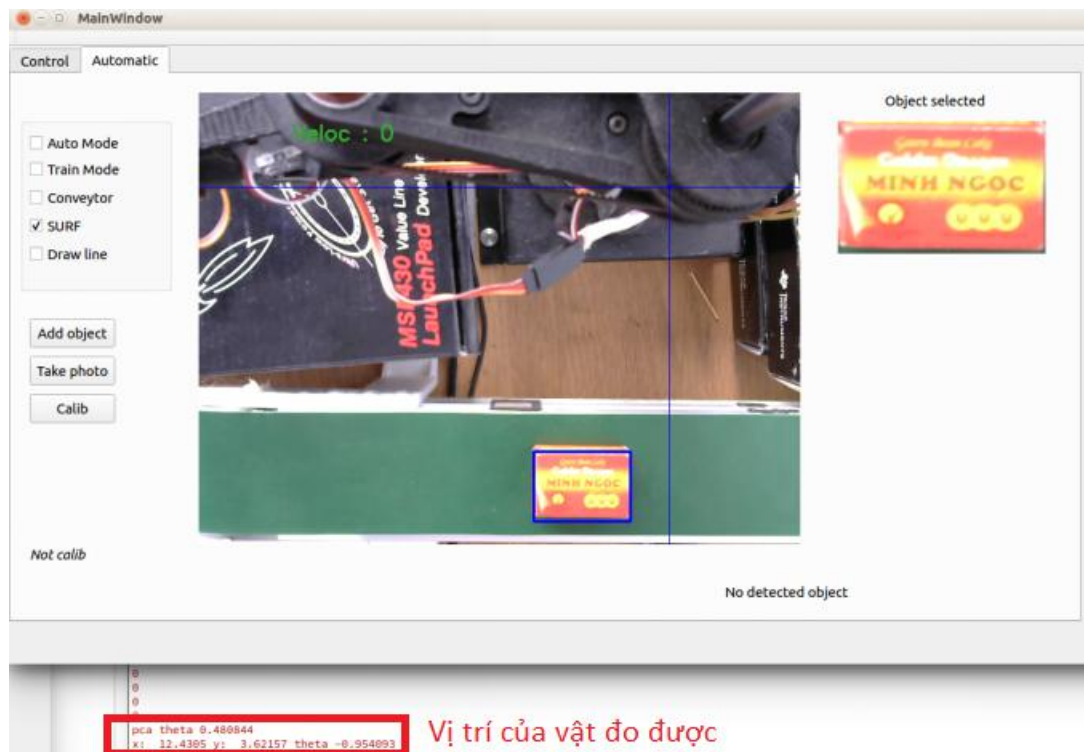
Sau khi thí nghiệm đo kết quả sai số điều khiển ta thu được bảng:

	Lần 1	Lần 2	Lần 3	Lần 4
(x,y,z)	(18,3,2)	(11,11,1)	(15,10,2)	(6,7,1)
Điều khiển				
(x,y,z)	(18.2,2.8,2.1)	(11.3,11.2,1.1)	(15.1,10.5,2.2)	(6.5,7.1,1.1)
Đo được				

Bảng 5.2 Kết quả thí nghiệm điều khiển vị trí

- Sai số trung bình theo trục x: 2.75 mm.
- Sai số trung bình theo trục y: 2.5 mm.
- Sai số trung bình theo trục z: 1.25 mm.

Kiểm tra độ chính xác của thuật toán xác định khoảng cách thực tế bằng camera:



Hình 5.4 Xác định vị trí của vật bằng camera



Điều kiện để mô hình có thể hoạt động tốt đó là: ánh sáng ít thay đổi, vật có tải trọng không lớn và vị trí của camera không thay đổi.

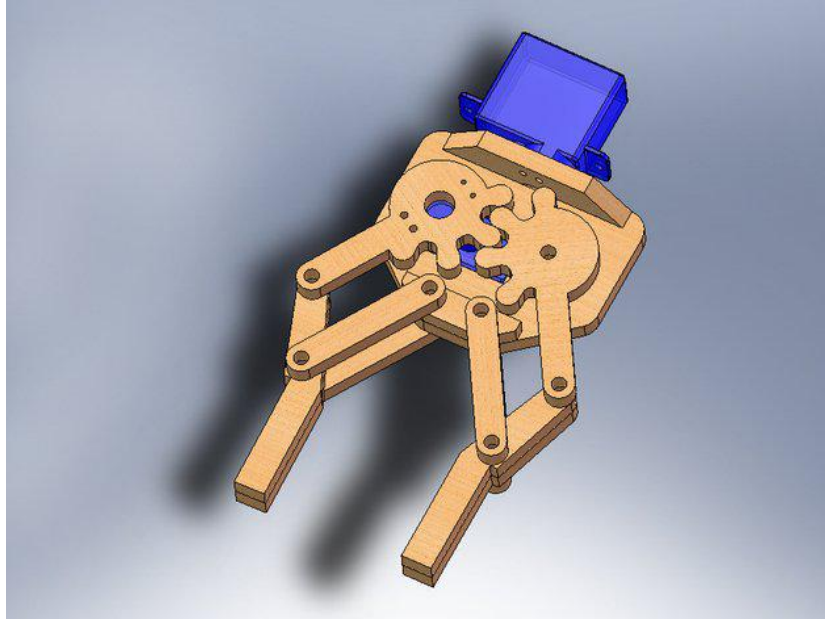
## **5.2. Thử nghiệm hoạt động của hệ thống phân loại hàng hóa**

Mô phỏng một ứng dụng xếp bao xi măng trong nhà máy thực tế. Các bao xi măng chạy trong nhà máy sau khi đóng gói sẽ được đưa ra theo băng chuyền. Robot ở cuối băng chuyền có nhiệm vụ gấp các bao xi măng, phân loại các bao dựa trên nhãn mác trên bao bì sản phẩm. Các bao xi-măng sau khi được phân loại robot sẽ xếp lên các xe tải theo thứ tự để chở đi tới các đại lý phân phối.



Hình 5.5 Robot xếp bao xi-măng trong thực tế

Do trong điều kiện mô hình, không có tay gấp phù hợp với bao xi măng nên tác giả thay thế bằng một tay gấp bình thường bằng vật liệu in 3D.



Hình 5.6 Tay gắp sử dụng trong đề tài

Hai vật thể được chọn để thử nghiệm là hộp cứng có đặc trưng khác nhau, vật thể có kích thước 4x2.5x1.5 cm.

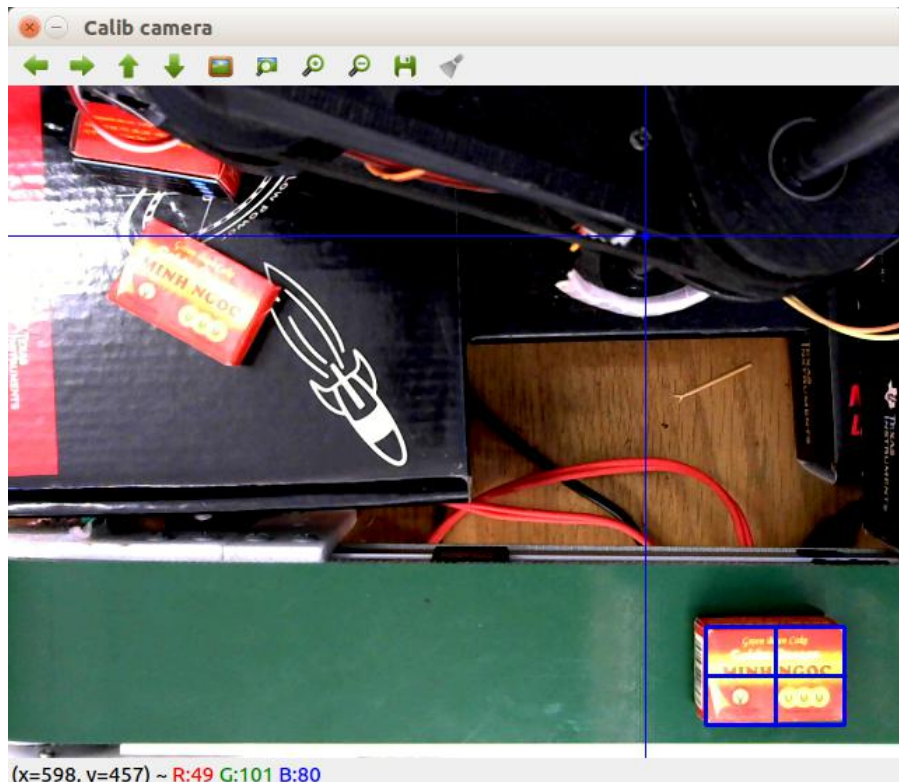


Hình 5.7 Vật thể chọn để phân loại

Trước khi hệ thống bắt đầu hoạt động, ta cần cài đặt một số thông số ban đầu trên giao diện theo các bước.

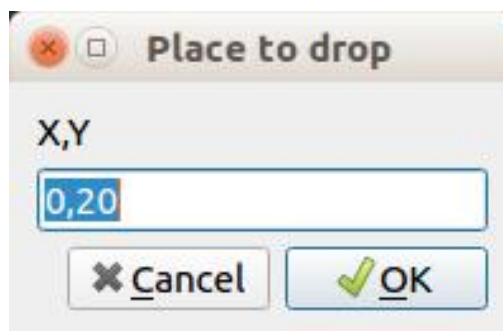
- Bước 1: Calib camera để chương trình có thể tính được tỉ lệ pixel/cm thực tế. Chọn 1 vật có cùng chiều cao với vật thể cần phân loại và nhập kích thước thực tế của nó vào giao diện.





Hình 5.8 Calib camera

- Bước 2: Chọn vật thể cần phân loại bằng cách nhấn nút “Add object” trên giao diện. Sau khi chọn xong hình ảnh của vật sẽ được hiện góc trên bên phải của giao diện. Nhập vào 2 thông số (x,y) là tọa độ để xếp vật sau khi đã gấp được.
- Bước 3: Chọn vị trí để xếp vật.



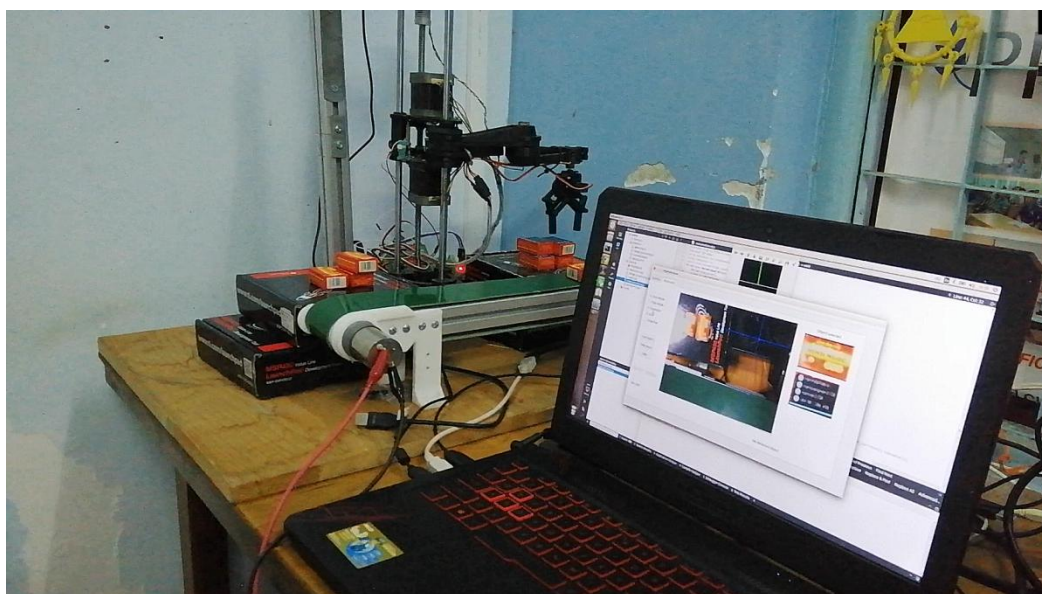
Hình 5.9 Chọn vị trí xếp vật

- Bước 4: Tích chọn ô “SURF” để bắt đầu chạy thuật toán nhận diện.
- Bước 5: Tích chọn ô “Conveyor” và “Auto Mode” để hệ thống bắt đầu hoạt động với chế độ tự động gấp vật trên băng tải.



Hình 5.10 Hệ thống bắt đầu hoạt động

Sau khi thực hiện đánh giá gấp 8 vật thể (4 vật thể mỗi loại) thì kết quả thu được là gấp được và phân loại đúng vị trí 7/8 vật.



Hình 5.11 Kết quả thu được

### 5.3. Đánh giá

Nhìn chung thì đề tài đã khá thành công, xây dựng thành công hệ thống phân loại hàng hóa trên băng tải. Mô hình hoạt động ổn định và có tốc độ di chuyển nhanh tốc độ mỗi khớp đạt 360 vòng/phút, sai số điều khiển vị trí theo trục  $x$ ,  $y$ ,  $z$  là dưới 3 mm. Mô hình có thể phân loại được vật di chuyển trên băng tải với tốc độ khoảng 3 cm/s và sắp xếp vật theo vị trí người dùng chọn.

Chương trình xây dựng trên máy tính có giao diện dễ sử dụng điều khiển được các hoạt động của hệ thống như: chọn vật cần gấp, lựa chọn giữa các chế độ điều khiển bằng tay hay gấp vật tự động, căn chỉnh camera,..

Tỉ lệ gấp thành công đạt khoảng 7/8 và phân loại đúng vị trí các vật thể. Tốc độ băng tải có thể thay đổi được, camera có thể đo vận tốc của băng tải để tính vị trí tay máy cần gấp. Tuy nhiên trong một vài trường hợp thuật toán nhận diện vật thể không xác định được chính xác hình chữ nhật bao quanh vật thể (có thể dài hơn hoặc rộng hơn) thì tay máy sẽ dễ gấp không chính xác.

## **Chương 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

### **6.1. Kết luận**

Qua các bước tìm hiểu, nghiên cứu, mô phỏng và xây dựng mô hình mặc dù gặp nhiều khó khăn trong quá trình thực hiện, nhưng đề tài cũng đã đạt được những mục tiêu đã đề ra.

Các mục tiêu đã đạt được:

- Xây dựng thành công mô hình cánh tay robot Scara 4 bậc tự do.
- Thiết kế mạch điều khiển hoạt động ổn định, điều khiển robot theo động học thuận và động học ngược với sai số điều khiển vị trí dưới 3 mm.
- Ứng dụng các thuật toán về xử lý ảnh để nhận diện và theo dõi xác định vị trí của vật thể trong không gian.
- Thiết kế được giao diện điều khiển hệ thống trên máy tính thân thiện dễ sử dụng cho nhiều đối tượng.
- Phân loại các vật thể chạy trên băng tải trong điều kiện môi trường ánh sáng ít thay đổi.
- Sản phẩm có khả năng ứng dụng trong công nghiệp nếu tăng kích thước và thay đổi các vật liệu cơ khí bằng kim loại.

Tuy nhiên luận văn vẫn còn một số điểm hạn chế:

- Chưa tính toán phương trình động lực học của tay máy.
- Xử lý ảnh chưa tốt trong các môi trường ánh sáng thay đổi.
- Chưa đánh giá được độ cứng vững của tay máy.

### **6.2. Hướng phát triển**

Với những kết quả đạt được hiện tại có thể trở thành nền tảng để xây dựng và phát triển thành công một cánh tay robot gấp hàng trên băng tải có thể hoạt động được trong môi trường công nghiệp. Có khả năng thích nghi và hoạt động ổn định với sự thay đổi của môi trường.

Một số giải pháp đưa ra để cải thiện:

- Tăng độ chính xác trong điều khiển vị trí thay thế vật liệu nhựa 3D bằng các chi tiết CNC chính xác kim loại.
- Sử dụng các thuật toán mạnh hơn về nhận diện vật thể như là các thuật toán về học sâu để làm cho phần xử lý ảnh bớt nhiễu với môi trường xung quanh.
- Xây dựng giao diện điều khiển với khả năng thu thập dữ liệu của hệ thống và phân tích các dữ liệu thu được để đưa ra những đánh giá.

## TÀI LIỆU THAM KHẢO

- [1] Ts. Nguyễn Đức Thành, “Slide môn kỹ thuật robot”.
- [2] Mark W. Spong, “Robot Dynamics and Control”, [Online].  
Available: <http://home.deib.polimi.it/gini/robot/docs/spong.pdf>
- [3] João F. Henriques, Rui Caseiro, Pedro Martins, Jorge Batista, “High-Speed Tracking with Kernelized Correlation Filters”, [Online].  
Available: <https://arxiv.org/abs/1404.7584>
- [4] Opencv document, “Introduction to SURF (Speeded-Up Robust Features)”.  
Available:  
[https://docs.opencv.org/3.0-beta/doc/py\\_tutorials/py\\_feature2d/py\\_surf\\_intro/py\\_surf\\_intro.html](https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_surf_intro/py_surf_intro.html)
- [5] Opencv document, “Introduction to Principal Component Analysis (PCA)”.  
Available:  
[https://docs.opencv.org/3.4.3/dl/dee/tutorial\\_introduction\\_to\\_pca.html](https://docs.opencv.org/3.4.3/dl/dee/tutorial_introduction_to_pca.html)
- [6] Wikipedia, “Lịch sử phát triển robot”, Available:  
<https://vi.wikipedia.org/wiki/Robot>

