

ĐẠI HỌC ĐÀ NẴNG  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA ĐIỆN

## ĐỒ ÁN CAPSTONE

NGÀNH: KỸ THUẬT ĐIỀU KHIỂN VÀ TỰ ĐỘNG HÓA

ĐỀ TÀI:

### CÁNH TAY ROBOT 3DOF ĐÀN HỒI VÀ THU THẬP DỮ LIỆU

Người hướng dẫn: TS. NGUYỄN HOÀNG MAI

Sinh viên thực hiện:

1. MAI XUÂN DŨNG – MSSV:105200296 – LỚP: 20TDH1
2. PHẠM VĂN TIẾN – MSSV: 105200315 – LỚP: 20TDH1

Đà Nẵng, 06/2025

## TÓM TẮT

Tên đề tài: Cánh tay Robot 3DoF đàn hồi và thu thập dữ liệu

Sinh viên thực hiện: Mai Xuân Dũng – MSSV: 105200296 – Lớp: 20TDH1

Phạm Văn Tiến – MSSV: 105200315 – Lớp: 20TDH1

Nội dung chính của đề tài, theo mô phỏng của nhóm gồm:

- Tìm hiểu về cấu trúc đàn hồi trong Robot
- Nghiên cứu động lực học của Robot
- Thiết kế, điều khiển chuyển động của cánh tay Robot đàn hồi.

**KHOA ĐIỆN**

**NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

TT	Họ tên sinh viên	Số thẻ SV	Lớp	Ngành
1	Mai Xuân Dũng	105200296	20TDH1	Kỹ thuật điều khiển và tự động hóa
2	Phạm Văn Tiến	105200315	20TDH1	Kỹ thuật điều khiển và tự động hóa

1. Tên đề tài đồ án:

*Cánh tay Robot 3DoF đàn hồi và thu thập dữ liệu*

2. Đề tài thuộc diện:  Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện

3. Các số liệu và dữ liệu ban đầu:

.....  
.....  
.....

4. Nội dung các phần thuyết minh và tính toán:

a. Phần chung:

TT	Họ tên sinh viên	Nội dung
1	Mai Xuân Dũng	- Đọc tài liệu tham khảo. - Nghiên cứu và lập kế hoạch cho đồ án.
2	Phạm Văn Tiến	- Tìm hiểu về Robot và tính đàn hồi trong Robot - Kiểm nghiệm và đánh giá kết quả mô hình.

b. Phần riêng

TT	Họ tên sinh viên	Nội dung
1	Mai Xuân Dũng	- Nghiên cứu động lực học Robot. - Mô phỏng Matlab. - Thiết kế giao diện điều khiển
2	Phạm Văn Tiến	- Thiết kế phần cứng cánh tay Robot - Tính toán, lựa chọn động cơ. - Thiết kế điều khiển chuyển động cánh tay Robot.

5. Các bản vẽ, đồ thị ( ghi rõ các loại và kích thước bản vẽ ):

*a. Phần chung:*

TT	Họ tên sinh viên	Nội dung
1	Mai Xuân Dũng	- Bản vẽ thiết kế mô hình
2	Phạm Văn Tiến	

*b. Phần riêng*

TT	Họ tên sinh viên	Nội dung
1	Mai Xuân Dũng	- Sơ đồ khối của hệ thống - Lưu đồ thuật toán giao diện điều khiển
2	Phạm Văn Tiến	- Vẽ sơ đồ mạch điều khiển - Lưu đồ thuật toán điều khiển

<i>6. Họ tên người hướng dẫn:</i>	<i>Phần/ Nội dung:</i>
TS. Nguyễn Hoàng Mai	- Nêu ra yêu cầu của dự án. - Hướng dẫn tư vấn giải pháp, giao nhiệm vụ thực hiện. - Đề xuất các giải pháp về phần cơ khí, điều khiển. - Hướng dẫn làm thuyết trình báo cáo dự án. - Theo dõi tiến độ dự án.

7. Ngày giao nhiệm vụ đồ án: 17/2/2025

8. Ngày hoàn thành đồ án: 15/6/2025

*Đà Nẵng, ngày 15 tháng 6 năm 2025*

**Trưởng bộ môn tự động hóa**

**Người hướng dẫn**

TS. Giáp Quang Huy

TS. Nguyễn Hoàng Mai

## PHIẾU KIỂM SOÁT TIẾN ĐỘ LÀM ĐỒ ÁN TỐT NGHIỆP

(Phiếu dành cho người hướng dẫn/sinh viên)

Họ tên sinh viên: Mai Xuân Dũng

Số thẻ SV: 105200296

Phạm Văn Tiến

Số thẻ SV: 100200315

Tên đề tài ĐATN: Cánh tay Robot 3DoF đàn hồi và thu thập dữ liệu

Họ tên người HD: TS. Nguyễn Hoàng Mai

Đơn vị: Khoa điện ĐHBK ĐHN

Tuần	Ngày	Khối lượng		GVHD ký tên
		đã thực hiện (%)	tiếp tục thực hiện (%)	
1	17/2/2025	Nhận đề tài	Nghiên cứu lý thuyết liên quan đến đề tài	
2	28/2/2025	Tìm hiểu, lựa chọn vật liệu cho cánh tay đàn hồi	Lựa chọn lò xo làm thanh cánh tay cho Robot	
3	5/3/2025	Thực hiện thiết kế cánh tay Robot để tiến hành gia công cơ khí	Viết báo cáo chương 1	
4	15/3/2025	Duyệt lần 1: Đánh giá khối lượng hoàn thành 20 %: Được tiếp tục làm ĐATN <input type="checkbox"/> Không tiếp tục thực hiện ĐATN <input type="checkbox"/>		
5	23/3/2025	Hoàn thành bản thiết kế cơ khí	Tới xưởng cơ khí để thực hiện gia công	
6	23/3/2025	Nguyên cứu tính toán động lực học Robot và mô phỏng trên matlab	Tìm hiểu về các tính toán động lực học liên quan đến đàn hồi	
7	2/4/2025	Nghiên cứu, tính toán bộ điều khiển PID cho cánh tay Robot	Viết báo cáo chương 2	
8	10/4/2025	Duyệt lần 2: Đánh giá khối lượng hoàn thành 50 %: Được tiếp tục làm ĐATN <input type="checkbox"/> Không tiếp tục thực hiện ĐATN <input type="checkbox"/>		
9	18/4/2025	Tính toán lựa chọn các thiết bị điều khiển cho mô hình	Tiến hành lắp ráp hoàn thiện mô hình thực nghiệm	
10	25/4/2025	Vẽ sơ đồ nguyên lý mạch điều khiển, xây dựng thuật toán điều khiển	Lắp ráp mạch điều khiển theo sơ đồ nguyên lý	

11	5/5/2025	Viết code điều khiển và giao diện cho mô hình thực nghiệm	Viết báo cáo chương 3	
12	11/5/2025	Duyệt lần 3: Đánh giá khối lượng hoàn thành 80 %: Được tiếp tục làm ĐATN <input type="checkbox"/> Không tiếp tục thực hiện ĐATN <input type="checkbox"/>		
13	20/5/2025	Tiếp tục tối ưu, hoàn thiện phần điều khiển mô hình	Kiểm nghiệm và đánh giá kết quả mô hình	
14	1/6/2025	Hoàn thiện mô hình thực nghiệm và các kết quả đánh giá	Viết báo cáo chương 4	
15	15/6/2025	Hoàn thiện thuyết minh ĐATN	Nộp bản thuyết minh, chuẩn bị slide báo cáo	

## LỜI NÓI ĐẦU VÀ CẢM ƠN

- Trong suốt thời gian học tập tại trường Đại học Bách Khoa – Đại Học Đà Nẵng, Ban giám hiệu nhà trường, Khoa Điện và Bộ môn Tự động hóa đã tạo điều kiện cho chúng em học tập, thực hành cùng các thầy cô đã tận tình giảng dạy truyền đạt kiến thức và kinh nghiệm quý báu cho chúng em. Đó sẽ là những kiến thức nền tảng, những hành trang vững chắc để chúng em tiếp tục học tập và làm việc. Chúng em xin chân thành cảm ơn quý thầy, cô.

- Sau một quá trình tìm hiểu và làm đề tài, dưới sự hướng dẫn từ Thầy Nguyễn Hoàng Mai đã giúp chúng em hoàn thành đề án tốt nghiệp này. Chúng em xin gửi lời cảm ơn chân thành và sâu sắc đến Thầy.

- Với điều kiện thời gian cũng như kiến thức vẫn còn hạn chế của mình, đề án tốt nghiệp này không thể tránh được những thiếu sót. Nhóm em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các thầy cô để nhóm em có điều kiện bổ sung, nâng cao ý thức của mình, phục vụ tốt hơn trong công việc thực tế sau này.

Chúng em xin chân thành cảm ơn!

## **LỜI CAM ĐOAN LIÊM CHÍNH HỌC THUẬT**

- Chúng tôi xin cam đoan rằng đồ án tốt nghiệp với đề tài “Cánh tay Robot 3DoF đàn hồi và thu thập dữ liệu” là công trình nghiên cứu của nhóm dưới sự hỗ trợ, giúp đỡ của thầy Nguyễn Hoàng Mai. Những phần có sử dụng tài liệu tham khảo trong đồ án đã được liệt kê và nêu rõ ràng tại phần tài liệu tham khảo. Đồng thời những số liệu hay kết quả trình bày trong đồ án đều mang tính chất trung thực, không sao chép.

- Nếu như sai sót chúng tôi xin chịu hoàn toàn trách nhiệm và chịu tất cả các kỷ luật của bộ môn cũng như nhà trường đề ra.

Sinh viên thực hiện

Mai Xuân Dũng

Phạm Văn Tiến

# MỤC LỤC

Tóm tắt .....	i
Nhiệm vụ đồ án tốt nghiệp .....	ii
Phiếu kiểm soát tiến độ làm đồ án tốt nghiệp .....	iv
Lời nói đầu và cảm ơn.....	vi
Lời cam đoan liên chính học thuật.....	vii
Mục lục.....	viii
Danh mục bảng biểu, hình ảnh.....	x
Mở đầu.....	1
<b>Chương 1: Tổng quan đề tài.....</b>	<b>3</b>
<b>1.1. Tổng quan.....</b>	<b>3</b>
1.1.1. Tổng quan về robot công nghiệp .....	3
1.1.2. Khái niệm về robot công nghiệp .....	3
1.1.3. Lịch sử phát triển.....	4
1.1.4. Phân loại robot công nghiệp .....	5
1.1.5. Ứng dụng robot công nghiệp.....	6
1.1.6. Bậc tự do của Robot .....	7
1.1.7. Các thành phần chính của Robot công nghiệp .....	8
<b>1.2 Tìm hiểu về tính đàn hồi trong cánh tay Robot.....</b>	<b>10</b>
1.2.1. Đặt vấn đề.....	10
1.2.2. Tay máy có tính đến độ đàn hồi.....	11
1.2.2.1. Ứng suất.....	11
1.2.2.2. Biến dạng.....	11
1.2.2.3. Uốn .....	12
1.2.2.4. Đàn hồi .....	13
1.2.3. Thuận lợi và khó khăn của Robot cánh tay đàn hồi .....	14
1.2.3.1. Thuận lợi.....	14
1.2.3.2. Khó khăn.....	14
1.2.4. Một số nghiên cứu về tay máy có tính đến độ đàn hồi.....	14
1.2.4.1. Trong nước.....	14
1.2.4.2. Ngoài nước .....	15
1.2.5. Ứng dụng thực tiễn của Robot cánh tay đàn hồi .....	16
1.2.6. Ý nghĩa khoa học và thực tiễn.....	17
<b>1.3 Kết luận chương 1.....</b>	<b>17</b>
<b>Chương 2: Xây dựng mô hình toán học.....</b>	<b>18</b>
<b>2.1 Cấu trúc.....</b>	<b>18</b>
<b>2.2 Nguyên lý hoạt động.....</b>	<b>18</b>

<b>2.3. Phương trình động lực học .....</b>	<b>19</b>
2.3.1. Tính toán động học thuận .....	20
2.3.2. Tính toán động học nghịch .....	21
2.3.3. Tính toán động lực học .....	23
2.3.4. Xây dựng mô hình toán học cho có xét đến đàn hồi .....	28
<b>2.4. Tổng quan về bộ điều khiển PID .....</b>	<b>30</b>
2.4.1. Cấu trúc chung của BDK.....	30
2.4.2. Phương pháp xác định tham số PID .....	31
2.4.3. Phương pháp Ziegler-Nichols.....	34
2.4.4. Điều khiển PID số.....	35
2.4.5. Sơ đồ cấu trúc HTĐK Robot sử dụng PID .....	37
<b>2.5. Đề xuất cho hệ thống .....</b>	<b>37</b>
2.5.1. Phần cứng .....	37
2.5.2. Phần mềm .....	38
<b>2.6. Kết luận chương 2.....</b>	<b>38</b>
<b>Chương 3: Thiết kế và tính chọn hệ thống điều khiển .....</b>	<b>39</b>
<b>3.1. Tính toán và lựa chọn thiết bị.....</b>	<b>39</b>
<b>3.2. Thiết kế phần cứng hệ thống .....</b>	<b>41</b>
<b>3.3. Sơ đồ và thuật toán điều khiển .....</b>	<b>42</b>
3.3.1. Sơ đồ nguyên lí mạch điều khiển .....	42
3.3.2. Lưu đồ thuật toán.....	43
<b>3.4. Thiết kế giao diện điều khiển.....</b>	<b>45</b>
<b>3.5. Chi phí đầu tư .....</b>	<b>46</b>
<b>3.6. Kết luận chương 3.....</b>	<b>46</b>
<b>Chương 4: Mô phỏng, thực nghiệm và đánh giá.....</b>	<b>47</b>
<b>4.1. Mô phỏng động lực học cánh tay robot .....</b>	<b>47</b>
<b>4.2. Mô hình thực nghiệm .....</b>	<b>50</b>
<b>4.3. Kết quả thực hiện .....</b>	<b>51</b>
4.3.1. Điều khiển theo góc đặt .....	51
4.3.2. Điều khiển theo vị trí.....	52
4.3.3. Điều khiển theo Vector .....	52
<b>KẾT LUẬN .....</b>	<b>54</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>1</b>
<b>PHỤ LỤC .....</b>	<b>2</b>

## DANH MỤC BẢNG BIỂU, HÌNH ẢNH

Bảng 1.1: Số lượng Robot các nước công nghiệp phát triển.....	5
Bảng 1.2: Đặc tính đàn hồi của vật liệu.....	13
Bảng 2.1: Lập bảng D-H (Denavit – Hartenberg). .....	20
Bảng 2.2: Ảnh hưởng của mỗi bộ điều khiển $K_p$ , $K_i$ , $K_d$ .....	33
Bảng 2.3: Các tham số PID theo phương pháp Ziegler-Nichols thứ nhất.....	34
Bảng 2.4: Các tham số PID theo phương pháp Ziegler-Nichols thứ 2.....	35
Bảng 3.1: Thông số các thiết bị.....	40
Bảng 3.2: Chi phí đầu tư .....	46
Hình 1.1: Robot trong công nghiệp .....	3
Hình 1.3: Robot phẫu thuật .....	6
Hình 1.4: Robot trinh thám.....	7
Hình 1.5: Robot ABB dùng trong giảng dạy Trường Đại học bách khoa Đà Nẵng.....	7
Hình 1.6: Các thành phần chính của hệ thống Robot.....	8
Hình 1.7: Robot kiểu tọa độ Đề các .....	9
Hình 1.8: Robot kiểu tọa độ trụ .....	9
Hình 1.9: Robot kiểu tọa độ cầu.....	9
Hình 1.10: Robot hoạt động theo hệ tọa độ góc .....	10
Hình 1.11: Robot kiểu SCARA .....	10
Hình 1.12: Vật liệu bị biến dạng.....	12
Hình 1.13: Vật liệu biến dạng uốn.....	13
Hình 1.14: Cầu trục tháp GM/GRT66 – 57 H .....	16
Hình 1.15: BionicSoftArm với bảy cơ cấu chấp hành khí nén và nhiều bậc tự do.....	17
Hình 2.1: Sơ đồ khối của hệ thống .....	18
Hình 2.3: Hệ trục tọa độ của tay máy .....	23
Hình 2.4: Hệ robot đàn hồi.....	28
Hình 2.5: Cấu trúc hệ thống điều khiển.....	30
Hình 2.6: Bộ điều khiển PID .....	31
Hình 2.7: Đồ thị đáp ứng theo thời gian với $K_p$ thay đổi.....	31
Hình 2.8: Đồ thị đáp ứng theo thời gian với $K_i$ thay đổi.....	32
Hình 2.9: Đồ thị đáp ứng theo thời gian với $K_d$ thay đổi.....	33

Hình 2.10: Đáp ứng nấc của hệ hở có dạng S .....	34
Hình 2.11: Xác định hằng số khuếch đại tới hạn.....	35
Hình 2.12: Đáp ứng nấc của hệ kín khi $k = k_{th}$ .....	35
Hình 2.13: Xấp xỉ đạo hàm của biến sai số $e$ .....	36
Hình 2.14: Xấp xỉ tích phân của biến sai số $e$ .....	36
Hình 2.15: Sơ đồ khối hệ thống điều khiển Robot sử dụng PID.....	37
Hình 3.1: Mô hình cánh tay Robot 3DoF đàn hồi .....	42
Hình 3.2: Sơ đồ nguyên lí mạch điều khiển .....	42
Hình 3.3: Lưu đồ thuật toán điều khiển.....	43
Hình 3.4: Lưu đồ thuật toán cho giao diện điều khiển .....	44
Hình 3.5: Giao diện điều khiển.....	45
Hình 4.1: Khối phương trình .....	48
Hình 4.2: Khối PID cho $\tau_1$ .....	48
Hình 4.3: Khối PID cho $\tau_2$ .....	49
Hình 4.4: Sơ đồ mô phỏng HTĐK Robot sử dụng bộ PID .....	49
Hình 4.5: Kết quả mô phỏng .....	50
Hình 4.6: Mô hình thực nghiệm .....	50
Hình 4.7: Đồ thị kết quả điều khiển theo góc đặt.....	51
Hình 4.8: Đồ thị kết quả điều khiển theo vị trí.....	52
Hình 4.9: Đồ thị kết quả điều khiển theo vecto điều khiển theo vector .....	53

## MỞ ĐẦU

### Tính cần thiết của đề tài

- Robot công nghiệp đang được ứng dụng rất rộng rãi trong các ngành sản xuất hiện đại. Với ưu điểm độ chính xác cao, thao tác lặp lại liên tục, tiêu tốn ít năng lượng và làm việc trong những môi trường khắc nghiệt. Các quốc gia phát triển hàng đầu thế giới đã ra sức ứng dụng robot vào các ngành công nghiệp chủ chốt nhằm đạt tới trình độ công nghiệp hóa và tự động hóa cao.

- Tuy nhiên Robot thường dùng trong công nghiệp có khối lượng rất lớn nhưng tải trọng thì nhỏ (ví dụ Robot tải trọng được 10kg thì Robot phải nặng ~ 50kg), tỉ lệ tải trọng mang được và khối lượng Robot là rất thấp. Việc sử dụng cánh tay to và nặng chỉ phù hợp với những Robot đứng yên một chỗ như trong nhà xưởng, còn những Robot di chuyển trên xe, máy bay... dùng cánh tay nặng sẽ không phù hợp. Do đó phải sử dụng những cánh tay nhẹ nhưng chịu được tải trọng cao và cánh tay đàn hồi là một trong những hướng phát triển có ứng dụng triển vọng, khối lượng cánh tay nhẹ nhưng có thể chịu tải trọng biến thiên lớn.

- Sự phát triển của robot với cơ cấu đàn hồi trong những năm gần đây đã tạo ra nhiều hướng tiếp cận mới trong thiết kế robot cánh tay. Không còn giới hạn trong các khớp kim loại truyền thống, các robot đàn hồi sử dụng vật liệu và cơ cấu có khả năng biến dạng linh hoạt, tăng khả năng tải trọng.

- Do đó, việc xây dựng một mô hình cánh tay robot 3DoF đàn hồi, kết hợp với hệ thống thu thập dữ liệu đầy đủ là hoàn toàn cần thiết để đáp ứng các yêu cầu thực tiễn và nghiên cứu chuyên sâu trong lĩnh vực robot học hiện đại.

### Mục tiêu nghiên cứu

- Nghiên cứu tổng quan về các cấu trúc đàn hồi trong robot học.
- Thiết kế cơ khí và lựa chọn vật liệu đàn hồi phù hợp.
- Thiết kế mạch điều khiển tích hợp PID
- Phát triển phần mềm thu thập và hiển thị dữ liệu (dựa trên Arduino)
- Thực nghiệm và phân tích số liệu: độ chính xác, sai số, khả năng lặp lại, phản hồi theo tải.

### Phương pháp nghiên cứu

- Nghiên cứu thiết kế và lắp ráp cơ khí.

+ Đảm bảo độ linh hoạt chuyên động trong không gian 3 chiều với ba khớp quay độc lập.

+ Đủ độ cứng vững và nhẹ.

+ Thiết kế đơn giản, dễ chế tạo, lắp ráp và bảo trì, phù hợp với điều kiện thực hiện.

- Nguyên cứu động lực học robot 3 DoF.

- Nguyên cứu ứng dụng các phần mềm điều khiển, và các thiết bị sử dụng.

- Nguyên cứu bộ điều khiển cho cánh tay robot.

### **Dự kiến kết quả đạt được**

- Hoàn thiện thiết kế và lắp ráp cơ khí cánh tay robot.

- Mô phỏng động lực học cánh tay robot.

- Điều khiển cánh tay robot.

- Phát triển hệ thống thu thập dữ liệu cảm biến, bao gồm đọc và lưu trữ tín hiệu encoder từ các khớp để phục vụ cho việc đánh giá hiệu suất và tinh chỉnh điều khiển.

### **Kết luận**

- Đề tài "*Cánh tay robot 3DOF đàn hồi và thu thập số liệu*" là một hướng tiếp cận có tính thực tiễn cao, phù hợp với xu thế phát triển hiện đại trong lĩnh vực robot học. Việc tích hợp cấu trúc đàn hồi không chỉ góp phần nâng cao độ linh hoạt và khả năng thích nghi của cánh tay robot, mà còn giúp tăng khả năng tải trọng.

- Thông qua quá trình thực hiện đề án, có cơ hội tiếp cận và ứng dụng kiến thức liên ngành gồm: thiết kế cơ khí, mô phỏng động học – động lực học, điều khiển động cơ, xử lý tín hiệu encoder, và thu thập dữ liệu cảm biến. Đây là nền tảng quan trọng giúp nâng cao năng lực chuyên môn, tư duy hệ thống, và khả năng triển khai kỹ thuật vào thực tế.

### **Cấu trúc của đề án tốt nghiệp**

- Chương 1: Tổng quan đề tài

- Chương 2: Xây dựng mô hình toán học

- Chương 3: Thiết kế và tính chọn hệ thống điều khiển

- Chương 4: Mô phỏng, thực nghiệm và đánh giá

## **Chương 1: TỔNG QUAN ĐỀ TÀI**

### **1.1. Tổng quan**

#### **1.1.1. Tổng quan về robot công nghiệp**

- Hiện nay đã có rất nhiều nghiên cứu cũng như ứng dụng của Robot trong cuộc sống và các ngành công nghiệp, nông nghiệp... Có thể nói Robot là tập hợp bao gồm hai phần là phần cứng và phần mềm. Phần cứng là toàn bộ các cơ cấu như là cơ cấu cơ khí, cơ cấu truyền động, vi mạch điều khiển, cảm biến... Phần mềm là chương trình điều khiển của Robot. Việc kết hợp phần cứng và các chương trình điều khiển đã mang lại khả năng ứng dụng rộng lớn cho Robot.

- Hiện nay robot ngày càng phát triển để đáp ứng cho nhu cầu thị trường về ngành công nghiệp trong các nhà máy. Robot không chỉ thay thế con người trong việc đáp ứng các nhu cầu của thao tác liên tục và lặp lại, mà còn thực hiện các tác vụ thông minh như một con người thực sự. Bởi sở hữu tốc độ nhanh và trí thông minh nhân tạo, robot có thể dễ dàng hoàn thành các nhiệm vụ trong thăm dò, làm việc tại những nơi nguy hiểm, cứu người trong tai nạn, hoặc lĩnh vực chuẩn đoán của thuốc.



Hình 1.1: Robot trong công nghiệp

#### **1.1.2. Khái niệm về robot công nghiệp**

- Robot công nghiệp có thể được định nghĩa theo một số tiêu chuẩn sau:

+ Theo tiêu chuẩn AFNOR của Pháp: Robot công nghiệp là một cơ cấu chuyển động tự động có thể lập trình, lặp lại các chương trình, tổng hợp các chương trình đặt ra trên các trục tọa độ, có khả năng định vị, định hướng, di chuyển các đối tượng vật chất như chi tiết, dụng cụ, gá lắp theo những hành trình thay đổi đã được chương trình hóa nhằm thực hiện các nhiệm vụ công nghệ khác nhau.

+ Theo tiêu chuẩn RIA của Mỹ (Robot Institute of America): Robot là một tay máy vạn năng có thể lập lại các chương trình, được thiết kế để di chuyển vật liệu, chi tiết, dụng cụ, hoặc các thiết bị chuyên dùng thông qua các chương trình chuyển động có thể thay đổi để hoàn thành các nhiệm vụ khác nhau.

+ Theo tiêu chuẩn TOCT 25686-85 của Nga: Robot công nghiệp là một máy tự động, được đặt cố định hoặc di động được, liên kết giữa một tay máy và một hệ thống điều khiển theo chương trình, có thể lập đi lập lại để hoàn thành các chức năng vận động và điều khiển trong quá trình sản xuất.

+ Do đó, robot công nghiệp có thể được hiểu là những thiết bị tự động linh hoạt, thực hiện các chức năng lao động công nghiệp của con người dưới một hệ thống điều khiển theo những chương trình đã được lập trình sẵn.

+ Với đặc điểm có thể lập trình lại được, robot công nghiệp là thiết bị tự động hóa và ngày càng trở thành bộ phận không thể thiếu được của các hệ thống sản xuất linh hoạt. Vì vậy, robot công nghiệp trở thành phương tiện hữu hiệu để tự động hóa, nâng cao năng suất lao động và giảm nhẹ cho con người những công việc nặng nhọc, độc hại dưới sự giám sát của con người.

### **1.1.3. Lịch sử phát triển**

- Thuật ngữ Robota đợc xuất hiện năm 1921 trong một tác phẩm văn học của nhà văn người Ba Lan – Karel Capek.

- Chính thuật ngữ “robota” này đã gợi ý cho con người phát triển Robot và một công ty ở Mỹ - AMF (Americal Machine and Foundry Company) quảng cáo mô phỏng một thiết bị mang dáng dấp và có một số chức năng như tay người điều khiển tự động thực hiện một số thao tác để sản xuất thiết bị có tên gọi Versatran.

Quá trình phát triển của Robot được tóm tắt như sau:

+ Từ những năm 1950 ở Mỹ đã xuất hiện sản phẩm có tên là Versatran của công ty AMF.

+ Ở Anh, người ta bắt đầu nghiên cứu và chế tạo Robot theo bản quyền của Mỹ từ những năm 1967.

+ Ở các nước Tây Âu khác như: Đức, Pháp, Ý, Thụy Điển thì bắt đầu chế tạo Robot từ những năm 1970.

- + Châu Á có Nhật Bản bắt đầu nghiên cứu ứng dụng Robot từ năm 1968.
- Tuy Mỹ là nước đầu tiên phát minh ra Robot nhưng Nhật Bản là nước sản xuất ra nhiều nhất.

Bảng 1.1: Số lượng Robot các nước công nghiệp phát triển

Nước sản xuất	1990	1998	2003	2006
Nhật	60.118	67.000	92.340	132.543
Mỹ	4.327	11.100	23.654	48.321
Đức	5.854	8.600	15.000	27.543
Anh	510	1.500	3.240	6.754
Pháp	1.488	2.000	4.563	9.123
Hàn Quốc	1.000	3.000	6.721	12.432

#### **1.1.4. Phân loại robot công nghiệp**

- Ngày nay, robot công nghiệp đã phát triển rất phong phú và đa dạng, vì vậy phân loại chúng không đơn giản. Có rất nhiều quan điểm khác nhau và mỗi quan điểm lại phục vụ một mục đích riêng. Dưới đây là các cách phân loại chính.

- Theo chủng loại, mức độ điều khiển, và nhận biết máy đã được sản xuất trên thế giới có thể phân loại các IR thành các thế hệ sau thông tin của tay máy - người.

+ Thế hệ 1: Thế hệ có kiểu điều khiển theo chu kỳ dạng chương trình cứng không có khả năng nhận biết thông tin.

+ Thế hệ 2: Thế hệ có kiểu điều khiển theo chu kỳ dạng chương trình mềm bước đầu đã có khả năng nhận biết thông tin.

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

- Phân loại theo kết cấu: theo kết cấu của tay máy người ta phân thành robot kiểu tọa độ Đề các, kiểu tọa độ trụ, kiểu tọa độ cầu, kiểu tọa độ góc, robot kiểu SCARA như đã trình bày ở trên.

- Phân loại theo ứng dụng: dựa vào ứng dụng của robot trong sản xuất có robot sơn, robot hàn, robot lắp ráp, robot chuyên phối...

- Phân loại theo cách thức và đặc trưng của phương pháp điều khiển: có robot điều khiển hở (mạch điều khiển không có các quan hệ phản hồi), robot điều khiển kín (hay điều khiển servo): sử dụng cảm biến, mạch phản hồi để tăng độ chính xác và mức độ linh hoạt khi điều khiển.

- Ngoài ra còn có thể có các cách phân loại khác tùy theo quan điểm và mục đích nghiên cứu.

### **1.1.5. Ứng dụng robot công nghiệp**

- Robot công nghiệp được áp dụng trong công nghiệp dưới góc độ thay thế con người, nhờ vậy các dây chuyền sản xuất được tăng năng suất và hiệu quả.

+ Ứng dụng trong công nghiệp: Robot được sử dụng trong công nghệ đúc, hàn, cắt, sơn, phun phủ, tháo lắp, vận chuyển...



Hình 1.2: Robot hàn trong công nghiệp

+ Ứng dụng trong Y học: Robot được sử dụng trong lĩnh vực y tế như nội soi, phẫu thuật, ...



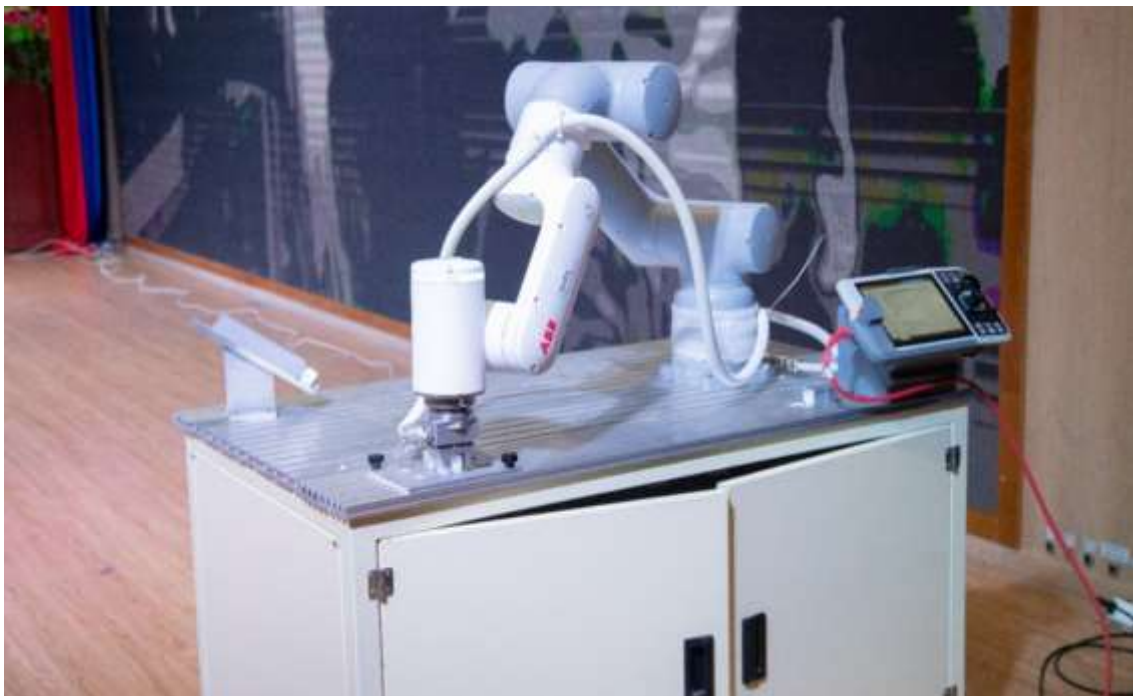
Hình 1.3: Robot phẫu thuật

+ Ứng dụng trong quân sự: Robot di chuyển hoặc tự động hoặc điều khiển từ xa, được thiết kế cho các ứng dụng quân sự từ vận chuyển, trinh sát, cứu hộ, tấn công...



Hình 1.4: Robot trinh thám

+ Trong giáo dục: Giúp phát triển kỹ năng, tư duy và sáng tạo trong thời đại công nghệ 4.0.



Hình 1.5: Robot ABB dùng trong giảng dạy Trường Đại học bách khoa Đà Nẵng

#### **1.1.6. Bậc tự do của Robot**

- Bậc tự do là số khả năng chuyển động của một cơ cấu (chuyển động quay hoặc tịnh tiến). Để dịch chuyển được một vật thể trong không gian, cơ cấu chấp hành của

robot phải đạt được một số bậc tự do. Nói chung cơ hệ của robot là một cơ cấu hở, do đó bậc tự do của nó có thể tính theo công thức:

$$W = 6n - \sum_{i=1}^5 ip_i \quad (1.1)$$

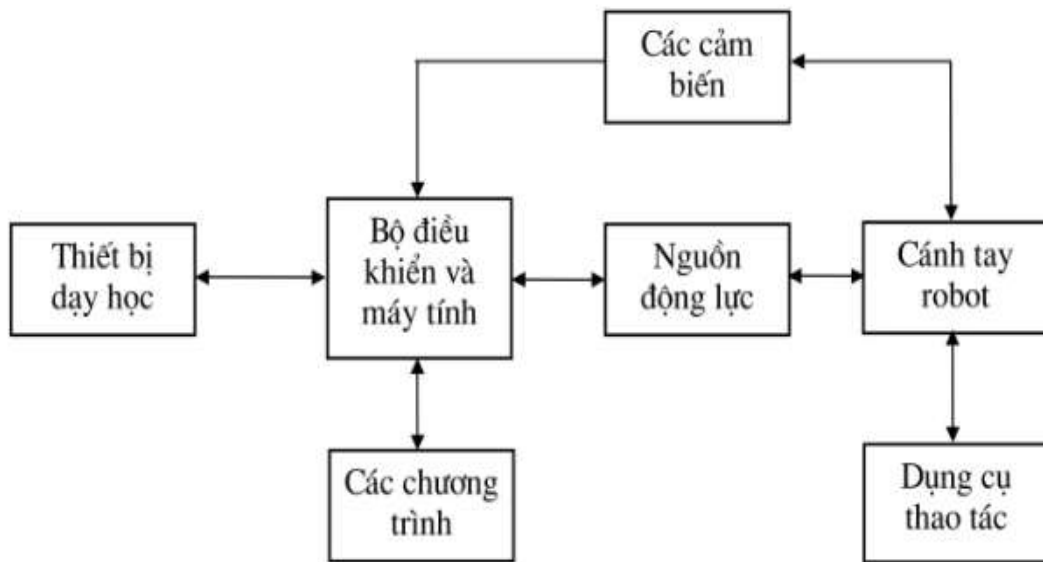
- Ở đây: n - Số khâu động;

Pi- Số khớp loại i (i = 1,2,...,5 : Số bậc tự do bị hạn chế)

- Đối với các cơ cấu có các khâu được nối với nhau bằng khớp quay hoặc tịnh tiến (khớp động loại 5) thì số bậc tự do bằng với số khâu động. Đối với cơ cấu hở, số bậc tự do bằng tổng số bậc tự do của các khớp động.

### 1.1.7. Các thành phần chính của Robot công nghiệp

#### a) Các thành phần chính



Hình 1.6: Các thành phần chính của hệ thống Robot

#### b) Kết cấu của tay máy

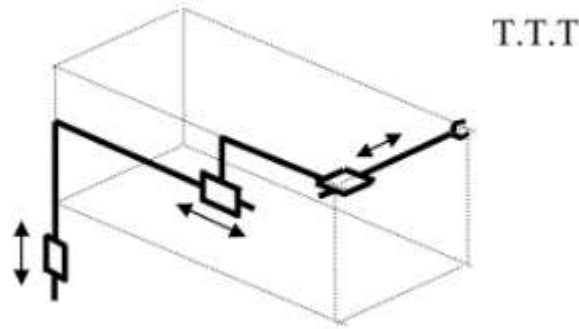
- Gồm 2 chuyển động cơ bản:

+ Chuyển động tịnh tiến (kí hiệu T)

+ Chuyển động quay (kí hiệu R)

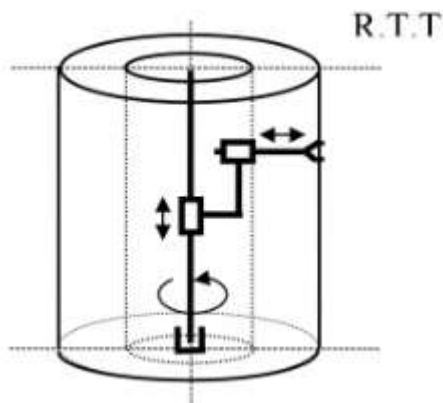
- Các kết cấu thường gặp của Robot là: Robot kiểu tọa độ đề các, tọa độ trụ, tọa độ cầu, robot kiểu SCARA, hệ tọa độ góc (phồng sinh),...

+ Robot kiểu tọa độ Đề các: là tay máy có 3 chuyển động cơ bản tịnh tiến theo phương của các trục hệ tọa độ góc (cấu hình T.T.T). Trường công tác có dạng khối chữ nhật. Do kết cấu đơn giản, loại tay máy này có độ cứng vững cao, độ chính xác cơ khí dễ đảm bảo vì vậy nó thường dùng để vận chuyển phôi liệu, lắp ráp, hàn trong mặt phẳng.



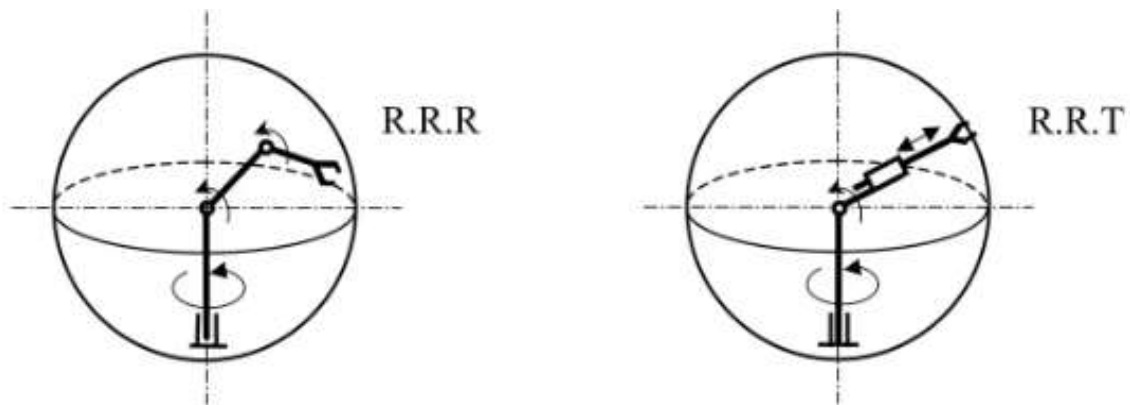
Hình 1.7: Robot kiểu tọa độ Đề các

+ Robot kiểu tọa độ trụ: vùng làm việc của Robot có dạng hình trụ rỗng. Thường khớp thứ nhất chuyển động quay.



Hình 1.8: Robot kiểu tọa độ trụ

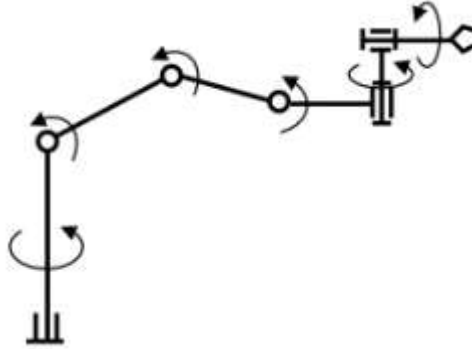
+ Robot kiểu tọa độ cầu: vùng làm việc của Robot có dạng hình cầu. Thường độ cứng vững của loại Robot này thấp hơn so với hai loại trên.



Hình 1.9: Robot kiểu tọa độ cầu

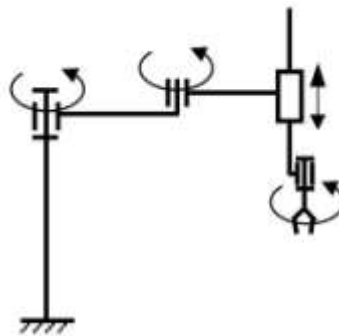
+ Robot kiểu tọa độ góc (Hệ tọa độ phỏng sinh): Đây là kiểu Robot được dùng nhiều hơn cả. Ba chuyển động đầu tiên là các chuyển động quay, trục quay thứ nhất vuông góc với hai trục kia. Các chuyển động định hướng khác cũng là các chuyển động quay. Vùng làm việc của tay máy này gần giống một phần khối cầu. Tất cả các khâu đều nằm trong mặt phẳng thẳng đứng nên các tính toán cơ bản là bài toán phẳng,

ưu điểm nổi bật của các loại robot hoạt động theo hệ tọa độ góc là gọn nhẹ, tức là có vùng làm việc tương đối lớn so với kích cỡ của bản thân Robot, độ linh hoạt cao.



Hình 1.10: Robot hoạt động theo hệ tọa độ góc

+ Robot kiểu SCARA: Robot SCARA ra đời vào năm 1979 tại trường đại học Yamanashi (Nhật Bản) là một kiểu robot mới nhằm đáp ứng sự đa dạng của các quá trình sản xuất. Tên gọi SCARA là viết tắt của "Selective Compliant Articulated Robot Arm": Tay máy mềm dẻo tùy ý. Loại robot này thường dùng trong công việc lắp ráp nên SCARA đôi khi được giải thích là từ viết tắt của "Selective Compliance Assembly Robot Arm". Ba khớp đầu tiên của kiểu Robot này có cấu hình R.R.T, các trục khớp đều theo phương thẳng đứng.



Hình 1.11: Robot kiểu SCARA

## 1.2 Tìm hiểu về tính đàn hồi trong cánh tay Robot

### 1.2.1. Đặt vấn đề

- Robot thường dùng trong công nghiệp có khối lượng rất lớn nhưng tải trọng thì nhỏ, tỉ lệ tải trọng mang được và khối lượng Robot là rất thấp. Ví dụ:

+ ABB IRB 1600: Nặng 250kg nhưng tải trọng tối đa chỉ khoảng 36kg.

+ KUKA KR 100-2 PA: nặng 1240 kg nhưng tải trọng tối đa chỉ khoảng 100kg

- Do đó việc nghiên cứu sử dụng cánh tay nhẹ nhưng chịu được tải trọng lớn là rất cần thiết.

- Trong những nghiên cứu cơ học đã chứng minh được sử dụng cánh tay đàn hồi có thể giảm khối lượng tối đa lên đến 80%.

### **1.2.2. Tay máy có tính đến độ đàn hồi**

Đầu tiên, ta sẽ xem xét một số khái niệm về biến đổi đặc tính cơ - lý của vật liệu làm tay máy trong thực tế. Vì đa số tay máy được chế tạo từ kim loại, nên chúng có khả năng bị biến dạng đàn hồi, uốn... khi có sự chuyển động. Ta sẽ lần lượt xem xét như sau:

#### **1.2.2.1. Ứng suất**

- Ứng suất là đại lượng biểu thị nội lực trong vật thể do tác động của các nguyên nhân bên ngoài. Nó có thể gây ra biến dạng kéo, nén, hoặc xoắn trong vật liệu.

$$\sigma = \frac{F}{A} \quad (1.2)$$

- Trong đó

+  $\sigma$ : Ứng suất (N/m<sup>2</sup> hoặc Pa)

+ F: Lực tác dụng (N)

+ A: Diện tích mặt cắt ngang (m<sup>2</sup>)

- Phân loại: Ứng suất được chia thành các loại chính dựa trên hướng và cách lực tác dụng

+ Ứng suất pháp tuyến (Normal Stress): xảy ra khi lực tác dụng vuông góc với bề mặt. Có 2 dạng:

Ứng suất kéo: Lực kéo dãn vật liệu (ví dụ: dây cáp chịu kéo).

Ứng suất nén: Lực nén ép vật liệu (ví dụ: cột bê tông chịu nén).

+ Ứng suất cắt: Xảy ra khi lực tác dụng song song với bề mặt, gây trượt các lớp vật liệu (Ví dụ: Lực cắt trong bu-lông hoặc trong vật liệu chịu xoắn.)

+ Ứng suất thể tích (Xảy ra khi vật liệu chịu áp suất đồng đều từ mọi hướng (như trong chất lỏng hoặc vật liệu chịu áp suất thủy tĩnh).

#### **1.2.2.2. Biến dạng**

- Sự dịch chuyển tương đối giữa các chất điểm, các phần tử của vật rắn dưới tác dụng của ngoại lực, nhiệt độ hoặc của một nguyên nhân nào đó dẫn đến sự thay đổi về hình dạng và kích thước của nó được gọi là biến dạng.

- Vật liệu dưới tác dụng của ngoại lực đều thay đổi hình dạng và kích thước mà không mất đi sự liên kết bền chặt của nó. Khả năng cho phép thực hiện một quá trình biến dạng dẻo được coi là một đặc tính quan trọng của kim loại.

$$\varepsilon = \frac{\Delta l}{l_0} = \frac{l_1 - l_0}{l_0} \quad (1.3)$$

Trong đó:  $\varepsilon$  là đường cong ứng suất- biến dạng

- Cơ tính của vật liệu được xác định bằng các phương pháp thử khác nhau, tùy thuộc vào bản chất tải (độ lớn, tốc độ...) và môi trường đặt tải (nhiệt độ, thời gian...). Thực tế khi hoạt động, chi tiết chịu tải trọng phức tạp với ứng suất ba chiều. Tuy nhiên,

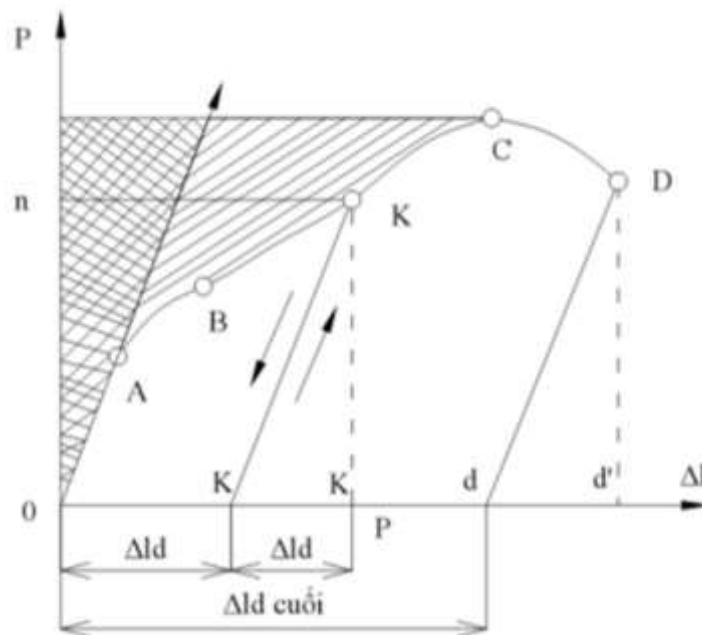
phương pháp thử đơn giản và thông dụng lại phản ánh được các đặc trưng cơ tính của vật liệu là thử kéo. Trong đó, mẫu thử được kéo một chiều, đúng tâm với tải trọng tăng dần cho tới khi bị đứt.

- Để thử, người ta tác động lên mẫu thử có tiết diện  $F_0$ , chiều dài lo một lực kéo  $P$ , sau đó lập quan hệ giữa lực kéo  $P$  và độ dẫn dài  $\Delta l = l - l_0$ ). Ta được biểu đồ kéo có dạng  $H_1$ , chúng gồm ba giai đoạn nối tiếp nhau như sau:

- Ban đầu, khi tải trọng tăng, độ dẫn dài tăng theo quy luật đường thẳng và chạm (đoạn OA). Khi bỏ tải trọng, kích thước mẫu lại trở về vị trí ban đầu. Giai đoạn này gọi là biến dạng đàn hồi.

- Khi tải trọng vượt quá giá trị nhất định (điểm A), biến dạng tăng nhanh, nếu bỏ tải trọng, kích thước mẫu dài hơn trị số ban đầu lo. Giai đoạn này gọi là biến dạng dẻo đi kèm biến dạng đàn hồi (ví dụ: điểm K trên biểu đồ).

- Khi tải trọng đã giá trị lớn nhất (điểm C), tiến vùng nào đó của mẫu xuất hiện biến dạng tạm trung, tiết diện mẫu giảm tại độ biến xuất hiện, tiếp theo vật nứt thành đoạn và hủy. Đó là giai đoạn phá hủy.

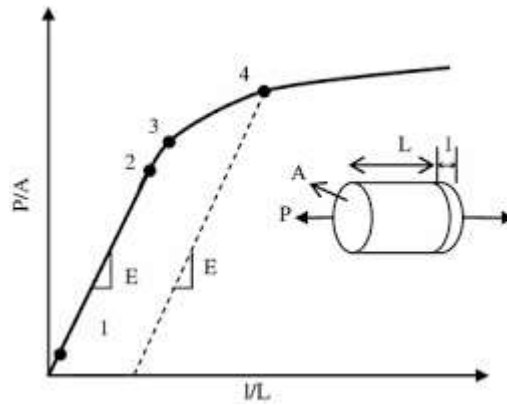


Hình 1.12: Vật liệu bị biến dạng

### 1.2.2.3. Uốn

- Trạng thái dẻo trong vật thể biến dạng chủ yếu được gây nên bởi trọng tải uốn.

- Nếu tải trọng tăng lên khiến ứng suất trong mẫu vượt quá giới hạn đàn hồi thì vật liệu bắt đầu quá trình biến dạng uốn. Trong vùng này, nếu bỏ tải trọng mà mẫu không phục hồi được hình dạng như ban đầu và vẫn bị uốn và đường cong ứng suất biến dạng được thể hiện bằng mức độ biến dạng dư. Trong kỹ thuật:



Hình 1.13: Vật liệu biến dạng uốn

#### 1.2.2.4. Đàn hồi

- Đàn hồi là khả năng của vật liệu trở lại hình dạng ban đầu sau khi lực tác dụng được gỡ bỏ (nếu chưa vượt quá giới hạn đàn hồi).

- Mối liên hệ giữa ứng suất và biến dạng (strain) trong giới hạn đàn hồi được mô tả bằng định luật Hooke:

$$\sigma = E \cdot \varepsilon \quad (1.4)$$

- Trong đó

+ E: Modul đàn hồi, đơn vị Pa

+  $\varepsilon$  : Biến dạng (không có đơn vị, là tỉ số)

- Với:  $\varepsilon = \frac{\Delta L}{L_0}$

+ Trong đó:  $\Delta L$  : Độ dãn hoặc co dài

$L_0$  : Chiều dài ban đầu

+ Tỉ số Poisson: Khi vật kéo dãn theo chiều dọc, nó co lại theo chiều ngang. Tỉ lệ giữa biến dạng ngang và biến dạng dọc

$$\nu = \frac{\varepsilon_{ngang}}{\varepsilon_{dọc}} \quad (1.5)$$

- Phân Loại:

+ Biến dạng kéo/nén: Xảy ra khi vật liệu bị kéo dãn hoặc nén dọc theo một trục.

+ Biến dạng cắt: Xảy ra khi các lớp vật liệu trượt tương đối với nhau, được mô tả bởi mô-đun cắt (G).

+ Biến dạng thể tích: Xảy ra khi vật liệu chịu áp suất đồng đều, được mô tả bởi mô-đun thể tích (K).

- Ví dụ về vật liệu và đặc tính đàn hồi:

Bảng 1.2: Đặc tính đàn hồi của vật liệu

Vật liệu	E	$\nu$
----------	---	-------

Thép	200	0.3
Nhôm	70	0.3
Cao su	0.1	0.49 (gần như không nén được)

### **1.2.3. Thuận lợi và khó khăn của Robot cánh tay đàn hồi**

#### **1.2.3.1. Thuận lợi**

- Tăng độ linh hoạt cho Robot, tăng khả năng chịu lực.
- Tiết kiệm vật liệu chế tạo.
- Khai thác hết khả năng cơ học của Robot.
- Đảm bảo độ bền cơ học và tăng tuổi thọ Robot.
- Giảm chi phí bảo trì, bảo dưỡng, thay thế.
- Nâng cao độ chính xác điều khiển khi Robot chuyển động theo những quỹ đạo phức tạp và biến đổi liên tục, có nhiều điểm kỳ dị.
- Giảm tối ưu hóa tính kinh tế khi sản xuất và ứng dụng Robot hàng loạt.

#### **1.2.3.2. Khó khăn**

- Tuy nhiên, một tay thao tác linh hoạt đòi hỏi nhiều các thuật toán điều khiển phức tạp cũng như các mô hình động chính xác hơn để có được độ ổn định và hiệu suất tốt. Nên việc điều khiển chính xác là rất khó.

### **1.2.4. Một số nghiên cứu về tay máy có tính đến độ đàn hồi**

#### **1.2.4.1. Trong nước**

- Việc điều khiển cánh tay được đề xuất bởi TS. Nguyễn Hoàng Mai và Bùi Tuấn Việt Linh (2017) [2] đã đưa ra mô hình toán học xây dựng phương pháp bù sai số do biến dạng đàn hồi của cấu trúc dưới ảnh hưởng của ngoại lực dựa trên công thức kết hợp giữa phương trình Lagrange-Euler với phương pháp giả định, sau đó thiết kế bộ điều khiển PID mở để điều khiển hệ thống. Xây dựng phương trình động lực học của tay máy công nghiệp, cụ thể là tay máy hai bậc tự do có tính đến đàn hồi, tổng hợp bộ điều khiển PID mở cho tay máy hai bậc tự do.

- Điều khiển trượt cho tay máy có tính đến độ đàn hồi được đề xuất bởi TS. Nguyễn Hoàng Mai và Phạm Minh Tuấn [3] đã nghiên cứu về xây dựng phương trình động lực học của tay máy công nghiệp, cụ thể là tay máy hai bậc tự do có tính đến đàn hồi.

- Tác giả Dương Xuân Biên, Chu Anh Mỹ, Phan Bùi Khởi (2017) [4] đã đưa ra mô hình động lực học tổng quát cho hệ tay máy có hai khâu đàn hồi. Hệ phương trình động lực học được xây dựng trên cơ sở phương pháp phân tử hữu hạn và cách tiếp cận Lagrange trong đó có tính đến yếu tố ma sát đàn hồi, mô men quán tính của các động

cơ, tài trọng giúp đưa mô hình động lực học sát với thực tế. Kết quả mô phỏng thể hiện được bản chất ứng xử động lực học của tay máy có khâu đàn hồi.

- Nghiên cứu, tính toán và điều khiển mô hình được nghiên cứu bởi tác giả Trần Đức Thuận, Trần Xuân Kiên, Bùi Hồng Huệ (2016) [5] đã nghiên cứu tay máy thường liên kết với nhau bằng các khớp nối và trong quá trình hoạt động xuất hiện sự biến dạng của các khâu cùng như sự biến dạng xoắn của các trục. Hiện tượng này xảy ra liên kết đàn hồi và tạo ra các tham số bất định trong mô hình chuyển động.

#### **1.2.4.2. Ngoài nước**

- Việc điều khiển cánh tay robot mềm được đề xuất bởi Cannon và Schmitz vào năm 1984 [6], họ đã đưa ra mô hình toán học và thực hiện một số thí nghiệm để giải quyết vấn đề điều khiển của robot một khâu. Vị trí của khâu chấp hành cuối được điều khiển bằng việc đo lường vị trí và sử dụng các đại lượng đo như 1 đơn vị chuẩn cho việc ứng dụng điều khiển momen quay đến điểm khác của cánh tay robot. Vì vậy, nó được trích dẫn trong những công trình của Harashima và Ueshiba, Wang và Vidyasagar, Sangveraphunsiri và nhiều những công trình khác. Trong tất cả những công trình trên cánh tay robot mềm được mô hình hóa như một khâu mà sai lệch được biểu diễn bằng một chuỗi hàm riêng hữu hạn.

Cannon và Schmith (1984) trong các nghiên cứu kể trên chỉ ra những hệ thống robot mềm này có dao động tắt dần xung quanh một vị trí cuối cùng mong muốn và đưa ra một luật phản hồi trong đó bao gồm các biến đàn hồi. Hệ thống được bù cuối cùng di chuyển nhanh về phía vị trí cuối cùng nhưng các chuyển động trung gian không phải là mịn, các nghiên cứu được tiến hành trong điều kiện các khâu có biến dạng và mômen dẫn động lớn. Để đạt được chuyển động mịn hơn, đã có đề xuất cần sử dụng các luật điều khiển phản hồi khác nhau và luôn tham chiếu tới một quỹ đạo đã chọn trong khi cố gắng để giữ cho robot dao động ít nhất

- Balas (1982) và Book (1989) [7] giới thiệu những cách tiếp cận khác nhau cho bài toán bám quỹ đạo của robot mềm. Bài toán cơ bản được rất nhiều tác giả nghiên cứu là vị trí của tâm bàn kẹp trong cơ cấu tay máy mềm có một liên kết, đây là cơ sở để mở rộng kết quả nghiên cứu cho các cơ hệ phức tạp hơn về sau này.

- Một kỹ thuật thay thế để giải quyết việc theo dõi quỹ đạo điểm cuối dựa trên giải pháp của bài toán động lực học ngược đối với các robot mềm (Lopez-Linares và cộng sự, 1991). Động lực học ngược tính toán mômen dẫn động tại khớp phải được áp dụng để có được một chuyển động theo chỉ định.

- Giải pháp đầu tiên cho bài toán động lực học ngược đối với một robot mềm một khâu đã được đưa ra bởi Bayo (1987) [8]. Kỹ thuật được đề xuất là sử dụng phương

pháp phần tử hữu hạn để rời rạc hoá biến dạng đàn hồi và phương pháp miền tần số để giải quyết bài toán số. Bayo và cộng sự (1989) mở rộng các bài toán động lực học ngược ở các hệ thống đa khâu, phẳng bằng một quy trình lặp đi lặp lại. Gần đây (Ledesma và cộng sự, 1994; Ledesma và Bayo, 1993, 1994) phương pháp đã được áp dụng cho các hệ thống mềm ba chiều, giữ các đặc điểm miền tần số và lặp đi lặp lại.

### **1.2.5. Ứng dụng thực tiễn của Robot cánh tay đàn hồi**

- Những Robot có dạng cánh tay dài phải đàn hồi để có thể dễ dàng làm việc hơn.

Ví dụ như: cần cẩu trong xây dựng nhà cao tầng:

+ Kết cấu thân, cần linh hoạt thuận tiện cho việc lắp đặt, tháo dỡ, vận chuyển.

+ Với kết cấu các module nhỏ gọn cho nên việc tháo lắp cần cẩu rất đơn giản, nhanh gọn và an toàn.

+ Trong xây dựng người ta thường sử dụng cần trục tháp có tải trọng nặng từ 3-10 tấn, tầm với 25m, chiều cao nâng đến 50m



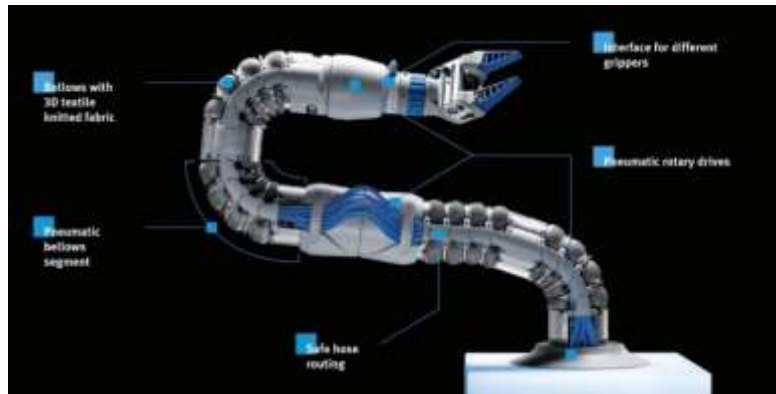
Hình 1.14: Cầu trục tháp GM/GRT66 – 57 H

- Robot di động với cánh tay mềm của Festo (BionicSoftArm):

+ Festo, một công ty công nghệ Đức, đã phát triển BionicSoftArm, một cánh tay robot mềm sử dụng các khoang khí nén để tạo ra chuyển động linh hoạt. Cánh tay này có thể được gắn trên một nền tảng di động để thực hiện các nhiệm vụ trong môi trường công nghiệp hoặc dịch vụ.

+ Cho dù là các chuyển động tự do và linh hoạt hay các quy trình xác định - nhờ cấu trúc mô-đun, robot nhẹ khí nén có thể được sử dụng cho nhiều ứng dụng. Kết hợp

với nhiều bộ kẹp thích ứng khác nhau, nó có thể gấp và xử lý nhiều loại đồ vật và hình dạng khác nhau. Đồng thời, nó vẫn có tính linh hoạt cao và không gây nguy hiểm cho người sử dụng ngay cả trong trường hợp va chạm.



Hình 1.15: BionicSoftArm với bảy cơ cấu chấp hành khí nén và nhiều bậc tự do

### **1.2.6. Ý nghĩa khoa học và thực tiễn**

- Ý nghĩa khoa học và thực tiễn:

+ Khoa học:

Góp phần phát triển lĩnh vực robot mềm và robot đàn hồi, thông qua việc nghiên cứu, mô phỏng và triển khai cấu trúc đàn hồi trong cánh tay robot 3 bậc tự do.

Cung cấp cơ sở dữ liệu và mô hình động lực học phục vụ cho các nghiên cứu sâu hơn về điều khiển phi tuyến, điều khiển thích nghi hoặc điều khiển dự báo trong các hệ cơ cấu linh hoạt.

Đóng vai trò như một mô hình thí nghiệm ứng dụng trong đào tạo và nghiên cứu khoa học về cơ điện tử, tự động hóa, điều khiển thông minh và cảm biến tích hợp.

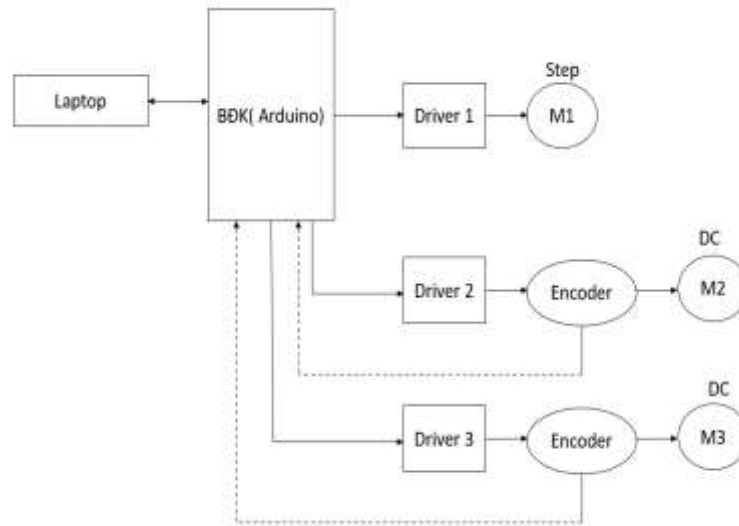
+ Thực tiễn: Hệ thống có thể làm nền tảng cho việc phát triển các robot có cánh tay dài như cần cẩu nâng hạ nhà cao tầng, robot di động, robot chữa cháy....

### **1.3 Kết luận chương 1**

- Chương 1 đã trình bày tổng quát về Robot công nghiệp từ lịch sử ra đời đến các giai đoạn phát triển, những ưu – nhược điểm trong các lĩnh vực nghiên cứu và ứng dụng của Robot. Tìm hiểu cấu trúc chung của Robot và các phương pháp điều khiển chuyển động của Robot để Robot ngày càng linh hoạt, thông minh đáp ứng nhu cầu sử dụng của con người trong mọi lĩnh vực: sản xuất, công nghiệp, dịch vụ... Thông qua đó còn tìm hiểu thêm về tính đàn hồi trong Robot, những ưu- nhược điểm và các công trình nghiên cứu trong và ngoài nước về cánh tay Robot có xét đến tính đàn hồi.

## Chương 2: XÂY DỰNG MÔ HÌNH TOÁN HỌC

### 2.1 Cấu trúc



Hình 2.1: Sơ đồ khối của hệ thống

\* Thành phần hệ thống:

- Laptop:

+ Kết nối với Arduino

+ Dùng để gửi lệnh điều khiển, giám sát, cập nhật /thu thập dữ liệu.

- BDK:

+ Điều khiển toàn bộ hệ thống.

+ Gửi tín hiệu điều khiển đến các driver motor.

+ Nhận phản hồi từ encoder để thực hiện điều khiển

- M1 – Stepper Motor (Động cơ bước):

+ Driver 1 nhận tín hiệu từ Arduino để điều khiển M1.

+ M1 là động cơ bước dùng cho đế xoay của robot, yêu cầu độ chính xác cao.

- M2 & M3 – DC Motor với Encoder:

+ Driver 2 và Driver 3 điều khiển hai động cơ DC M2 và M3.

+ Encoder gắn liền với động cơ để phản hồi vị trí/góc quay về cho Arduino.

+ Arduino dùng thông tin này để điều chỉnh tốc độ/góc quay theo PID.

### 2.2 Nguyên lý hoạt động

- Cánh tay robot 3 bậc tự do (3DoF) đàn hồi được thiết kế để thực hiện các chuyển động xoay và gập tại ba khớp cơ bản, tương ứng với ba trục tự do trong không gian. Các

khâu nối giữa các khớp được tích hợp cấu trúc đàn hồi nhằm tăng độ linh hoạt và khả năng thích nghi khi robot thao tác với môi trường hoặc tiếp xúc với vật thể.

- Truyền động:

+ Cánh tay robot sử dụng các động cơ điện (stepper motor cho trục đế, DC motor cho các khớp còn lại), và cơ cấu dẫn động để tạo ra chuyển động tại các khớp.

+ Các khâu đàn hồi có khả năng biến dạng trong phạm vi cho phép, hấp thụ dao động hoặc tải bất ngờ, giúp bảo vệ cơ cấu và tăng độ linh hoạt.

- Đo lường và phản hồi:

+ Mỗi khớp được gắn encoder để ghi nhận vị trí góc quay theo thời gian thực.

+ Dữ liệu từ encoder được truyền về vi điều khiển để so sánh với giá trị đặt trước (setpoint).

- Điều khiển:

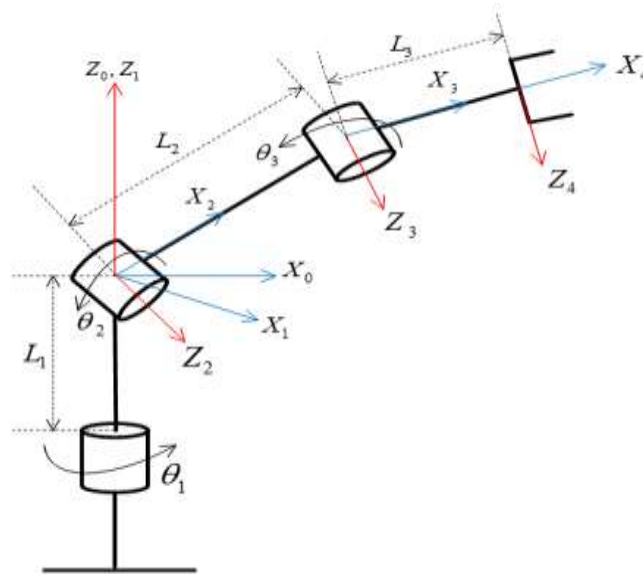
+ Vi điều khiển thực hiện thuật toán điều khiển (PID) để hiệu chỉnh sai lệch vị trí và điều khiển động cơ sao cho cánh tay đạt được chuyển động chính xác theo yêu cầu.

+ Trong quá trình hoạt động, hệ thống có khả năng tự cân bằng dao động phát sinh do tính đàn hồi của các khâu.

+ Thu thập dữ liệu: hệ thống thu thập dữ liệu từ các, lưu trữ hoặc truyền về máy tính để phục vụ cho việc giám sát, phân tích và đánh giá hiệu suất.

- Tác vụ thực hiện: cánh tay có thể thực hiện các tác vụ đơn giản như gấp – di chuyển có khối lượng khoảng 0.5kg.

### 2.3. Phương trình động lực học



Hình 2.2: Mô hình 2D cánh tay robot 3 bậc tự do

### 2.3.1. Tính toán động học thuận

- Mục đích của bài toán động học thuận là xác định vị trí của khâu tác động cuối của tay máy khi biết các biến khớp của tay máy.

Bảng 2.1: Lập bảng D-H (Denavit – Hartenberg).

i	$a_{i-1}$	$\alpha_{i-1}$	$d_i$	$\theta_i$
1	0	0	0	$\theta_1$
2	0	90	0	$\theta_2$
3	$L_2$	0	0	$\theta_3$
4	$L_3$	0	0	0

- Trong đó: i là thứ tự khớp của robot;  $a_{i-1}$  là khoảng cách của trục  $Z_i$  và  $Z_{i-1}$ ;  $\alpha_{i-1}$  là độ xoắn (được xác định bằng độ lệch của 2 trục  $Z_i$  và  $Z_{i-1}$ );  $d_i$  là khoảng cách của  $X_i$  và  $X_{i-1}$  cùng vuông góc với trục  $Z_i$ ;  $\theta_i$  là góc khớp (góc giữa 2 trục  $X_i$  và  $X_{i-1}$ ).

-  $L_2, L_3 = L_{cd}$  (cố định) +  $\Delta L$  (có thể tăng hoặc giảm)

- Công thức tổng quát ma trận biến đổi đồng nhất từ hệ trục {i} so với {i-1} là:

$${}^{i-1}T_i = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

- Trong đó:  $c\theta_i = \cos(\theta_i), s\theta_i = \sin(\theta_i), s\alpha_i = \sin(\alpha_i), c\alpha_i = \cos(\alpha_i)$

- Ta có ma trận chuyển đổi giữa 2 hệ trục tọa độ liên tiếp như sau:

$$\text{Từ } \{0\} \text{ sang } \{1\}: {}^0T_1 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

$$\text{Từ } \{1\} \text{ sang } \{2\}: {}^1T_2 = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

$$\text{Từ } \{2\} \text{ sang } \{3\}: {}^2T_3 = \begin{bmatrix} c_3 & -s_3 & 0 & L_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

$$\text{Từ } \{3\} \text{ sang } \{4\}: \quad {}_4^3T = \begin{bmatrix} 1 & 0 & 0 & L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

$$\text{Từ } \{0\} \text{ sang } \{3\}: \quad {}_3^0T = \begin{bmatrix} c_1 c_{23} & -c_1 s_{23} & s_1 & L_2 c_1 c_2 \\ s_1 c_{23} & -s_1 s_{23} & -c_1 & L_2 s_1 c_2 \\ s_{23} & c_{23} & 0 & L_2 s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

- Ma trận chuyển đổi tổng quát từ hệ thế 0 sang hệ thứ 4 như sau:

$${}_4^0T = {}_1^0T {}_2^1T {}_3^2T {}_4^3T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

$${}_4^0T = \begin{bmatrix} c_1 c_{23} & -c_1 s_{23} & s_1 & c_1 (L_3 c_{23} + L_2 c_2) \\ s_1 c_{23} & -s_1 s_{23} & -c_1 & s_1 (L_3 c_{23} + L_2 c_2) \\ s_{23} & c_{23} & 0 & L_3 s_{23} + L_2 s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

- Tọa độ vị trí cơ cấu cuối:

$$\begin{cases} P_x = c_1 (L_3 c_{23} + L_2 c_2) \\ P_y = s_1 (L_3 c_{23} + L_2 c_2) \\ P_z = L_3 s_{23} + L_2 s_2 \end{cases} \quad (2.9)$$

### 2.3.2. Tính toán động học nghịch

- Mục đích của bài toán động học nghịch là tìm các biến khớp của tay máy khi biết vị trí khâu tác động cuối của tay máy. Có 2 phương pháp để giải bài toán động học nghịch là phương pháp hình học và phương pháp đại số. Ở đây nhóm đã áp dụng phương pháp đại số để giải. Các bước thực hiện được trình bày dưới đây:

- Từ phương trình (2.7), biến đổi tương đương ta được phương trình sau:

$${}_1^0T^{-1} \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}_2^1T {}_3^2T {}_4^3T \quad (2.10)$$

$$\begin{bmatrix} r_{11}c_1 + r_{21}s_1 & r_{12}c_1 + r_{22}s_1 & r_{13}c_1 + r_{23}s_1 & P_x c_1 + P_y s_1 \\ r_{21}c_1 - r_{11}s_1 & r_{22}c_1 - r_{12}s_1 & r_{23}c_1 - r_{13}s_1 & P_y c_1 - P_x s_1 \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{23} & -s_{23} & 0 & L_3 c_{23} + L_2 c_2 \\ 0 & 0 & -1 & 0 \\ s_{23} & c_{23} & 0 & L_3 s_{23} + L_2 s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.11)$$

- *Tính theta1:*

Cân bằng hai ma trận từ phương trình (2.11), phần tử hàng 2 cột 4:

$$P_y c_1 - P_x s_1 = 0 \quad (2.12)$$

Tương đương, ta có:

$$\frac{s_1}{c_1} = \frac{P_y}{P_x} = \frac{-P_y}{-P_x} \quad (2.13)$$

Ta thu được 2 nghiệm tính góc  $\theta_1$ :

$$\begin{cases} \theta_1 = a \tan 2(P_y, P_x) \\ \theta_1 = a \tan 2(-P_y, -P_x) \end{cases} \quad (2.14)$$

- *Tính theta3:*

Cân bằng hai ma trận từ phương trình (2.11), phần tử hàng 1 cột 4 và phần tử hàng 3 cột 4, ta được:

$$\begin{cases} L_3 c_{23} + L_3 c_2 = P_x c_1 + P_y s_1 \\ L_3 s_{23} + L_2 s_2 = P_z \end{cases} \quad (2.15)$$

Bình phương 2 vế hệ phương trình (2.15), rồi cộng vế theo vế, ta được:

$$L_2^2 + L_3^2 + 2L_2 L_3 c_3 = (P_x c_1 + P_y s_1)^2 + P_z^2 \quad (2.16)$$

Thực hiện chuyển vế ta rút được  $c_3$ , như sau:

$$c_3 = \frac{(P_x c_1 + P_y s_1)^2 + P_z^2 - (L_2^2 + L_3^2)}{2L_2 L_3} \quad (2.17)$$

$$s_3 = \pm \sqrt{1 - c_3^2}$$

Ta thu được 2 nghiệm cho góc Theta3:

$$\begin{cases} \theta_3 = a \tan 2(s_3, c_3) \\ \theta_3 = a \tan 2(-s_3, c_3) \end{cases} \quad (2.18)$$

- *Tính theta2:*

Áp dụng công thức lượng giác, ta biến đổi phương trình (2.15) như sau:

$$\begin{cases} (L_3 c_3 + L_2) c_2 + (-L_3 s_3) s_2 = P_x c_1 + P_y s_1 \\ (L_3 s_3) c_2 + (L_3 c_3 + L_2) s_2 = P_z \end{cases} \quad (2.19)$$

Đặt:  $a_{11} = L_3c_3 + L_2$ ,  $a_{12} = -L_3s_3$ ,  $a_{21} = L_3s_3$ ,  $a_{22} = L_3c_3 + L_2$ ,  $a = P_x c_1 + P_y s_1$ ,  
 $b = P_z$

Phương trình (2.19) tương đương:

$$\begin{cases} a_{11}c_2 + a_{12}s_2 = a \\ a_{21}c_2 + a_{22}s_2 = b \end{cases} \quad (2.20)$$

Áp dụng phương pháp Cramer giải phương trình (2.20):

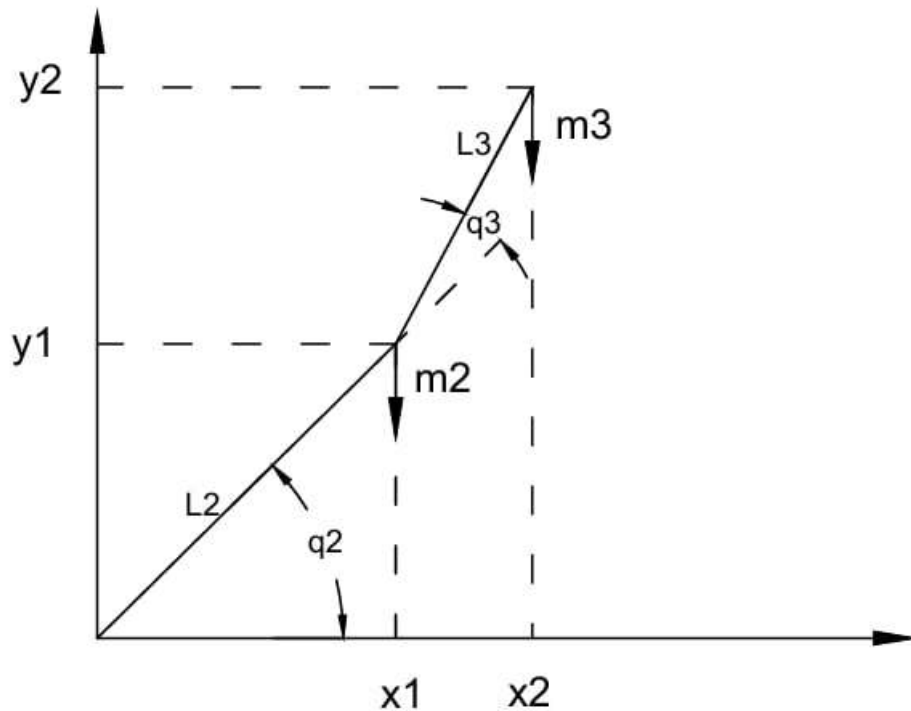
$$c_2 = \frac{aa_{22} - ba_{12}}{a_{11}a_{22} - a_{21}a_{12}} \quad (2.21)$$

$$s_2 = \frac{a_{11}b - a_{21}a}{a_{11}a_{22} - a_{21}a_{12}} \quad (2.22)$$

Ta tính được 1 nghiệm cho góc  $\theta_2$ :

$$\theta_2 = a \tan 2(s_2, c_2) \quad (2.23)$$

### 2.3.3. Tính toán động lực học



Hình 2.3: Hệ trục tọa độ của tay máy

$q_2, q_3$ : Góc quay của từng khâu

$l_2, l_3$ : Chiều dài của từng khâu

$m_2, m_3$ : Khối lượng toàn bộ của mỗi khâu

a) Vị trí

$$\text{Khâu 1: } \begin{cases} x_1 = l_2 \cos q_2 \\ y_1 = l_2 \sin q_2 \end{cases} \quad (2.24)$$

$$\text{Khâu 2: } \begin{cases} x_2 = l_2 \cos q_2 + l_3 \cos(q_2 + q_3) \\ y_2 = l_2 \sin q_2 + l_3 \sin(q_2 + q_3) \end{cases} \quad (2.25)$$

b) Vận tốc

$$\text{Vận tốc của Robot: } v^2 = \dot{x}^2 + \dot{y}^2 \quad (2.26)$$

$$\text{Với: } \begin{cases} v_{x_1} = \dot{x}_1 = -l_2 \sin q_2 \dot{q}_2 \\ v_{y_1} = \dot{y}_1 = l_2 \cos q_2 \dot{q}_2 \end{cases} \quad (2.27)$$

$$\begin{cases} v_{x_2} = \dot{x}_2 = -l_2 \sin q_2 \dot{q}_2 - l_3 (\dot{q}_2 + \dot{q}_3) \cdot \sin q_2 (q_2 + q_3) \\ v_{y_2} = \dot{y}_2 = l_2 \cos q_2 \dot{q}_2 - l_3 \cos q_2 (q_2 + q_3) (\dot{q}_2 + \dot{q}_3) \end{cases} \quad (2.28)$$

$$\text{Vận tốc khâu 1: } v_1^2 = \dot{x}_1^2 + \dot{y}_1^2 \quad (2.29)$$

$$\Rightarrow v_1^2 = (l_2 \dot{q}_2 \cos q_2)^2 + (l_2 \dot{q}_2 \sin q_2)^2$$

$$\Leftrightarrow v_1^2 = l_2^2 \dot{q}_2^2 \cos^2 q_2 + l_2^2 \dot{q}_2^2 \sin^2 q_2 \quad (2.30)$$

$$\Leftrightarrow v_1^2 = l_2^2 \dot{q}_2^2 (\cos^2 q_2 + \sin^2 q_2)$$

$$\Leftrightarrow v_1^2 = l_2^2 \dot{q}_2^2$$

$$\text{Vận tốc khâu 2: } v_2^2 = \dot{x}_2^2 + \dot{y}_2^2 \quad (2.31)$$

$$\Rightarrow v_2^2 = [-l_2 \sin q_2 \dot{q}_2 - l_3 (\dot{q}_2 + \dot{q}_3) \cdot \sin q_2 (q_2 + q_3)]^2 + [-l_2 \cos q_2 \dot{q}_2 - l_3 \cos q_2 (q_2 + q_3) (\dot{q}_2 + \dot{q}_3)]^2$$

$$\Leftrightarrow v_2^2 = l_2^2 \dot{q}_2^2 \cos^2 q_2 + l_3^2 \dot{q}_2^2 \sin^2 q_2 + l_3^2 (\dot{q}_2 + \dot{q}_3)^2 \cos^2 (q_2 + q_3) + l_3^2 (\dot{q}_2 + \dot{q}_3)^2 \sin^2 (q_2 + q_3) +$$

$$[2l_2 \dot{q}_2 \dot{l}_3 (\dot{q}_2 + \dot{q}_3) \cos q_2 \cos (q_2 + q_3)] + [2l_2 \dot{q}_2 \dot{l}_3 (\dot{q}_2 + \dot{q}_3) \sin q_2 \sin (q_2 + q_3)]$$

$$\Leftrightarrow v_2^2 = l_2^2 \dot{q}_2^2 + l_3^2 (\dot{q}_2 + \dot{q}_3)^2 + 2l_2 \dot{q}_2 \dot{l}_3 (\dot{q}_2 + \dot{q}_3) [\cos(q_2 - q_2 + q_3)]$$

$$\Leftrightarrow v_2^2 = l_2^2 \dot{q}_2^2 + l_3^2 (\dot{q}_2 + \dot{q}_3)^2 + 2l_2 \dot{q}_2 \dot{l}_3 (\dot{q}_2 + \dot{q}_3) \cos q_3 \quad (2.32)$$

$$\Leftrightarrow v_2^2 = l_2^2 \dot{q}_2^2 + l_3^2 (\dot{q}_2 + \dot{q}_3)^2 + 2l_2 \dot{l}_3 (\dot{q}_2^2 + \dot{q}_3 \dot{q}_2) \cos q_3$$

c) Động năng

$$K = \frac{1}{2} m v^2 \quad (2.33)$$

$$\text{Khâu 1: } K_1 = \frac{1}{2} m_2 v_1^2 = \frac{1}{2} m_2 l_2^2 \dot{q}_2^2 \quad (2.34)$$

$$\text{Khâu 2: } K_2 = \frac{1}{2} m_3 v_2^2 = \frac{1}{2} m_3 [l_2^2 \dot{q}_2^2 + l_3^2 (\dot{q}_2 + \dot{q}_3)^2 + 2l_2 \dot{l}_3 (\dot{q}_2^2 + \dot{q}_3 \dot{q}_2) \cos q_3] \quad (2.35)$$

$$\text{Tổng động năng: } \sum K = K_1 + K_2 \quad (2.36)$$

$$K = \frac{1}{2}m_2l_2^2\dot{q}_2^2 + \frac{1}{2}m_3[l_2^2\dot{q}_2^2 + l_3^2(\dot{q}_2 + \dot{q}_3)^2 + 2l_2l_3(\dot{q}_2^2 + \dot{q}_3\dot{q}_2)\cos q_3]$$

$$\Rightarrow K = \frac{1}{2}l_2^2\dot{q}_2^2(m_2 + m_3) + \frac{1}{2}m_3l_3[l_3(\dot{q}_2 + \dot{q}_3)^2 + 2l_2(\dot{q}_2^2 + \dot{q}_3\dot{q}_2)\cos q_3]$$
(2.37)

d) Thế năng

$$P = mgy \tag{2.38}$$

$$\text{Khâu 1 } P_1 = m_2gy_1 = m_2gl_2 \sin q_2 \tag{2.39}$$

$$\text{Khâu 2: } P_2 = m_3gy_2 = m_3g[l_2 \sin q_2 + l_3 \sin(q_2 + q_3)] \tag{2.40}$$

$$\text{Tổng thế năng: } \sum P = P_1 + P_2 \tag{2.41}$$

$$P = m_2gl_2 \sin q_2 + m_3g[l_2 \sin q_2 + l_3 \sin(q_2 + q_3)] \tag{2.42}$$

$$\Rightarrow P = gl_2 \sin q_2(m_2 + m_3) + l_3m_3g \sin(q_2 + q_3)$$

- Hàm lagrange:

$$L = \sum K - \sum P \tag{2.43}$$

$$L = \frac{1}{2}l_2^2\dot{q}_2^2(m_2 + m_3) + \frac{1}{2}m_3l_3[l_3(\dot{q}_2 + \dot{q}_3)^2 + 2l_2(\dot{q}_2^2 + \dot{q}_3\dot{q}_2)\cos q_3] -$$

$$[gl_2 \sin q_2(m_2 + m_3) + l_3m_3g \sin(q_2 + q_3)] \tag{2.44}$$

$$L = \frac{1}{2}l_2^2\dot{q}_2^2(m_2 + m_3) + \frac{1}{2}m_3l_3^2(\dot{q}_2 + \dot{q}_3)^2 + m_3l_2l_3(\dot{q}_2^2 + \dot{q}_3\dot{q}_2)\cos q_3 -$$

$$(m_2 + m_3)gl_2 \sin q_2 - m_3gl_3 \sin(q_2 + q_3)$$

e) Phương trình động lực học tay máy

$$F = \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} \tag{2.45}$$

Ta có

$$\frac{\partial L}{\partial q_2} = -(m_2 + m_3)gl_2 \sin q_2 - m_3g \sin(q_2 + q_3)$$

$$\frac{\partial L}{\partial \dot{q}_2} = (m_2 + m_3)l_2^2\dot{q}_2 + m_3l_3^2(\dot{q}_2 + \dot{q}_3) + m_3l_2l_3(2\dot{q}_2 + \dot{q}_3)\cos q_3 \tag{2.46}$$

$$\frac{\partial L}{\partial q_3} = -m_3l_2l_3(\dot{q}_2^2 + \dot{q}_2\dot{q}_3)\sin q_3 - m_3l_3g \cos(q_2 + q_3)$$

$$\frac{\partial L}{\partial \dot{q}_3} = m_3l_3^2(\dot{q}_2 + \dot{q}_3) + m_3l_2l_3\dot{q}_2 \cos q_3$$

$$\frac{d}{dt} * \frac{\partial L}{\partial \dot{q}_2} = (m_2 + m_3)l_2^2\ddot{q}_2 + m_3l_3^2(\ddot{q}_2 + \ddot{q}_3) + m_3l_2l_3(2\ddot{q}_2 + \ddot{q}_3)\cos q_3 - m_3l_2l_3(2\dot{q}_2\dot{q}_3 + \dot{q}_3^2)\sin q_3 \tag{2.47}$$

$$\frac{d}{dt} * \frac{\partial L}{\partial \dot{q}_3} = m_3l_3^2(\ddot{q}_2 + \ddot{q}_3) + m_3l_2l_3\ddot{q}_2 \cos q_3 - m_3l_2l_3\dot{q}_2\dot{q}_3 \sin q_3$$

$$F_1 = \frac{d}{dt} * \frac{\partial L}{\partial \dot{q}_2} - \frac{\partial L}{\partial q_2}$$

$$\Rightarrow F_1 = [(m_2 + m_3)l_2^2 \ddot{q}_2 + m_3 l_3^2 (\ddot{q}_2 + \ddot{q}_3) + m_3 l_2 l_3 (2\dot{q}_2 + \dot{q}_3) \cos q_3 - m_3 l_2 l_3 (2\dot{q}_2 \dot{q}_3 + \dot{q}_3^2) \sin q_3] + (m_2 + m_3)l_2 g \cos q_3 + m_3 l_3 g \cos(q_2 + q_3) \quad (2.48)$$

$$\Rightarrow F_1 = [(m_2 + m_3)l_2^2 + m_3 l_3^2 + 2m_3 l_2 l_3 \cos q_3] \ddot{q}_2 + [m_3 l_3^2 + m_3 l_2 l_3 \cos q_3] \ddot{q}_3 - m_3 l_2 l_3 (2\dot{q}_2 \dot{q}_3 + \dot{q}_3^2) \sin q_3 + (m_2 + m_3)l_2 g \cos q_2 + m_3 l_3 g \cos(q_2 + q_3)$$

$$F_2 = \frac{d}{dt} * \frac{\partial L}{\partial \dot{q}_3} - \frac{\partial L}{\partial q_3}$$

$$\Rightarrow F_2 = m_3 l_3^2 (\ddot{q}_2 + \ddot{q}_3) + m_3 l_2 l_3 \ddot{q}_2 \cos q_3 - m_3 l_2 l_3 \dot{q}_2 \dot{q}_3 \sin q_3 + m_3 l_2 l_3 (\dot{q}_2^2 + \dot{q}_2 \dot{q}_3) \sin q_3 + m_3 l_3 g \cos(q_2 + q_3) \quad (2.49)$$

$$\Rightarrow F_2 = [m_3 l_3^2 + m_3 l_2 l_3 \cos q_3] \ddot{q}_2 + m_3 l_3^2 \ddot{q}_3 + m_3 l_2 l_3 \dot{q}_2^2 \sin q_3 + m_3 l_3 g \cos(q_2 + q_3)$$

$$F = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = M(q) * \ddot{q} + N(q)\dot{q} + G(q) \quad (2.50)$$

- Trong đó:

M: Ma trận cấp (nxn) đặc trưng cho quán tính

N: Ma trận cấp (nxn) đặc trưng cho lực li tâm và lực Coriolit

G: Ma trận cấp (nx1) lực trọng trường

- Với:

$$\tau_1 = [(m_2 + m_3)l_2^2 + m_3 l_3^2 + 2m_3 l_2 l_3 \cos q_3] \ddot{q}_2 + [m_3 l_3^2 + m_3 l_2 l_3 \cos q_3] \ddot{q}_3 - m_3 l_2 l_3 (2\dot{q}_2 \dot{q}_3 + \dot{q}_3^2) \sin q_3 + (m_2 + m_3)l_2 g \cos q_2 + m_3 l_3 g \cos(q_2 + q_3) \quad (2.51)$$

$$\tau_2 = [m_3 l_3^2 + m_3 l_2 l_3 \cos q_3] \ddot{q}_2 + m_3 l_3^2 \ddot{q}_3 + m_3 l_2 l_3 \dot{q}_2^2 \sin q_3 + m_3 l_3 g \cos(q_2 + q_3) \quad (2.52)$$

- Ta có:

$$F = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} (m_2 + m_3)l_2^2 + m_3 l_3^2 + 2m_3 l_2 l_3 \cos q_3 & m_3 l_3^2 + m_3 l_2 l_3 \cos q_3 \\ m_3 l_3^2 + m_3 l_2 l_3 \cos q_3 & m_3 l_3^2 \end{bmatrix} \begin{bmatrix} \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + \begin{bmatrix} -m_3 l_2 l_3 (2\dot{q}_2 \dot{q}_3 + \dot{q}_3^2) \sin q_3 \\ m_3 l_2 l_3 \dot{q}_2^2 \sin q_3 \end{bmatrix} + \begin{bmatrix} (m_2 + m_3)g l_2 \cos q_2 + m_3 g l_3 \cos(q_2 + q_3) \\ m_3 g l_3 \cos(q_2 + q_3) \end{bmatrix} \quad (2.53)$$

$$M(q) = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}$$

$$m_{11} = (m_2 + m_3)l_2^2 + m_3 l_3^2 + 2m_3 l_2 l_3 \cos q_3 \quad (2.54)$$

$$m_{12} = m_{32} = m_3 l_3^2 + m_3 l_2 l_3 \cos q_3 = m_3 (l_3^2 + l_2 l_3 \cos q_3)$$

$$m_{22} = m_3 l_3^2$$

$$N(q)\dot{q} = \begin{bmatrix} n_{11} & n_{12} \\ n_{21} & n_{22} \end{bmatrix}$$

$$n_{11} = -m_3 l_2 l_3 (2\dot{q}_2 \dot{q}_3 + \dot{q}_3^2) \sin q_3 \quad (2.55)$$

$$n_{12} = n_{21} = m_3 l_2 l_3 \dot{q}_2^2 \sin q_3$$

$$n_{22} = 0$$

$$G(q) = \begin{bmatrix} g(q_2) \\ g(q_3) \end{bmatrix}$$

$$g(q_2) = (m_2 + m_3)gl_2 \cos q_2 + m_3 gl_3 \cos(q_2 + q_3) \quad (2.56)$$

$$g(q_3) = m_3 gl_3 \cos(q_2 + q_3)$$

### Hệ phương trình trạng thái:

Đặt biến trạng thái cho khớp 1 và khớp 2 như sau:

$$\begin{cases} X_1 = \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix} = \begin{bmatrix} q_1 \\ \dot{q}_1 \end{bmatrix} \\ X_2 = \begin{bmatrix} x_{21} \\ x_{22} \end{bmatrix} = \begin{bmatrix} q_2 \\ \dot{q}_2 \end{bmatrix} \end{cases} \quad (2.57)$$

Và tín hiệu vào U:  $U = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$  (2.58)

Hệ phương trình vi phân trạng thái của khớp 1 và khớp 2 được viết:

$$\text{Khớp 1: } \begin{cases} \dot{x}_{11} = \dot{x}_{12} \\ \dot{x}_{12} = a_1(X) + \sum_{j=1}^2 b_{1j} u_j \end{cases} \quad (2.59)$$

$$\text{Khớp 2: } \begin{cases} \dot{x}_{21} = \dot{x}_{22} \\ \dot{x}_{22} = a_2(X) + \sum_{j=1}^2 b_{2j} u_j \end{cases} \quad (2.60)$$

Khi đó phương trình được viết lại:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} n_{11} & n_{12} \\ n_{21} & n_{22} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} \quad (2.61)$$

Đặt:

$$x_1 = q_2; x_{12} = \dot{q}_2; \dot{x}_{12} = \ddot{q}_2; x_2 = q_3; x_{22} = \dot{q}_3; \dot{x}_{22} = \ddot{q}_3; u_1 = \tau_1; u_2 = \tau_2$$

$$\Rightarrow \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} \dot{x}_{12} \\ \dot{x}_{22} \end{bmatrix} + \begin{bmatrix} n_{11} & n_{12} \\ n_{21} & n_{22} \end{bmatrix} \begin{bmatrix} x_{12} \\ x_{22} \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} \quad (2.62)$$

Vậy:

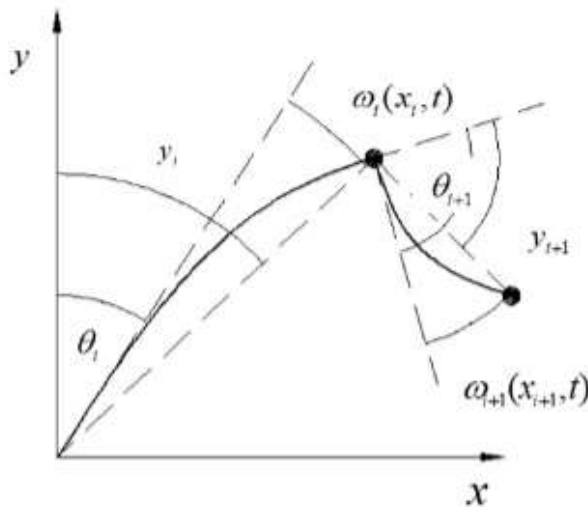
$$\begin{aligned} \dot{x}_{12} &= \frac{h_{12}u_2 - h_{22}u_1 + h_{22}n_{11}x_{12} + h_{22}n_{12}x_{22} + h_{22}g_1 - h_{12}n_{21}x_{12} - h_{12}g_2}{h_{12}h_{21} - h_{22}h_{11}} \\ \dot{x}_{22} &= \frac{h_{11}u_2 - h_{21}u_1 + h_{21}n_{11}x_{12} + h_{21}n_{12}x_{22} + h_{21}g_1 - h_{11}n_{21}x_{12} - h_{11}g_2}{h_{22}h_{11} - h_{12}h_{21}} \end{aligned} \quad (2.63)$$

Như vậy mô hình tay máy robot là một hệ một đầu vào một đầu ra, được mô tả bằng hai hệ nhỏ, mỗi hệ tương ứng với từng khớp 1 và 2, được đặc trưng bởi hai hệ phương trình vi phân trạng thái.

#### 2.3.4. Xây dựng mô hình toán học cho có xét đến đàn hồi

- Khi các cánh tay robot có kích thước lớn hoạt động (như máy nâng hạ nhiều bậc tự do) thì độ đàn hồi vật liệu sẽ xuất hiện và gây ra những hiệu ứng phi tuyến mạnh, dẫn đến khó điều khiển chính xác vị trí nếu dùng các phương trình mô tả cánh tay máy thông thường như phương trình Lagrange II cho tay máy cứng. Do đó việc xây dựng một phương pháp tính toán bù sai số điểm cuối do biến dạng đàn hồi của cấu trúc dưới ảnh hưởng của ngoại lực là rất cần thiết.

- Hình trình bày sơ đồ hệ thống robot đàn hồi và sai lệch vị trí khâu cuối do ảnh hưởng của biến dạng đàn hồi. Hai cánh tay đàn hồi được nối với nhau bằng các khớp quay được giả định là trong cùng mặt phẳng chuyển động.



Hình 2.4: Hệ robot đàn hồi

#### \*Xác Định Biến Dạng Đàn Hồi Bằng Phương Pháp Chế Độ Giả Định

- Phương pháp chế độ giả định là một phương pháp phổ biến vì nó hạn chế số bậc tự do của đối tượng từ vô hạn đến hữu hạn với một số lượng tương đối nhỏ các tọa độ tổng quát. Các hình thức giả định thường được sử dụng như là dùng các hàm cơ sở để xấp xỉ một chuyển vị liên tục. Các biến dạng đàn hồi của cánh tay được đưa ra bởi sự kết hợp tuyến tính của các hàm cơ sở và tọa độ tổng quát.

$$v_i(x_{i,t}) = \sum_1^{N_{ei}} \varnothing_{ij}(x_i) \delta_{ij}(t) \quad (2.64)$$

Với:

-  $\delta$  là tọa độ tổng quát của vector biến dạng đàn hồi.

-  $\varnothing_i(x_{ij})$  là hàm cơ sở thứ j của khâu i.

- Góc quay thực tế của từng khâu xấp xỉ với góc  $y_i$  và được mô tả bởi (với  $\theta_i$  góc quay trên lý thuyết của khâu i):

$$y_i = \theta_i \pm v_i(x_{i,t})$$

- Sử dụng giả thiết (I), phương trình động lực của liên kết tại một điểm không gian tùy ý  $x_i$  dọc theo liên kết tại một thời điểm t có thể được viết bằng lý thuyết Euler-Beam trong tài liệu [9] như sau:

$$(EI)_i \frac{\partial^4 v_i(x_{i,t})}{\partial x_i^4} + \rho \frac{\partial^2 v_i(x_{i,t})}{\partial t^2} = 0 \quad (2.65)$$

Với  $\rho$  là mật độ khối lượng của khâu thứ i.

Thay  $v_i(x_{i,t})$  từ phương trình (2.64) vào (2.65) ta được:

$$\frac{(EI)_i}{\rho_i \varnothing_{ij}(x_i)} \frac{d^4 \varnothing_{ij}(x)}{dx_i^4} = - \frac{1}{\delta_{ij}(t)} \frac{d^4 \delta_{ij}(t)}{dt^2} = \omega_{ij}^2 \quad (2.66)$$

Với  $\omega_{ij}^2$  là hằng số, từ phương trình (2.66) ta suy ra:

$$\frac{d^2 (EI)_i (d^2 \varnothing_{ij} / dx_i^2)}{dx_i^2} - \omega_{ij}^2 \rho_i \varnothing_{ij} = 0 \quad (2.67)$$

Và:

$$\frac{d^2 \delta_{ij}(t)}{dt^2} + \omega_{ij}^2 \delta_{ij}(t) = 0 \quad (2.68)$$

Từ phương trình (2.68) ta suy ra:

$$\delta_{ij}(t) = \exp(\omega_{ij}(t)) \quad (2.69)$$

Và từ phương trình (2.67) ta có:

$$\varnothing_{ij}(x_i) = N_{ij} [\cos(\beta_{ij} x_i) - \cosh(\beta_{ij} x_i) + \gamma_{ij} \sinh(\beta_{ij} x_i) - \cosh(\beta_{ij} x_i)] \quad (2.70)$$

Với  $\gamma_{ij}$  được cho bởi tài liệu [6]:

$$\gamma_{ij} = \frac{\sin \beta_{ij} - \sinh \beta_{ij} + M_{Ei} \beta_{ij} (\cos \beta_{ij} - \cosh \beta_{ij}) - M_{DEi} \beta_{ij}^2 (\sin \beta_{ij} + \sinh \beta_{ij})}{\cos \beta_{ij} + \cosh \beta_{ij} - M_{Ei} \beta_{ij} (\sin \beta_{ij} - \sinh \beta_{ij}) - M_{DEi} \beta_{ij}^2 (\cos \beta_{ij} - \cosh \beta_{ij})} \quad (2.71)$$

Và  $\beta_{ij}$  được tính bởi phương trình sau, cho bởi tài liệu [9]:

$$1 + \cosh \beta_{ij} l_i \cos \beta_{ij} l_i - M_{Ei} \beta_{ij} (\sinh \beta_{ij} l_i - \cosh \beta_{ij} l_i \sin \beta_{ij} l_i) - I_{Ei} \beta_{ij}^3 (\sinh \beta_{ij} l_i + \cosh \beta_{ij} l_i \sin \beta_{ij} l_i) + M_{Ei} I_{Ei} \beta_{ij}^4 (1 - \cosh \beta_{ij} l_i \sin \beta_{ij} l_i) + M_{DEi}^2 \beta_{ij}^4 (1 - \cosh \beta_{ij} l_i \sin \beta_{ij} l_i) - 2M_{DEi}^2 \beta_{ij}^4 \sinh \beta_{ij} l_i = 0 \quad (2.72)$$

- Trong đó:  $M_{Ei}$ ,  $I_{Ei}$  lần lượt là khối lượng hiệu dụng, mô men quán tính của điểm cuối khâu thứ  $i$ ,  $M_{DEi}$  là khối lượng các thành phần thuộc các khâu sau.  $\beta_{ij}$  là tham số điều kiện biên của khâu  $i$ ,  $N_{ij}$  là hằng số được cho bởi:

$$N_{ij} = \int_0^{l_i} [\varnothing_{ij}(x_i)]^2 dx_i = m_i \quad (2.73)$$

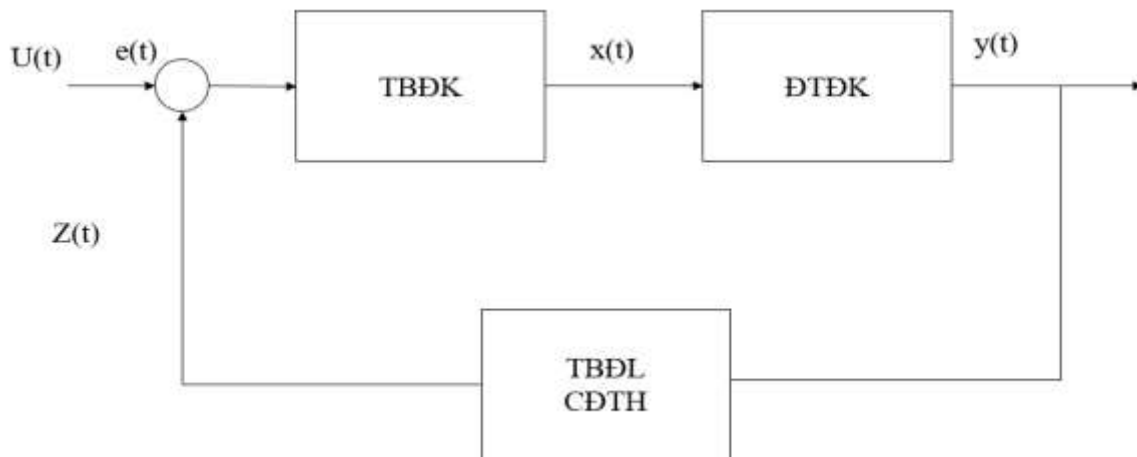
- Vậy hằng số  $\omega_{ij}^2$  sẽ được xác định bởi công thức sau, tài liệu [9]:

$$\beta_{ij}^4 = \frac{\omega_{ij}^2 \rho_i}{(EI)_i} \quad (2.74)$$

- Từ các phương trình (2.68) đến (2.74) ta sẽ xác định được hàm biến dạng đàn hồi trong từng khâu của robot.

## 2.4. Tổng quan về bộ điều khiển PID

### 2.4.1. Cấu trúc chung của BDK



Hình 2.5: Cấu trúc hệ thống điều khiển

- Trong đó:

+ ĐTĐT: Đối tượng điều khiển

+ TBĐK: Thiết bị điều khiển

+ TBĐL – CDTH: Thiết bị đo lường và chuyển đổi tín hiệu.

+  $U(t)$ : Là tín hiệu vào của hệ thống – còn gọi là tín hiệu đặt hay đại lượng chủ đạo để xác định điểm làm việc của hệ thống.

+  $y(t)$ : Tín hiệu đầu ra của hệ thống. Đây chính là đại lượng được điều chỉnh.

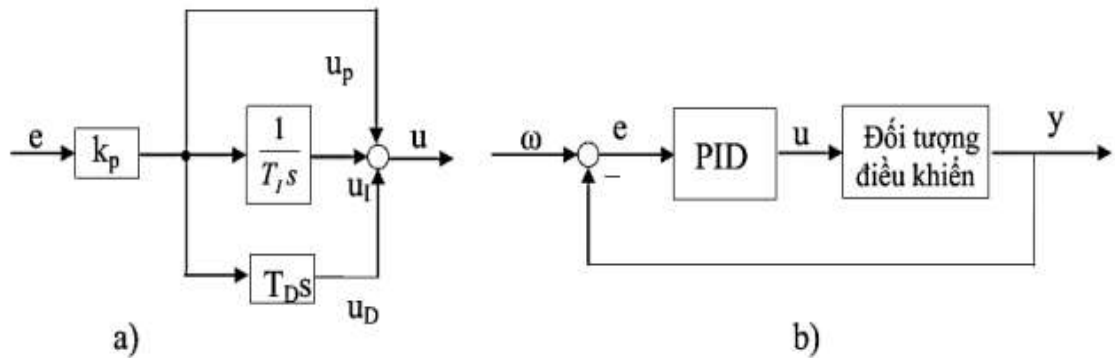
+  $x(t)$ : Là tín hiệu điều khiển tác động lên đối tượng.

+  $Z(t)$ : Là tín hiệu phản hồi.

- Thiết bị điều khiển là thành phần quan trọng nhất duy trì chế độ làm việc cho cả hệ thống điều khiển.

### 2.4.2. Phương pháp xác định tham số PID

- Tên gọi PID là chữ viết tắt của ba thành phần cơ bản có trong bộ điều khiển gồm khâu khuếch đại (P), khâu tích phân (I), và khâu vi phân (D).

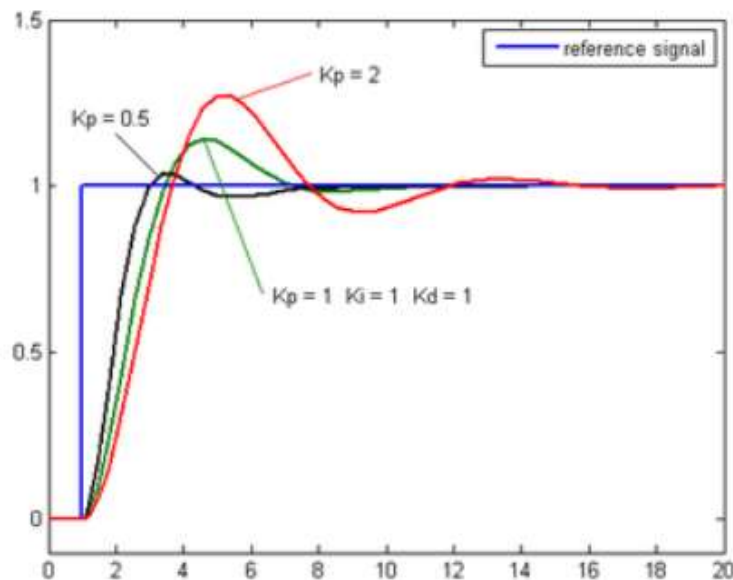


Hình 2.6: Bộ điều khiển PID

- Bộ điều khiển PID được sử dụng khá rộng rãi để điều khiển đối tượng SISO theo nguyên lý hồi tiếp. Bộ PID được sử dụng rộng rãi nhờ tính đơn giản của nó cả về cấu trúc và nguyên lý làm việc. Bộ PID có nhiệm vụ đưa sai lệch  $e(t)$  của hệ thống về 0 sao cho quá trình quá độ thỏa mãn các yêu cầu cơ bản về chất lượng.

- Các thành phần của bộ PID

#### a) Khâu tỉ lệ



Hình 2.7: Đồ thị đáp ứng theo thời gian với  $K_p$  thay đổi

- Khâu tỷ lệ (đôi khi còn được gọi là độ lợi) làm thay đổi giá trị đầu ra, tỷ lệ với giá trị sai số hiện tại. Đáp ứng tỷ lệ có thể được điều chỉnh bằng cách nhân sai số đó với một hằng số  $K_p$ , được gọi là độ lợi tỷ lệ.

- Khâu tỷ lệ được cho bởi:

$$P_{out} = K_p \cdot e(t) \quad (2.75)$$

Trong đó:

$P_{out}$ : thứ số tỷ lệ của đầu ra

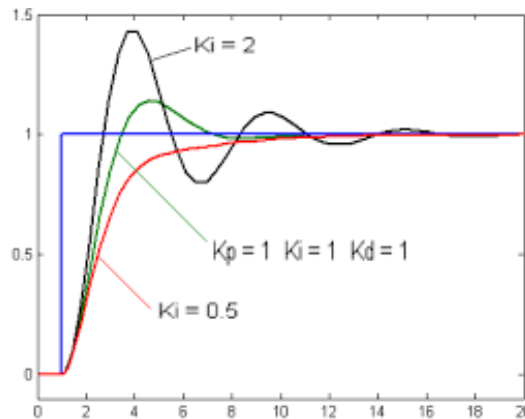
$K_p$ : Độ lợi tỷ lệ, thông số điều chỉnh

$e$ : sai số

$t$ : thời gian hay thời gian tức thời (hiện tại)

- Nếu độ lợi của khâu tỷ lệ quá cao, hệ thống sẽ không ổn định. Ngược lại, độ lợi nhỏ và làm cho bộ điều khiển kém nhạy, hoặc đáp ứng chậm, không đáp ứng với các nhiễu của hệ thống.

b) *Khâu tích phân*



Hình 2.8: Đồ thị đáp ứng theo thời gian với  $K_i$  thay đổi

- Phân phối của khâu tích phân (đôi khi còn gọi là reset) tỷ lệ thuận với cả biên độ sai số lẫn khoảng thời gian xảy ra sai số. Biến độ phân phối của khâu tích phân trên tất cả các động điều chỉnh được xác định bởi độ lợi tích phân,  $K_i$ .

- Thừa số tích phân được cho bởi:

$$I_{out} = K_i \int_0^t e(\tau) d\tau \quad (2.76)$$

- Trong đó:

$I_{out}$ : thừa số tích phân của đầu ra.

$K_i$ : độ lợi tích phân.

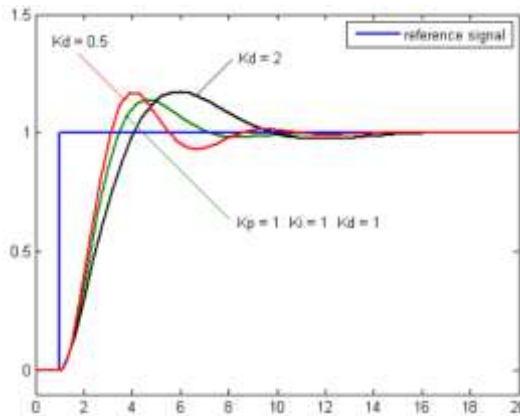
$e$ : sai số.

$t$ : thời gian hoặc thời gian tức thời (hiện tại).

$\tau$ : một biến tích phân trung gian.

- Khâu tích phân (khi cộng thêm khâu tỷ lệ) làm tăng tốc độ đáp ứng. Tuy nhiên, vì khâu tích phân là đáp ứng của sai số tích lũy trong quá khứ, nếu  $K_p K_i$  lớn  $\Rightarrow$  vọt lố.

c) Khâu vi phân



Hình 2.9: Đồ thị đáp ứng theo thời gian với Kd thay đổi

- Khâu vi phân xác định tốc độ thay đổi của sai số.
- Thừa số vi phân được cho bởi:

$$D_{out} = K_d \frac{d}{dt} e(t) \tag{2.77}$$

- Trong đó:

*Dout*: thừa số vi phân của đầu ra.

*Kd*: Độ lợi vi phân, một thông số điều chỉnh.

*e*: Sai số.

*t*: thời gian hoặc thời gian tức thời (hiện tại).

- Khâu vi phân làm chậm tốc độ thay đổi của đầu ra bộ điều khiển -> giảm biên độ vọt lố được tạo ra bởi thành phần tích phân và tăng cường độ ổn định của bộ điều khiển hợp lý.

d) Tóm tắt

- Điều khiển tỉ lệ (KP) có ảnh hưởng làm giảm thời gian lên và sẽ làm giảm nhưng không loại bỏ sai số xác lập. Điều khiển tích phân (KI) sẽ loại bỏ sai số xác lập nhưng có thể làm đáp ứng quá độ xấu đi. Điều khiển vi phân (KD) có tác dụng làm tăng sự ổn định của hệ thống, giảm vọt lố và cải thiện đáp ứng quá độ. Ảnh hưởng của mỗi bộ điều khiển KP, KI và KD lên hệ thống vòng kín được tóm tắt ở bảng bên dưới:

Bảng 2.2: Ảnh hưởng của mỗi bộ điều khiển Kp, Ki, Kd

Đáp ứng vòng kín	Thời gian lên	Lọt vó	Thời gian xác lập	Sai số xác lập
Kp	Giảm	Tăng	Thay đổi nhỏ	Giảm
Ki	Giảm	Tăng	Tăng	Loại bỏ
Kd	Thay đổi nhỏ	Giảm	Giảm	Thay đổi nhỏ

- Bộ điều khiển PID được mô tả bằng mô hình vào ra:

$$u(t) = k_p \left[ e(t) + \frac{1}{T_I} \int_0^t e(\tau) d\tau + T_D \frac{de(t)}{dt} \right] \quad (2.78)$$

- Trong đó  $e(t)$  là tín hiệu đầu vào,  $u(t)$  là tín hiệu đầu ra,  $k_p$  được gọi là hệ số khuếch đại,  $T_I$  là hằng số tích phân,  $T_D$  là hằng số vi phân.

- Từ mô hình vào ra trên ta có được hàm truyền đạt của bộ điều khiển PID:

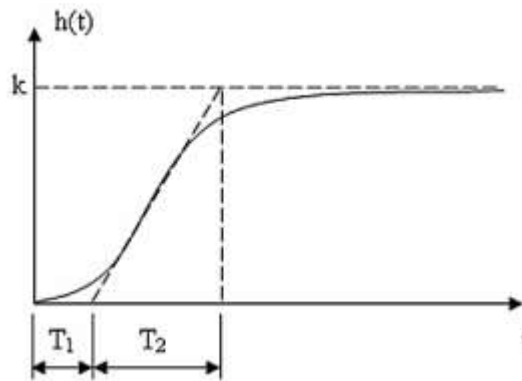
$$R(s) = k_p \left( 1 + \frac{1}{T_I s} + T_D s \right) \quad (2.79)$$

- Chất lượng hệ thống phụ thuộc vào các tham số  $k_p$ ,  $T_I$ ,  $T_D$ .

### 2.4.3. Phương pháp Ziegler-Nichols

- Phương pháp Ziegler-Nichols là pháp thực nghiệm để xác định tham số bộ điều khiển P, PI, hoặc PID bằng cách dựa vào đáp ứng quá độ của đối tượng điều khiển. Tùy theo đặc điểm của từng đối tượng, Ziegler và Nichols đưa ra hai phương pháp lựa chọn tham số của bộ điều khiển:

- Phương pháp Ziegler-Nichols thứ nhất: Phương pháp này áp dụng cho các đối tượng có đáp ứng đối với tín hiệu vào là hàm nấc có dạng chữ S (hình 1.9) như nhiệt độ lò nhiệt, tốc độ động cơ...

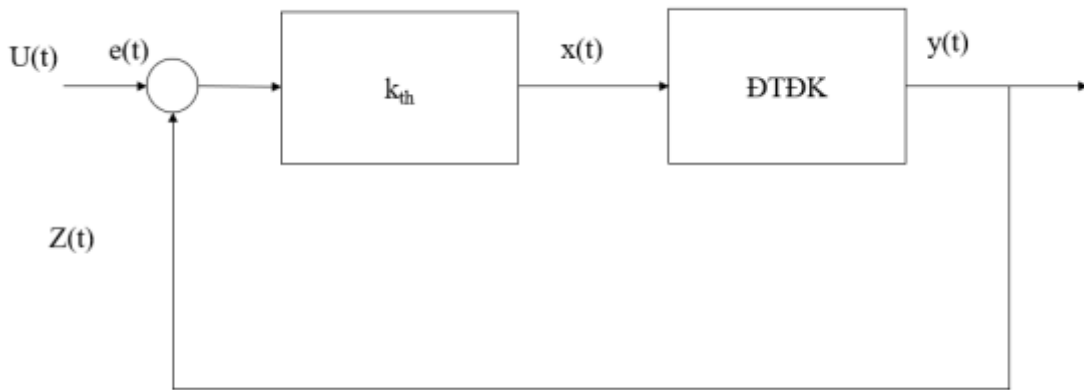


Hình 2.10: Đáp ứng nấc của hệ hờ có dạng S

Bảng 2.3: Các tham số PID theo phương pháp Ziegler-Nichols thứ nhất

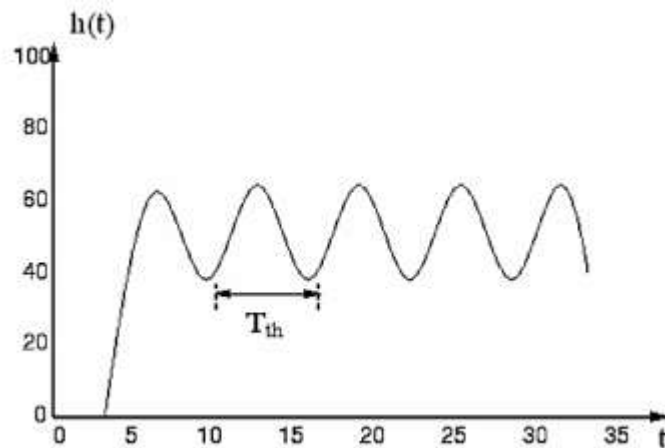
	<b>K<sub>P</sub></b>	<b>T<sub>I</sub></b>	<b>T<sub>D</sub></b>
P	$T_2/(k \cdot T_1)$	-	-
PI	$0.9T_2/(k \cdot T_1)$	$T_1/0.3$	-
PID	$1.2T_2/(k \cdot T_1)$	$2T_1$	$0.5 T_1$

- Phương pháp Ziegler-Nichols thứ hai: Phương pháp này áp dụng cho đối tượng có khâu tích phân lý tưởng như mực chất lỏng trong bồn chứa, vị trí hệ truyền động dùng động cơ... Đáp ứng quá độ của hệ hờ của đối tượng tăng đến vô cùng. Phương pháp này được thực hiện như sau:



Hình 2.11: Xác định hằng số khuếch đại tới hạn

- Thay bộ điều khiển PID trong hệ kín bằng bộ khuếch đại.
- Tăng hệ số khuếch đại tới giá trị tới hạn  $k_{th}$  để hệ kín ở chế độ biên giới ổn định, tức là  $h(t)$  có dạng dao động điều hòa.
- Xác định chu kỳ  $T_{th}$  của dao động:



Hình 2.12: Đáp ứng nấc của hệ kín khi  $k = k_{th}$

Bảng 2.4: Các tham số PID theo phương pháp Ziegler-Nichols thứ 2

	<b>K<sub>P</sub></b>	<b>T<sub>I</sub></b>	<b>T<sub>D</sub></b>
P	$0.5k_{th}$	-	-
PI	$0.45k_{th}$	$0.85 T_{th}$	-
PID	$0.6 k_{th}$	$0.5T_{th}$	$0.125 T_{th}$

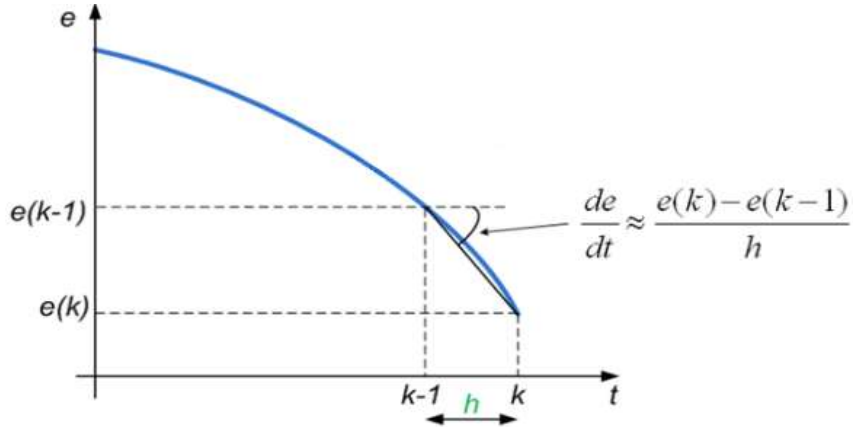
#### 2.4.4. Điều khiển PID số

Để sử dụng bộ điều khiển PID trong vi điều khiển hoặc trên máy tính, cần chuyển đổi từ mô hình liên tục sang rời rạc bằng cách xấp xỉ phương trình liên tục thành dạng rời rạc.

Rời rạc hóa thành phần đạo hàm:

$$\frac{de}{dt} = (e(k) - e(k-1)) / h \quad (2.80)$$

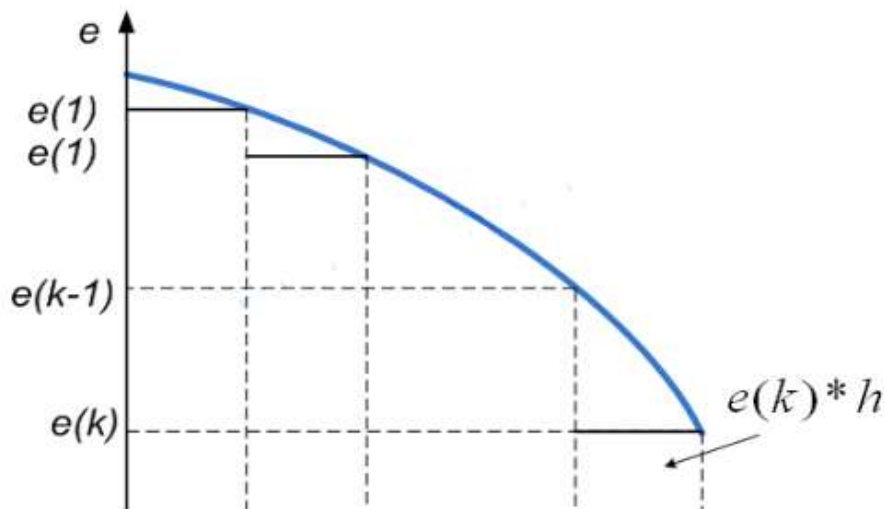
Trong đó  $e(k)$  là giá trị hiện tại của  $e$ ,  $e(k-1)$  là giá trị của  $e$  trong lần lấy mẫu trước đó và  $h$  là khoảng thời gian lấy mẫu ( $h$  là hằng số).



Hình 2.13: Xấp xỉ đạo hàm của biến sai số  $e$

- Rời rạc hóa thành phần tích phân:

$$\int_e^t e \cdot dt = \sum_0^k e(k) * h \quad (2.81)$$



Hình 2.14: Xấp xỉ tích phân của biến sai số  $e$

- Tổng hợp các xấp xỉ, công thức của bộ điều khiển PID số được trình bày:

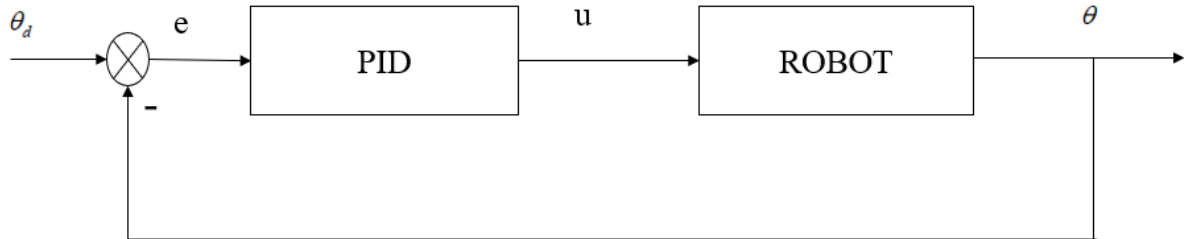
$$U(t) = K_p \cdot e + \frac{e(k) - e(k-1)}{h} + \sum_0^k e(k) * h \quad (2.82)$$

- Trong đó  $U(t)$  là đại lượng output của bộ điều khiển. Để đơn giản hóa việc tính thành phần tích phân, chúng ta nên dùng phương pháp “cộng dồn” (hay đệ quy):

$$I(k) = I(k-1) + e(k) * h \quad (2.83)$$

- Với  $I(k)$  là thành phần tích phân hiện tại,  $I(k-1)$  là thành phần tích phân trước đó. Các công thức trên rất dễ dàng để thực hiện bằng vi xử lý. Do đó, đến bước này chúng ta đã có thể đưa bộ PID số vào chip xử lý.

#### 2.4.5. Sơ đồ cấu trúc HTĐK Robot sử dụng PID



Hình 2.15: Sơ đồ khối hệ thống điều khiển Robot sử dụng PID

- Trong đó:

$\theta_d$  là vectơ góc quay đặt của biến khớp.

$\theta$  là vectơ góc đáp ứng của biến khớp.

- Cấu trúc bộ điều khiển PID gồm các luật điều khiển tỷ lệ - đạo hàm - tích phân như sau:

$$\tau = K_p e + K_D \dot{e} + K_i \int_0^t e(t) d\tau \quad (2.84)$$

- Trong đó:

+  $K_p$  - ma trận đường chéo các hệ số khuếch đại của 2 khớp động:

$$K_p = \text{diag}(K_{p1}, K_{p2}) \quad (2.85)$$

+  $K_D$  - ma trận đường chéo của hệ số đạo hàm của 2 khớp động:

$$K_D = \text{diag}(K_{D1}, K_{D2}) \quad (2.86)$$

+  $K_i$  - ma trận đường chéo của hệ số tích phân của 2 khớp động:

$$K_i = \text{diag}(K_{i1}, K_{i2}) \quad (2.87)$$

$e$  - sai số vị trí của khớp động:  $e = \theta_d - \theta$

$\dot{e}$  - sai số tốc độ:  $\dot{e} = \dot{\theta}_d - \dot{\theta}$

- Để điều chỉnh bộ điều khiển PID có thể sử dụng phương pháp xấp xỉ tuyến tính, phương pháp điều khiển tách kênh. Ngoài ra cũng có thể sử dụng phương pháp chỉnh định thông số qua mô phỏng trên Matlab – Simulink.

### 2.5. Đề xuất cho hệ thống

#### 2.5.1. Phần cứng

- Dùng lò xo làm khớp cánh tay.

- Dùng hộp số để tăng momen quay cho động cơ.
- Dùng động cơ DC giảm tốc kết hợp với encoder để điều khiển chuyển động các khớp cánh tay.
- Dùng động cơ step kết hợp với bánh răng làm cơ cấu chuyển động cho phần đế xoay.
- Dùng Arduino để làm bộ điều khiển cho hệ thống.

### **2.5.2. Phần mềm**

- Solidworks: để thiết kế mô hình 3D cho cánh tay robot.
- AutoCAD: dùng để vẽ các chi tiết 2D, sử dụng để cắt CNC làm sản phẩm cơ khí.
- Matlab: dùng để mô phỏng động lực học của cánh tay.
- Visual Studio Code: Viết giao diện điều khiển và hiển thị dữ liệu.
- Arduino IDE: để viết chương trình điều khiển.

### **2.6. Kết luận chương 2**

Ở chương này ta nghiên cứu phương trình động lực học của tay máy 3 bậc tự do cũng như hàm biến dạng đàn hồi tổng quát của một khâu. Để từ đó xác định được lực của khâu thứ  $i$ , đây là thành phần đóng vai trò quan trọng khi xây dựng sơ đồ khối để thiết lập hàm điều khiển cho tay máy 2 bậc tự do. Tìm hiểu về bộ điều khiển PID và đề xuất phần mềm, phần cứng cho hệ thống.

## Chương 3: THIẾT KẾ VÀ TÍNH CHỌN HỆ THỐNG ĐIỀU KHIỂN

### 3.1. Tính toán và lựa chọn thiết bị

#### 1. Động cơ DC

- Tính chọn động cơ:

+ Ta có khối lượng tải: 0,5kg

+ Khối lượng khâu 3: 1kg, chiều dài l3: 30cm

+ Khối lượng khâu 2: 2kg, chiều dài l2: 35cm

+ Hộp số sử dụng cho động cơ DC loại tua vít có tỷ số truyền 1:20

- Tính mô-men xoắn tại từng khớp

$$T'_3 = (m_3 + m_{tải}).g.L_3 = (1+0,5).9,81.0,3 = 4,41Nm$$

$$T'_2 = m_2.g.L_2 + (m_3 + m_{tải}).g.(L_2 + L_3) = 2.9,81.0,35 + (1+0,5).9,81.(0,3+0,35) = 16,43Nm \quad (3.1)$$

- Momen xoắn khi dùng hộp số, hiệu suất hộp số là 90%

$$T_3 = \frac{T'_3}{20.0,9} = \frac{4,41}{20.0,9} = 0,25Nm$$

$$T_2 = \frac{T'_2}{20.0,9} = \frac{16,43}{20.0,9} = 0,91Nm$$

$$T_3 = \frac{0,25}{0,6} = 0,42Nm$$

$$T_2 = \frac{0,91}{0,6} = 1,52Nm$$

- Do lực ma sát giữa các cơ cấu và đảm bảo an toàn, nên lực chọn động cơ có momen xoắn lớn hơn 30-40% so với tính toán nên momen xoắn cần chọn cho từng động cơ:

$$T_3 = \frac{0,25}{0,6} = 0,42Nm$$

$$T_2 = \frac{0,91}{0,6} = 1,52Nm$$

- Chọn động cơ JGB37-545 24V 110rpm có momen xoắn định mức 1,8Nm và JGB37-545 24V 320rpm có momen xoắn định mức 0,7Nm.

- Lý do chọn:

+ Khả năng chạy quá tải: Động cơ DC có thể chịu được tình trạng quá tải trong thời gian ngắn (150% đến 200% công suất định mức) mà không hỏng hóc ngay lập tức, phù hợp cho các ứng dụng cánh tay robot cần lực mạnh trong thời gian ngắn.

+ Điều khiển linh hoạt: Động cơ DC dễ điều chỉnh tốc độ và mô-men xoắn thông qua điện áp hoặc dòng điện, phù hợp cho các chuyển động mượt mà của cánh tay robot.

+ Chi phí thấp: Động cơ DC thường rẻ hơn so với động cơ bước và servo, đặc biệt trong các ứng dụng không yêu cầu độ chính xác cao, giúp giảm chi phí tổng thể cho hệ thống robot.

## 2. Nguồn điện

- Tính chọn nguồn điện:

Bảng 3.1: Thông số các thiết bị

Thiết bị	Điện áp(V)	Số lượng	Dòng mỗi thiết bị(A)	Tổng dòng(A)	Công suất(W)
Động cơ DC JGB37	24	2	4(chạy quá tải)	8	192
Động cơ bước 57HS	24	1	4	4	96
Servo MG90	5	1	1	1	5
Encoder Omron	5	2	0.1	0.2	1
Arduino mega 2560	5	1	0.5	0.5	2,5
Drive TBS7960	5	2	0.05	0.1	0,5
Drive TB6600	5	1	0,05	0,05	0,25
Tổng				13,85	297,25

a) Khi hệ thống hoạt động ở chế độ chờ

- Lúc này, chỉ có Arduino Mega, 2 encoder, và các driver điều khiển động cơ hoạt động. Nên dòng tiêu thụ vào khoảng:  $I = 0,5 + 2 \times 0,1 + 0,05 \times 3 = 0,85A$  (3.4)

- Công suất tiêu thụ tương ứng:  $P = 5 \times 0,85 = 4,25W$  (3.5)

b) Khi hệ thống hoạt động bình thường

- Các động cơ DC, STEP và servo hoạt động, dòng điện tiêu thụ lúc này vào khoảng:

$$I = 0,85 + 2 \times 2 + 4 + 1 = 9,85A \quad (3.6)$$

- Công suất tiêu thụ tương ứng:  $P = 5 \times (0,85 + 1) + 24 \times 8 = 201,25W$  (3.7)

c) Khi hệ thống hoạt động quá tải

- Khi động cơ bắt đầu khởi động, hoặc hoạt động trong quá trình quá tải thì dòng điện tiêu thụ lúc ở mức cao nhất:

$$I = 0,85 + 2 \times 4 + 4 + 1 = 13,85A \quad (3.8)$$

$$P = 5 \times (0,85 + 1) + 24 \times 13 = 297,25W \quad (3.9)$$

- Chọn nguồn tổ ong 24V-25A (dư 40% so với nhu cầu thực tế giúp hệ thống ổn định, không bị sụt áp và Tăng tuổi thọ nguồn).

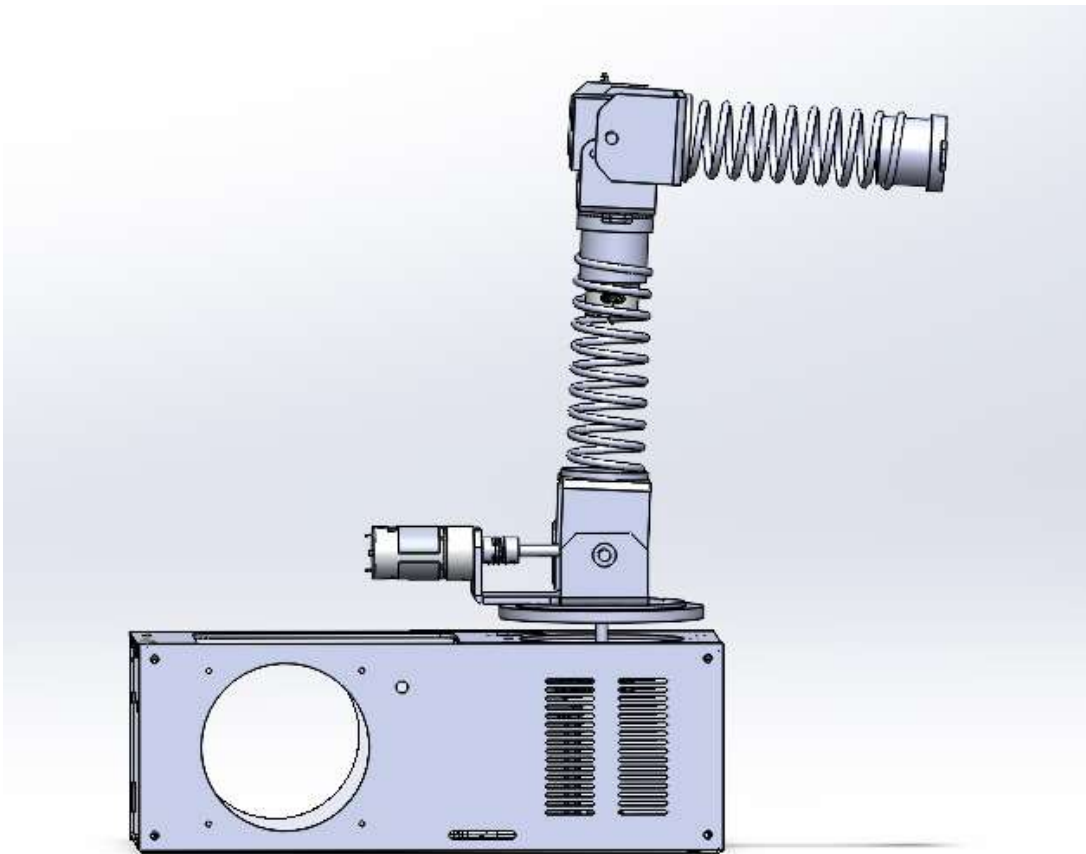
### **3.2. Thiết kế phần cứng hệ thống**

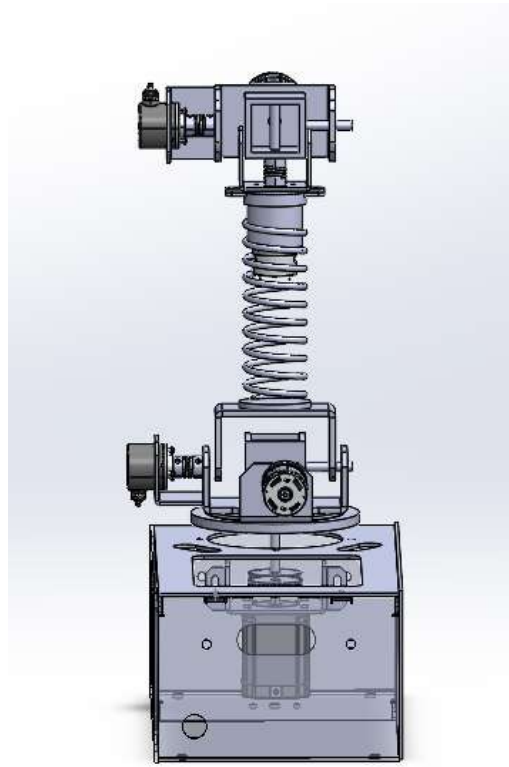
- Robot được thiết kế gồm 3 khớp xoay, sử dụng cơ cấu bánh răng cho để quay và sử dụng hộp số cho khớp 2 và 3 để tăng momen quay của động cơ.

- Chiều rộng đế 150mm, chiều dài 350mm, chiều cao 150mm

- Chiều dài link 2 khoảng 300mm, rộng 50mm

- Chiều dài link 3 khoảng 350mm, rộng 50mm

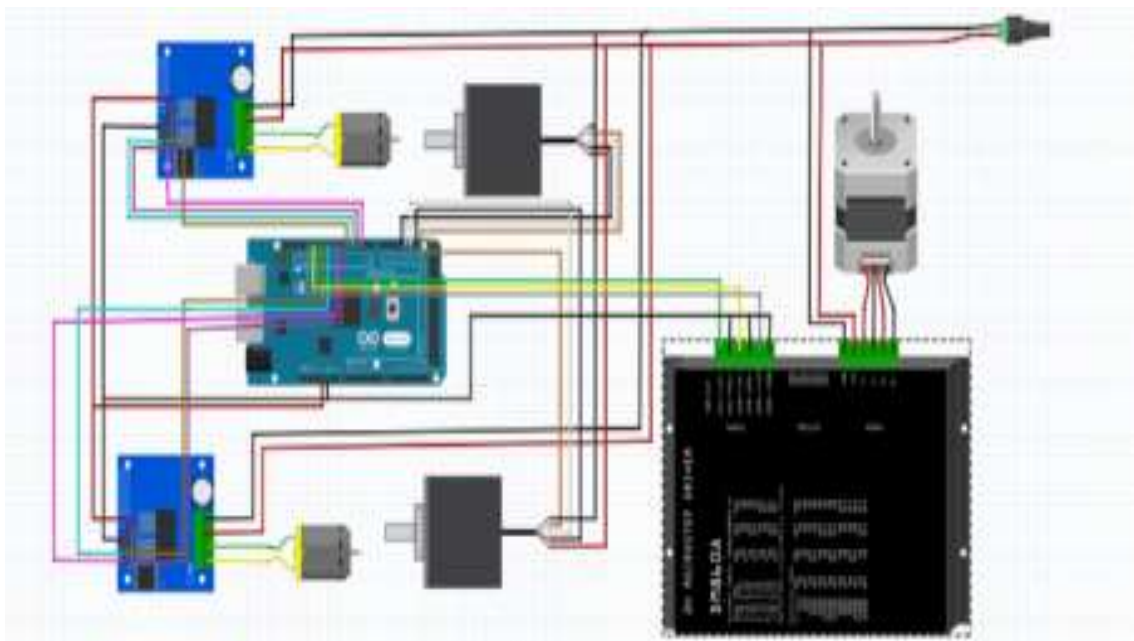




Hình 3.1: Mô hình cánh tay Robot 3DoF đàn hồi

### **3.3. Sơ đồ và thuật toán điều khiển**

#### **3.3.1. Sơ đồ nguyên lý mạch điều khiển**



Hình 3.2: Sơ đồ nguyên lý mạch điều khiển

- Nguồn cấp đầu vào là nguồn tổ ong 24VDC-25A cung cấp điện cho toàn bộ hệ thống, sử dụng module giảm áp LM2596 để biến đổi điện áp đầu vào cung cấp cho Encoder Omron và BTS7960.

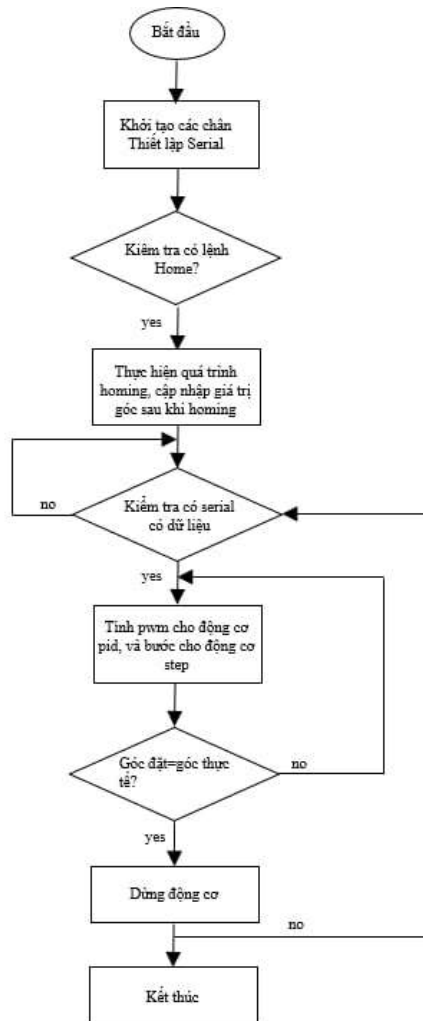
- Thứ tự nối chân:

+Chân PUL+ và chân DIR+ của TB6600 được nối lần lượt vào chân 6, 7 của arduino.

+ Chân Len, Ren BTS7960 cho động cơ DC khớp 2 được kích mức cao được nối vào chân 8,9 của arduino; các chân điều khiển hướng quay Lpwm và Rpwm được nối vào chân 10, 11 của arduino. Encoder được nối vào chân 2,4 của arduino.

+ Chân Len, Ren BTS7960 cho động cơ DC khớp 3 được kích mức cao được nối vào chân 12,13; các chân điều khiển hướng quay Lpwm và Rpwm được nối vào chân 3, 5 của arduino. Encoder được nối vào chân 20, 21 của arduino.

### **3.3.2. Lưu đồ thuật toán**

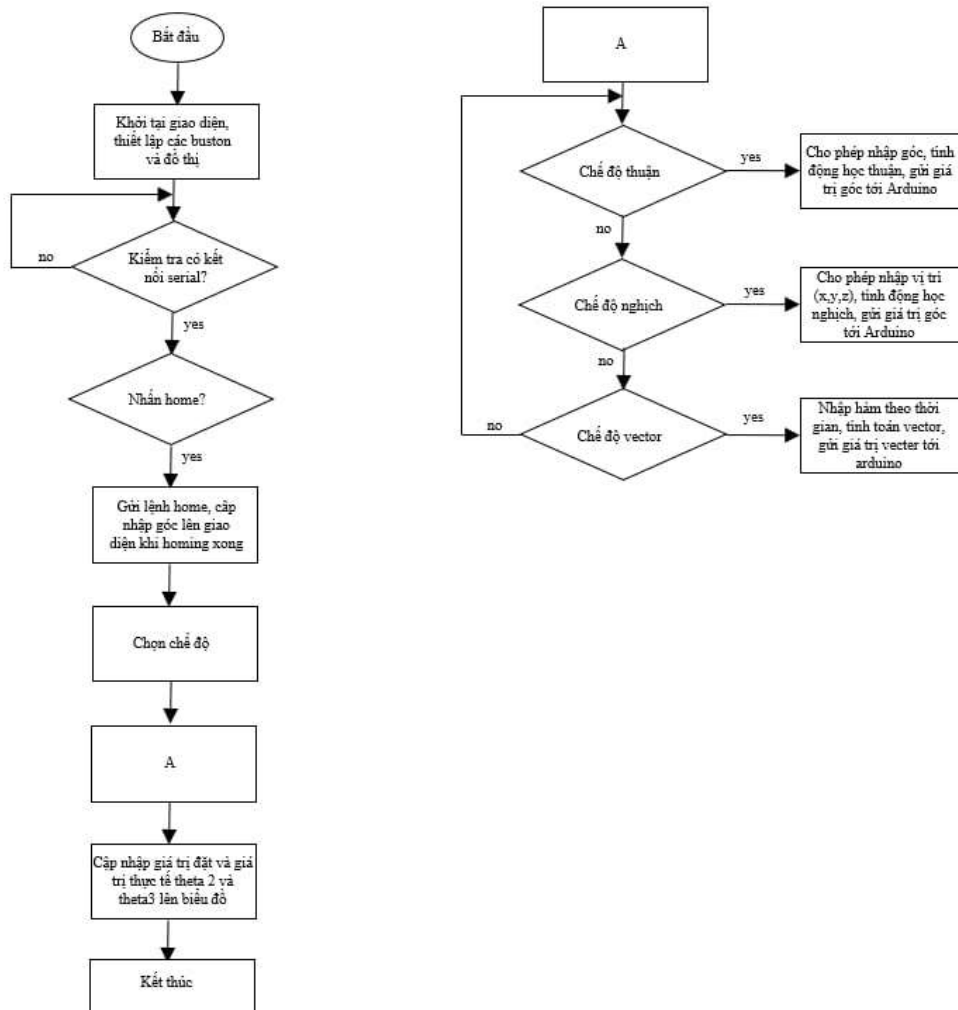


Hình 3.3: Lưu đồ thuật toán điều khiển

- Khi bắt đầu chạy, Arduino được kết nối với máy tính, thiết lập các chân điều khiển cho động cơ và cảm biến đo góc. Từ đó, Arduino liên tục đọc giá trị từ cảm biến để tính góc quay hiện tại của động cơ và gửi thông tin này cho máy tính đều đặn sau mỗi 100 mili giây.

- Trong khi đó, Arduino chờ đợi lệnh từ máy tính. Nếu chưa nhận được lệnh gì, nó tiếp tục gửi dữ liệu. Khi có lệnh gửi đến, Arduino sẽ kiểm tra nội dung lệnh. Nếu là lệnh “HOME”, Arduino thực hiện homing bằng cách di chuyển từng động cơ đến vị trí gốc dựa trên công tắc giới hạn, sau đó di chuyển đến vị trí mặc định ( $0^\circ$ ,  $60^\circ$ ,  $-45^\circ$ ). Nếu lệnh yêu cầu động cơ quay tới một góc cụ thể, Arduino sẽ tính toán hướng và khoảng cách cần quay, sau đó điều khiển động cơ bằng tín hiệu PWM (cho động cơ DC) hoặc tín hiệu bước (cho stepper) để quay đến đúng vị trí.

- Trong suốt quá trình quay, Arduino liên tục kiểm tra xem động cơ đã đạt đến góc mong muốn chưa. Nếu chưa, nó tiếp tục điều chỉnh; khi đã khớp góc, Arduino sẽ tự động dừng động cơ. Quá trình này được lặp đi lặp lại cho đến khi Arduino bị tắt.



Hình 3.4: Lưu đồ thuật toán cho giao diện điều khiển

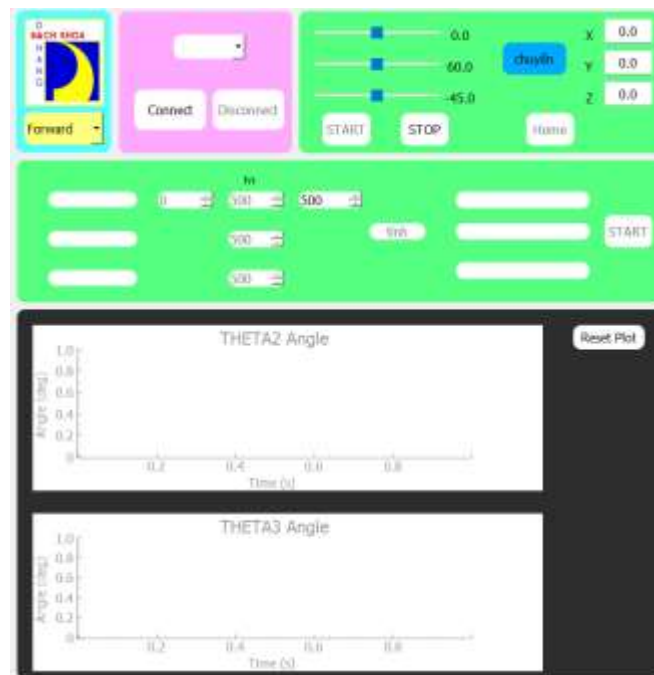
- Khi giao diện bắt đầu chạy, kiểm tra xem có kết nối Serial giữa máy tính và Arduino hay không. Nếu không có kết nối, hệ thống sẽ chờ và yêu cầu kiểm tra lại phần cứng. Nếu kết nối thành công, các chức năng giao diện sẽ được bật lên.

- Sau khi giao diện được bật lên, Nhấn "Home" gửi lệnh "HOME" qua Serial, chờ phản hồi thông báo quá trình homing hoàn tất. Sau đó người dùng có thể chọn giữa ba chế độ: chế độ thuận, chế độ nghịch hoặc chạy theo vector. Trong chế độ thuận, người dùng sẽ nhập vào các giá trị góc theta1, theta2 và theta3 rồi gửi các giá trị đó đến Arduino để điều khiển trực tiếp. Ngược lại, trong chế độ nghịch, người dùng sẽ nhập tọa độ đích (tọa độ điểm cần di chuyển đến), hệ thống sẽ tính toán các góc tương ứng thông qua thuật toán động học nghịch và sau đó gửi xuống Arduino để thực hiện. Còn ở chế độ vector, người dùng nhập hàm theo thời gian cho từng khớp, sau đó giao diện sẽ tính toán ra chuỗi vector (có thể thay đổi giá trị trong vector theo thời gian lấy mẫu trên giao diện).

- Sau khi dữ liệu đã được nhập vào và gửi đi, giao diện sẽ kiểm tra nút Start bật chưa để bắt đầu quá trình điều khiển. Nếu nút chưa được nhấn, giao diện sẽ tiếp tục chờ. Khi người dùng nhấn nút Start, giao diện sẽ tiến hành cập nhật các giá trị thực tế từ Arduino và hiển thị góc đặt và góc đo thực tế từ encoder cho các góc theta 2 và theta 3 lên biểu đồ (thời gian cập nhật mỗi lần là 100ms). Nếu start chưa được nhấn hoặc nhấn nút stop giao diện sẽ dừng việc hiển thị biểu đồ và không cập nhật dữ liệu mới.

### 3.4. Thiết kế giao diện điều khiển

- Sử dụng Python và QT Design để thiết kế giao diện người dùng, kết nối với máy tính.



Hình 3.5: Giao diện điều khiển

### 3.5. Chi phí đầu tư

- Chi phí đầu tư:

Bảng 3.2: Chi phí đầu tư

STT	Tên thiết bị	Số lượng	Đơn giá (VNĐ)	Thành tiền (VNĐ)
1	Arduino Mega 2560	1	450k/set	450.00
2	Động cơ DC jgb37	2	150.000	300.000
3	Động cơ Step 57hs	1	500.000	500.00
4	Động cơ servo mg995	1	80.000	80.000
3	Encoder omron 2000 xung	2	900.000	1.800.000
4	Hộp số giảm tốc tua vít (1:20)	2	520.000	1.040.000
	Tay gấp	1	90.000	90.000
5	Driver TBS7960	2	85.000	170.000
6	Driver TB6600	1	100.000	100.000
7	Nguồn tổ ong 24V-25A	1	500.000	500.000
7	LM5596	1	18.000	18.000
8	Bánh rang 1:1	1	40.000	40.000
9	Lò xo	2	50.000	100.000
	Vòng chặn trục+nối trục(6mm-8mm)			120.000
10	Cát laeser			1.000.000
11	ốc vít			30.000
12	Dây ruột gà(5m)	1		10.000
13	Dây điện			30.000
Tổng cộng (VNĐ)				6.378.000

### 3.6. Kết luận chương 3

Ở chương 3 đã trình bày quá trình thiết kế và lựa chọn hệ thống điều khiển cho cánh tay 3DoF đàn hồi. Thông qua việc phân tích, lựa chọn thiết bị, xây dựng sơ đồ điều khiển, thiết kế phần cứng và tính toán chi phí cho hệ thống.

## Chương 4: MÔ PHỎNG, THỰC NGHIỆM VÀ ĐÁNH GIÁ

### 4.1. Mô phỏng động lực học cánh tay robot

- Hệ phương trình động lực học Lagrange của cánh tay được viết dưới dạng ma trận sau:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} n_{11} & n_{12} \\ n_{21} & n_{22} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}$$

- Trong đó  $\tau_1$  và  $\tau_2$  là các mômen điều khiển tác động lên khâu 1 và khâu 2. Các thành phần của ma trận được tính theo các biểu thức dưới đây:

$$h_{11} = J_2 + m_2 l_{g1}^2 + m_3 (l_2^2 + l_{g2}^2 + 2l_2 l_3 C_3) + J_3 + J_t + m_t (l_2^2 + l_{g2}^2 + 2l_2 l_3 C_3)$$

$$h_{12} = h_{21} = m_3 (l_{g2}^2 + 2l_2 l_3 C_3) + J_3 + J_t + m_t l_3^2$$

$$h_{22} = m_2 l_{g2}^2 + J_2$$

$$g_1 = m_2 g l_{g1} C_2 + m_3 g (l_2 C_2 + l_{g2} C_{23}) + m_t g (l_2 C_2 + l_3 C_{23})$$

$$g_2 = m_3 g l_{g2} C_{23} + m_t g l_3 C_{23}$$

$$n_{11} = -m_3 l_2 l_3 (2\dot{q}_2 \dot{q}_3 + \dot{q}_3^2) \sin q_3$$

$$n_{12} = n_{21} = m_3 l_2 l_3 \dot{q}_2^2 \sin q_3$$

$$n_{22} = 0$$

\* Với các thông số:

- Chiều dài khớp 1:  $l_1 = 0.35$  m

- Chiều dài khớp 2:  $l_2 = 0.3$  m

- Khối lượng khớp 1:  $m_1 = 2$  kg

- Khối lượng khớp 2:  $m_2 = 1$  kg

- Momen quán tính khớp 1:  $J_1 = 0,05$  kgm<sup>2</sup>

- Momen quán tính khớp 2:  $J_2 = 0,05$  kgm<sup>2</sup>

- Chiều dài từ trục quay đến trọng tâm khớp 1:  $l_{g1} = 0,0175$  m

- Chiều dài từ trục quay đến trọng tâm khớp 2:  $l_{g2} = 0,015$  m

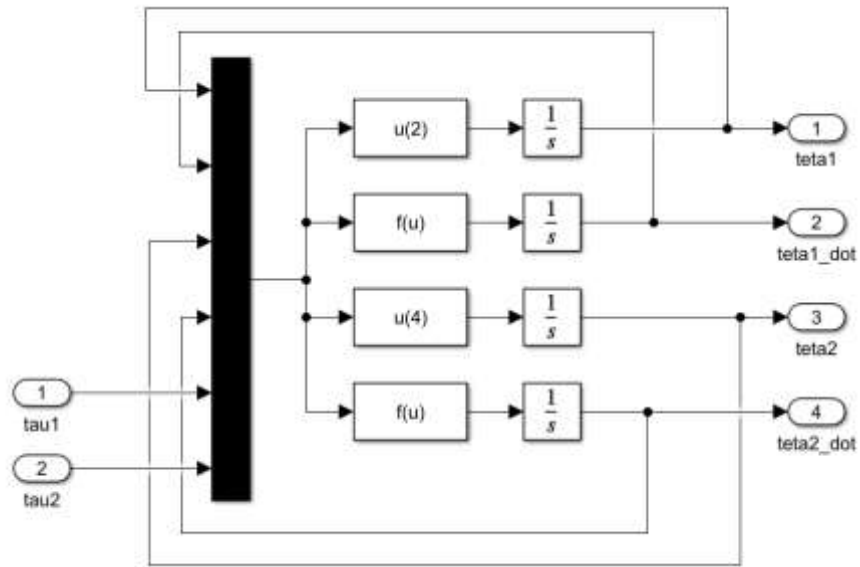
- Khối lượng tải:  $m_t = 1$  kg

- Momen quán tính tải:  $J_t = 0.1$  kgm<sup>2</sup>

- Gia tốc trọng trường:  $g = 9,81$  m/s<sup>2</sup>

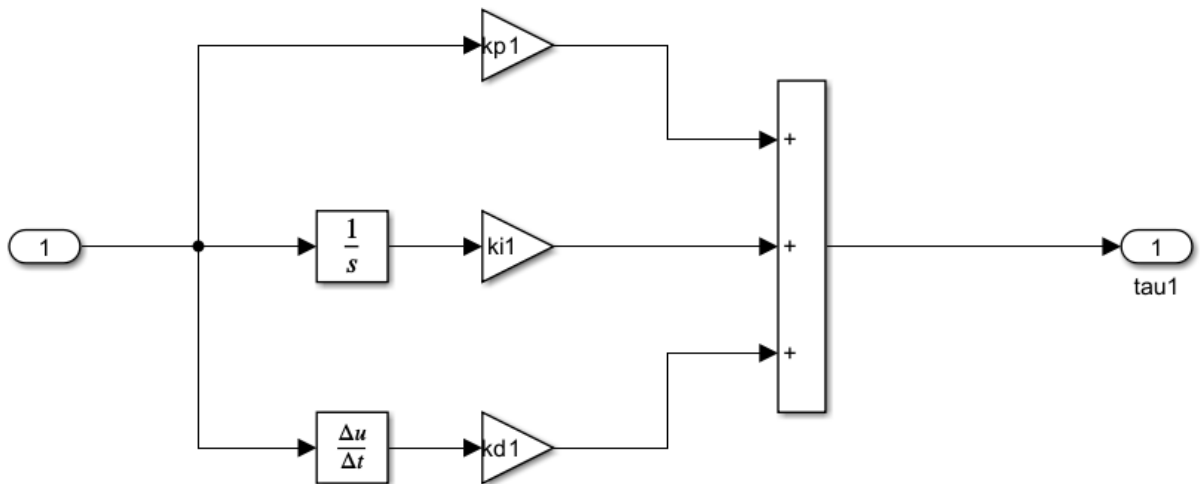
- Hệ số biến dạng đàn hồi:  $(EI)_1 = (EI)_2 = 1$  Nm<sup>2</sup>

\*Tiến hành thiết lập khối phương trình trên mô phỏng:

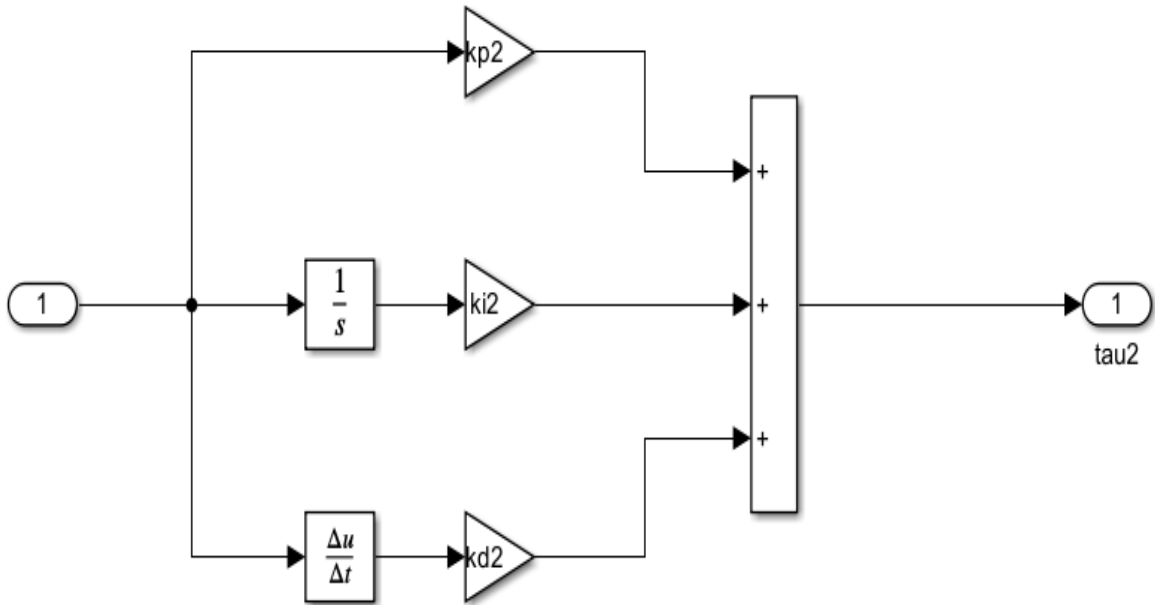


Hình 4.1: Khối phương trình

- Với:
- + tau1, tau2 là 2 giá trị ngõ vào  $\tau_1$  và  $\tau_2$
- + 2 khối f(u) chính là  $f_1$  và  $f_2$
- + Các khối integrator từ trên xuống lần lượt là giá trị của teta1, teta1\_dot, teta2 và teta2\_dot tương ứng với các ngõ ra.
- + Sau đó, ta tiến hành mô phỏng khối PID cho  $\tau_1$  và  $\tau_2$

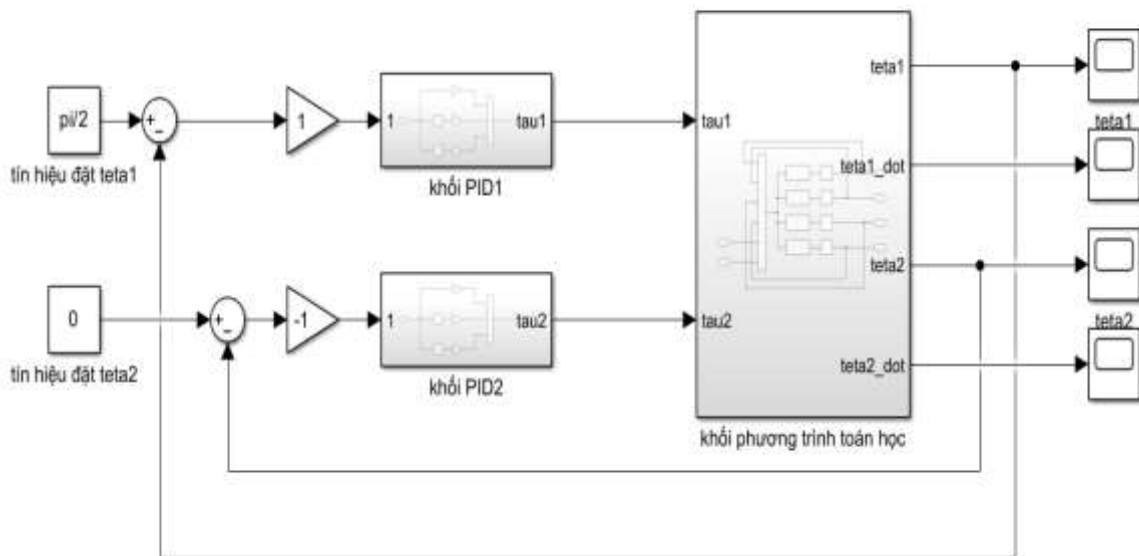


Hình 4.2: Khối PID cho  $\tau_1$



Hình 4.3: Khối PID cho  $\tau_2$

\* Sơ đồ mô phỏng HTĐK Robot sử dụng bộ PID như sau:



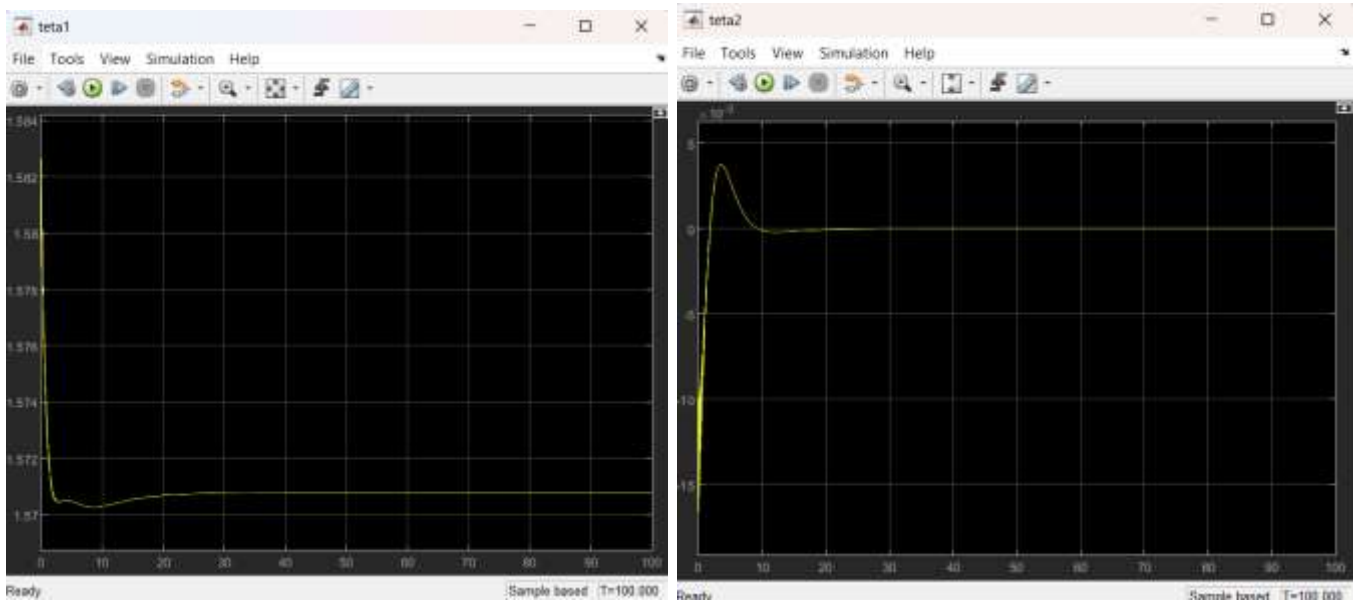
Hình 4.4: Sơ đồ mô phỏng HTĐK Robot sử dụng bộ PID

- Với tham số PID như sau:

+ Tham số bộ PID1:  $K_{p1} = 47.5$ ;  $K_{I1} = 4$ ;  $K_{D1} = 27.66$

+ Tham số bộ PID2:  $K_{p1} = 30$ ;  $K_{I1} = 1.5$ ;  $K_{D1} = 3.61$

- Kết quả mô phỏng:



Hình 4.5: Kết quả mô phỏng

- Nhận xét: nhìn vào kết quả hiển thị trên scope ta có thể thấy rằng sau 1 khoảng  $t_{x1}$  gần bằng 20s thì hệ thống của chúng ta cũng ổn định. Giá trị xác lập cũng gần bằng với giá trị đặt.

#### 4.2. Mô hình thực nghiệm



Hình 4.6: Mô hình thực nghiệm

\* Quá trình thực hiện:

- Trong quá trình hoạt động, khi động cơ quay gần tới góc đặt hoặc quay góc nhỏ thì điện áp cung cấp nhỏ làm momen động cơ không đủ quay tải, dẫn đến động cơ chạy nhưng cánh tay không hoạt động.

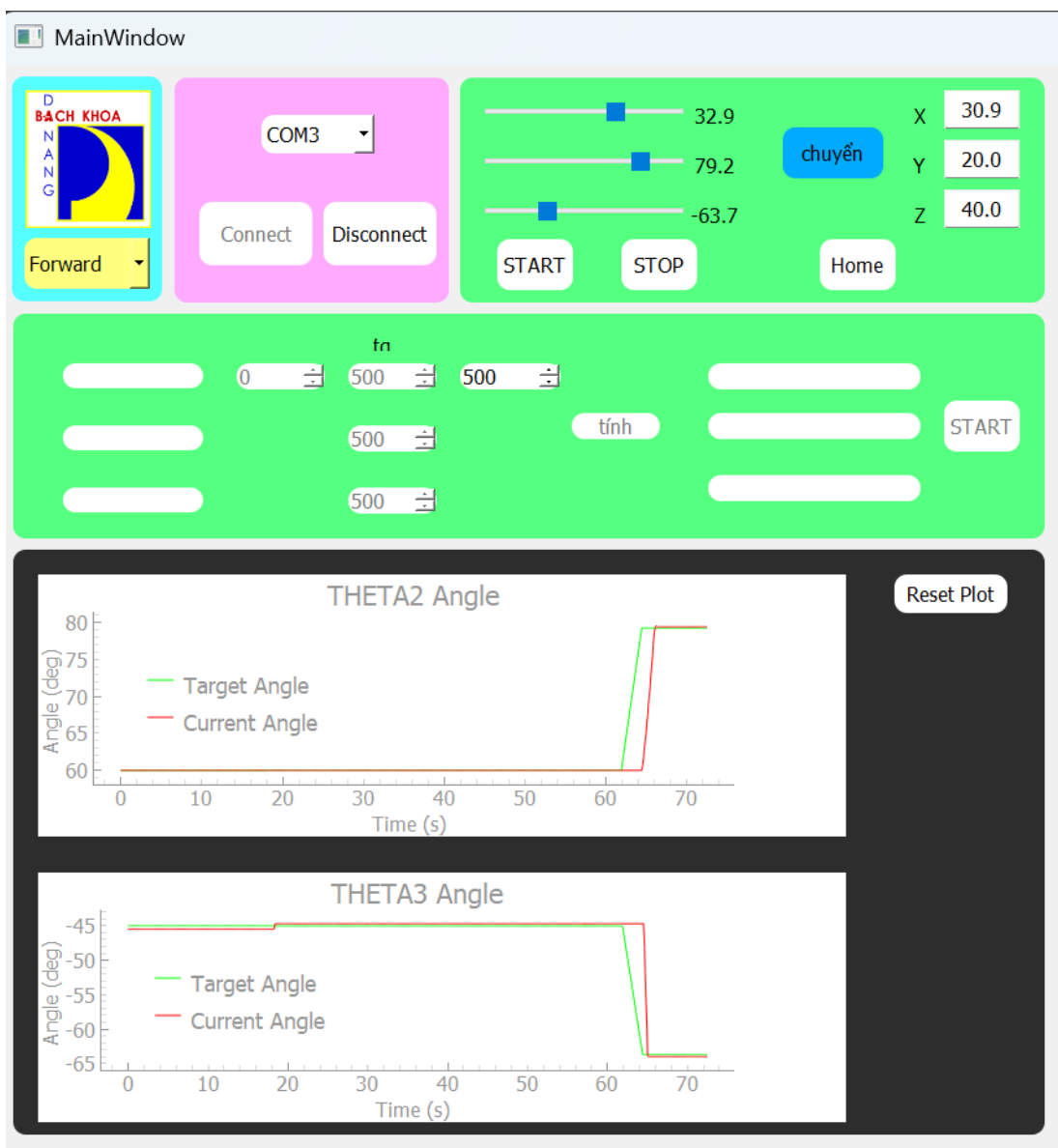
=>> Giải pháp: Giới hạn xung PWM tối thiểu điều khiển cho động cơ.

- Tuy bộ điều khiển đã sử dụng thuật toán PID nhưng do cơ cấu cánh tay lò xo, cánh tay vẫn còn giao động mạnh, đặc biệt khi chênh lệch giữa góc đặt và góc thực tế càng lớn dao động càng mạnh.

=>> Giải pháp: Tăng thời gian đáp ứng, và tìm ra bộ số  $K_p$ ,  $K_i$ ,  $K_d$  phù hợp trong thực nghiệm

### 4.3. Kết quả thực hiện

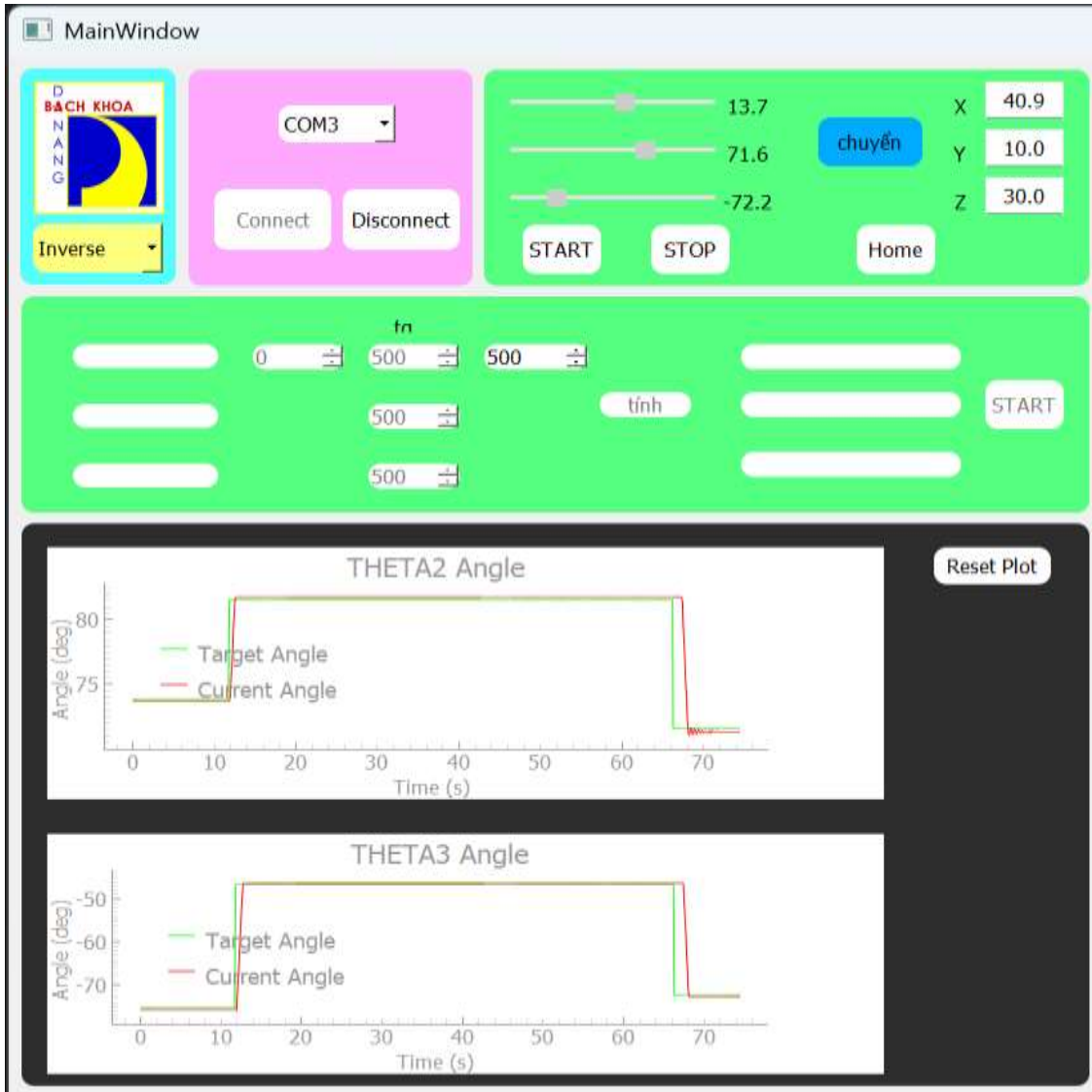
#### 4.3.1. Điều khiển theo góc đặt



Hình 4.7: Đồ thị kết quả điều khiển theo góc đặt

- Nhận xét: Nhìn vào kết quả đồ thị ta thấy với góc đặt (32,9; 79,2; -63,7) cánh tay đáp ứng được góc đặt, thời gian đáp ứng khoảng 1,5 giây. Nhìn chung THETA2 và THETA3 đáp ứng tương đối tốt, tuy nhiên vẫn còn giao động nhỏ do quán tính của cánh tay và sự đàn hồi của lò xo.

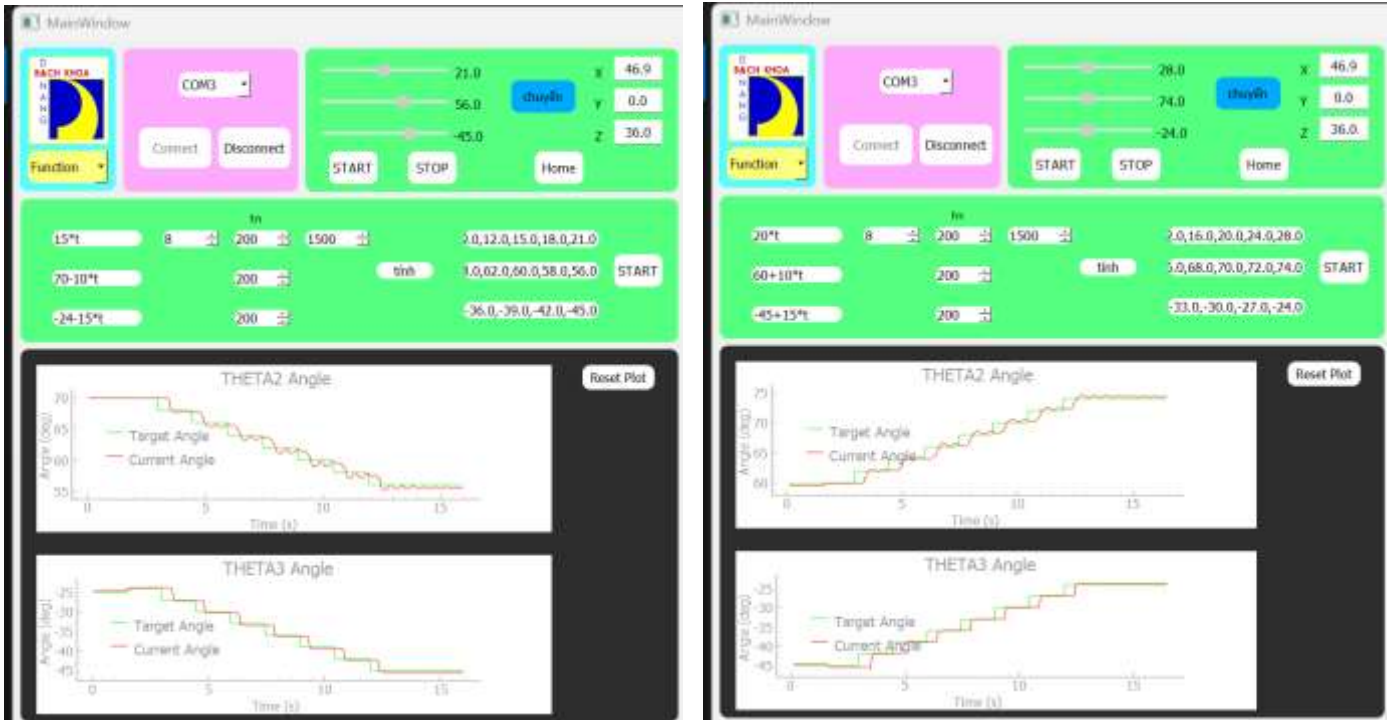
#### 4.3.2. Điều khiển theo vị trí



Hình 4.8: Đồ thị kết quả điều khiển theo vị trí

-Nhận xét: Nhìn vào kết quả đồ thị ta thấy với vị trí đầu cuối (40,9; 10; 30) cánh tay đáp ứng được góc đặt, thời gian đáp ứng khoảng 1,5 giây. Nhìn chung THETA2 và THETA3 đáp ứng tương đối tốt, tuy nhiên vẫn còn giao động nhỏ do quán tính của cánh tay và sự đàn hồi của lò xo.

#### 4.3.3. Điều khiển theo Vector



Hình 4.9: Đồ thị kết quả điều khiển theo vecto điều khiển theo vector

- Nhận xét: Nhìn vào kết quả đồ thị, với vector đặt mong muốn, THETA2 và THETA3 đáp ứng tương đối với góc đặt thay đổi liên tục(1,5s). THETA2 và THETA3 vẫn còn dao động, góc thực tế THETA2 thể hiện rõ sự biến thiên do khối lượng đặt lên THETA2 tương đối lớn(toàn bộ khối lượng cánh tay) cộng với góc quay thay đổi liên tục cùng với sự đàn hồi lò xo. Nhìn chung hệ thống điều khiển vẫn đảm bảo được độ chính xác trong toàn bộ quá trình chuyển động.

## **KẾT LUẬN**

### **1. Kết quả đạt được**

- Hoàn thành việc thiết kế và chế tạo cánh tay robot 3DoF đàn hồi dựa trên cơ sở mô phỏng và lý thuyết.

- Giải quyết được các phương trình động học, động lực học. Xây dựng được thuật toán điều khiển robot trên Matlab.

- Đã điều khiển được cánh tay theo góc đặt và vị trí

### **2. Các hạn chế còn gặp phải**

- Việc mô phỏng và tính toán trên Matlab đã đáp ứng được nhiệm vụ của đề tài. Tuy nhiên, khi gia công thực tế sẽ tồn tại nhiều sai số dẫn đến kết cấu của robot không ổn định.

- Lập trình điều khiển chưa tối ưu.

- Do kết cấu cánh tay Robot là lò xo nên quá trình hoạt động gặp nhiều dao động, không ổn định.

### **3. Hướng phát triển đề tài**

- Tối ưu hóa phần điều khiển.

- Sử dụng thêm Loadcell để xác định được khối lượng tải.

- Sử dụng công tác hành trình quang tối ưu độ chính xác và mở rộng phạm vi hoạt động của cánh tay

# TÀI LIỆU THAM KHẢO

## 1. Tiếng việt

[1] Nguyễn Thiện Phúc, Robot công nghiệp- nhà xuất bản khoa học và kỹ thuật.

[2] Nguyễn Hoàng Mai, Bùi Tuấn Việt Linh, Thiết kế bộ điều khiển mở lai điều khiển vị trí tay máy có tính đến module đàn hồi của cánh tay, Đại học Bách Khoa Đà Nẵng, 2017.

[3] Nguyễn Hoàng Mai, Phạm Minh Tuấn, Điều khiển trượt cho tay máy có tính đến độ đàn hồi, Đại học Bách Khoa Đà Nẵng

[4] Dương Xuân Biên, Chu Anh Mỹ, Phan Bùi Khôi, Xây dựng hệ phương trình động lực học hệ tay máy có khâu đàn hồi, Học viện Kỹ Thuật Quân Sự, 2017.

[5] Trần Đức Thuận, Trần Xuân Kiên, Bùi Hồng Huế, Tổng hợp hệ điều khiển tay máy có khớp đàn hồi ứng dụng vi cảm biến vị cơ quán tính, Viện Khoa Học và Công Nghệ Quân Sự, 2016.

## 2. Tiếng anh

[6] R.H. Cannon Jr. and E. Schmitz, *Initial experiments on the end-point control of flexible one-link robot*, Internat. J. Robotics Research 3(3) (1984) 62-75.

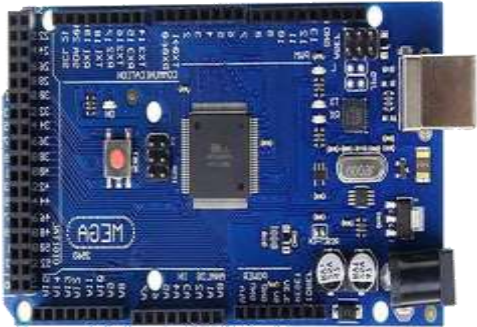


[7] W.J. Book, *Modeling, design and control of flexible manipulators arms: Status and trends*, NASA Conf. on Space Telerobotics, Vol. 3 (1989) 11-24.

[8] E. Bayo and M.A. Serna, *Penalty formulation for the dynamic analysis of elastic mechanisms*, *J. Mechanisms, Transmission and Automation in Design* 3 (1989) 321-327.





[9] B. Subudhi, A.S. Morris “Dynamic modelling, simulation and control of a manipulator with flexible links and joints”, Department of Automatic Control & Systems Engineering, University of Sheffield, Mappin Street, Sheffield S1 3JD, UK, 17 July 2002.



## PHỤ LỤC

### 1. Lựa chọn thiết bị

STT	Tên Thiết bị	Hình ảnh	Thông số chính
1	Arduino Mega 2560		<ul style="list-style-type: none"> <li>- Vi điều khiển chính: ATmega2560</li> <li>- Nguồn nuôi mạch: 5VDC từ cổng USB</li> <li>- Số chân Digital I/O: 54 (15 PWM)</li> <li>- Số chân Analog Input: 16</li> <li>- Flash Memory: 256 KB trong đó 8 KB sử dụng cho bootloader.</li> <li>- SRAM: 8 KB</li> <li>- EEPROM: 4 KB</li> </ul>
2	động cơ JGB37-545		<ul style="list-style-type: none"> <li>- Tỉ số truyền 110:1</li> <li>- Dòng không tải: 200mA</li> <li>- Dòng chịu đựng tối đa khi có tải: 4A</li> <li>- Tốc độ không tải: 110RPM</li> <li>- Lực léo Moment tối đa: 20KG.CM</li> <li>- Chiều dài hộp số L: 26.5mm</li> </ul>
3	Động cơ step		<ul style="list-style-type: none"> <li>- Đường kính trục động cơ: 8mm</li> <li>- Góc bước động cơ: 1.8 độ.</li> <li>- Dòng điện: 4.0V.</li> <li>- Điện áp động cơ: 24VDC.</li> <li>- Mô-men xoắn: 1.8Nm</li> </ul>

4	Encoder Omron E6B2 – CWZ6C		<ul style="list-style-type: none"> <li>- Đường kính trục: 6mm</li> <li>- Đường kính thân: 40mm</li> <li>- Điện áp hoạt động: 5...24VDC</li> <li>- Độ phân giải: 2000 xung/vòng</li> <li>- Pha đầu ra: A, B và Z</li> <li>- Tần số đáp ứng: 100 KHz</li> </ul>
5	Hộp số		<ul style="list-style-type: none"> <li>- Vật liệu gia công: Thép carbon, đồng</li> <li>- Vật liệu khung vỏ: Hợp kim nhôm</li> <li>- Đỡ trục: Bạc đạn</li> <li>- Đường kính trục: 08mm</li> <li>- Tỉ số truyền: 1:20</li> <li>- Khoảng cách tâm: 18mm</li> <li>- Tổng khối lượng: 0.25kg</li> </ul>
6	BTS7960		<ul style="list-style-type: none"> <li>- Nguồn cấp: 6 ~ 27VDC</li> <li>- Dòng điện tải mạch: 43A</li> <li>- Tín hiệu logic điều khiển: 3.3 ~ 5VDC.</li> <li>- Tự động shutdown khi điện áp thấp: để tránh điều khiển động cơ ở mức điện áp thấp thiết bị sẽ tự shutdown. Nếu điện áp &lt; 5.5VDC, driver sẽ tự ngắt điện và sẽ mở lại sau khi điện áp &gt; 5.5VDC.</li> <li>- Bảo vệ quá nhiệt: BTS7960 bảo vệ chống quá nhiệt bằng cảm biến nhiệt tích hợp bên trong.</li> </ul>

			<p>Đầu ra sẽ bị ngắt khi có hiện tượng quá nhiệt.</p> <ul style="list-style-type: none"> <li>- Kích thước: 40 x 50 x 12mm.</li> </ul>
7	TB6600		<ul style="list-style-type: none"> <li>- Nguồn đầu vào là 9V – 42V.</li> <li>- Dòng cấp tối đa là 4A.</li> <li>- Ngõ vào có cách ly quang, tốc độ cao.</li> <li>- Có tích hợp đo quá dòng quá áp.</li> </ul>
7	Thanh trục		Thanh trục 8mm, dài 170mm
8	Nguồn tổ ong 24V-25A		<ul style="list-style-type: none"> <li>- Điện áp đầu vào: 110/220VAC, 50Hz</li> <li>- Điện áp đầu ra: 24VDC</li> <li>- Dòng điện đầu ra: 0 - 25A</li> <li>- Công suất đầu ra: 600W</li> <li>- Kích thước (Dài x Rộng x Cao): 238 x 124 x 65 mm</li> </ul>
9	Bánh răng 1:1		Trục 8mm

10	Vòng chặn trục		
11	Lò xo		Dài 150mm, rộng 50mm

## 2. Code điều khiển Arduino

```
#define STEP_PULSE 6
#define STEP_DIR 7
#define R_EN1 8
#define L_EN1 9
#define RPWM1 10
#define LPWM1 11
#define R_EN2 12
#define L_EN2 13
#define RPWM2 3
#define LPWM2 5
#define ENC_A1 2
#define ENC_B1 4
#define ENC_A2 20
#define ENC_B2 21
#define LIMIT_SWITCH1 26
#define LIMIT_SWITCH2 27
#define LIMIT_SWITCH3 28
#define LIMIT_SWITCH4 29
#define LIMIT_SWITCH5 30
#define LIMIT_SWITCH6 31
#define LIMIT_ACTIVE_LOW true
volatile long encoderCount1 = 0;
volatile long encoderCount2 = 0;
long stepCount = 0;
const float PULSES_PER_REV_DC = 4000.0;
const float DEG_PER_PULSE_DC = 360.0 / PULSES_PER_REV_DC;
const float STEPS_PER_REV = 3200.0;
const float DEG_PER_STEP = 360.0 / STEPS_PER_REV;
float currentAngle1 = 0.0;
float currentAngle2 = 0.0;
float currentAngle3 = 0.0;
```

```

float targetAngle1 = -90.0;
float targetAngle2 = 90.0;
float targetAngle3 = 0.0;
bool isRunning1 = false;
bool isRunning2 = false;
bool isRunning3 = false;

const float Kp = 4.0;
const float Ki = 0.001;
const float Kd = 0.5;
const float Kp3 = 7.0;
const float Ki3 = 0.02;
const float Kd3 = 0.4;

float errorSum2 = 0.0, errorSum3 = 0.0;
float lastError2 = 0.0, lastError3 = 0.0;

const unsigned long MIN_PULSE_DELAY = 1500;
const unsigned long MAX_PULSE_DELAY = 4000;
const int ACCEL_STEPS = 150;
const int MAX_PWM_DC = 255;
const int MIN_PWM_DC = 150;
const int MIN_PWM_THETA2_POS = 160;
const int MIN_PWM_THETA3_POS = 120;
const int MIN_PWM_THETA3_NEG = 120;
const int PWM_ACCEL_STEP = 2;

int currentPwm2 = 0;
int currentPwm3 = 0;

const float DEACCEL_DISTANCE = 10.0;

void setup() {
  Serial.begin(9600);
  pinMode(STEP_PULSE, OUTPUT);
  pinMode(STEP_DIR, OUTPUT);
  digitalWrite(STEP_PULSE, LOW);
  digitalWrite(STEP_DIR, LOW);

```

```

pinMode(R_EN1, OUTPUT);
pinMode(L_EN1, OUTPUT);
pinMode(RPWM1, OUTPUT);
pinMode(LPWM1, OUTPUT);
digitalWrite(R_EN1, HIGH);
digitalWrite(L_EN1, HIGH);
analogWrite(RPWM1, 0);
analogWrite(LPWM1, 0);
pinMode(R_EN2, OUTPUT);
pinMode(L_EN2, OUTPUT);
pinMode(RPWM2, OUTPUT);
pinMode(LPWM2, OUTPUT);
digitalWrite(R_EN2, HIGH);
digitalWrite(L_EN2, HIGH);
analogWrite(RPWM2, 0);
analogWrite(LPWM2, 0);
pinMode(ENC_A1, INPUT_PULLUP);
pinMode(ENC_B1, INPUT_PULLUP);
pinMode(ENC_A2, INPUT_PULLUP);
pinMode(ENC_B2, INPUT_PULLUP);
attachInterrupt(digitalPinToInterrupt(ENC_A1), readEncoder1, CHANGE);
attachInterrupt(digitalPinToInterrupt(ENC_A2), readEncoder2, CHANGE);
pinMode(LIMIT_SWITCH1, INPUT_PULLUP);
pinMode(LIMIT_SWITCH2, INPUT_PULLUP);
pinMode(LIMIT_SWITCH3, INPUT_PULLUP);
pinMode(LIMIT_SWITCH4, INPUT_PULLUP);
pinMode(LIMIT_SWITCH5, INPUT_PULLUP);
pinMode(LIMIT_SWITCH6, INPUT_PULLUP);
}

```

```

void homeMotors() {

```

```

const unsigned long HOMING_TIMEOUT = 60000;
const unsigned long MOVE_TIMEOUT = 15000;
// Homing THETA1
unsigned long startTime = millis();
digitalWrite(STEP_DIR, LOW);
while ((LIMIT_ACTIVE_LOW ? digitalRead(LIMIT_SWITCH1) == HIGH :
digitalRead(LIMIT_SWITCH1) == LOW) && millis() - startTime < HOMING_TIMEOUT)
{
    digitalWrite(STEP_PULSE, HIGH);
    delayMicroseconds(2500);
    digitalWrite(STEP_PULSE, LOW);
    delayMicroseconds(2500);
}
if (LIMIT_ACTIVE_LOW ? digitalRead(LIMIT_SWITCH1) == HIGH :
digitalRead(LIMIT_SWITCH1) == LOW) {
} else {
    stepCount = (-90.0 / DEG_PER_STEP);
    currentAngle1 = -90.0;
    targetAngle1 = -90.0;
}
// Homing THETA2
startTime = millis();
analogWrite(RPWM1, 160);
analogWrite(LPWM1, 0);
while ((LIMIT_ACTIVE_LOW ? digitalRead(LIMIT_SWITCH2) == HIGH :
digitalRead(LIMIT_SWITCH2) == LOW) && millis() - startTime < HOMING_TIMEOUT)
{
    delay(10);
}
analogWrite(RPWM1, 0);
if (LIMIT_ACTIVE_LOW ? digitalRead(LIMIT_SWITCH2) == HIGH :
digitalRead(LIMIT_SWITCH2) == LOW) {
} else {
    noInterrupts();

```

```

encoderCount1 = (90.0 / DEG_PER_PULSE_DC);
currentAngle2 = 90.0;
targetAngle2 = 90.0;
interrupts();
}
// Homing THETA3
startTime = millis();
analogWrite(RPWM2, 0);
analogWrite(LPWM2, 120);
while ((LIMIT_ACTIVE_LOW ? digitalRead(LIMIT_SWITCH6) == HIGH :
digitalRead(LIMIT_SWITCH6) == LOW) && millis() - startTime < HOMING_TIMEOUT)
{
    delay(10);
}
analogWrite(LPWM2, 0);
if (LIMIT_ACTIVE_LOW ? digitalRead(LIMIT_SWITCH6) == HIGH :
digitalRead(LIMIT_SWITCH6) == LOW) {
} else {
    noInterrupts();
    encoderCount2 = (-90.0 / DEG_PER_PULSE_DC); // Đặt -90° tại LIMIT_SWITCH6
    currentAngle3 = -90.0;
    targetAngle3 = -90.0;
    interrupts();
}
isRunning1 = isRunning2 = isRunning3 = false;
digitalWrite(STEP_PULSE, LOW);
analogWrite(RPWM1, 0);
analogWrite(LPWM1, 0);
analogWrite(RPWM2, 0);
analogWrite(LPWM2, 0);
currentPwm2 = 0;
currentPwm3 = 0;

```

```

// Đợi 1 giây để ổn định
delay(1000);
// Di chuyển đến (0, 60, -45)
targetAngle1 = 0.0;
isRunning1 = true;
startTime = millis();
while (isRunning1 && millis() - startTime < MOVE_TIMEOUT) {
    float error = targetAngle1 - currentAngle1;
    if (abs(error) > 0.05) {
        digitalWrite(STEP_DIR, error > 0 ? HIGH : LOW);
        int stepsToMove = abs(error) / DEG_PER_STEP;
        unsigned long adjustedMinPulseDelay = MIN_PULSE_DELAY;
        if (abs(error) <= 10.0) {
            adjustedMinPulseDelay = map(abs(error), 0, 10, 4000, MIN_PULSE_DELAY);
        }
        for (int i = 0; i < stepsToMove; i++) {
            unsigned long pulseDelay = (i < ACCEL_STEPS) ? map(i, 0, ACCEL_STEPS - 1,
                MAX_PULSE_DELAY, adjustedMinPulseDelay) :
                (i >= stepsToMove - ACCEL_STEPS) ? map(i, stepsToMove -
                ACCEL_STEPS, stepsToMove - 1, adjustedMinPulseDelay, MAX_PULSE_DELAY) :
                adjustedMinPulseDelay;
            digitalWrite(STEP_PULSE, HIGH);
            delayMicroseconds(pulseDelay);
            digitalWrite(STEP_PULSE, LOW);
            delayMicroseconds(pulseDelay);
            stepCount += (error > 0 ? 1 : -1);
            currentAngle1 = stepCount * DEG_PER_STEP;
        }
    }
    isRunning1 = false;
    digitalWrite(STEP_PULSE, LOW);
}

```

```

targetAngle2 = 60.0;
isRunning2 = true;
startTime = millis();
while (isRunning2 && millis() - startTime < MOVE_TIMEOUT) {
  noInterrupts();
  currentAngle2 = (encoderCount1 * DEG_PER_PULSE_DC);
  interrupts();
  float error = targetAngle2 - currentAngle2;
  if (abs(error) > 0.1) {
    if (error < 0 && currentAngle2 > targetAngle2 + 0.5) {
      errorSum2 = 0.0;
    } else if (error > 0 && currentAngle2 < targetAngle2 - 0.5) {
      errorSum2 = 0.0;
    }
    errorSum2 += error;
    float errorDiff = error - lastError2;
    float pidOutput = Kp * error + Ki * errorSum2 + Kd * errorDiff;
    int minPwm = (pidOutput > 0) ? MIN_PWM_THETA2_POS : MIN_PWM_DC;
    int desiredPwm = constrain(abs(pidOutput), minPwm, MAX_PWM_DC);
    if (error < 0 && (currentAngle2 - targetAngle2) > DEACCEL_DISTANCE) {
      desiredPwm = MIN_PWM_THETA2_POS;
    } else if (error > 0 && (targetAngle2 - currentAngle2) > DEACCEL_DISTANCE) {
      desiredPwm = MIN_PWM_DC;
    }
    currentPwm2 = desiredPwm;
    if (pidOutput > 0) {
      analogWrite(RPWM1, currentPwm2);
      analogWrite(LPWM1, 0);
    } else {
      analogWrite(RPWM1, 0);
      analogWrite(LPWM1, currentPwm2);
    }
  }
}

```

```

    lastError2 = error;
} else {
    isRunning2 = false;
    analogWrite(RPWM1, 0);
    analogWrite(LPWM1, 0);
    currentPwm2 = 0;
    noInterrupts();
    encoderCount1 = (60.0 / DEG_PER_PULSE_DC);
    currentAngle2 = 60.0;
    interrupts();
}
}
targetAngle3 = -45.0;
isRunning3 = true;
startTime = millis();
while (isRunning3 && millis() - startTime < MOVE_TIMEOUT) {
    noInterrupts();
    currentAngle3 = (encoderCount2 * DEG_PER_PULSE_DC);
    interrupts();
    float error = targetAngle3 - currentAngle3;
    if (abs(error) > 0.1) {
        if (error < 0 && currentAngle3 > targetAngle3 + 0.5) {
            errorSum3 = 0.0;
        } else if (error > 0 && currentAngle3 < targetAngle3 - 0.5) {
            errorSum3 = 0.0;
        }
        errorSum3 += error;
        float errorDiff = error - lastError3;
        float pidOutput = Kp3 * error + Ki3 * errorSum3 + Kd3 * errorDiff;
        int minPwm = (pidOutput > 0) ? MIN_PWM_THETA3_POS :
MIN_PWM_THETA3_NEG;
        int desiredPwm = constrain(abs(pidOutput), minPwm, MAX_PWM_DC);

```

```

if (error < 0 && (currentAngle3 - targetAngle3) > DEACCEL_DISTANCE) {
    desiredPwm = MIN_PWM_THETA3_POS;
} else if (error > 0 && (targetAngle3 - currentAngle3) > DEACCEL_DISTANCE) {
    desiredPwm = MIN_PWM_THETA3_NEG;
}
currentPwm3 = desiredPwm;
if (pidOutput > 0) {
    analogWrite(RPWM2, currentPwm3);
    analogWrite(LPWM2, 0);
} else {
    analogWrite(RPWM2, 0);
    analogWrite(LPWM2, currentPwm3);
}
lastError3 = error;
} else {
    isRunning3 = false;
    analogWrite(RPWM2, 0);
    analogWrite(LPWM2, 0);
    currentPwm3 = 0;
    noInterrupts();
    encoderCount2 = (-45.0 / DEG_PER_PULSE_DC);
    currentAngle3 = -45.0;
    interrupts();
}
}
Serial.println("Homing đã xong");
}

```

```

void loop() {
    static bool initialMove = true;

    noInterrupts();

```

```

currentAngle1 = (stepCount * DEG_PER_STEP);
long encoderValue1 = encoderCount1;
long encoderValue2 = encoderCount2;
currentAngle2 = (encoderValue1 * DEG_PER_PULSE_DC);
currentAngle3 = (encoderValue2 * DEG_PER_PULSE_DC);
interrupts();

if (!initialMove) {
    if (currentAngle1 > 90.0) {
        stepCount = 90.0 / DEG_PER_STEP;
        currentAngle1 = 90.0;
        isRunning1 = false;
    } else if (currentAngle1 < -90.0) {
        stepCount = -90.0 / DEG_PER_STEP;
        currentAngle1 = -90.0;
        isRunning1 = false;
    }
    if (currentAngle2 > 90.0) {
        encoderValue1 = 90.0 / DEG_PER_PULSE_DC;
        currentAngle2 = 90.0;
        isRunning2 = false;
    } else if (currentAngle2 < 30.0) {
        encoderValue1 = 30.0 / DEG_PER_PULSE_DC;
        currentAngle2 = 30.0;
        isRunning2 = false;
    }
    if (currentAngle3 > 0.0) {
        encoderValue2 = 0.0 / DEG_PER_PULSE_DC; // LIMIT_SWITCH3 = 0°
        currentAngle3 = 0.0;
        isRunning3 = false;
    } else if (currentAngle3 < -90.0) {
        encoderValue2 = -90.0 / DEG_PER_PULSE_DC; // LIMIT_SWITCH6 = -90°
    }
}

```

```

    currentAngle3 = -90.0;
    isRunning3 = false;
}
}
static unsigned long lastSendTime = 0;
if (millis() - lastSendTime >= 100) {
    Serial.print("THETA1:");
    Serial.print(currentAngle1, 1);
    Serial.print(", THETA2:");
    Serial.print(currentAngle2, 1);
    Serial.print(", THETA3:");
    Serial.println(currentAngle3, 1);
    lastSendTime = millis();
}
if (Serial.available() > 0) {
    String command = Serial.readStringUntil('\n');
    command.trim();
    if (command.startsWith("ANGLE1:")) {
        int idx1 = command.indexOf(',');
        int idx2 = command.indexOf(',', idx1 + 1);
        if (idx1 != -1 && idx2 != -1) {
            String angleStr1 = command.substring(7, idx1);
            String angleStr2 = command.substring(idx1 + 1, idx2);
            String angleStr3 = command.substring(idx2 + 1);
            float angles[] = {angleStr1.toFloat(), angleStr2.toFloat(), angleStr3.toFloat()};
            bool isValid[] = {true, true, true};
            for (int i = 0; i < 3; i++) {
                String& str = (i == 0) ? angleStr1 : (i == 1) ? angleStr2 : angleStr3;
                for (unsigned int j = (str[0] == '-' ? 1 : 0); j < str.length(); j++) {
                    if (!isDigit(str[j]) && str[j] != '-') isValid[i] = false;
                }
                if (isValid[i]) {

```

```

float& target = (i == 0) ? targetAngle1 : (i == 1) ? targetAngle2 : targetAngle3;
bool& running = (i == 0) ? isRunning1 : (i == 1) ? isRunning2 : isRunning3;
target = constrain(angles[i], (i == 0 ? -90.0 : (i == 1 ? 30.0 : -90.0)), (i == 0 ? 90.0 : (i
== 1 ? 90.0 : 0.0)));
    running = true;
    if (i == 1) { errorSum2 = 0.0; lastError2 = 0.0; }
    if (i == 2) { errorSum3 = 0.0; lastError3 = 0.0; }
    }
    }
    initialMove = false;
}
} else if (command == "STOP") {
    isRunning1 = isRunning2 = isRunning3 = false;
    digitalWrite(STEP_PULSE, LOW);
    analogWrite(RPWM1, 0);
    analogWrite(LPWM1, 0);
    analogWrite(RPWM2, 0);
    analogWrite(LPWM2, 0);
    currentPwm2 = 0;
    currentPwm3 = 0;
    initialMove = false;
    Serial.println("Stopped");
} else if (command == "HOME") {
    initialMove = true;
    homeMotors();
}
}

if (isRunning1 || isRunning2 || isRunning3) {
    unsigned long currentTime = millis();

    if (isRunning1) {

```

```

float error = targetAngle1 - currentAngle1;

bool limitTriggered1 = (LIMIT_ACTIVE_LOW ? digitalRead(LIMIT_SWITCH1) ==
LOW : digitalRead(LIMIT_SWITCH1) == HIGH);

bool limitTriggered2 = (LIMIT_ACTIVE_LOW ? digitalRead(LIMIT_SWITCH4) ==
LOW : digitalRead(LIMIT_SWITCH4) == HIGH);

if (limitTriggered1 || limitTriggered2) {
    isRunning1 = false;
    digitalWrite(STEP_PULSE, LOW);
    if (limitTriggered1) {
        stepCount = -90.0 / DEG_PER_STEP;
        currentAngle1 = -90.0;
        targetAngle1 = -90.0;
        Serial.println("Hit limit switch THETA1 at -90°");
    } else if (limitTriggered2) {
        stepCount = 90.0 / DEG_PER_STEP;
        currentAngle1 = 90.0;
        targetAngle1 = 90.0;
        Serial.println("Hit limit switch THETA1 at 90°");
    }
} else if (abs(error) > 0.05) {
    digitalWrite(STEP_DIR, error > 0 ? HIGH : LOW);
    int stepsToMove = abs(error) / DEG_PER_STEP;
    unsigned long adjustedMinPulseDelay = MIN_PULSE_DELAY;
    if ((error < 0 && (currentAngle1 - targetAngle1) > DEACCEL_DISTANCE) || (error > 0
&& (targetAngle1 - currentAngle1) > DEACCEL_DISTANCE)) {
        adjustedMinPulseDelay = MAX_PULSE_DELAY;
    } else if (abs(error) <= 10.0) {
        adjustedMinPulseDelay = map(abs(error), 0, 10, MAX_PULSE_DELAY,
MIN_PULSE_DELAY);
    }
    for (int i = 0; i < stepsToMove; i++) {
        unsigned long pulseDelay = (i < ACCEL_STEPS) ? map(i, 0, ACCEL_STEPS - 1,
MAX_PULSE_DELAY, adjustedMinPulseDelay) :

```

```

        (i >= stepsToMove - ACCEL_STEPS) ? map(i, stepsToMove -
ACCEL_STEPS, stepsToMove - 1, adjustedMinPulseDelay, MAX_PULSE_DELAY) :
adjustedMinPulseDelay;

    digitalWrite(STEP_PULSE, HIGH);
    delayMicroseconds(pulseDelay);
    digitalWrite(STEP_PULSE, LOW);
    delayMicroseconds(pulseDelay);
    stepCount += (error > 0 ? 1 : -1);
    currentAngle1 = stepCount * DEG_PER_STEP;
}
} else {
    isRunning1 = false;
    digitalWrite(STEP_PULSE, LOW);
}
}
if (isRunning2) {
    float error = targetAngle2 - currentAngle2;

    bool limitTriggered1 = (LIMIT_ACTIVE_LOW ? digitalRead(LIMIT_SWITCH2) ==
LOW : digitalRead(LIMIT_SWITCH2) == HIGH);

    bool limitTriggered2 = (LIMIT_ACTIVE_LOW ? digitalRead(LIMIT_SWITCH5) ==
LOW : digitalRead(LIMIT_SWITCH5) == HIGH);

    if (limitTriggered1 || limitTriggered2) {
        isRunning2 = false;
        analogWrite(RPWM1, 0);
        analogWrite(LPWM1, 0);
        currentPwm2 = 0;
        if (limitTriggered1) {
            encoderCount1 = 90.0 / DEG_PER_PULSE_DC;
            currentAngle2 = 90.0;
            targetAngle2 = 90.0;
            Serial.println("Hit limit switch THETA2 at 90°");
        } else if (limitTriggered2) {
            encoderCount1 = 30.0 / DEG_PER_PULSE_DC;

```

```

    currentAngle2 = 30.0;
    targetAngle2 = 30.0;
    Serial.println("Hit limit switch THETA2 at 30°");
}
} else if (abs(error) > 0.1) {
    if (error < 0 && currentAngle2 > targetAngle2 + 0.5) {
        errorSum2 = 0.0;
    } else if (error > 0 && currentAngle2 < targetAngle2 - 0.5) {
        errorSum2 = 0.0;
    }
    errorSum2 += error;
    float errorDiff = error - lastError2;
    float pidOutput = Kp * error + Ki * errorSum2 + Kd * errorDiff;
    int minPwm = (pidOutput > 0) ? MIN_PWM_THETA2_POS : MIN_PWM_DC;
    int desiredPwm = constrain(abs(pidOutput), minPwm, MAX_PWM_DC);
    if (error < 0 && (currentAngle2 - targetAngle2) > DEACCEL_DISTANCE) {
        desiredPwm = MIN_PWM_THETA2_POS;
    } else if (error > 0 && (targetAngle2 - currentAngle2) > DEACCEL_DISTANCE) {
        desiredPwm = MIN_PWM_DC;
    }
    currentPwm2 = desiredPwm;
    if (pidOutput > 0) {
        analogWrite(RPWM1, currentPwm2);
        analogWrite(LPWM1, 0);
    } else {
        analogWrite(RPWM1, 0);
        analogWrite(LPWM1, currentPwm2);
    }
    lastError2 = error;
} else {
    isRunning2 = false;
    analogWrite(RPWM1, 0);
}

```

```

    analogWrite(LPWM1, 0);
    currentPwm2 = 0;
}
}
if (isRunning3) {
    float error = targetAngle3 - currentAngle3;

    bool limitTriggered1 = (LIMIT_ACTIVE_LOW ? digitalRead(LIMIT_SWITCH3) ==
LOW : digitalRead(LIMIT_SWITCH3) == HIGH); // 0°

    bool limitTriggered2 = (LIMIT_ACTIVE_LOW ? digitalRead(LIMIT_SWITCH6) ==
LOW : digitalRead(LIMIT_SWITCH6) == HIGH); // -90°

    if (limitTriggered1 || limitTriggered2) {
        isRunning3 = false;
        analogWrite(RPWM2, 0);
        analogWrite(LPWM2, 0);
        currentPwm3 = 0;
        if (limitTriggered1) {
            encoderCount2 = 0.0 / DEG_PER_PULSE_DC;
            currentAngle3 = 0.0;
            targetAngle3 = 0.0;
            Serial.println("Hit limit switch THETA3 at 0°");
        } else if (limitTriggered2) {
            encoderCount2 = -90.0 / DEG_PER_PULSE_DC;
            currentAngle3 = -90.0;
            targetAngle3 = -90.0;
            Serial.println("Hit limit switch THETA3 at -90°");
        }
    } else if (abs(error) > 0.1) {
        if (error < 0 && currentAngle3 > targetAngle3 + 0.5) {
            errorSum3 = 0.0;
        } else if (error > 0 && currentAngle3 < targetAngle3 - 0.5) {
            errorSum3 = 0.0;
        }
    }
    errorSum3 += error;
}
}

```

```

float errorDiff = error - lastError3;
float pidOutput = Kp3 * error + Ki3 * errorSum3 + Kd3 * errorDiff;
int minPwm = (pidOutput > 0) ? MIN_PWM_THETA3_POS :
MIN_PWM_THETA3_NEG;
int desiredPwm = constrain(abs(pidOutput), minPwm, MAX_PWM_DC);

if (error < 0 && (currentAngle3 - targetAngle3) > DEACCEL_DISTANCE) {
    desiredPwm = MIN_PWM_THETA3_POS;
} else if (error > 0 && (targetAngle3 - currentAngle3) > DEACCEL_DISTANCE) {
    desiredPwm = MIN_PWM_THETA3_NEG;
}
currentPwm3 = desiredPwm;
if (pidOutput > 0) {
    analogWrite(RPWM2, currentPwm3);
    analogWrite(LPWM2, 0);
} else {
    analogWrite(RPWM2, 0);
    analogWrite(LPWM2, currentPwm3);
}
lastError3 = error;
} else {
    isRunning3 = false;
    analogWrite(RPWM2, 0);
    analogWrite(LPWM2, 0);
    currentPwm3 = 0;
}
}
}
}

void readEncoder1() {
    if (digitalRead(ENC_A1) == digitalRead(ENC_B1)) {

```

```
    encoderCount1--;  
  } else {  
    encoderCount1++;  
  }  
}  
  
void readEncoder2() {  
  if (digitalRead(ENC_A2) == digitalRead(ENC_B2)) {  
    encoderCount2--;  
  } else {  
    encoderCount2++;  
  }  
}
```