

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

**TRẦN GIA TUẤN**

**ĐỒ ÁN TỐT NGHIỆP**  
**XÂY DỰNG BỘ ĐIỀU KHIỂN CHO CÁNH TAY ROBOT**  
**5 BẬC TỰ DO**  
**BUILDING CONTROL SYSTEM FOR 5 DOFS ROBOTIC**  
**MANIPULATOR**

**CỬ NHÂN NGÀNH KỸ THUẬT ĐIỀU KHIỂN & TỰ ĐỘNG HÓA**

**TP. HỒ CHÍ MINH, 2024**

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

**TRẦN GIA TUẤN – 2012357**

**ĐỒ ÁN TỐT NGHIỆP**  
**XÂY DỰNG BỘ ĐIỀU KHIỂN CHO CÁNH TAY ROBOT**  
**5 BẬC TỰ DO**  
**BUILDING CONTROL SYSTEM FOR 5 DOFS ROBOTIC**  
**MANIPULATOR**

**CỬ NHÂN NGÀNH KỸ THUẬT ĐIỀU KHIỂN & TỰ ĐỘNG HÓA**

**GIẢNG VIÊN HƯỚNG DẪN**  
**GS. TS. Hồ Phạm Huy Ánh**

**TP. HỒ CHÍ MINH, 2024**

**CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI**  
**TRƯỜNG ĐẠI HỌC BÁCH KHOA –ĐHQG -HCM**

Cán bộ hướng dẫn Khóa luận tốt nghiệp : .....  
(Ghi rõ họ, tên, học hàm, học vị và chữ ký)

Cán bộ chấm nhận xét 1 : .....  
(Ghi rõ họ, tên, học hàm, học vị và chữ ký)

Cán bộ chấm nhận xét 2 : .....  
(Ghi rõ họ, tên, học hàm, học vị và chữ ký)

Khóa luận tốt nghiệp được bảo vệ tại Trường Đại học Bách Khoa, ĐHQG  
Tp.HCM ngày ..... tháng ..... năm .....

Thành phần Hội đồng đánh giá khoá luận tốt nghiệp gồm:  
(Ghi rõ họ, tên, học hàm, học vị của Hội đồng chấm bảo vệ khóa luận tốt nghiệp)

1. ....
2. ....
3. ....
4. ....
5. ....

Xác nhận của Chủ tịch Hội đồng đánh giá khóa luận tốt nghiệp và Chủ nhiệm Bộ  
môn sau khi luận văn đã được sửa chữa (nếu có).

**CHỦ TỊCH HỘI ĐỒNG**

**CHỦ NHIỆM BỘ MÔN.....**

**TP. HỒ CHÍ MINH, 2024**

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:**

**Xây dựng bộ điều khiển cho cánh tay robot 5 bậc tự do**

**Nhóm Sinh viên thực hiện:**

Trần Gia Tuấn

2012357

**Cán bộ hướng dẫn:**

GS. TS. Hồ Phạm Huy Ánh

**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:

<nhận xét về định dạng, cách thức viết báo cáo, phân bố nội dung, chương mục có hợp lý không, ...>

2. Về nội dung luận văn:

<nhận xét về kiến thức, phương pháp mà sinh viên đã tìm hiểu, nghiên cứu, nhận xét ưu điểm và hạn chế>

3. Về tính ứng dụng:

<nhận xét về việc xây dựng ứng dụng demo, nhận xét ưu điểm và hạn chế>

4. Về thái độ làm việc của sinh viên:

<nhận xét về thái độ, ưu khuyết điểm của từng sinh viên tham gia>

**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:**

**Trần Gia Tuấn ..... /10**

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

TP. HCM, ngày....tháng.....năm.....

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

**Tên luận văn:**

**Xây dựng bộ điều khiển cho cánh tay robot 5 bậc tự do**

**Nhóm Sinh viên thực hiện:**

Trần Gia Tuấn

**Cán bộ phản biện:**

2012357

**Đánh giá Luận văn**

5. 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:

*<nhận xét về định dạng, cách thức viết báo cáo, phân bố nội dung, chương mục có hợp lý không, ...>*

6. Về nội dung luận văn:

*<nhận xét về kiến thức, phương pháp mà sinh viên đã tìm hiểu, nghiên cứu, nhận xét ưu điểm và hạn chế>*

7. Về tính ứng dụng:

*<nhận xét về việc xây dựng ứng dụng demo, nhận xét ưu điểm và hạn chế>*

8. Về thái độ làm việc của sinh viên:

*<nhận xét về thái độ, ưu khuyết điểm của từng sinh viên tham gia>*

**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:**

**Trần Gia Tuấn ..... /10**

**Người nhận xét**  
(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.....

### ĐỀ CƯƠNG CHI TIẾT

<b>TÊN LUẬN VĂN:</b> Xây dựng bộ điều khiển cho cánh tay robot 5 bậc tự do
<b>Cán bộ hướng dẫn:</b> GS. TS. Hồ Phạm Huy Ánh
<b>Thời gian thực hiện:</b> Từ ngày 01/09/2024 đến ngày
<b>Sinh viên thực hiện:</b> <b>Trần Gia Tuấn – 2012357</b>
<b>Nội dung đề tài:</b> <ul style="list-style-type: none"><li>Mục tiêu:</li><li>Đối tượng và phạm vi nghiên cứu:</li></ul>



[illegible][illegible]

<p><b>Xác nhận của Cán bộ hướng dẫn</b></p> <p>(Ký tên và ghi rõ họ tên)</p>	<p>TP. HCM, ngày....tháng .....năm.....</p> <p><b>Sinh viên</b></p> <p>(Ký tên và ghi rõ họ tên)</p>
--	--

## **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

TPHCM, ngày    tháng    năm

Sinh viên

## MỤC LỤC

## DANH MỤC HÌNH VẼ

**DANH MỤC BẢNG BIỂU**

## **DANH MỤC TỪ VIẾT TẮT**

DH: Denavit – Hartenberg.

DMA: Direct Memory Access.

DOF: Degree Of Freedom.

FOV: Field Of View.

GNU GPL: GNU General Public License.

IDE: Integrated Development Environment.

LSPB: Linear Segment with Parabol Blend.

PID: Proportional Integral Derivative.

PWM: Pulse Width Modulation.

ROI: Regions of Interest.

SIFT: Scale-Invariant Feature Transform.

SoC: System on Chip.

SP: Set Point.

SURF: Speeded-Up Robust Features.

TTL: Transistor – transistor logic.

UART: Universal Asynchronous Receiver – Transmitter.



## **TÓM TẮT LUẬN VĂN BẰNG TIẾNG VIỆT**

Trong thời đại hiện nay, với sự phát triển vượt bậc của khoa học và công nghệ, mọi ngành công nghiệp đang chuyển dịch theo xu hướng tự động hóa. Phần lớn các robot công nghiệp được đề cập trong các dây chuyền sản xuất thường là sự kết hợp của cánh tay robot và thị giác máy. Cánh tay robot có khả năng làm việc ổn định và chính xác cao bởi chúng sẽ không bị ảnh hưởng bởi cảm xúc như con người, hơn thế nữa, những cánh tay này có thể làm việc được dưới những môi trường khắc nghiệt và độc hại với sức khỏe con người như là phân loại hạt trong buồng UV,... Việc nghiên cứu và phát triển cánh tay robot ứng dụng thị giác máy trở thành một trong những vấn đề thiết yếu trong các nhà máy thông minh nhằm cải thiện tính đa dạng của lĩnh vực ứng dụng đồng thời tăng hiệu suất cũng như độ chính xác cho cánh tay robot.

Đồ án này tập trung vào vấn đề áp dụng lý thuyết điều khiển robot vào mô hình thực tế, kết hợp với các giải thuật xử lý ảnh để xử lý yêu cầu gấp vật theo vật mẫu bỏ vào vùng quy định. Trong suốt quá trình hiện thực đề tài, những sự điều chỉnh về mặt cơ khí của khung tay máy, về xung điều khiển động cơ cũng như hiệu chỉnh vị trí và kích thước vật khi được đọc từ camera sẽ luôn được tiến hành nhằm giúp tay máy có thể đến chính xác vị trí và gấp vật.

## **ABSTRACT**

In the current era, with the rapid advancement of science and technology, all industries are shifting towards automation. Most industrial robots mentioned in production lines are often a combination of robotic manipulators and machine vision systems. Robotic manipulators are capable of working stably with high precision as they are unaffected by emotions like humans. Moreover, these arms can operate in harsh and hazardous environments that are detrimental to human health, such as sorting grains in UV chambers. Research and development of robotic arms integrated with machine vision have become essential in smart factories to improve the versatility of application fields, while also increasing the efficiency and precision of robotic arms.

This project focuses on applying robot control theory to a real-world model, integrating image processing algorithms to handle object-picking tasks based on given sample objects and designated areas. Throughout the implementation, adjustments in the robot arm's mechanical frame, motor control pulses, as well as object position and size calibration from the camera, will be continuously carried out. These adjustments aim to ensure the robotic arm can accurately reach the specified positions and grasp objects effectively.

# **CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI**

## **1.1 Đặt vấn đề**

## **1.2. Tình hình nghiên cứu**

## **1.3. Lý do chọn đề tài**

## **1.4. Mục tiêu của đề tài**

## **1.5. Cấu trúc của đề tài**

Cấu trúc của đề tài gồm có:

- **Chương 1:** Giới thiệu đề tài.
- **Chương 2:** Tổng quan về mô hình cánh tay robot.
- **Chương 3:** Thiết kế bộ điều khiển cho cánh tay robot.
- **Chương 4:** Kết quả thực hiện.

## CHƯƠNG 2: TỔNG QUAN VỀ MÔ HÌNH CÁNH TAY ROBOT

Chương này giới thiệu các linh kiện và vật liệu chính trong mô hình cánh tay robot, bao gồm các khối nguồn, khối điều khiển và khối thị giác máy. Sự kết hợp này tạo nên cấu trúc bền vững và hệ thống điều khiển chính xác, đáp ứng yêu cầu về hiệu suất và tính linh hoạt cho robot.

### 2.1. Tổng quan

Hệ thống được chia làm hai phần chính cần thiết kế và điều khiển, gồm: phần cánh tay robot và phần thị giác máy.

Phần cánh tay robot được lắp ghép từ các động cơ servo với bộ não điều khiển là mạch vi điều khiển stm32 thông qua PCA 9685. Mỗi động cơ sẽ đảm bảo một vai trò cho từng góc quay của các khớp robot được lập trình phối hợp với nhau để chuyển động mượt mà. Nguồn sử dụng chính trong phần cánh tay cần được cân nhắc dựa trên nguồn sử dụng của các loại servo, mạch điều khiển và PCA 9685. Nguồn cũng cần đảm bảo đủ dòng cho mỗi servo trong cánh tay. Hệ thống cần được tinh gọn để không làm ảnh hưởng đến quy trình chuyển động của cánh tay robot.

Phần thị giác máy sẽ được điều khiển bởi máy tính nhúng là Raspberry Pi với các tác vụ xử lý ảnh nhằm cung cấp chính xác tọa độ cho mạch stm32. Raspberry Pi có giá thành khá cao và là một phần linh kiện quan trọng trong cả hệ thống, vì vậy nguồn cấp cho mạch phải đúng chính xác như các thông số cung cấp của nhà phát hành. Camera sử dụng sẽ được chọn lựa sao cho thích ứng với máy tính nhúng tốt nhất. Bên cạnh đó, phần này còn yêu cầu phải thiết kế lắp đặt camera ở nơi có thể quan sát cung cấp tọa độ cần thiết mà không bị ảnh hưởng bởi workspace của robot.

### 2.2. Khối cánh tay robot

Yêu cầu của bài toán đặt ra là thiết kế hệ thống điều khiển cánh tay robot có khả năng di chuyển đến vị trí tương ứng của vật và gắp vật.

#### 2.2.1. Nguồn

Để vận hành các động cơ servo cũng như các mạch điều khiển, ta cần một nguồn cấp 5V và dòng đủ lớn để điều khiển các servo, vì thế nguồn tổ ong 5V – 20A sẽ được sử dụng trong đồ án này để cấp nguồn cho các thành phần liên quan đến cánh tay robot.

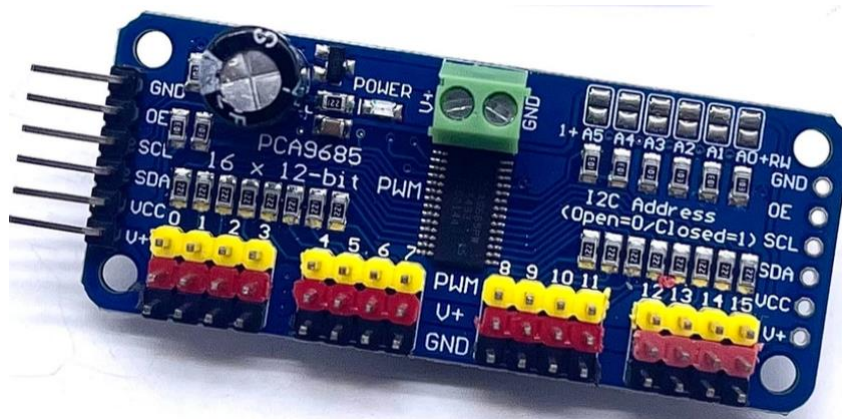


Hình 2.1. Nguồn tổ ong 5V -20A.

Điện áp đầu vào	180 VAC – 240 VAC
Công suất	100W
Dòng đầu ra tối đa	20A
Nhiệt độ làm việc	10 ~ 60 độ C
Kích thước	198 x 48 x 42 mm

Bảng 2.1. Thông số kỹ thuật nguồn tổ ong.

Để thuận tiện cho việc thiết kế hệ thống cũng như trong quá trình điều khiển, các động cơ servo được điều khiển thông qua PCA9685. Mạch điều khiển 16 Chanel PWM PCA9685 được sử dụng để có thể xuất ra đồng thời 16 xung PWM từ 16 cổng khác nhau thông qua giao tiếp I2C sử dụng IC PCA9685.



Hình 2.2. Mạch điều khiển 16 kênh PWM PCA9685.

IC chính	PCA 9685
Điện áp sử dụng	2.3 ~ 5.5 VDC
Số kênh PWM	16 kênh
Tần số	40 ~ 1000 Hz
Độ phân giải PWM	12 bit
Giao tiếp	I2C (chấp nhận mức Logic TTL 3 ~ 5 VDC)
Kích thước	62.5 x 25.4 x 3 mm

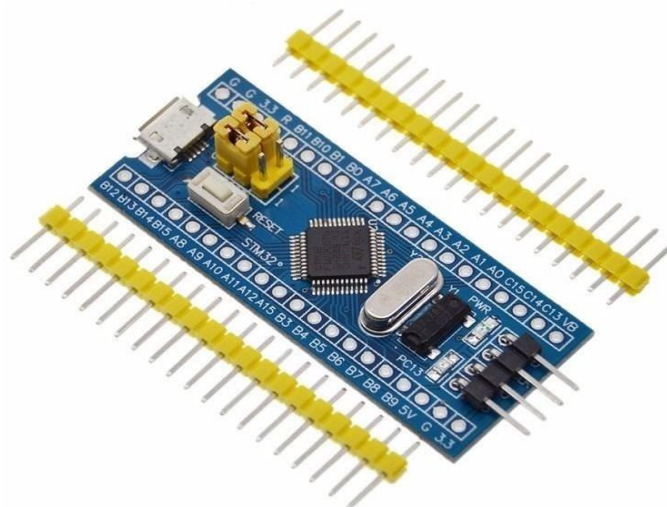
Bảng 2.2. Thông số kỹ thuật của PCA 9685.

### 2.2.2. Mạch điều khiển robot

Đối với khối điều khiển, ta cần các tác vụ giao tiếp I2C với PCA 9685 và giao tiếp UART với Raspberry Pi, vì thế đề xuất được đưa ra ở dự án này là mạch STM32F103C8T6 Blue Pill dùng làm bộ não chính của cánh tay robot.

Là một phiên bản nhỏ gọn trong các dòng chip phổ biến của ST nhưng vẫn giữ được các chức năng cơ bản, tương tự và cần thiết so với các loại vi điều khiển khác. STM32F103 thuộc họ F1 với lõi ARM 32-bit COTEX-M3 RISC cùng bộ dao động bên trong 4-16MHz, là một chip công nghệ flash CMOS.

Với số lượng các Timers, USART, và DMA được tích hợp trong vi điều khiển này cho phép nhận và truyền dữ liệu giữa lớp trên và lớp dưới một cách hợp lý.



Hình 2.3. Mạch vi điều khiển STM32F103C8T6.

Vi điều khiển	ARM 32-bit COTEX-M3 RISC
Điện áp hoạt động	2.0 → 3.6V
Dòng điện nguồn	6mA
Số lượng chân	47
Bộ nhớ flash	64/128 KB
Bộ dao động bên trong	4 → 16 MHz
Tốc độ CPU	72 MHz (tối đa)
Số Timer	3 Timers 16 bit hỗ trợ mode Input Capture/Output Compare/PWM, 1 Timer 16 bit hỗ trợ điều khiển động cơ, 2 Watchdog Timer, 1 Systick Timer 24 bit.
Số bộ I2C	2
Số bộ USART	3
Số kênh DMA	7

Bảng 2.3. Thông số kỹ thuật của STM32F103C8T6.

### 2.2.3. Động cơ

#### Động cơ RC Servo TowerPro MG996R

Động cơ RC Servo MG996R là loại động cơ servo đến từ hãng TowerPro với lực

kéo, độ bền, độ chính xác và độ ổn định cao, động cơ có cấu tạo hộp số hoàn toàn bằng kim loại với trục chính bằng nhôm hợp kim có độ cứng cao 6061-T6 (màu bạc) giúp giảm trọng lượng, ngoài ra trục chính và các trục quay trong hộp số còn được hỗ trợ bạc đạn và các vòng đệm kim loại giúp tăng độ bền và độ chính xác của động cơ khi hoạt động.

*Động cơ Digital RC Servo TowerPro MG996R* là dạng Digital RC Servo với các đặc tính vượt trội và độ chính xác cao, thích hợp với các mô hình robot phức tạp.



Hình 2.4. Động cơ RC Servo MG996R.

Điện áp hoạt động	4.8 → 6.6V
Lực kéo	9.4 kg/cm ở mức 4.8VDC 11 kg/cm ở mức 6VDC
Tốc độ quay	0.19s/60° (4.8VDC không tải) 0.15s/60° (6VDC không tải)
Kích thước	40 x 19.7 x 42.9 mm
Trọng lượng	55g

Bảng 2.4. Thông số kỹ thuật của Động cơ RC Servo MG966.



### *Động cơ Digital RC Servo HiWonder HPS-2027*

Vì khớp thứ 2 của cánh tay robot chịu moment quán tính lớn khiến cho *động cơ servo MG996R* không đủ khả năng ghi lại làm cho cánh tay bị ngã về phía trước dù đã cung cấp đủ dòng điện cần tiêu thụ. Việc ngã về phía trước sẽ dẫn đến sai lệch vị trí gấp của cánh tay hoặc hỏng cả hệ thống. Do đó, *động cơ servo HiWonder HPS-2027* được sử dụng như một sự thay thế cần thiết, giúp khớp thứ 2 của cánh tay robot có đủ khả năng chịu lực, giảm sai số do cơ khí và xung chạy của servo gây ra cho hệ thống.

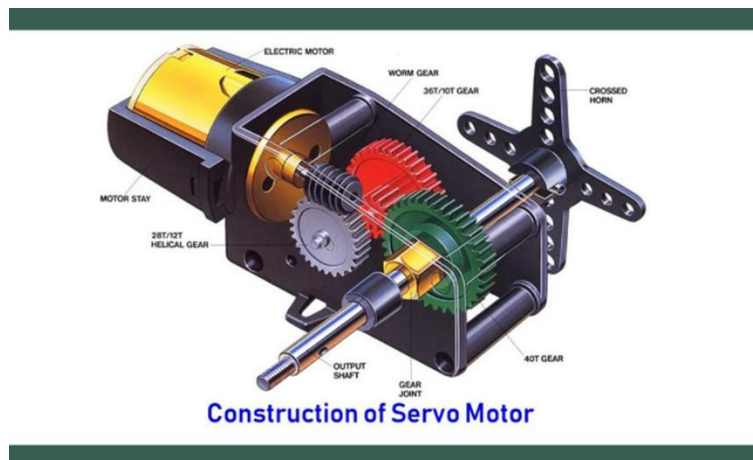
*Động cơ Digital RC Servo HiWonder HPS-2027* được sử dụng trong các Robot có cấu trúc phức tạp: Humanoid Robot, Biped Robot, Robot nhện,...vì các đặc tính vượt trội của Digital RC Servo so với các loại Analog RC Servo truyền thống *MG995*, *MG996*: lực kéo khỏe hơn, phản ứng nhanh hơn, chạy mượt hơn,... *Động cơ Digital RC Servo HiWonder HPS-2027* có hộp số bánh răng kim loại, tốc độ phản ứng nhanh với lực kéo khỏe theo thông số nhà sản xuất moment lên đến 20Kg.cm.



*Hình 2.5. Động cơ Digital RC Servo HiWonder HPS-2027.*

Điện áp hoạt động	6 ~7.4VDC
Lực kéo	20Kg.cm tại 7.4VDC
Tốc độ quay	0.2sec / 60 ° tại 7.4VDC
Kích thước	54.5 x 20 x 47.5mm
Trọng lượng	66 g
Tần số PWM	500 ~2500 us tương ứng 0 ~270 độ

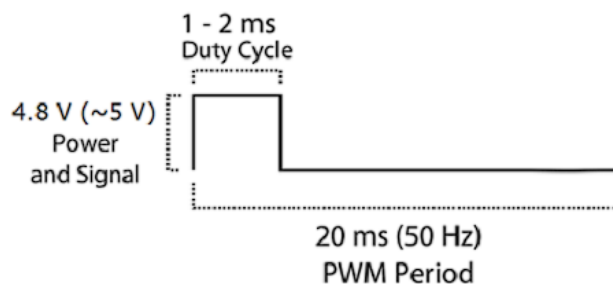
Hình 2.5. Thông số kỹ thuật động cơ Digital RC Servo HiWonder HPS-2027.



Hình 2.6. Cấu tạo của 1 động cơ servo.

Động cơ servo được điều khiển bằng cách gửi một xung điện có độ rộng thay đổi hoặc điều chế độ rộng xung (PWM), thông qua dây điều khiển. Có một xung tối thiểu, một xung tối đa và tốc độ lặp lại. Một động cơ servo thường chỉ có thể quay 90 độ theo một trong hai hướng cho tổng chuyển động là 180 độ. Vị trí trung tính của động cơ được xác định là vị trí mà servo có cùng số vòng quay tiềm năng theo cả chiều kim đồng hồ hoặc ngược chiều kim đồng hồ. PWM được gửi đến động cơ xác định vị trí của trục và dựa trên thời gian của xung gửi qua dây điều khiển, motor sẽ chuyển sang vị trí mong muốn của người điều khiển.

Tín hiệu điều khiển servo là dạng tín hiệu tương tự như Hình 2.7.



Hình 2.7. Dạng tín hiệu điều khiển góc quay động cơ MG996R tại 50 Hz.

Theo mô tả của hình trên:

- Ứng với 1ms sẽ là 0°.
- Ứng với 2ms sẽ là 180°.

Đối với động cơ *Digital RC Servo HiWonder HPS-2027* thì sẽ là:

- Ứng với 500  $\mu s$  sẽ là 0°.
- Ứng với 2500  $\mu s$  sẽ là 270°.

Để tính xung cần cấp tương ứng với mỗi góc bất kỳ, ta sử dụng công thức như sau:

$$Pulse\ Width\ (ms) = Min\ Pulse\ Width + \left( \frac{Max\ Pulse\ Width - Min\ Pulse\ Width}{180} \times Góc \right)$$

### 2.3. Khởi thị giác máy

Raspberry Pi chạy với hệ điều hành Raspberry Pi OS (trước đây gọi là Raspbian), một phiên bản của Debian Linux được tối ưu hóa cho phần cứng của Raspberry Pi. Khi chọn camera tương ứng, ta cũng cần tìm hiểu về thư viện điều khiển camera đó trên Raspberry. Có hai lựa chọn để có thể điều khiển Raspberry để ta cân nhắc khi mua linh kiện: remote control và direct control. Nếu chọn direct control, ta cần mua màn LCD để hiển thị cũng như cần chuột và bàn phím riêng để lập trình trên máy tính nhúng.

#### 2.3.1. Nguồn

Đối với máy tính nhúng Raspberry Pi 4B, cần sử dụng nguồn riêng của hãng để tránh gây ra hư hỏng.



Hình 2.8. Nguồn Raspberry Pi 4 -5.1 V – 3 A.

Điện áp đầu ra	5.1 VDC
Dòng tối thiểu	0A
Dòng tối đa	3A
Công suất tối đa	15.3 W
Điều tiết tải	$\pm 5\%$
Quy định dòng	$\pm 2\%$
Bảo vệ	<ul style="list-style-type: none"> <li>• Bảo vệ ngắn mạch</li> <li>• Bảo vệ quá dòng</li> <li>• Bảo vệ quá nhiệt</li> </ul>
Hiệu quả	Tối thiểu 81% ( dòng đầu ra từ 100% 75% 50% 25% ) tối thiểu 72% ở 10% tải
Chuẩn kết nối đầu ra	USB Type - C

Bảng 2.6. Thông số kỹ thuật của nguồn Raspberry Pi 4.

### 2.3.2. Camera

Camera đóng vai trò là đôi mắt cho cánh tay robot hoạt động. Đối với dự án này, thị giác sẽ thực hiện tác vụ đọc được tọa độ thực của vật trên bề mặt phẳng, từ đó sẽ truyền thông tin về khối điều khiển để xử lý.

Camera sẽ đóng vai trò vô cùng quan trọng trong việc điều hướng robot sao cho đi đúng đến vị trí vật, vì lẽ đó lựa chọn camera nào cần được cân nhắc trước khi đưa ra quyết định. Camera được lựa chọn ở đây sẽ là Raspberry Pi Camera module V3 75°.



Hình 2.9. Cảm biến USB Camera.

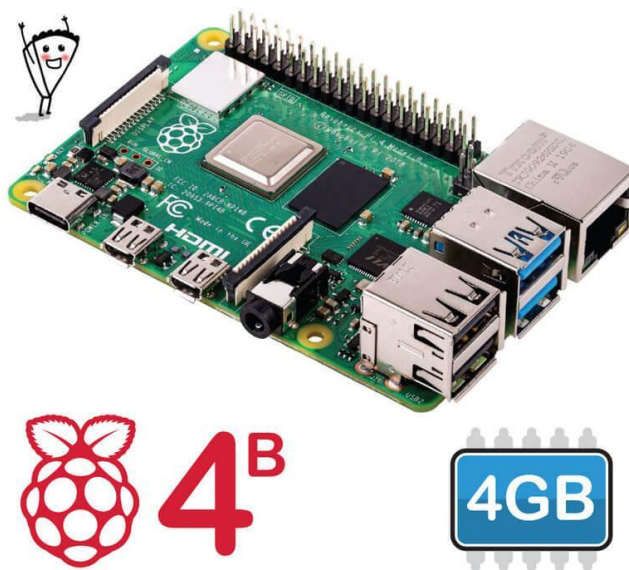
Độ phân giải	12 MP
Độ phân giải ảnh	4608 x 2592
Độ phân giải video	1080p@50fps
Cảm biến hình ảnh	IMX708
Góc nhìn	75°
Kích thước	25 x 23.862 mm

Bảng 2.7. Thông số kỹ thuật của USB Camera.

### 2.3.3. Máy tính nhúng

Để có thể thu nhận hình ảnh từ camera về xử lý để cho ra tọa độ vật, máy tính nhúng lựa chọn được đưa ra vì mạch điều khiển STM32F103C8T6 sẽ không đủ khả năng để xử lý công việc giao tiếp với camera. Máy tính nhúng lựa chọn sẽ là Raspberry Pi model 4B phiên

bản 4GB.



Hình 2.10. Raspberry Pi 4 Model B – 4GB RAM.

RAM	4GB
SoC	Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit @ 1.5GHz
Nguồn điện	DC 5V – 3A DC chuẩn USB – C
Wifi	Chuẩn 2.4 GHz và 5.0 GHz IEE 802.11ac
Kích thước	85 x 56 mm

Bảng 2.8. Thông số kỹ thuật của Raspberry Pi 4 Model B.

## 2.2. Mô hình cánh tay và hệ thống

Hình 2.17. Mô hình cánh tay robot 5 bậc tự do.

Hình 2.18. Giá đỡ camera.

Hình 2.19. Bộ đỡ của cánh tay robot.

## 2.3. Sơ đồ giao tiếp giữa các khối với nhau trong mô hình

Hình 2.20. Sơ đồ nối dây tổng thể của hệ thống.

Hình 2.21. Minh họa cơ chế hoạt động của DMA.

## CHƯƠNG 3: TÍNH TOÁN ĐỘNG HỌC CHO CÁNH TAY ROBOT

Trong chương này, tính động học và hoạch định quỹ đạo của cánh tay robot sẽ được trình bày để làm rõ cách xác định vị trí, hướng của các khớp và lập trình quỹ đạo di chuyển. Bắt đầu với các khái niệm động học thuận và ngược, chương giúp cung cấp nền tảng tính toán chính xác chuyển động cho cánh tay robot. Ngoài ra, chương sẽ đề cập đến việc hoạch định quỹ đạo nhằm đảm bảo chuyển động mượt mà, giảm thiểu dao động và tối ưu thời gian di chuyển, qua đó nâng cao hiệu suất và độ chính xác của robot trong thao tác thực tiễn.

### 3.1. Tổng quan

Phần này cung cấp cái nhìn toàn diện về động học và hoạch định quỹ đạo trong hệ thống điều khiển cánh tay robot, đóng vai trò nền tảng trong việc thiết lập chuyển động chính xác và hiệu quả cho các ứng dụng thực tiễn. Đầu tiên, chương trình bày các nguyên lý động học thuận và động học ngược. Động học thuận là quá trình tính toán vị trí và hướng của hiệu dụng cụ từ các giá trị góc khớp đã biết, còn động học ngược cho phép xác định các góc khớp dựa trên vị trí mong muốn của hiệu dụng cụ. Hai kỹ thuật này bổ trợ lẫn nhau, giúp xây dựng mối quan hệ giữa không gian khớp và không gian tác vụ, đồng thời là cơ sở cho việc xác định chính xác các tọa độ và góc của mỗi khớp trong không gian ba chiều.

Tiếp theo, chương tập trung vào các phương pháp hoạch định quỹ đạo để đảm bảo cánh tay robot có thể di chuyển theo lộ trình liên tục và tối ưu. Hoạch định quỹ đạo không chỉ yêu cầu xác định lộ trình mà còn đòi hỏi tính toán vận tốc và gia tốc hợp lý tại các điểm chuyển tiếp, từ đó giảm thiểu các dao động đột ngột và duy trì sự ổn định trong chuyển động. Việc xây dựng quỹ đạo với các đoạn chuyển tiếp mượt mà giúp tối ưu thời gian di chuyển, đảm bảo các thao tác nhanh chóng và an toàn mà vẫn đạt độ chính xác cao. Bằng cách áp dụng các kỹ thuật động học và hoạch định quỹ đạo được đề cập trong chương này, người thiết kế và điều khiển robot sẽ có cơ sở vững chắc để triển khai cánh tay robot trong các nhiệm vụ đòi hỏi hiệu suất cao và tính chính xác, như trong các hệ thống sản xuất tự động hay ứng dụng dịch vụ.

## 3.2. Cơ sở lý thuyết về cánh tay robot 5 bậc tự do

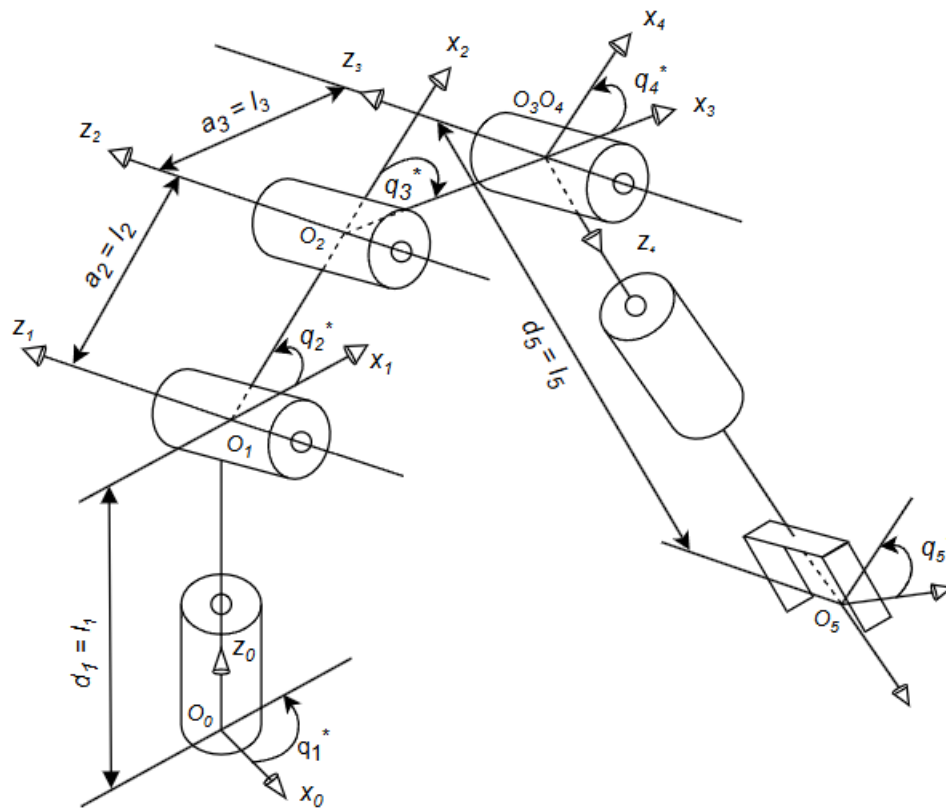
### 3.2.1. Đặt hệ trục tọa độ lên cánh tay robot

Để có thể đặt được trục, ta cần tuân thủ vài nguyên tắc cơ bản về đặt trục của phương pháp DH như sau:

- Trục z sẽ nằm trùng với trục xoay của động cơ
- Trục x sẽ song song với pháp tuyến chung của  $Z_n$  và  $Z_{n-1}$  như sau

$$X_n = Z_n \times Z_{n-1}.$$

- Hướng của  $x_n$  sẽ hướng từ  $Z_{n-1}$  đến  $Z_n$
- Trục y được xác định bằng quy tắc bàn tay phải



Hình 3.1. Hệ trục tọa độ từng khớp của cánh tay robot trong không gian 3 chiều.



### 3.2.2. Thiết lập bảng DH

Các thông số DH tương ứng với các khớp sẽ được xác định dựa trên quy tắc sau:

- $\theta_i$  là góc quay quanh trục  $z_{i-1}$  từ  $x_{i-1}$  sang  $x_i$
- $d_i$  là khoảng cách từ điểm giao nhau của trục  $z_{i-1}$  với trục  $x_i$  đến gốc  $O_{i-1}$
- $\alpha_i$  là góc xoắn quay quanh trục  $x_i$  từ  $z_{i-1}$  sang  $z_i$
- $a_i$  là khoảng cách dọc theo pháp tuyến chung giữa trục  $z_{i-1}$  và trục  $z_i$

Khâu	$a_i$	$\alpha_i$	$d_i$	$\theta_i$
1	0	$-\frac{\pi}{2}$	$d_1$	$\theta_1^*$
2	$a_2$	0	0	$\theta_2^*$
3	$a_3$	0	0	$\theta_3^*$
4	0	$\frac{\pi}{2}$	0	$\theta_4^*$
5	0	0	$d_5$	$\theta_5^*$

Bảng 3.1. Bảng DH.

Để có được các thông số điền vào và hoàn thiện bảng DH, ta tiến hành đo đặc kính thước cho tay máy và có được kết quả như Bảng 3.2.

Thông số	$d_1$	$a_2$	$a_3$	$d_5$
Kích thước (cm)				

Bảng 3.2. Kích thước tay máy.

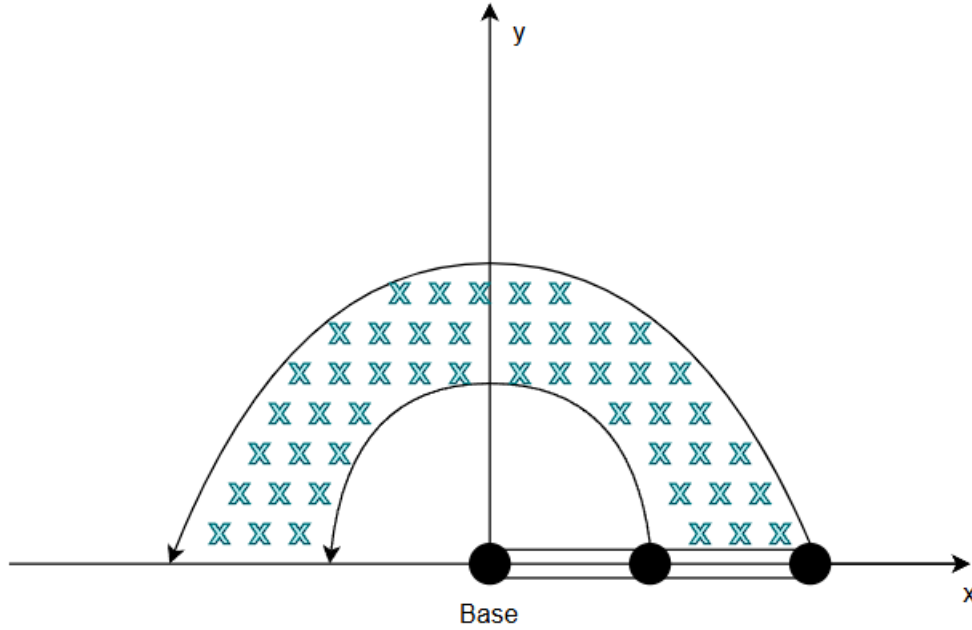
### 3.2.3. Không gian làm việc của cánh tay robot

*Vùng giới hạn của theta 1*

Không gian làm việc của cánh tay robot được định nghĩa là một tập hợp các điểm

nằm trong đó mà cánh tay robot có thể vươn tới được. Trong đồ án này, cánh tay robot được thiết lập để gấp theo chiều vuông góc với mặt sàn, vì vậy không gian sẽ khác với khi robot hoạt động tự do.

Khớp thứ nhất có khả năng quay được  $[0^\circ; 180^\circ]$ , nên ta có được vùng hoạt động của cánh tay robot khi chiếu lên mặt phẳng Oxy như Hình 3.2.



Hình 3. 2. Vùng hoạt động giới hạn của  $\theta_1$ .

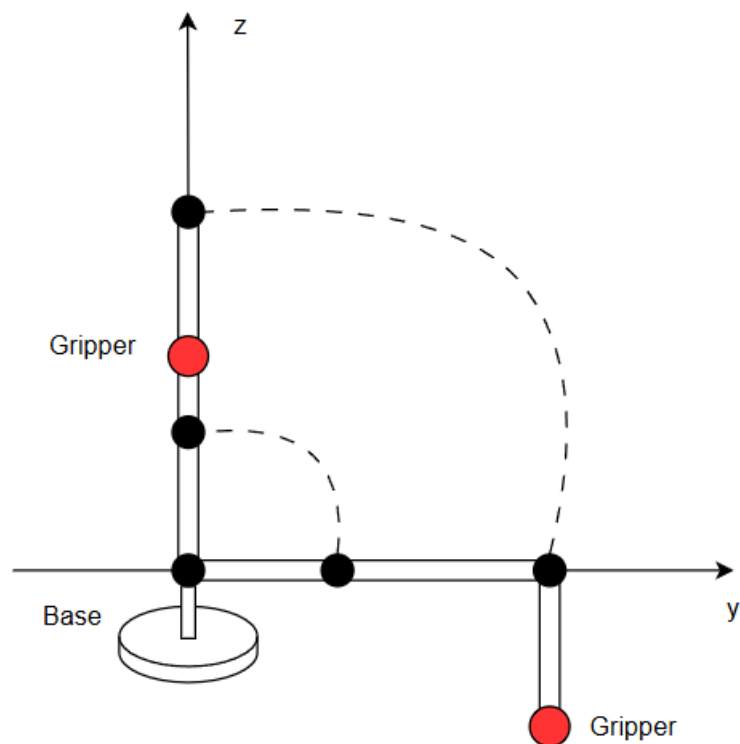
Vùng đánh dấu sẽ là vùng tồn tại cặp tọa độ  $(x, y)$  mà cánh tay robot có thể vươn tới được. Từ đó ta tiến hành đặt  $D_{\min}$  là khoảng cách ngắn nhất từ gốc tọa độ mà cánh tay có thể vươn tới;  $D_{\max}$  là khoảng cách xa nhất từ gốc tọa độ mà cánh tay có thể vươn tới.

$$\text{Đặt } R = \sqrt{x^2 + y^2}; \theta_1 = \arctan 2(y, x).$$

$$\text{Vậy } D_{\min} < R < D_{\max}; 0^\circ < \theta_1 < 180^\circ.$$

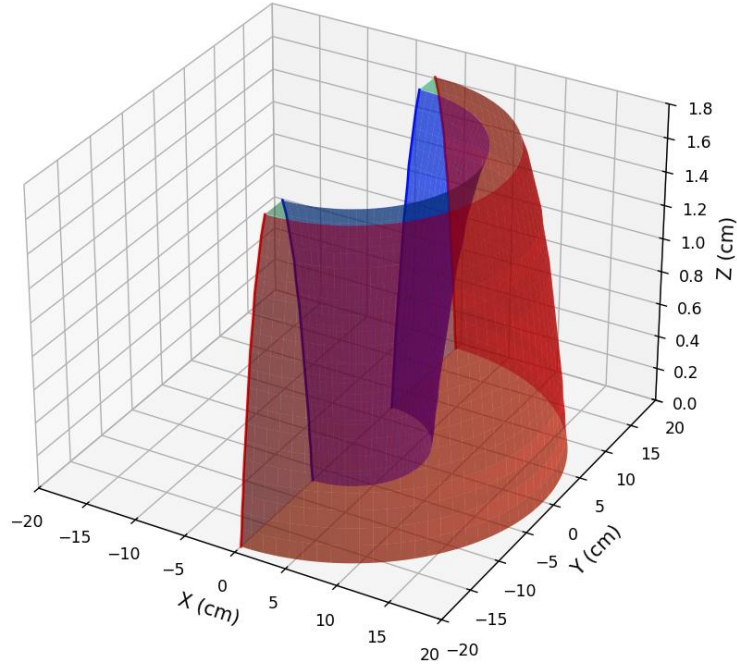
Mà theo như phương pháp gấp vật của đồ án này, cánh tay sẽ gấp thẳng đứng tương ứng với việc  $D_{\min} = a_2; D_{\max} = a_3$ .

### Vùng giới hạn của $\theta_2$



Hình 3. 3. Vùng hoạt động giới hạn của  $\theta_2$ .

Để hoạt động tốt và trơn tru khớp thứ hai được giới hạn từ  $[0^\circ; 90^\circ]$ , khi vị trí khớp



Hình 3.3. Không gian làm việc của robot khi gấp thẳng đứng.

### 3.2.4. Tính toán động học thuận của cánh tay robot

Để tính toán động học thuận của cánh tay robot, ta cần nhớ công thức để tính ma trận chuyển tiếp như sau

$$A_i = \begin{bmatrix} c_i & -c_{\alpha_i} \cdot s_i & s_{\alpha_i} \cdot s_i & a_i \cdot c_i \\ s_i & c_{\alpha_i} \cdot c_i & -s_{\alpha_i} \cdot c_i & a_i \cdot s_i \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Trong đó:  $c_i = \cos(\theta_i)$ ,  $s_i = \sin(\theta_i)$ ,  $c_{\alpha_i} = \cos(\alpha_i)$ ,  $s_{\alpha_i} = \sin(\alpha_i)$ .

Dựa vào các thông số trên bảng DH cho cánh tay robot 5 bậc tự do:

$$A_1 = \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2 \cdot c_2 \\ s_2 & c_2 & 0 & a_2 \cdot s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} c_3 & -s_3 & 0 & a_3 \cdot c_3 \\ s_3 & c_3 & 0 & a_3 \cdot s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$A_4 = \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$A_5 = \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

$$T_5 = A_1 \cdot A_2 \cdot A_3 \cdot A_4 \cdot A_5,$$

$$T_5 = \begin{bmatrix} c_1 \cdot c_{234} \cdot c_5 + s_1 \cdot s_5 & -c_1 \cdot c_{234} \cdot s_5 + s_1 \cdot c_5 & -c_1 \cdot s_{234} & c_1 \cdot (-d_5 \cdot s_{234} + a_3 \cdot c_{23} + a_2 \cdot c_2) \\ c_1 \cdot c_{234} \cdot c_5 - s_1 \cdot s_5 & -s_1 \cdot c_{234} \cdot s_5 - c_1 \cdot s_5 & -s_1 \cdot s_{234} & s_1 \cdot (-d_5 \cdot s_{234} + a_3 \cdot c_{23} + a_2 \cdot c_2) \\ -s_{234} \cdot c_5 & s_{234} \cdot s_5 & -c_{234} & d_1 - a_2 \cdot s_2 - a_3 \cdot s_{23} - d_5 \cdot c_{234} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Trong đó:  $c_{ijk} = \cos(q_i + q_j + q_k)$ ,  $s_{ijk} = \sin(q_i + q_j + q_k)$ .

### 3.2.5. Tính toán động học nghịch của cánh tay robot

Đối với bài toán động học nghịch (vấn đề cần quan tâm trong dự án này), yêu cầu đặt ra sẽ là ta cần tìm góc của các khớp sao để thỏa mãn với vị trí và hướng xoay được cho bởi ma trận

$$H = \begin{bmatrix} n_x & O_x & a_x & p_x \\ n_y & O_y & a_y & p_y \\ n_z & O_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Để giải quyết vấn đề trên ta sẽ có hai hướng tiếp cận: phương pháp đại số và phương pháp hình học.

### *Phương pháp đại số*

Phương pháp này sẽ chủ yếu là đồng nhất hệ số khi ta cho hai ma trận bằng nhau để giải:

$$T_5 = H.$$

Có thể thấy để giải được ra các góc, ta phải giải 1 hệ 5 phương trình 5 ẩn vô cùng phức tạp. Điều này dẫn đến 1 giải pháp hay hơn là giải bằng hình học.

### *Phương pháp hình học*

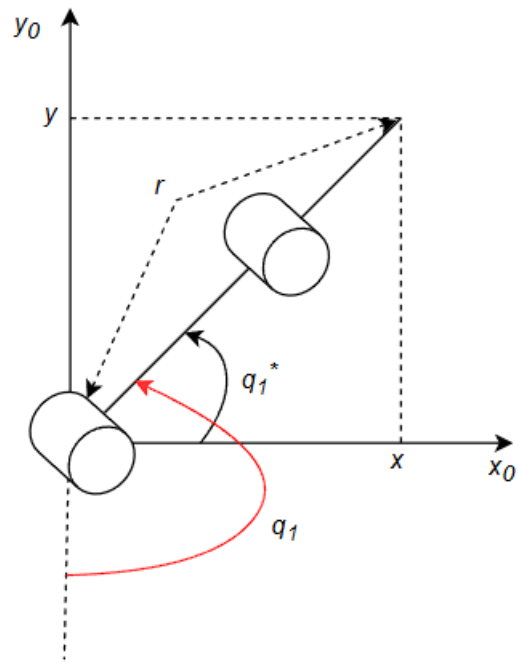
Trước tiên ta cần biết 1 điều, chỉ có 3 khớp đầu tiên sẽ có trách nhiệm với tọa độ vị trí điểm cuối, còn 2 khớp còn lại sẽ ảnh hưởng đến hướng xoay. Vì thế ta sẽ áp dụng phương pháp Kinematic Decoupling để tách ra làm 2 phần để giải.

- Giải động học ngược cho vị trí ( tìm góc của 3 khớp đầu tiên )

Từ ma trận H ta có được các tọa độ x, y, z của vật tương ứng là  $p_x$ ,  $p_y$ ,  $p_z$ .

Vì tọa độ mà 3 khớp đầu tiên khi xoay mang lại sẽ là tọa độ của khớp thứ 4 nên khi giải ta sẽ lấy tọa độ thực biến đổi về tọa độ khớp thứ 4 rồi mới giải

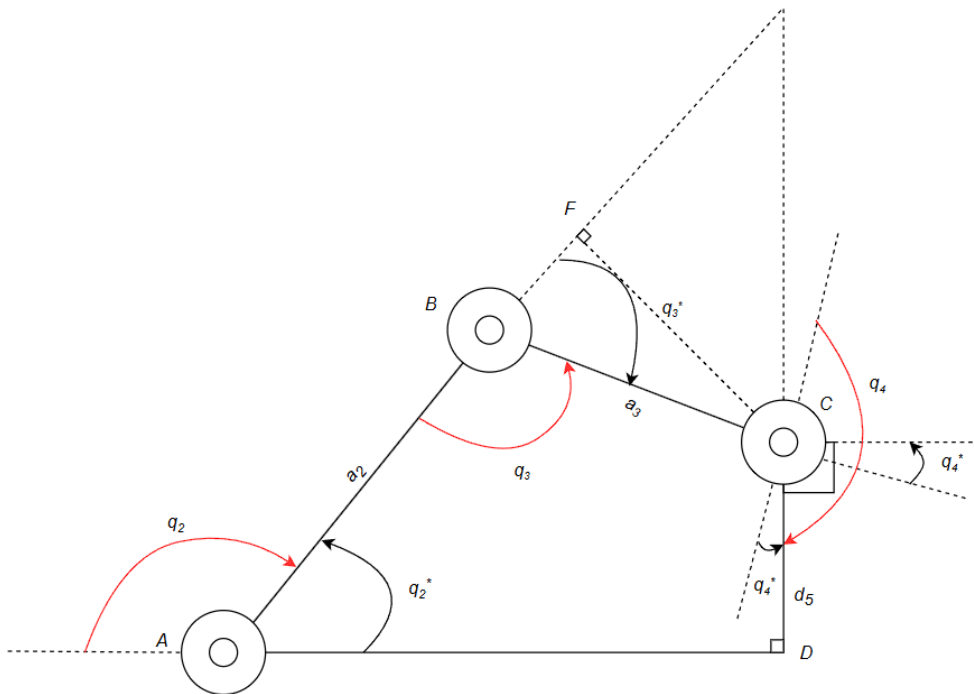




Hình 3.6. Góc chiếu thẳng đứng từ trên xuống của cánh tay robot

Từ hình trên ta tìm được góc thứ nhất của cánh tay robot bằng phương trình

$$\theta_1 = \arctan 2(y, x),$$





Hình 3.7. Góc nhìn ngang của cánh tay robot

Từ hình trên

$$r = \sqrt{x^2 + y^2},$$

$$s = z - d_1,$$

$$\cos(\widehat{ABC}) = \frac{r^2 + s^2 - a_2^2 - a_3^2}{-2 \cdot a_2 \cdot a_3} := D,$$

$$\rightarrow \theta_3 = \arctan2(\sqrt{1 - D^2}, D),$$

$$\widehat{BAD} = \pi - \theta_2,$$

$$\widehat{BAD} = \arctan2(s, r) + \arctan2(a_3 \cdot \sin(\pi - \theta_3), a_2 + a_3 \cdot \cos(\pi - \theta_3)),$$

$$\rightarrow \theta_2 = \pi - \arctan2(s, r) - \arctan2(a_3 \cdot \sin(\pi - \theta_3), a_2 + a_3 \cdot \cos(\pi - \theta_3)),$$

$$\theta_4 = 2\pi - \frac{\pi}{2} - \theta_3 - \widehat{BAD},$$

$$\theta_4 = \frac{\pi}{2} - \theta_3 + \theta_2$$

$\theta_5 = 0 + \frac{\pi}{2}$ . (vì các servo khi mới mua về đều phải preset 1 góc 90° rồi mới lắp đặt, các góc bên trên đã được công thêm khi vẽ hình)

### 3.2.6. Hoạch định quỹ đạo

Hoạch định quỹ đạo là thiết kế quỹ đạo chuyển động cho cánh tay robot. Để xác định được đường đi mong muốn của cánh tay robot theo thời gian, quỹ đạo có thể được tính toán và thiết kế trong hệ tọa độ truyền thống Oxyz ( Cartesian Space ) hoặc là trong không gian chứa các biến khớp ( Configuration Space ).

Trong thiết kế quỹ đạo cho cánh tay robot dựa trên các ứng dụng của chúng, ta thường gặp hai trường hợp sau:

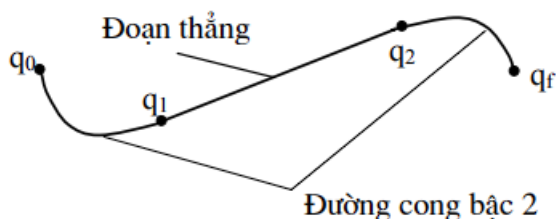
- Trường hợp 1: Đối với các ứng dụng chỉ quan tâm chủ yếu đến vị trí và hướng tại đích đến, điển hình là các công việc gấp thả của cánh tay robot,... Khi đó,

các điểm đầu và cuối sẽ là những điểm quan trọng của hành trình, còn dạng đường đi sẽ chỉ là vấn đề thứ yếu.

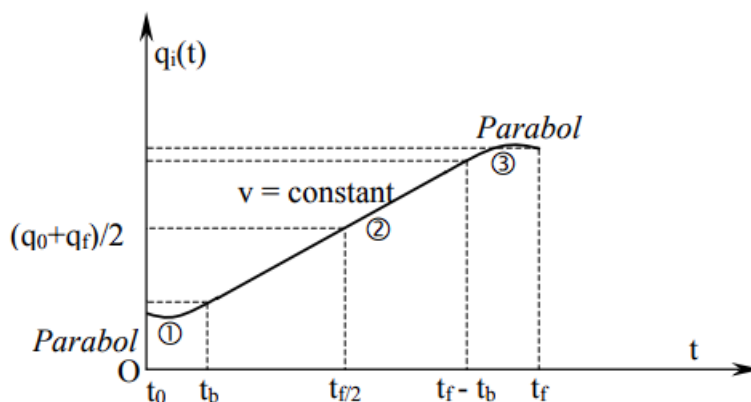
- Trường hợp 2: Đối với các ứng dụng quan tâm đến hành trình của cánh tay robot, phải luôn xác định đúng các điểm trên quỹ đạo xác định theo thời gian thực, có thể kể đến là sơn, hàn, cắt kim loại,... Vấn đề thiết kế quỹ đạo trong các trường hợp này là rất quan trọng. Quỹ đạo sẽ là đường sơn, vết hàn hay nét cắt trên sản phẩm, do đó nó quyết định trực tiếp đến chất lượng đầu ra của sản phẩm.

Đối với đồ án này, ta chỉ cần quan tâm đến ứng dụng của trường hợp 1. Quỹ đạo sẽ được thiết kế trong không gian chứa các biến khớp ( Configuration Space ), tức là xác định quy luật chuyển động của các biến khớp sao quỹ đạo của biến khớp sau khi hoàn thành chuyển động giống với quỹ đạo đã xác định.

Quỹ đạo được chọn để thiết kế ở đây là quỹ đạo LSPB: phối hợp đa thức bậc 2 với đa thức bậc 1.



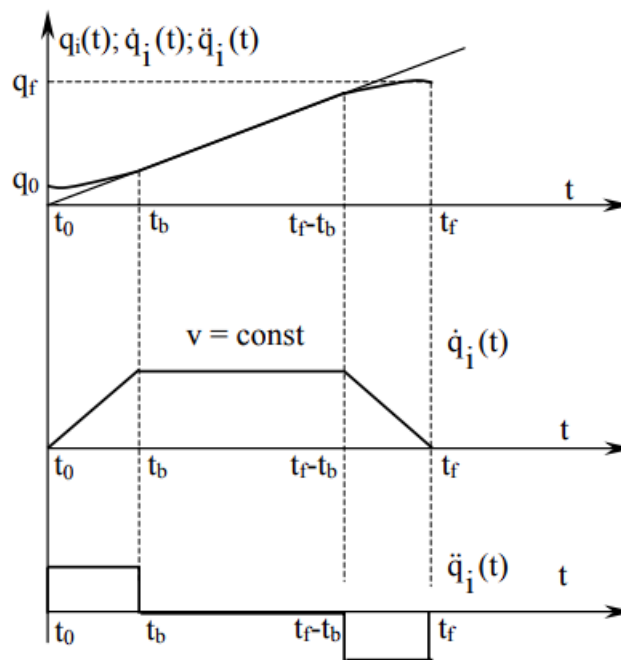
Hình . Quỹ đạo LSPB 1



LSPB sẽ quy hoạch quỹ đạo góc theo ba giai đoạn:

- Giai đoạn 1: Góc quay sẽ thay đổi nhanh dần, có quỹ đạo là đa thức bậc 2, dạng là đường parabol, vận tốc góc sẽ tăng đều tuyến tính theo gia tốc góc, gia tốc góc sẽ là  $+ a_{\max}$ .
- Giai đoạn 2: Góc quay sẽ thay đổi tuyến tính với vận tốc góc không đổi, vì vận tốc góc không đổi nên gia tốc góc sẽ bằng 0.
- Giai đoạn 3: Góc quay sẽ thay đổi chậm dần, có quỹ đạo là đa thức bậc 2, dạng là đường parabol, vận tốc góc sẽ giảm đều và tuyến tính theo gia tốc góc, gia tốc góc sẽ là  $- a_{\max}$ .

Các đặc tính chuyển động theo quỹ đạo LSPB của khớp  $q_i$  như sau:



Bài toán đặt ra với các tham số đầu vào cho trước như sau:

$q_0$  : Góc hiện tại.

$q_f$  : Góc tới mục tiêu.

$t_f$  : Thời gian đạt đến góc mục tiêu.

$v_{\max}$  : Tốc độ góc tối đa đạt được.

$a$  : Gia tốc góc tối đa.

Phương trình quỹ đạo góc theo thời gian như sau:

$$q(t) = \begin{cases} q_0 + \frac{1}{2}at^2, & 0 < t \leq t_b \\ q(t_b) + v_{max}(t - t_b) - t_b, & t_b < t \leq t_f - t_b \\ q_f - \frac{1}{2}at_f^2 + at_ft - \frac{1}{2}at^2, & t_f - t_b < t \leq t_f \end{cases}$$

Phương trình vận tốc góc theo thời gian như sau:

$$v(t) = \begin{cases} at, & 0 < t \leq t_b \\ v_{max}, & t_b < t \leq t_f - t_b \\ -at, & t_f - t_b < t \leq t_f \end{cases}$$

## CHƯƠNG 4: GIẢI THUẬT XỬ LÝ ẢNH

Trong chương này, ta sẽ đi sâu vào những phương pháp để xử lý ảnh đầu vào để có thể cung cấp đầu ra là tọa độ vật cần gấp. Mục tiêu của chương là xử lý từng frame ảnh theo thời gian thực và dùng giải thuật để khớp các đặc trưng với vật mẫu. Bên cạnh đó chương cũng sẽ đưa ra các giải pháp tinh chỉnh hệ tọa độ vật sao cho đúng với hệ tọa độ thực của cánh tay robot.

### 4.1. Tổng quan ( Thư viện và nền tảng lập trình )

Phần mềm được phát triển dựa trên OpenCV, một thư viện mã nguồn mở mạnh mẽ dành cho xử lý ảnh và thị giác máy, bằng ngôn ngữ lập trình Python. Việc xây dựng và biên dịch mã nguồn được thực hiện trên trình soạn thảo Geany trong môi trường hệ điều hành Linux.

Geany là chương trình soạn thảo văn bản đa hệ sử dụng nền tảng GTK+ dựa trên Scintilla với Môi trường phát triển tích hợp (IDE) cơ bản. Nó được thiết kế giới hạn phụ thuộc vào các gói riêng lẻ để có thời gian nạp ngắn. Nó có mặt rộng rãi trên các hệ điều hành, như Windows, Linux, BSD và Solaris. Các ngôn ngữ lập trình được hỗ trợ (theo tài liệu) là C, Java, PHP, HTML, Python, Perl và Pascal. Geany là một trong những trình soạn thảo đầy đủ chức năng cho Linux, vì các trình soạn thảo văn bản cho Linux thường có mức đơn giản tối đa. Nó gần giống với các trình soạn thảo văn bản cho Windows như NoteTab hoặc ConTEXT. Geany là phần mềm tự do được cấp theo giấy phép GNU GPL.

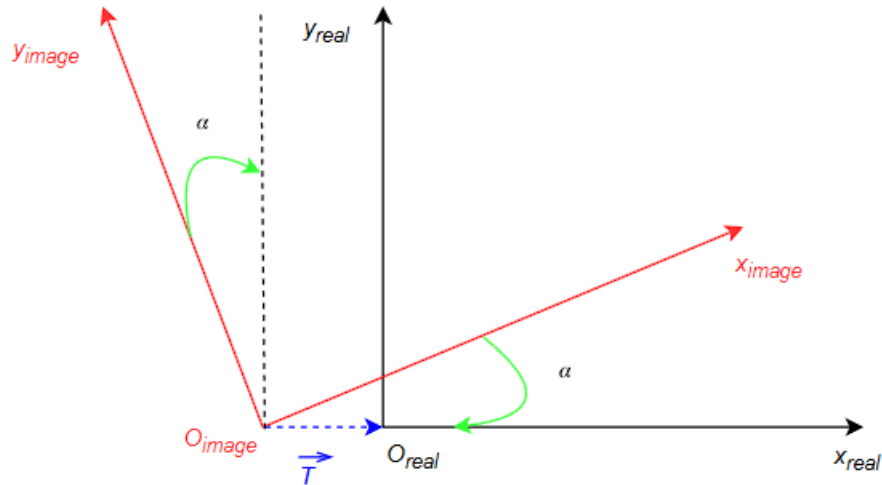
Python là một ngôn ngữ lập trình cấp cao, đa mục đích. Triết lý thiết kế của nó nhấn mạnh khả năng đọc mã với việc sử dụng thụt lề có ý nghĩa. Python có kiểu động và tự động thu gom rác. Nó hỗ trợ nhiều mô hình lập trình, bao gồm lập trình có cấu trúc (đặc biệt là thủ tục), lập trình hướng đối tượng và lập trình hàm. Python thường được mô tả là ngôn ngữ "đầy đủ công cụ" nhờ thư viện chuẩn phong phú của nó.

### 4.2. Hiệu chỉnh hệ tọa độ camera với hệ tọa độ tay máy

Vì khi xử lý các thuật toán và đọc vật về bằng camera, tọa độ sẽ có đơn vị pixel tương ứng với vị trí của vật trên frame ảnh. Do đó để tay máy có thể hiểu được và di chuyển đến chính xác vị trí vật, ta cần calib hệ tọa độ camera sao cho quy đổi về đúng với hệ tọa độ

thực của tay máy.

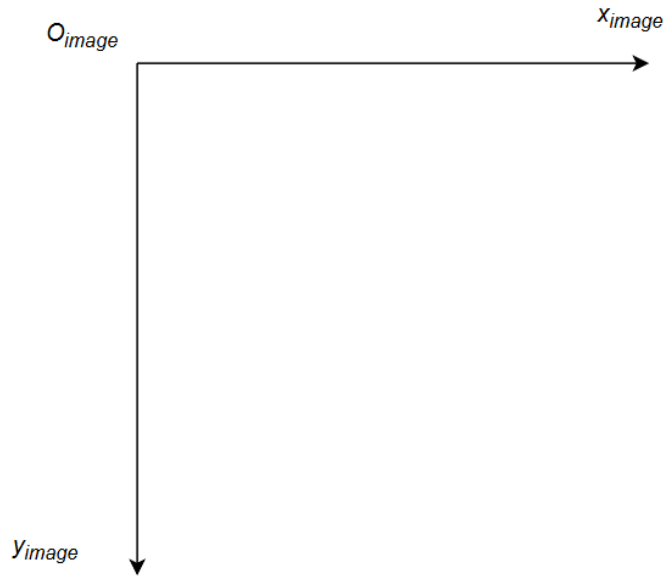
$$\begin{bmatrix} x_{real} \\ y_{real} \end{bmatrix} = S \cdot \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \cdot \begin{bmatrix} x_{image} \\ y_{image} \end{bmatrix} + \begin{bmatrix} t_{11} \\ t_{21} \end{bmatrix}.$$



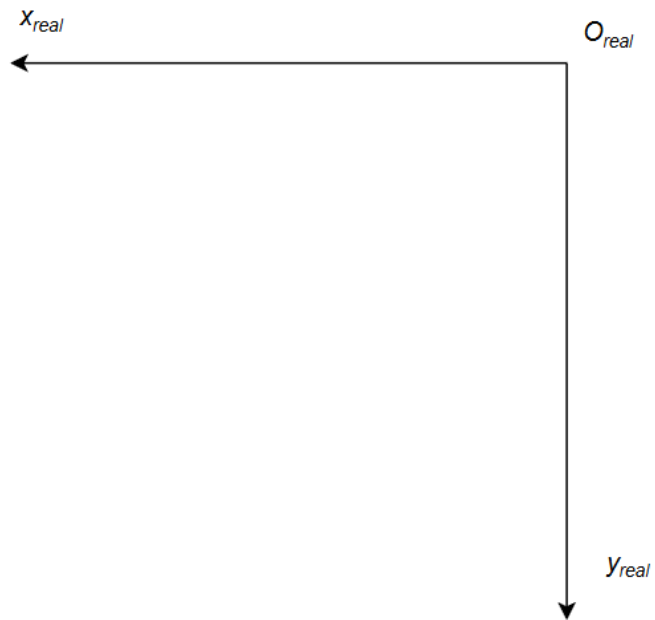
Hệ số S được tính bằng tỉ lệ giữa chiều rộng FOV và kích thước chiều rộng của frame ảnh dưới đơn vị pixel.

$$S = \frac{width(FOV)}{width(resolution)}$$

Mã trận xoay R sẽ mô tả hệ tọa độ thực tế sẽ lệch như thế nào so với hệ tọa độ trong ảnh về góc. Đối với đề án này, vì python khi xử lý tọa độ của khung ảnh hệ tọa độ ảnh sẽ có hình như sau:



Hệ tọa độ thực tế sẽ như sau:



Vì vậy ta sẽ tính ra được ma trận xoay R là:

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Ma trận dịch chuyển T sẽ mô tả độ dịch chuyển của gốc tọa độ so với gốc tọa độ trên ảnh.

### 4.3. Khớp vật với vật mẫu bằng giải thuật SURF

Thuật toán feature – matching SURF được dùng để phát hiện ra vật cần gấp trong đề tài này. Giải thuật tập trung vào việc khai thác và nhận diện các đặc trưng của một bức ảnh từ đó có thể tìm và match với những đặc trưng đang hiển thị theo thời gian thực, qua đó sẽ khoanh vùng được vật cần gấp phục vụ cho các bước tiếp sau đó.

SURF (Speeded Up Robust Features) là thuật toán được đề xuất bởi Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF là một phiên bản nâng cấp so với SIFT, với các ưu điểm là nhanh hơn và ổn định hơn khi nhận diện các đặc trưng của vật dưới góc nhìn khác hoặc bị ảnh hưởng bởi chuyển dịch và thay đổi.

SURF cũng tuân thủ theo các bước cơ bản như SIFT nhưng có đôi phần khác biệt khi ta đào sâu vào từng bước. SURF bao gồm ba bước chính:

- *Phát hiện điểm đặc trưng*

Điểm đặc trưng thường được phát hiện dựa trên điểm đó có phải cực trị địa phương hay không. Trong SIFT, người ta dùng DoG để phát hiện ra các cực trị địa phương. Trong SURF, thay vì sử dụng Difference of Gaussian (DoG) như trong SIFT, thuật toán này dùng Fast-Hessian – một phương pháp nhanh hơn dựa trên determinant của ma trận Hessian để phát hiện điểm đặc trưng. Tại mỗi điểm ảnh, ta sẽ được cung cấp tọa độ  $p = (x, y)$ , ma trận Hessian tại điểm  $p$  với mức scale  $\sigma$ :

$$H(p, \sigma) = \begin{bmatrix} L_{xx}(p, \sigma) & L_{xy}(p, \sigma) \\ L_{yx}(p, \sigma) & L_{yy}(p, \sigma) \end{bmatrix}.$$

Sau đó, ta sẽ tiến hành tính định thức ma trận để có thể đánh giá điểm tại đó:

$$\det(H) = L_{xx} \cdot L_{yy} - (0.9 \cdot L_{xy})^2.$$

Hệ số 0.9 được thêm vào nhằm điều chỉnh ảnh hưởng của đạo hàm hỗn hợp  $L_{xy}$ , giúp giảm thiểu nhiễu từ các vùng không phải là đặc trưng rõ rệt.

- $\det(H) > 0$  và  $L_{xx} > 0 \rightarrow$  cực trị là cực tiểu địa phương.

- $\det(H) > 0$  và  $L_{xx} < 0 \rightarrow$  cực trị là cực đại địa phương.



- $\det(H) < 0$   $\rightarrow$  điểm đang xét là điểm yên ngựa.
- $\det(H) = 0$   $\rightarrow$  không thể kết luận điều gì.

Để tính toán ma trận Hessian tại một điểm trong ảnh, ta cần phải sử dụng các bộ lọc Gaussian bậc hai (second-order Gaussian derivatives) để tính toán các đạo hàm này. Tuy nhiên, việc tính toán các đạo hàm bậc hai trực tiếp có thể rất tốn thời gian. Thay vì sử dụng các bộ lọc Gaussian phức tạp, SURF sử dụng box filters để xấp xỉ các bộ lọc Gaussian bậc hai. Ảnh tích phân giúp tính toán các box filters này một cách nhanh chóng.

Trong SURF, mỗi điểm đặc trưng được mô tả bằng một vector đặc trưng dựa trên các đặc trưng Haar-like (tương tự như trong các thuật toán nhận diện khuôn mặt của Viola-Jones). Đặc trưng Haar-like là các phép toán tích phân trên các vùng ảnh hình chữ nhật. Với ảnh tích phân, ta có thể tính tổng của pixel trong một vùng hình chữ nhật chỉ bằng việc tham chiếu tới các giá trị tại bốn góc của vùng đó. Điều này cho phép tính toán các đặc trưng Haar-like một cách nhanh chóng và hiệu quả mà không cần phải tính toán lại tổng của các pixel trong từng vùng cho mỗi điểm đặc trưng.

Công thức ảnh tích phân (integral image) sẽ có dạng như sau:

$$S(x, y) = \sum_{i=0}^x \sum_{j=0}^y I(i, j).$$

Trong SURF, không gian tỷ lệ được xây dựng bằng cách sử dụng box filters thay vì pyramid ảnh. Các lớp tỷ lệ được tạo ra bằng cách áp dụng bộ lọc có kích thước tăng dần ( $9 \times 9$ ,  $15 \times 15$ ,  $21 \times 21$ , v.v.). Tỷ lệ bộ lọc được ước tính bằng công thức:

$$\sigma_{approx} = \text{current filter size} \times \left( \frac{\text{base filter scale}}{\text{base filter size}} \right).$$

- *Mô tả vùng lân cận cục bộ*

Mục tiêu của một descriptor (đặc trưng) là cung cấp một mô tả duy nhất và mạnh mẽ cho một đặc trưng ảnh, ví dụ, bằng cách mô tả phân bố cường độ của các điểm

ảnh trong vùng lân cận của điểm quan tâm. Các descriptor thường được tính toán theo cách cục bộ, nghĩa là mỗi điểm quan tâm đã được xác định trước đó sẽ có một mô tả riêng.

Kích thước của descriptor có ảnh hưởng trực tiếp đến độ phức tạp tính toán và độ chính xác khi so khớp điểm. Một descriptor ngắn có thể giúp tăng tính mạnh mẽ trước sự thay đổi hình ảnh nhưng có thể không đủ phân biệt, dẫn đến nhiều trường hợp sai.

*Xác định phương hướng:* Để đạt được tính bất biến khi xoay, phương hướng của điểm quan tâm cần phải được xác định. Các phản hồi của sóng Haar theo cả hướng x và y trong một vùng tròn có bán kính  $6s$  xung quanh điểm quan tâm được tính toán, trong đó  $s$  là tỉ lệ mà điểm quan tâm được phát hiện. Các phản hồi này được trọng số bằng một hàm Gaussian và được vẽ trên không gian hai chiều. Phương hướng chủ yếu được xác định bằng cách tính tổng tất cả các phản hồi trong cửa sổ hướng trượt có kích thước  $\pi/3$ . Vectơ phương hướng dài nhất sẽ xác định phương hướng của điểm quan tâm.

*Descriptor dựa trên tổng phản hồi sóng Haar:* Để mô tả vùng xung quanh điểm, một vùng vuông được cắt ra, trung tâm là điểm quan tâm và hướng theo phương hướng đã xác định trước đó. Kích thước của cửa sổ này là  $20s$ . Vùng quan tâm được chia thành các vùng con vuông  $4 \times 4$ , và tại mỗi vùng con, các phản hồi sóng Haar được tính tại các điểm mẫu đều  $5 \times 5$ . Các phản hồi này được trọng số bằng Gaussian để tăng tính mạnh mẽ đối với biến dạng, nhiễu và dịch chuyển.

- *Khớp đặc trưng*

Bằng cách so sánh các descriptors có được từ các bức ảnh khác nhau, các cặp giống nhau sẽ được ghép lại để hoàn tất việc khớp đặc trưng.

SURF (Speeded Up Robust Features) và SIFT (Scale-Invariant Feature Transform) là hai thuật toán phổ biến trong xử lý ảnh để phát hiện và mô tả các đặc trưng, nhưng SURF có một số ưu điểm so với SIFT:

- *Tốc độ xử lý nhanh hơn:* SURF được thiết kế để tối ưu hóa tốc độ so với SIFT.

SURF sử dụng bộ lọc Haar wavelet trong tính toán các đặc trưng, giúp giảm đáng kể thời gian xử lý, đặc biệt là trong các ứng dụng thời gian thực.

- *Khả năng phát hiện đặc trưng mạnh mẽ hơn:* SURF cho phép phát hiện các đặc trưng có độ phân giải thấp hơn so với SIFT, giúp phát hiện được nhiều đặc trưng hơn trong một ảnh. Điều này đặc biệt hữu ích khi xử lý ảnh có độ phân giải thấp hoặc hình ảnh bị mờ.
- *Khả năng chống nhiễu tốt:* SURF có khả năng chống nhiễu tốt hơn SIFT trong nhiều trường hợp, đặc biệt là khi có các thay đổi nhỏ về độ sáng hoặc độ tương phản.
- *Khả năng sử dụng vector mô tả ngắn hơn:* SURF tạo ra các vector đặc trưng ngắn hơn so với SIFT (64 chiều thay vì 128 chiều), giúp tiết kiệm tài nguyên khi lưu trữ và tăng tốc độ tính toán khoảng cách giữa các đặc trưng.

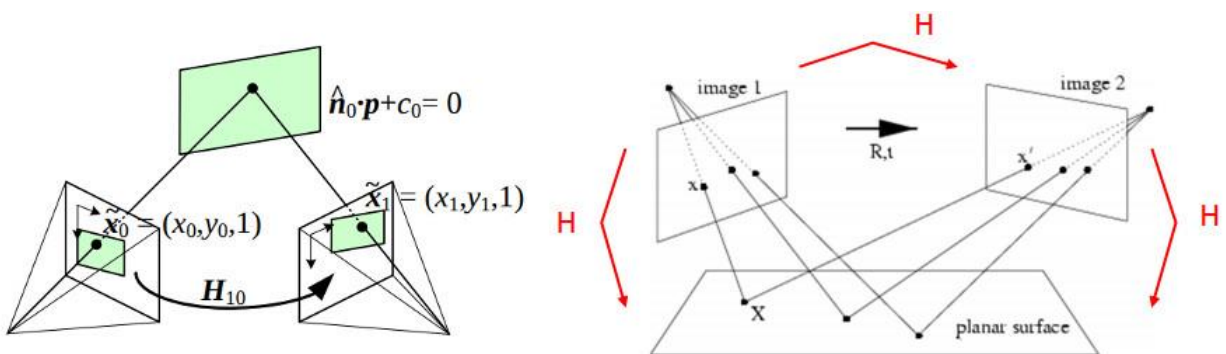


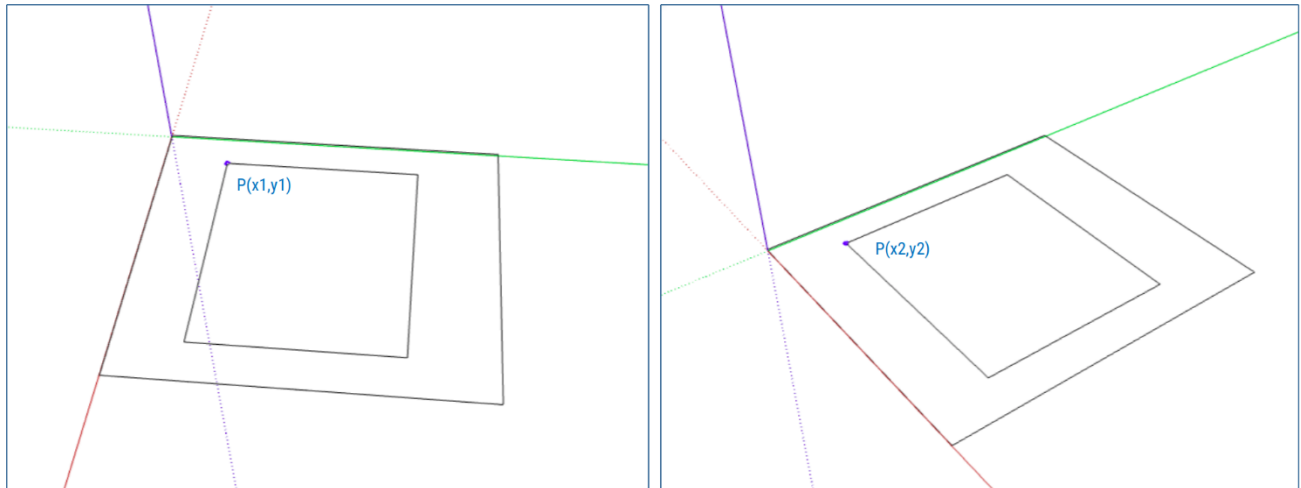
SURF đã được dùng để tìm ra các đặc trưng của nắp chai và khớp các đặc trưng lại

với nhau. Khi số đặc trưng tốt (good\_matches) lớn hơn hoặc bằng một số lượng cố định, ta có thể xác định được vùng tập hợp các đặc trưng đó sẽ có khả năng cao là vật cần tìm, sau đó sẽ dùng phương pháp Homography được nêu ở mục 4.4 để có thể xác định đúng hình dạng cũng như vị trí vật.

#### 4.4. Tìm homography

Vì trong khung hình video, các nắp chai không có vị trí hay hướng cố định, đồng thời ánh sáng ngoại cảnh cũng có thể tác động đến độ nhận diện, nên chúng ta sẽ áp dụng phương pháp ánh xạ hình ảnh từ ảnh mẫu sang ảnh trong khung hình để xác định vị trí, hướng, hoặc góc nhìn của camera so với vật. Để giải quyết vấn đề này, chúng ta có thể sử dụng homography, một phép biến đổi hình học mạnh mẽ giúp ánh xạ các điểm trong ảnh mẫu sang ảnh trong khung hình. Homography cho phép xác định mối quan hệ giữa các điểm trong không gian 2D của ảnh mẫu và ảnh thực tế, giúp chúng ta nhận diện được vị trí, góc quay và hướng của các nắp chai, bất chấp sự thay đổi về góc nhìn hay điều kiện ánh sáng. Bằng cách tính toán ma trận homography từ các điểm tương ứng giữa ảnh mẫu và ảnh trong khung hình, chúng ta có thể xác định chính xác vị trí của đối tượng và điều chỉnh góc nhìn của camera để cải thiện hiệu quả nhận diện.





Mục đích của homography là:

- *Ánh xạ điểm giữa hai ảnh:* Khi chúng ta có hai ảnh của cùng một đối tượng từ các góc nhìn khác nhau, homography giúp ánh xạ các điểm tương ứng giữa ảnh mẫu (reference image) và ảnh thực tế (input image).
- *Xử lý ảnh thay đổi phối cảnh:* Khi camera di chuyển hoặc đối tượng thay đổi góc nhìn, homography giúp chúng ta hiểu được sự thay đổi đó và điều chỉnh ảnh sao cho phù hợp.
- *Chuyển đổi mặt phẳng:* Mặt phẳng 2D trong ảnh mẫu có thể bị biến dạng khi nhìn từ một góc độ khác, và ma trận homography giúp chuyển các điểm trên mặt phẳng này sang một mặt phẳng khác trong ảnh.

Ma trận homography là một ma trận  $3 \times 3$  có khả năng ánh xạ điểm từ một ảnh này sang ảnh khác. Để minh họa quá trình này, giả sử chúng ta có hai điểm trong hai ảnh khác nhau, điểm  $P_1 = (x_1, y_1)$  trong ảnh 1 và điểm  $P_2 = (x_2, y_2)$  trong ảnh 2, mối quan hệ giữa chúng có thể được mô tả bởi ma trận homography  $H$ .

Công thức ánh xạ điểm giữa hai ảnh được mô tả như sau:

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = H \cdot \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}.$$

Trong đó:

- $P_1 = (x_1, y_1)$  là tọa độ điểm trong ảnh mẫu.
- $P_2 = (x_2, y_2)$  là tọa độ điểm tương ứng trong ảnh thực tế.
- $H$  là ma trận homography  $3 \times 3$ , với các phần tử như sau:

$$H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}.$$

Điểm  $P_1$  và  $P_2$  sẽ có được từ trích xuất đặc trưng ảnh mẫu và ảnh lấy được từ mỗi frame của live video.

Dựa trên các điểm tương ứng, ta có thể tạo ra các hệ phương trình để tính các phần tử trong ma trận homography  $H$ . Quá trình này có thể được giải quyết bằng phương pháp least squares hoặc RANSAC (để loại bỏ các điểm nhiễu và xác định homography chính xác).

Sau khi có ma trận  $H$ , ta có thể sử dụng nó để chuyển đổi một điểm từ ảnh mẫu sang ảnh thực tế bằng cách áp dụng công thức ánh xạ ở trên. Quá trình này cho phép ta tìm kiếm các điểm tương ứng giữa các ảnh hoặc biến đổi ảnh mẫu sao cho khớp với ảnh thực tế.



Ảnh trên cung cấp ví dụ cụ thể về sử dụng Homography để tìm ảnh ánh xạ. Nắp chai trong khung hình camera được chêm lên bởi một mảnh kim loại làm nắp chai không được đặt song song với mặt đất, tuy nhiên với ánh xạ thông qua ma trận Homography, nắp chai vẫn được khoanh vùng và trả về kết quả đúng như mong đợi.

#### 4.5. Đọc bán kính và tọa độ vật

Sau khi chọn đúng được vật cần gấp, thông tin cần được đọc tiếp theo là tọa độ và bán kính của vật. Đây là giai đoạn cần thiết nhằm loại bỏ bớt những thông tin không quan trọng và nhiễu. Việc xử lý ở giai đoạn này sẽ đưa về dạng thông tin sơ cấp của hình ảnh với ảnh nhị phân và các đường biên nhằm phục vụ dễ dàng cho việc trích xuất tọa độ vật.

Giai đoạn này sẽ bao gồm các công việc như: loại bỏ nhiễu, tăng độ tương phản, trích xuất cạnh.

Box filter có thể giúp giảm nhiễu trong ảnh nhưng không hoàn toàn loại bỏ được. Đây là một loại bộ lọc trung bình, trong đó giá trị của mỗi điểm ảnh trong ảnh được tính bằng trung bình cộng của các giá trị của các điểm ảnh xung quanh nó trong một cửa sổ hình vuông hoặc hình chữ nhật (tùy vào kích thước của cửa sổ).

Kết quả của box filter là làm mượt ảnh và giảm các biến động đột ngột của độ sáng, giúp giảm bớt nhiễu ngẫu nhiên (như nhiễu salt-and-pepper). Tuy nhiên, nó không thể loại bỏ nhiễu một cách hoàn hảo và có thể làm mất chi tiết của ảnh, vì mỗi điểm ảnh mới là trung bình của các điểm ảnh xung quanh, dẫn đến làm mờ các cạnh của các đối tượng trong ảnh.

Điều chỉnh độ sáng là thay đổi đồng thời giá trị các mức xám của mỗi kênh màu trong ảnh lên một không nhất định, hiển thị rõ các thông tin có mức sáng thấp đồng thời làm mờ các thông tin có mức sáng cao – là các thông tin không cần thiết trong phạm vi đề tài. Phép biến đổi độ sáng được biểu diễn ở công thức

$$g(x, y) = \alpha \cdot f(x, y) + \beta.$$

Trong đó:

$f, g$  là mức xám trước và sau phép biến đổi.

$x, y$  là tọa độ pixel trên ảnh.

$\alpha, \beta$  là thông số của phép biến đổi.

Điều chỉnh độ tương phản là giãn cách sự phân bố mức xám của hai vùng là đối tượng và nền, nhằm tạo thành một histogram phân bố rõ ràng thành hai thái cực vùng mức xám, làm cho đối tượng nổi bật, dễ phát hiện và xử lý. Phép điều chỉnh độ tương phản được biểu diễn ở công thức:

$$g(x, y) = \left( \frac{f(x, y)}{255} \right)^\gamma \cdot 255.$$

Trong đó:

$f, g$  là mức xám trước và sau phép biến đổi.

$x, y$  là tọa độ pixel trên ảnh.

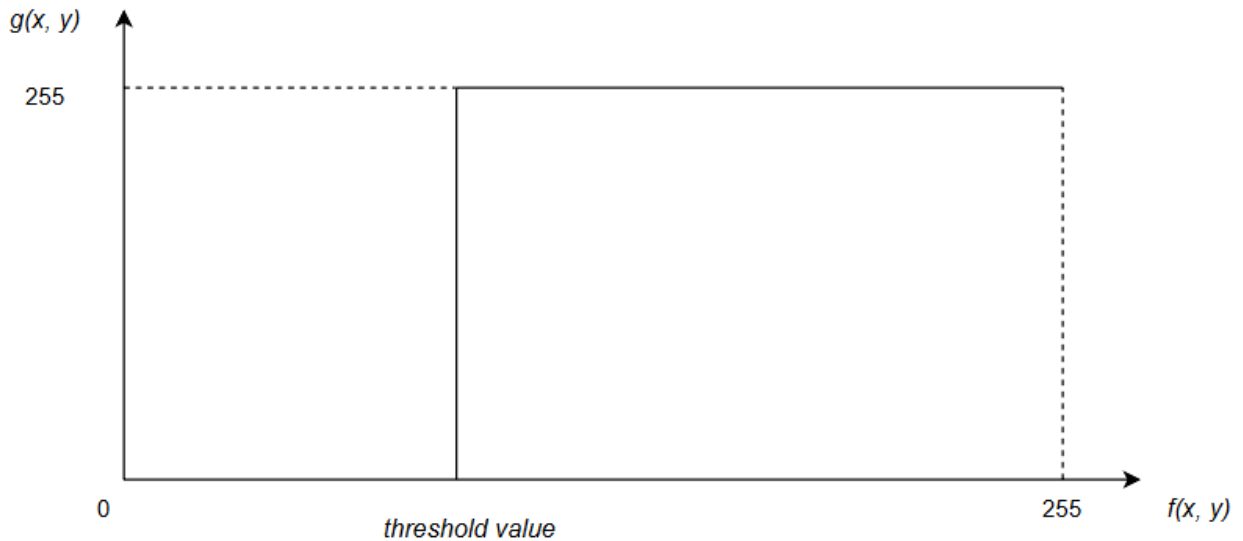
$\gamma$  là thông số của phép biến đổi.





Lấy ngưỡng ảnh là phương pháp trích xuất đặc trưng cơ bản của ảnh với yêu cầu đầu ra là cô lập và xác định được đối tượng cần trích xuất. Lấy ngưỡng ảnh có nhiều hình thức như lấy ngưỡng tĩnh, lấy ngưỡng thích nghi, lấy ngưỡng otsu theo vùng,...

Trong phạm vi đề tài sử dụng lấy ngưỡng tĩnh với mức xám lấy ngưỡng là 75 do sau khi xử lý với các bước trước đó, ta đã thấy hầu như tách biệt được phần nền với vật do độ sáng được đẩy lên cao, các chi tiết cần quan tâm được làm bật lên trên nền trắng, do đó chỉ cần lấy ngưỡng tĩnh với mức ngưỡng trên đã có thể cô lập vật.



Kết quả sau khi lấy ngưỡng:



Contours (đường viền) trong xử lý ảnh là các đường cong liên tục dùng để biểu diễn biên của các vật thể trong ảnh, thường được xác định dựa trên sự thay đổi mức xám từ nền sang viền. Điều này giúp đưa ra các ý tưởng về phát hiện đường viền dựa trên đạo hàm. Contours giúp xác định hình dạng, kích thước, và vị trí của các đối tượng, rất hữu ích trong các ứng dụng như phát hiện đối tượng, phân đoạn ảnh và theo dõi chuyển động.

Hàm `cv2.findContours` trong OpenCV được sử dụng để tìm contours trong một ảnh nhị phân. Hàm này nhận vào ảnh nhị phân và trả về danh sách các contours tìm được, mỗi contour là một tập hợp các điểm biên liên tiếp của một đối tượng trong ảnh. Cú pháp của hàm:

```
contours, hierarchy = cv2.findContours(image, mode, method)
```

Trong đó:

- `image`: Ảnh nhị phân đầu vào (ảnh trắng đen), thường là ảnh kết quả sau khi áp dụng `threshold` hoặc `edge detection`.
- `mode`: Chế độ lấy contours, ví dụ `cv2.RETR_EXTERNAL` (lấy các contours ngoài cùng) hoặc `cv2.RETR_TREE` (lấy toàn bộ contours theo cấu trúc phân cấp).
- `method`: Phương pháp xấp xỉ contours, như `cv2.CHAIN_APPROX_SIMPLE` (giảm bớt các điểm dư thừa) hoặc `cv2.CHAIN_APPROX_NONE` (lưu toàn bộ điểm trên contour).

Khi có được các tọa độ của các điểm trên contour, trọng tâm của vật được tính bằng cách lấy trung bình tọa độ tất cả các pixel của vật như công thức:

$$center(x, y) = \frac{1}{n} \cdot \sum_{i=0}^{n-1} pixel_i(x, y).$$

Trong đó:  $n$  là tổng số pixel được tổng hợp thuộc đường viền.

Bán kính của vật được tính bằng trung bình khoảng cách từ trọng tâm đến các điểm ngoài đường bao của vật như công thức:

$$r = \frac{1}{n} \sum_{i=0}^{n-1} norm(center(x, y) - contour(x, y)).$$



```
[0:48:07.119601545] [3284] INFO Camera camera_manager.cpp:325 libcamera v0.3.2+  
27-7330f29b  
[0:48:07.151820312] [3306] WARN RPiSdn sdn.cpp:40 Using legacy SDN tuning - ple  
ase consider moving SDN inside rpi.denoise  
[0:48:07.154275652] [3306] INFO RPI vc4.cpp:447 Registered camera /base/soc/i2c  
0mux/i2c@1/imx708@1a to Unicam device /dev/media3 and ISP device /dev/media0  
[0:48:07.154508799] [3306] INFO RPI pipeline_base.cpp:1126 Using configuration  
file '/usr/share/libcamera/pipeline/rpi/vc4/rpi_apps.yaml'  
[0:48:07.162329242] [3284] INFO Camera camera.cpp:1197 configuring streams: (0)  
640x480-XPB8888 (1) 1536x864-SBGR10_CSI2P  
[0:48:07.162875184] [3306] INFO RPI vc4.cpp:622 Sensor: /base/soc/i2c0mux/i2c@1  
/imx708@1a - Selected sensor format: 1536x864-SBGR10_1X10 - Selected unicam for  
mat: 1536x864-pBAA  
Nhấn 'q' để thoát.  
12.1  
64  
Gửi tọa độ x: 12.1, y: 8.2  
█
```

## **CHƯƠNG 5: GIẢI THUẬT ĐIỀU KHIỂN**

Trong chương này, chúng ta sẽ đi sâu vào thiết kế bộ điều khiển cho từng servo từ các góc quay đã được tính toán thông qua động học robot. Bên cạnh đó sẽ là giải thuật chung điều khiển toàn bộ hệ thống