**BỘ GIÁO DỤC VÀ ĐÀO TẠO XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI *Độc lập – Tự do – Hạnh phúc***

***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****

**NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

***1.Thông tin sinh viên thực hiện đồ án***

**Nguyễn Thanh Hải MSSV: 20138602 - Khóa 58 - Lớp NUT12 - Viện SIE**

**Nguyễn Văn Chinh MSSV: 20138050 - Khóa 58 - Lớp NUT12 - Viện SIE**

**Nguyễn Hoàng Phương MSSV: 20138609 - Khóa 58 - Lớp NUT12 - Viện SIE**

***2.Đề tài đồ án***

**“THIẾT KẾ CHẾ TẠO ROBOT 6 BẬC TỰ DO SỬ DỤNG IN 3D”**

***3.Nội dung yêu cầu***

* ***Thiết kế, xây dựng mô hình thực nghiệm robot 6 bậc tự do.***
* ***Thiết kế bộ điều khiển.***
* ***Viết chương trình điều khiển.***

***4.Nội dung thuyết minh***

* ***Tổng quan về robot công nghiệp.***
* ***Tính toán thiết kế chế tạo robot.***
* ***Thiết kế bộ điều khiển.***
* ***Thiết kế chương trình điều khiển.***

***5.Các bản vẽ kĩ thuật***

* ***Bản vẽ lắp.***

**Hà Nội, ngày tháng 06 năm 2018**

**Trưởng Bộ Môn Cán bộ hướng dẫn Sinh viên thực hiện**

***(Ký, ghi rõ họ tên) (Ký, ghi rõ họ tên) (Ký, ghi rõ họ tên)***

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

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

**Hà Nội, ngày tháng 06 năm 2018**

**Giáo viên hướng dẫn**

***(Ký, ghi rõ họ tên)***

**NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN**

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

**Hà Nội, ngày tháng 06 năm 2018**

**Giáo viên phản biện**

***(Ký, ghi rõ họ tên)***

**MỤC LỤC**

[**LỜI MỞ ĐẦU 3**](#_Toc516660024)

[**ĐẶT VẤN ĐỀ 4**](#_Toc516660025)

[**CHƯƠNG 1 : TỔNG QUAN VỀ ROBOT CÔNG NGHIỆP 5**](#_Toc516660026)

[**1. Khái niệm và xu hướng phát triển của Robot công nghiệp 5**](#_Toc516660027)

[**1.1. Khái niệm 5**](#_Toc516660028)

[**1.2. Xu hướng phát triển của Robot công nghiệp 6**](#_Toc516660029)

[**1.3 Cấu trúc chung của Robot công nghiệp 6**](#_Toc516660030)

[**2. Phân loại Robot 7**](#_Toc516660031)

[**2.1. Phân loại theo dạng hình học của không gian hoạt động 8**](#_Toc516660032)

[**2.2. Phân loại theo thế hệ 9**](#_Toc516660033)

[**2.3. Phân loại theo hệ thống truyền động 9**](#_Toc516660034)

[**3. Ứng dụng của Robot công nghiệp 9**](#_Toc516660035)

[**3.1. Ứng dụng Robot trong các thao tác cần khuếch đại lực 9**](#_Toc516660036)

[**3.2. Ứng dụng Robot trong các thao tác phức tạp 10**](#_Toc516660037)

[**3.3. Làm việc trong môi trường khắc nghiệt,độc hại. 11**](#_Toc516660038)

[**CHƯƠNG 2: TÍNH TOÁN THIẾT KẾ CHẾ TẠO ROBOT 12**](#_Toc516660039)

[**Yêu cầu nhiệm vụ thiết kế 12**](#_Toc516660040)

[**1. Tính toán 12**](#_Toc516660041)

[**1.1. Động học 12**](#_Toc516660042)

[**1.1.1. Các tham số động học DH ( Denavit Hartenberg ) 12**](#_Toc516660043)

[**1.1.2. Thiết lập ma trận trạng thái khâu thao tác 16**](#_Toc516660044)

[**1.1.3. Hệ phương trình động học độc lập 18**](#_Toc516660045)

[**1.2. Động học vi phân của Robot 19**](#_Toc516660046)

[**1.2.1. Bài toán động học thuận 19**](#_Toc516660047)

[**1.2.2. Bài toán động học ngược 22**](#_Toc516660048)

[**1.3. Tĩnh học 24**](#_Toc516660049)

[**1.4. Động lực học robot 25**](#_Toc516660050)

[**1.4.1. Ma trận khối lượng 27**](#_Toc516660052)

[**1.4.2. Lực quán tính Coriolis và quán tính li tâm 28**](#_Toc516660053)

[**1.4.3. Thế năng, lực thế 28**](#_Toc516660054)

[**1.4.4. Lực suy rộng của các lực ko thế 29**](#_Toc516660055)

[**2. Thiết kế 29**](#_Toc516660056)

[**2.1. Phương án thiết kế 29**](#_Toc516660057)

[**2.2. Xây dựng mô hình trên Solidworks 30**](#_Toc516660058)

[**3. Chọn vật liệu 33**](#_Toc516660059)

[**3.1. Chọn phương pháp chế tạo 33**](#_Toc516660060)

[**3.2. Chọn động cơ 34**](#_Toc516660061)

[**3.3. Chọn hệ thống dẫn động 37**](#_Toc516660062)

[**3.4. Chọn ổ bi 37**](#_Toc516660063)

[**3.5. Chọn tay gắp 38**](#_Toc516660064)

[**4. Chế tạo và lắp ráp 40**](#_Toc516660065)

[**4.1. Chế tạo 40**](#_Toc516660066)

[**4.2. Lắp ráp 41**](#_Toc516660067)

[**CHƯƠNG 3: THIẾT KẾ BỘ ĐIỀU KHIỂN 42**](#_Toc516660068)

[**1. Mạch Arduino 42**](#_Toc516660069)

[**2. Mạch Điều khiển robot 42**](#_Toc516660070)

[**2.1. Mạch Arduino ATmega 2560 42**](#_Toc516660071)

[**2.2. Driver điều khiển động cơ 44**](#_Toc516660072)

[**2.3. Sơ đồ nối mạch 47**](#_Toc516660073)

[**CHƯƠNG 4: THIẾT KẾ CHƯƠNG TRÌNH ĐIỀU KHIỂN 51**](#_Toc516660074)

[**1. Điều khiển Robot 51**](#_Toc516660075)

[**2. Lập trình điều khiển Robot 53**](#_Toc516660076)

[**2.1. Lập trình trên Andruino 53**](#_Toc516660077)

[**2.2. Lập trình trên Python 53**](#_Toc516660078)

[**2.2.1. Giới thiệu ngôn ngữ Python 53**](#_Toc516660079)

[**2.2.2. Lập trình GUI bằng Tkinter 54**](#_Toc516660080)

[**2.2.3. Xây dựng giao diện chương trình 55**](#_Toc516660081)

[**KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN 60**](#_Toc516660082)

[**1. Kết quả 60**](#_Toc516660083)

[**2. Phương hướng phát triển 60**](#_Toc516660084)

[**PHỤ LỤC 61**](#_Toc516660085)

[**1. Code chương trình 61**](#_Toc516660086)

[**1.1. Code Arduino 61**](#_Toc516660087)

[**1.2. Lập trình giao diện 66**](#_Toc516660088)

[**1.3. Code điều khiển chính 70**](#_Toc516660089)

[**TÀI LIỆU THAM KHẢO 84**](#_Toc516660090)

**DANH MỤC HÌNH ẢNH**

*Hình 1.1 : Các loại tự động hóa công nghiệp 5*

*Hình 1.2 : Sơ đồ cấu trúc chung của Robot công nghiệp 7*

*Hình 1.3 Các thế hệ Robot 9*

*Hình 1.4 Robot bốc dỡ hàng hóa 10*

*Hình 1.5a Robot lắp ráp trong sản xuất ô tô 10*

*Hình 1.5b Robot hàn trong công nghiệp 10*

*Hình 2.1 Sơ đồ trình tự thiết kế 12*

*Hình 2.2 Chiều dài và góc xoắn của một khâu 13*

*Hình 2.3 Các thông số của khâu: θ, d, a và α. 13*

*Hình 2.4 Hệ trục tọa độ của robot trong mô hình hình học 14*

*Hình 2.5 Đồ thị quỹ đạo chuyển động của điểm tác động cuối 19*

*Hình 2.6 Đồ thị vận tốc, gia tốc điểm tác động cuối E 20*

*Hình 2.7 Vị trí robot 21*

*Hình 2.8 Bốn vị trí hình học của Robot 21*

*Hình 2.9 Mô hình Robot 24*

*Hình 2.10 Robot Stanford cấu hình RRTRRR 28*

*Hình 2.11 Robot KUKA cấu hình RRRRRR 28*

*Hình 2.12 Robot Elbow cấu hình RRRRRR 28*

*Hình 2.13 Mô hình 3D trên Solidworks 29*

*Hình 2.14 Khâu 1 29*

*Hình 2.15 Khâu 2 29*

*Hình 2.16 Khâu 3 30*

*Hình 2.17 Khâu 4 30*

*Hình 2.18 Khâu 5 30*

*Hình 2.19 Khâu 6 30*

*Hình 2.20 Robot sau khi liên kết 31*

*Hình 2.21 Động cơ bước 34*

*Hình 2.22 Ổ đũa côn 37*

*Hình 2.23 Khả năng tiếp nhận tải trọng của ổ đũa côn 37*

*Hình 2.24 Các loại tay gắp 39*

*Hình 2.25  Cơ cấu tay kẹp 39*

*Hình 2.26 Động cơ servo 40*

*Hình 2.27 Mô phỏng in 3D 40*

*Hình 2.28 Một số chi tiết sau khi in 3D 41*

*Hình 2.29 Robot sau khi lắp ráp 41*

*Hình 3.1 Một số ứng dụng nổi bật của Arduino 42*

*Hình 3.2 Cấu trúc cơ bản của Mega 2560 42*

*Hình 3.3 Module TB6560 điều khiển động cơ bước 44*

*Hình 3.4 Thông số mạch AT 2560 46*

*Hình 3.5 Sơ đồ khái quát về hệ thống sử dụng động cơ bước 47*

*Hình 3.6 Sơ đồ kết nối giữa TB6560, mạch điều khiển và động cơ bước 48*

*Hình 3.7 Sơ đồ nối mạch điều khiển cánh tay robot 49*

*Hình 3.8 Mạch điều khiển sau khi nối hoàn chỉnh 50*

*Hình 4.1 Sơ đồ điều khiển vòng hở cho robot. 51*

*Hình 4.2 Lưu đồ thuật toán quá trình Robot calibration 52*

*Hình 4.3 Giao diện phần mềm Arduino IDE 53*

*Hình 4.4 : Cửa sổ đơn giản với Python Tkinter 54*

*Hình 4.5 : Giao diện máy tính đơn giản với Python Tkinter 55*

*Hình 4.6 Giao diện điều khiển chính được viết trên Python 55*

*Hình 4.7 Khối chạy / dừng chương trình 56*

*Hình 4.8 Khối điều khiển góc khớp của robot 56*

*Hình 4.9 Chức năng “Jog in steps” 56*

*Hình 4.10 Khối điều khiển vị trí khâu cuối 56*

*Hình 4.11 Khối tùy chỉnh chương trình 57*

*Hình 4.12 Khối điều chỉnh thông số của robot 58*

*Hình 4.13 Khối calibration 58*

# **LỜI MỞ ĐẦU**

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Trong suốt thời gian từ khi bắt đầu học tập ở giảng đường đại học đến nay, em đã nhận được rất nhiều sự quan tâm, giúp đỡ rất nhiều từ quý thầy cô, gia đình và bạn bè

Trước tiên em xin được gửi lời cám ơn và tri ân đến ba mẹ, người đã sinh và nuối em lớn không đên ngà nay, nhờ có họ là niềm phấn đấu vươn lên của em. Em xin gửi lời cảm ơn đến các quý thầy cô trong trường Đại học Bách Khoa Hà Nội cũng như các thầy cô ở viện đào tạo quốc tế SIE và viện Cơ Khí, đã nhiệt tình dạy dỗ và truyền đạt cho em nhiều kiến thức nền tảng để em có được hành trang tốt chuẩn bị bước vào đời. Đặc biệt là em muốn tỏ lòng cảm ơn thật nhiều đến thầy **TS.Hoàng Hồng Hải**, người thầy đã ân cần, tận tụy hướng dẫn em trong suốt thời gian thực hiện đồ án này. Trong quá trình làm việc, với sự hướng dẫn tận tình của thầy **TS.Hoàng Hồng Hải** và nỗ lực của bản thân, đến nay công việc của chúng em đã hoàn thành. Trong quá trình tìm hiểu, tính toán và lập trình có thể có sai sót. Chúng em rất mong sự chỉ bảo của thầy để đề tài được hoàn thiện hơn. Nhờ sự hướng dẫn tận tình của thầy mà em đã hoàn thành tốt đồ án tốt nghiệp này, một lần nữa cho em gửi lời cảm ơn sâu sắc đến thầy, người lái đò sống mãi với sự nghiệp giáo dục, dìu dắt những thế hệ trẻ trên con đường học tập và nghiên cứu khoa học.

Sau cùng, em xin kính chúc quý thầy cô trong viện Cơ Khí và thầy **TS.Hoàng Hồng Hải** thật dồi dào sức khỏe, niềm tin để tiếp tục thực hiện sứ mệnh cao đẹp của mình là truyền đạt kiến thức cho thế hệ mai sau.

**Hà Nội, ngày tháng 06 năm 2018**

**Sinh viên thực hiện**

Nguyễn Thanh Hải

Nguyễn Văn Chinh

Nguyễn Hoàng Phương

# **ĐẶT VẤN ĐỀ**

Ngày nay, Robot không còn là cái gì đó quá xa lạ với mọi người. Chúng dần dần được giới kỹ thuật hình dung như những chiếc máy đặc biệt, được con người phỏng tác theo cấu tạo và hoạt động của chính mình, dùng để thay thế mình trong một số công việc xác định. Để hoàn thành nhiệm vụ đó, Robot cần có khả năng cảm nhận các thông số trạng thái của môi trường và thực hiện các động tác tương tự nhưcon người.

Khả năng hoạt động của Robot được đảm bảo bởi hệ thống cơ khí, gồm cơ cấu vận động để đi lại và cơ cấu hành động để có thể làm việc. Việc thiết kế và chế tạo hệ thống này thuộc lĩnh vực khoa học về cơ cấu truyền động, chấp hành và vật liệu cơ khí.

 Chức năng cảm nhận, gồm thu nhận tín hiệu về trạng thái môi trường và trạng thái bản thân hệ thống, do các cảm biến (sensor) và các thiết bị liên quan thực hiện. Hệ thống này được gọi là hệ thống thu nhận và xử lý tín hiệu, hay đơn giản là hệ thống cảm biến.

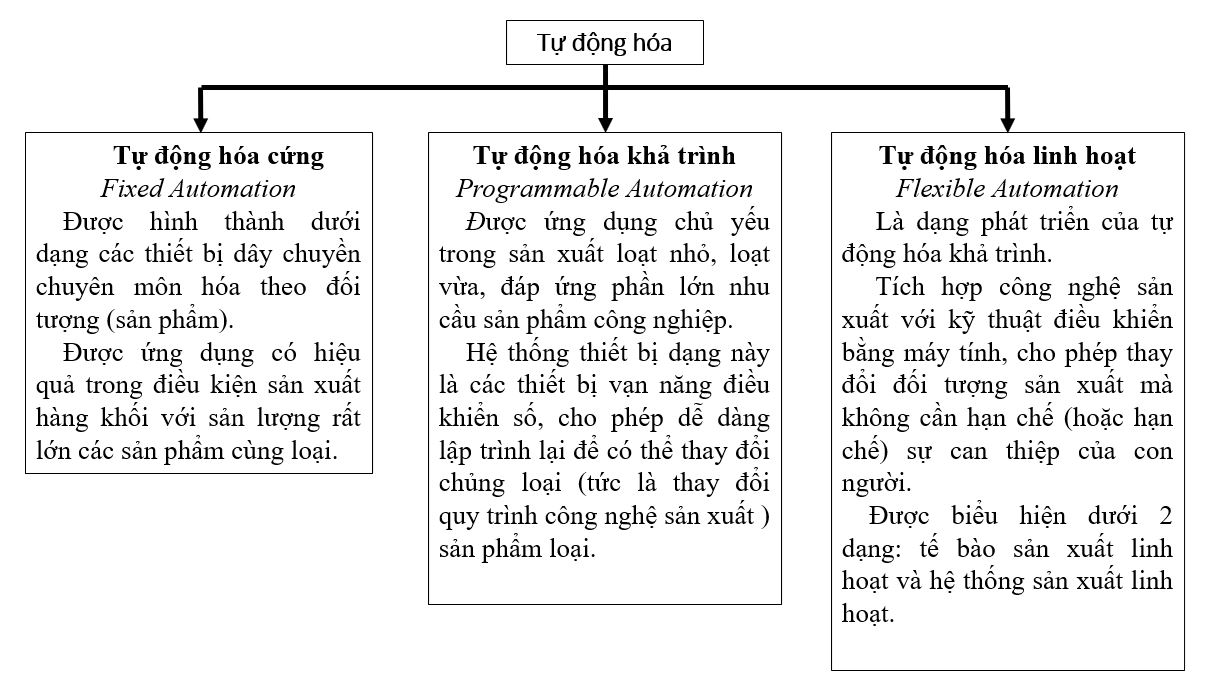
Một cách đơn giản, Robotics được hiểu là một ngành khoa học có nhiệm vụ nghiên cứu về thiết kế, chế tạo các Robot và ứng dụng chúng trong các lĩnh vực hoạt động khác nhau của xã hội loài người, như nghiên cứu khoa học - kỹ thuật, kinh tế, quốc phòng và dân sinh.

Ngày nay, khái niệm về Robot đã mở rộng hơn khái niệm nguyên thủy rất nhiều. Sự phỏng tác về kết cấu, chức năng, dáng vẻ của con người là cần thiết nhưng không còn ngự trị trong kỹ thuật Robot nữa. Kết cấu của nhiều con Robot khác xa với kết cấu các bộ phận của cơ thể người và chúng cũng có thể thực hiện được những việc vượt xa khả năng của con người.

# **CHƯƠNG 1 : TỔNG QUAN VỀ ROBOT CÔNG NGHIỆP**

1. **Khái niệm và xu hướng phát triển của Robot công nghiệp**
   1. **Khái niệm**

Mặc dù, như định nghĩa chung về Robot đã nêu, không có gì giới hạn phạm vi ứng dụng của Robot, nhưng có một thực tế là hầu hết Robot hiện đang có đều được dùng trong công nghiệp. Chúng có đặc điểm riêng về kết cấu, chức năng đã thống nhất hóa, thương mại hóa rộng rãi. Lớp Robot này được gọi là Robot công nghiệp.

Kỹ thuật tự động hóa trong công nghiệp đã đạt tới trình độ rất cao: không chỉ tự động hóa các quá trình vật lý mà cả quá trình xử lý thông tin. Vì vậy, tự động hóa trong công nghiệp tích hợp công nghệ sản xuất, kỹ thuật điện, điện tử, kỹ thuật điều khiển tự động trong đó có tự động hóa nhờ máy tính.

*Hình 1.1 Các loại tự động hóa công nghiệp*

Robot công nghiệp có 2 đặc trưng cơ bản:

* Là thiết bị vạn năng, được tự động hóa theo chương trình và có thể lập trình lại để đáp ứng một cách linh hoạt, khéo léo các nhiệm vụ khác nhau.
* Được ứng dụng trong những trường hợp mang tính công nghiệp đặc trưng như vận chuyển và xếp dỡ nguyên vật liệu, lắp ráp, đo lường,…

Do thể hiện 2 đặc trưng cơ bản trên của Robot công nghiệp, hiện nay định nghĩa sau đây về Robot công nghiệp do viện nghiên cứu Robot của Mỹ đề xuất được sử dụng rộng rãi:

*Robot công nghiệp là tay máy vạn năng, hoạt động theo chương trình và có thể lập trình lại để hoàn thành và nâng cao hiệu quả hoàn thành các nhiệm vụ khác nhau trong công nghiệp, như vận chuyển nguyên vật liệu, chi tiết, dụng cụ hoặc các thiết bị chuyên dùng khác.*

Hiện nay nhu cầu sử dụng Robot trong công nghiệp ngày càng tăng , các loại Robot được chế tạo ngày càng đa dạng , độ chính xác cao hơn, linh hoạt hơn, giá cả phù hợp hơn, năng suất và tuổi thọ cao hơn. Mặc dù dải ứng dụng của Robot ngày càng mở rộng, nhưng theo thống kê vào năm 2000 tại Mỹ , lượng Robot hàn và xử lý phôi chiếm khoảng 78% tổng số lượng Robot công nghiệp sử dụng tại thời điểm đó. Số lượng Robot lắp ráp chiếm khoảng 10% , phần còn lại là của các loại Robot công nghiệp khác.

* 1. **Xu hướng phát triển của Robot công nghiệp**

Nhằm mục tiêu thiết kế, chế tạo các thế hệ Robot công nghiệp ngày càng thông minh, linh hoạt, nhỏ gọn, tin cậy, chính xác, tuổi thọ cao... một số vấn đề sau đang được các nhà khoa học quan tâm giải quyết :

* Tối ưu hóa cấu trúc cơ khí với việc chú ý tới việc sử dụng vật liệu nhẹ, độ bền cao
* Các bài toán cơ học : động hoc, động lực học-điều khiển, cân bằng…
* Các cơ cấu dẫn động và cảm nhận tín hiệu cũng là vấn đề quan tâm lớn trong kỹ thuật Robot.
* Điều khiển thông minh là hướng phát triển lớn của kỹ thuật Robot, thu hút nhiều sự quan tâm của nhiều nhà khoa học trên thế giới.

### **1.3 Cấu trúc chung của Robot công nghiệp**

Mọi RBCN được cấu thành bởi các hệ thống chính sau:

* Tay máy (manipulator) là cơ cấu cơ khí gồm các khâu, khớp. Chúng hình thành cánh tay để tạo các chuyển động cơ bản, cổ tay tạo nên sự khéo léo, linh hoạt và bàn tay (end effector) để trực tiếp hoàn thành các thao tác trên đối tượng.
* Cơ cấu chấp hành tạo chuyển động cho các khâu của tay máy. Nguồn động lực của các cơ cấu chấp hành là động cơ các loại: điện, thủy lực, khí nén hoặc kết hợp giữa chúng.
* Hệ thống cảm biến gồm các sensor và thiết bị chuyển đổi tín hiệu cần thiết khác. Các Robot cần hệ thống sensor trong để nhận biết trạng thái của bản thân các cơ cấu của Robot và các sensor ngoài để nhận biết trạng thái của môi trường.
* Hệ thống điều khiển (controller) và giao diện người dùng hiện nay thường là máy tính để giám sát và điều khiển hoạt động của Robot.



*Hình 1.2 Sơ đồ cấu trúc chung của Robot công nghiệp*

Tay máy là phần cơ sở, quyết định khả năng làm việc của RBCN. Đó là thiết bị cơ khí đảm bảo cho Robot khả năng chuyển động trong không gian và khả năng làm việc như nâng hạ vật, lắp ráp,…Ban đầu,việc thiết kế và chế tạo tay máy là phỏng tác cấu tạo và chức năng của tay người. Về sau, đây không còn là điều bắt buộc nữa. Tay máy hiện nay rất đa dạng và có nhiều loại có dáng vẻ khác rất xa với tay người. Tuy nhiên, trong kỹ thuật Robot người ta vẫn dùng các thuật ngữ quen thuộc như: vai, cánh tay, cổ tay, bàn tay và các khớp… để chỉ tay máy và các bộ phận của nó.

  Trong thiết kế và sử dụng tay máy, người ta quan tâm đến các thông số có ảnh hưởng lớn đến khả năng làm việc của chúng như:

-Sức nâng, độ cứng vững, lực kẹp của tay,…

-Tầm với hay vùng làm việc: kích thước và hình dạng vùng mà phần công tác có thể với tới.

-Sự khéo léo, nghĩa là khả năng định vị và định hướng phần công tác trong vùng làm việc. Thông số này liên quan đến số bậc tự do của phần công tác.

  Ngoài ra để định vị và định hướng phần công tác một cách tùy ý trong không gian 3 chiều cần có 6 bậc tự do, trong đó 3 bậc tự do để định vị trí, 3 bậc tự do để định hướng.

1. **Phân loại Robot**

Trong công nghiệp người ta sử dụng những đặc điểm khác nhau cơ bản nhất của Robot để giúp cho việc nhận biết được dễ dàng. Có bốn yếu tố chính để phân loại Robot như sau :

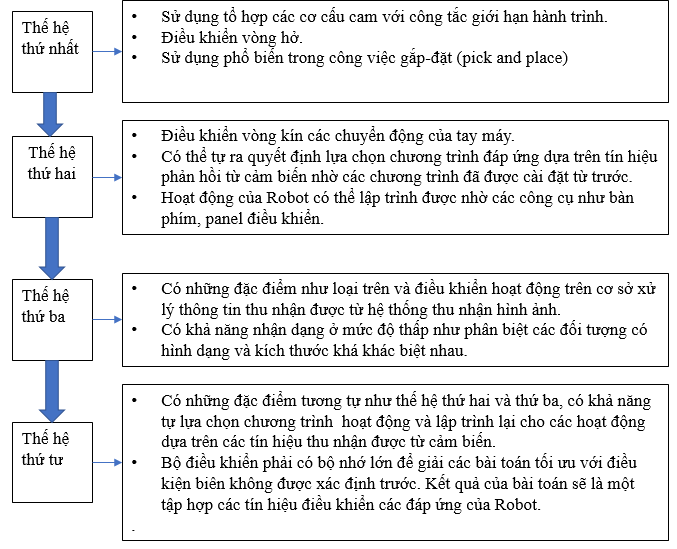
* 1. **Phân loại theo dạng hình học của không gian hoạt động**

|  |  |  |  |
| --- | --- | --- | --- |
| Robot tọa độ góc (cartesian Robot) | Robot tọa độ trụ  (cylindrical Robot) | Robot toa độ cầu (spherical Robot) | Robot khớp bản lề (articular Robot) |
|  |  |  |  |
| Sử dụng 3 đường trượt vuông góc nhau trong không gian 3 trục tọa độ x, y, z. Robot sử dụng các khớp tịnh tiến chuyển động theo 3 trục tọa độ, phạm vi làm việc của Robot mở rộng theo hình chữ nhật. | Trục cơ bản là trục dọc, Robot di chuyển lên và xuống dọc theo trục, quay quanh trục. Phạm vi làm việc của Robot được mở rộng theo 1 hình trụ quanh trục cơ bản. | Sử dụng các khớp lồng vào nhau, giúp Robot có khả năng chuyển động lên hoặc xuống theo chiều ngang trục quay. Dạng đa khớp nối cho phép Robot mở rộng không gian làm việc theo hình cầu. | Đó là một kiểu tay máy có cấu tạo đặc biệt gồm 2 khớp quay và 1 khớp trượt, nhưng cả 3 khớp đều có trục song song với nhau. |
| Do sự đơn giản về kết cấu, tay máy kiểu này có độ cứng vững cao, độ chính xác được đảm bảo đồng đều trong toàn bộ vùng làm việc nhưng ít khéo léo. Vì vậy,tay máy kiểu tọa độ góc được dùng để vận chuyển lắp ráp. | Độ cứng vững cơ học của tay máy trụ tốt thích hợp với tải nặng, nhưng độ chính xác định vị góc trong mặt phẳng nằm ngang giảm khi tầm với tăng. | Độ cứng vững của tay máy này thấp hơn 2 loại trên và độ chính xác định vị phụ thuộc vào tầm với. Tuy nhiên loại này có thể nhặt được vật dưới nền. | Kết cấu này làm tay máy cứng vững hơn theo phương đứng nhưng kém cứng vững theo phương ngang. Loại này chuyên dùng cho công việc lắp ráp với tải trọng nhỏ theo phương đứng. |

*Bảng 1.1 Các dạng Robot theo hình học không gian*

* 1. **Phân loại theo thế hệ**

Theo quá trình phát triển của Robot, ta có thể chia ra theo các thế hệ sau đây:



*Hình 1.3 Các thế hệ Robot*

* 1. **Phân loại theo hệ thống truyền động**

- Robot truyền động điều khiển bằng điện.

- Robot truyền động điều khiển bằng khí nén.

- Robot truyền động điều khiển bằng thủy lực.

- Robot truyền động điều khiển hỗn hợp

1. **Ứng dụng của Robot công nghiệp**
   1. **Ứng dụng Robot trong các thao tác cần khuếch đại lực**

Không như con người,Robot hoàn toàn không biết mệt.Vì vậy,một ứng dụng phổ biến nhất của Robot là để bốc dỡ hàn hóa, vật liệu, phôi có trọng lượng lớn, cồng kềnh trong các ngành công nghiệp nặng. Robot loại  này có thể nâng tải trọng lên đến tối đa một tấn một cách dễ dàng với độ chính xác vị trí nhỏ hơn 1mm



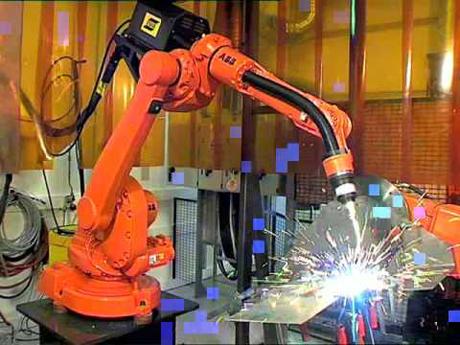
*Hình 1.4 Robot bốc dỡ hàng hóa*

* 1. **Ứng dụng Robot trong các thao tác phức tạp**

Con người cần 1 khoảng thời gian rất lâu để thành thạo 1 công việc gồm nhiều các thao tác phức tạp trong khi Robot có thể học được toàn bộ kiến thức đó chỉ trong vòng một vài giờ đồng hồ.Việc thay thế con người trong các công việc như vậy là một ứng dụng khá phổ biến hiện nay,điển hình như trong rắp láp chi tiết, hàn,…

Việc lắp ráp liên quan đến nhiều xử lý khác nhau: đưa một chi tiết vào một chi tiết kia, đặt một chi tiết trên một chi tiết khác, siết chặt đai ốc, siết vít, hay phun keo, v.v... Khi đã có một trình tự lắp ráp nhất định,việc thực thi nó làm sao để đạt năng suất cao nhất lại là một vấn đề khó.Trong khi người công nhân cần nhiều thời gian để được đào tạo và làm quen với các thao tác lắp ráp chi tiết thì Robot chỉ cần 1 lần “hướng dẫn” là có thể bắt đầu làm việc với hiệu suất cao nhất.Do đó việc ứng dụng Robot trong các dây chuyền sản xuất ko còn gì xa lạ trên thế giới.

Trong các nhà máy sản xuất xe hơi thì hàn điểm là công việc sử dụng Robot nhiều nhất: khung xe được cố định vào một xe được điều khiển từ xa đi khắp nhà máy. Khi xe đến trạm hàn, kẹp sẽ cố định các chi tiết đúng vào vị trí cần hàn, trong khi đó Robot di chuyển dọc theo các điểm hàn được lập trình trước.



*Hình 1.5a Robot lắp ráp trong sản xuất ô tô 1.5b Robot hàn trong công nghiệp*

* 1. **Làm việc trong môi trường khắc nghiệt,độc hại.**

Robot được ứng dụng rộng rãi trong nhiều ngành công nghiệp. Một trong số đó là nhiệt luyện và phun sơn…

Một trong những công việc kém năng suất nhất của con người là rèn luyện kim loại. Đó là do yêu cầu công việc đòi hỏi làm việc trong điều kiện nhiệt độ cao.Nhưng đây lại không phải là vấn đề đối với Robot công nghiệp.Robot ban đầu đã được sử dụng để thay thế công nhân làm việc trong điều kiện môi trường ngặt nghèo như trong lò đúc, xưởng rèn.

Sơn là một công việc nặng nhọc và độc hại đối với sức khỏe của con người, nhưng lại hoàn toàn không nguy hiểm đối với Robot.

**Nhận xét** :

Trên thế giới,Robot ngày càng được sử dụng rộng rãi trong các ngành công nghiệp.Với trình độ phát triển của khoa học công nghệ hiện nay,khả năng của Robot là gần như bất tận.Các nhà khoa học đang ngày đêm nghiên cứu để có thể phát minh ra các loại Robot chuyên dụng có thể thay thế con người trong nhiều lĩnh vực khác nhau.Việc phát triển và ứng dụng Robot vào sản xuất là một trong những vấn đề lớn của nhiều quốc gia trên thế giới. Tuy nhiên việc chế tạo và điều khiển một robot thực tế không hề đơn giản. Để đánh giá cũng như xác định độ phức tạp đó cần phải có những nghiên cứu kỹ lưỡng từ những mô hình robot đơn giản nhất. Đây cũng chính là nội dung mà đồ án tốt nghiệp đề cập tới.

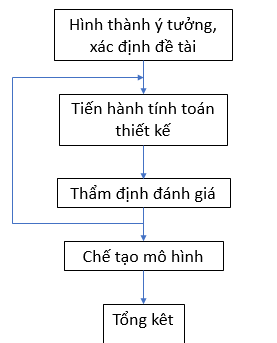
# **CHƯƠNG 2: TÍNH TOÁN THIẾT KẾ CHẾ TẠO ROBOT**

## Yêu cầu nhiệm vụ thiết kế

Với đề tài được chọn là Robot 6 bậc tự do có nhiệm vụ là gắp các sản phẩm được định trước. Như vậy ta có chu trình làm việc của robot như sau: sau khi nhận được tín hiệu bắt đầu hoạt động ( có sản phẩm tới ), robot sẽ gắp sản phẩm bỏ lên các khay.

Để gắp được 1 vật robot phải thực hiện các chuyển động sau: bao gồm chuyển động xoay thân tới phía có vật, gập thân tiếp cận vật cần gắp, kẹp vật, nâng thân lên, xoay về vị trí khay, gập thân xuống và thả vật vào trong khay chứa.

**Trình tự thiết kế**



*Hình 2.1 Sơ đồ trình tự thiết kế*

1. **Tính toán**
   1. **Động học**
      1. **Các tham số động học DH ( Denavit Hartenberg )**

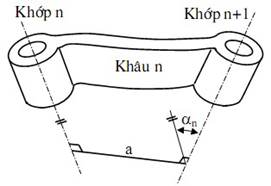
Một Robot nhiều khâu cấu thành từ các khâu nối tiếp nhau thông qua các khớp động. Gốc chuẩn (Base) của một Robot là khâu số 0 và không tính vào số các khâu. Khâu 1 nối với khâu chuẩn bởi khớp 1 và không có khớp ở đầu mút của khâu cuối cùng. Bất kỳ khâu nào cũng được đặc trưng bởi hai kích thước:

Độ dài pháp tuyến chung: an.

Góc giữa các trục trong mặt phẳng vuông góc với an : αn.

Thông thường, người ta gọi an là chiều dài và αn là góc xoắn của khâu. Phổ biến là hai khâu liên kết với nhau ở chính trục của khớp.

Mỗi trục sẽ có hai pháp tuyến với nó, mỗi pháp tuyến dùng cho mỗi khâu (trước và sau một khớp). Vị trí tương đối của hai khâu liên tiếp như thế được xác định bởi dn là khoảng cách giữa các pháp tuyến đo dọc theo trục khớp n và θn là góc giữa các pháp tuyến đo trong mặt phẳng vuông góc với trục. dn và θn thường được gọi là khoảng cách và góc giữa các khâu.



*Hình 2.2 Chiều dài và góc xoắn của một khâu*

**

*Hình 2.3 Các thông số của khâu: θ, d, a và α.*

Để mô tả mối quan hệ giữa các khâu ta gắn vào mỗi khâu một hệ tọa độ. Nguyên tắc chung để gắn hệ tọa độ lên các khâu như sau:

Gốc của hệ tọa độ gắn lên khâu thứ n đặt tại giao điểm của pháp tuyến an với khớp thứ n+1. Trường hợp hai trục khớp cắt nhau, gốc tọa độ sẽ đặt tại chính điểm cắt đó. Nếu các trục khớp song song với nhau, gốc tọa độ được chọn trên trục khớp của khâu kế tiếp, tại điểm thích hợp.

Trục z của hệ tọa độ gắn lên khâu thứ n đặt dọc theo trục khớp thứ n+1.

Trục x thường được đặt dọc theo pháp tuyến chung và hướng từ khớp n

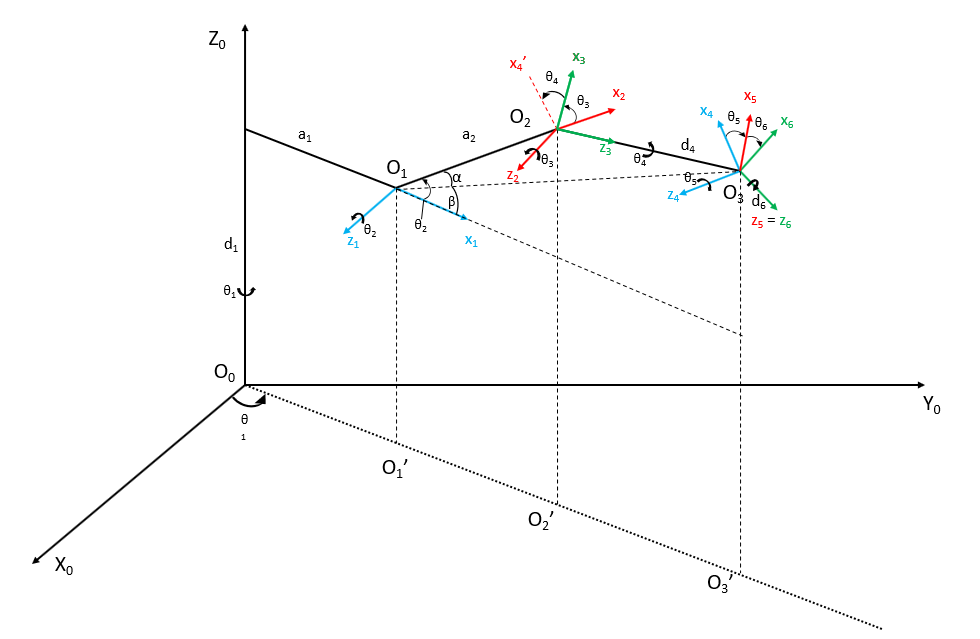
đến n+1.Trong trường hợp các trục khớp cắt nhau thì trục x chọn theo tích vectơ

Trường hợp khớp quay thì θn là các biến khớp, trong trường hợp khớp tịnh tiến

thì dn là biến khớp và an bằng 0.

**->** Các thông số an, αn, dn và θn được gọi là bộ thông số DH.

Thiết lập các tham số động học Denavit – Hartenberg



*Hình 2.4 Hệ trục tọa độ của robot trong mô hình hình học*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Khâu** |  |  |  |  |
| **1** |  |  |  |  |
| **2** |  |  |  |  |
| **3** |  |  | 0 |  |
| **4** |  |  | 0 |  |
| **5** |  | 0 | 0 |  |
| **6** |  |  | 0 | 0 |

*Bảng 2.1 Bảng Denavit Hartenberg các tham số động học của robot*

* Trong đó  là các biến khớp, còn  là các hằng số.

Và  là vector biểu diễn vị trí của bàn kẹp trong hệ cố định.

với  là các góc xoay các biến khớp.

* Dạng tổng quát của ma trận truyền biến đổi tọa độ thuần nhất Denavit – Hartenberg cho các khâu:

** (2-1)

* Ma trận truyền biến đổi tọa độ thuần nhất Denavit – Hartenberg của hệ tọa độ  đối với hệ tọa độ cố định  (hệ 0):

 (2-2)

* Ma trận truyền biến đổi tọa độ thuần nhất Denavit – Hartenberg của hệ tọa độ  đối với hệ tọa độ:

 (2-3)

* Ma trận truyền biến đổi tọa độ thuần nhất Denavit – Hartenberg của hệ tọa độ  đối với hệ tọa độ:

 (2-4)

* Ma trận truyền biến đổi tọa độ thuần nhất Denavit – Hartenberg của hệ tọa độ  đối với hệ tọa độ:

 (2-5)

* Ma trận truyền biến đổi tọa độ thuần nhất Denavit – Hartenberg của hệ tọa độ  đối với hệ tọa độ:

 (2-6)

* Ma trận truyền biến đổi tọa độ thuần nhất Denavit – Hartenberg của hệ tọa độ  đối với hệ tọa độ:

 (2-7)

* + 1. **Thiết lập ma trận trạng thái khâu thao tác**

Từ các ma trận (2-2) đến (2-7) ta xác định được ma trận biến đổi tọa độ thuần nhất *0* biểu diễn trạng thái khâu thao tác:

 (2-8)

Ma trận  cho ta biết hướng và vị trí của khâu thao tác trong hệ tọa độ cố định hay nói cách khác là vị trí của điểm tác động cuối và hướng của hệ tọa độ động gắn vào khâu tại điểm tác động cuối trong hệ tọa độ cố định. Vì thế nó còn được biểu diễn qua thông số các biến khớp ta tạm gọi là . Trong bài toán cụ thể thì nó là các khớp xoay , với .Khi đó, ma trận (3) được kí hiệu thành 

 (2-9)

Với

***c11(q)*** = (((C1\*C2\*C3-C1\*S2\*S3)\*C4 + S1\*S4)\*C5+(-C1\*C2\*S3-C1\*S2\*C3)\*S5)\*C6+ (-(C1\*C2\*C3-C1\*S2\*S3)\*S4+S1\*C4)\*S6

***c12(q)*** =(((S1\*C2\*C3-S1\*S2\*S3)\*C4-C1\*S4)\*C5 + (-S1\*C2\*S3-S 1\*S2\*C3)\*S5)\*C6+(-(S1\*C2\*C3-S1\*S2\*S3)\*S4-C 1\*C4)\*S6

c13***(q)*** = ((C1\*C2\*C3-C1\*S 2\*S 3)\*C4 +S1\*S4)\*S5-(-C1\*C2\*S3-C1\*S2\*C3)\*C5

***c21(q)*** = (((S1\*C2\*C3-S1\*S2\*S3)\*C4+ C1\*S4)\*C5+(S1\*C2\*S3+ S1\*S2\*C3)\*S5)\*C6 +(-(S1\*C2\*C3-S1\*S2\*S3)\*S4+C1\*C4)\*S6

***c22(q)*** =-(((S1\*C2\*C3-S1\*S2\*S3)\*C4-C1\*S4)\*C5+(-S1\*C2\*S3-S1\*S2\*C3)\*S5)\*S6+ (-(S1\*C2\*C3-S1\*S2\*S3)\*S4-C1\*C4)\*C6

***c23(q)*** = -((S1\*C2\*C3-S1\*S2\*S3)\*C4+ C1\*S4)\*S5+(S1\*C2\*S3+S1\*S2\*C3)\*C5

***c31(q)*** = ((S2\*C3+C2\*S3)\*C4\*C5+(-S2\*S3+ C2\*C3)\*S5) \* C6-(S2\*C3+ C2\*S3)\*S4\*S6

***c32(q)*** = -((S2\*C3+C2\*S3)\*C4\*C5+(-S2\*S3+ C2\*C3)\*S5)\*S6-(S2\*C3+C2\*S3)\*S4\*C6

***c33(q)*** = (S2\*C3+C2\*S3)\*C4\*S5-(-S2\*S3+ C2\*C3)\*C5

***x(q)*** = (((C1\*C2\*C3-C1\*S2\*S3)\*C4+S1\*S4)\*S5-(-C1\*C2\*S3-C1\*S2\*C3)\*C5)\*d6+ (C1\*C2\*S3+C1\*S2\*C3)\*d4+C1\*C2\*a2+C1\*a1

=(C1\*C23\*C4+S1\*S4)\*S5+C1\*S23\*C5\*d6+C1\*S23\*d4

+C1\*C2\*a2+C1\*a1

***y(q)*** = (((S1\*C2\*C3-S1\*S2\*S3)\*C4-C1\*S4)\*S5- (-S1\*C2\*S3-S1\*S2\*C3)\*C5)\*d6+ (S1\*C2\*S3+S1\*S2\*C3)\*d4+S1\*C2\*a2+S1\*a1

=(S1\*C23\*C4-C1\*S4)\*S5+S1\*S23\*C5\*d6+S1\*S23\*d4+S1\*C2\*a2+S1\*a1

***z(q)*** = ((S2\*C3+C2\*S3)\*C4\*S5- (-S2\*S3+ C2\*C3)\*C5)\*d6+ (S2\*S3-C2\*C3)\*d4+S2\*a2+d1  
=S23\*C4\*S5 +C23\*C5\*d6-C23\*d4+S2\*a2+d1

Sử dụng các góc Cardan xác định hướng của vật rắn, gọi  là giá trị mô tả trực tiếp vị trí và hướng của  so với hệ tọa độ . Trong đó:  là các tọa độ điểm tác động cuối *E* và là các góc quay Cardan của  so với hệ tọa độ .   
Do các tọa độ thao tác đều là hàm của thời gian. Nên ta có thể biểu diễn:

** (2-10)

Với:

** là ma trận Cardan mô tả hướng  so với hệ tọa độ .

** là vectơ mô tả vị trí của điểm tác động cuối trong hệ tọa độ .

 (2-11)

 (2-12)

* Do ma trận  biểu diễn vị trí và hướng của khâu thao tác trong hệ tọa độ cố định thông qua biến khớp  (Ma trận trạng thái khâu thao tác theo cấu trúc động học). Còn ma trận  cũng mô tả vị trí và hướng của khâu thao tác thông qua hệ tọa độ khâu thao tác. Ở đây ta chọn cách biểu diễn thông qua các góc Cardan.

Từ đó, ta có phương trình động học ROBOT có dạng:



* + 1. **Hệ phương trình động học độc lập**

Từ các hệ thức (2-9), (2-10), (2-11), (2-12) và (2-13), ta xây dựng được hệ 6 phương trình độc lập như sau:

 (2.14)

* 1. **Động học vi phân của Robot**
     1. **Bài toán động học thuận**

Ta có tọa độ của điểm tác động cuối:

 (2-15)

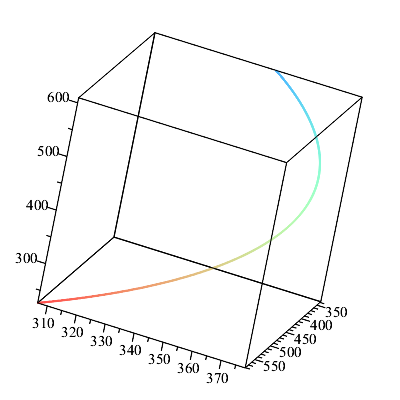
1. **Xây dựng quy luật chuyển động từng khâu từ đó vẽ quỹ đạo, vận tốc và gia tốc điểm E**

* Đưa vào các thông số hình học:



Chọn 

* Để thuận tiện cho việc tính toán ta sử dụng phần mềm tính toán **maple**. Thay các giá trị  và các thông số hình học vào biểu thức của 
* Ta có thể vẽ đồ thị của điểm thao tác cuối qua các tọa độ như sau:

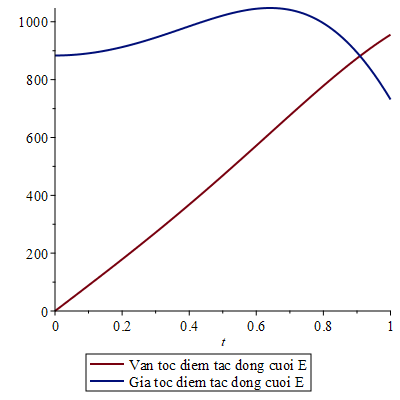


*Hình 2.5 Đồ thị quỹ đạo chuyển động của điểm tác động cuối trong không gian.*

* Ta tính được: 



Dùng maple ta vẽ được các đồ thị sau:



*Hình 2.6 Đồ thị vận tốc, gia tốc điểm tác động cuối E*

1. **Xác định vị trí điểm tác động cuối E và vận tốc góc khâu thao tác EE.**

* Từ phương trình động học thuận, ta có vị trí điểm tác động cuối *E*: (với



****

****

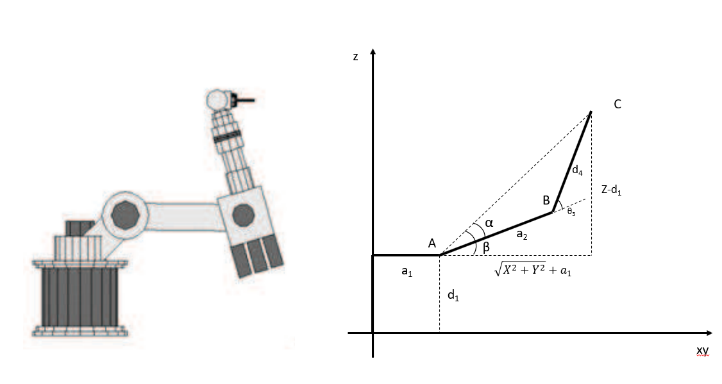
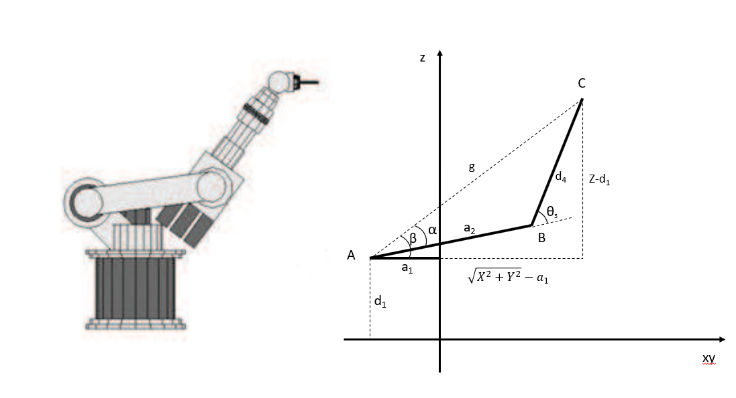
Với 



* + 1. **Bài toán động học ngược**

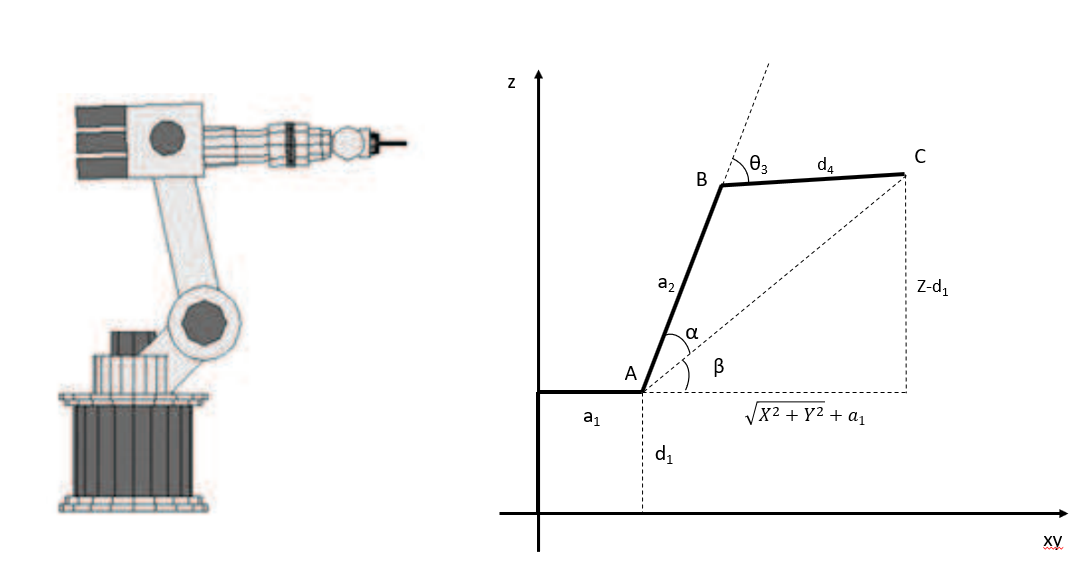


*Hình 2.7 Vị trí robot*Điểm A,B,C lần lượt là vị trí khớp 2,3 6.ta có thể dễ dàng tính được ngay góc q1

q1 = attan2(Y,X) hoặc q1 = attan2(Y,X)+ π

b)

a)



d)

c)

*Hình 2.8  Bốn vị trí hình học của robot*

Góc q2 và q3 có thể tính được theo từ các công thức sau  
a)   b) 

c)   d) 

với  ;  (a , b : + ; c , d : -)

 (a , b : - ; c , d :+)

Để tính 3 góc còn lại ta sử dụng 3 góc RPY(Roll-Pitch-Yaw)



Ta tính





So sánh 2 ma trận và ta tính được các góc 

\* 

\*< 0 

* 1. **Tĩnh học**

Dữ kiện đầu bài

|  |  |
| --- | --- |
| Thông số | Giá trị |
| m1 | 3 kg |
| m2 | 2 kg |
| m3 | 1 kg |
| m4 | 0.75 kg |
| m5 | 0.7 kg |
| m6 | 0.5 |

*Bảng 2.2 Các tham số động học*

Cơ sở lí thuyết

Tính lực (mô-men) dẫn động tại các khớp đảm bảo robot cân bằng tĩnh. Hệ phương trình cân bằng trong hệ tọa độ cơ sở:



Trong đó:

 là lực do khâu i-1 tác dụng lên khâu i ở khớp thứ i trong hệ tọa độ cơ bản.

 là mô-men do khâu i-1 tác dụng lên khâu i ở khớp thứ i trong hệ tọa độ cơ bản.

 là trọng lực của khâu i trong hệ tọa độ cơ bản,

 là véc-tơ có gốc là Oo nối với Oi trong hệ tọa độ cơ bản.

 là ma trận quay biến đổi từ hệ tọa độ 0 sang hệ tọa độ thứ i.

 là véc-tơ có gốc Oi-1 nối với Oi trong hệ tọa độ khâu i.

Tính toán

Ta giả sử lực tác động khâu cuối coi như bằng không:

, 

Lực tĩnh của từng khâu

, ,,,, ,

Momen tĩnh động cơ cần để robot cân bằng:

Với các số liệu: 

Ta thu được các kết quả:

(N.m) (N.m)

(N.m) (N.m)

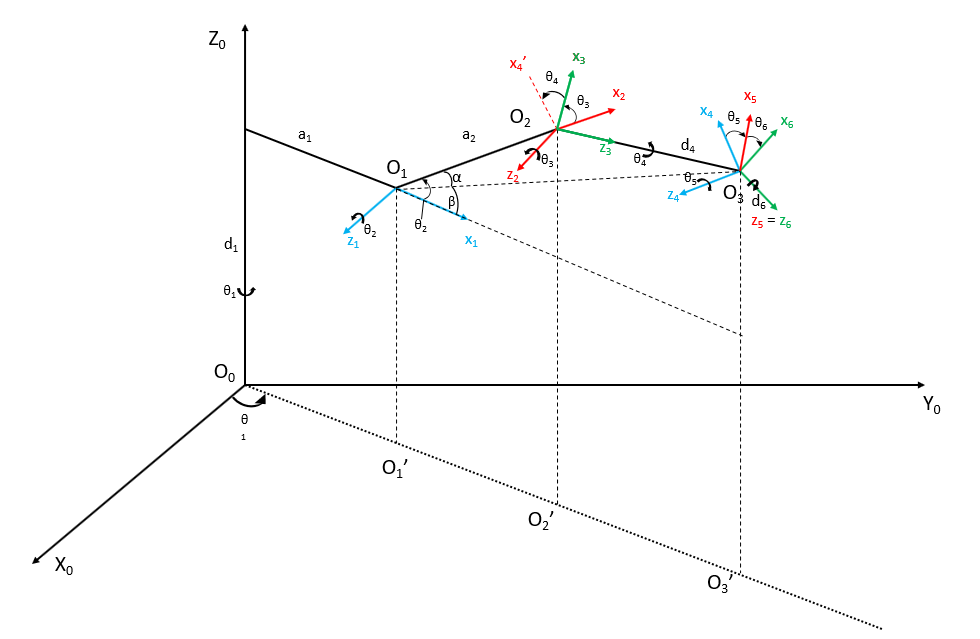
(N.m) (N.m)

* 1. **Động lực học robot**

**Xây dựng cấu trúc động lực học và các thành phần cần thiết để viết phương trình động lực học**

Các tham số động lực học sau xét trên hệ trục gắn với khâu (coi các khâu là thanh mảnh đồng chất các biểu thức I đối với khối tâm  trong hệ  song song với hệ tọa độ khâu).





*Hình 2.9 Mô hình Robot*

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Khâu | Vị trí trọng tâm | | | Khối lượng | Ma trận mômen quán tính | | | | | |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |

*Bảng 2.3 Chiều dài và momen quán tính các khâu*

Ta có phương trình Lagrange loại II :

Dạng ma trận : 

Với 

 : Ma trận khối lượng

Ri: ma trận cosin chỉ hướng của khâu i so với hệ cơ sở

JTi , JRi :Lần lượt là các ma trận Jacobian tịnh tiến và Jacobian quay được tính bằng công thức :

Và phương trình Lagrange được đưa về dạng :



* + 1. **Ma trận khối lượng**

****

**Khâu 1**

 ; 

Ma trận biến đổi tọa độ thuần nhất của khối tâm khâu 1 so vơi htd Oxyz

 **=> **

Đạo hàm riêng cho lần lượt 

**;**

**; ;**

Ma trận sóng

*  =>

Đạo hàm riêng cho lần lượt 

 ; ; 

Ma trận khối lượng 



Tương tự ta cũng tính được 



* + 1. **Lực quán tính Coriolis và quán tính li tâm**

 ****(n=6)

Ký hiệu 

* + 1. **Thế năng, lực thế**

 ;

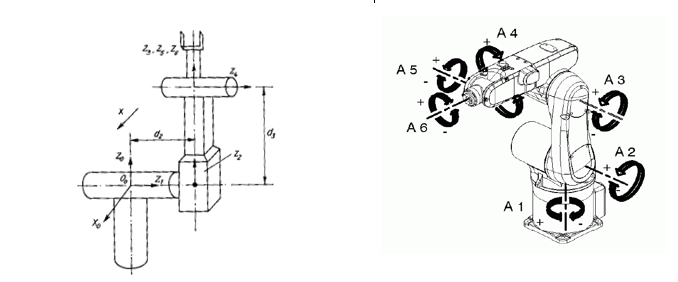




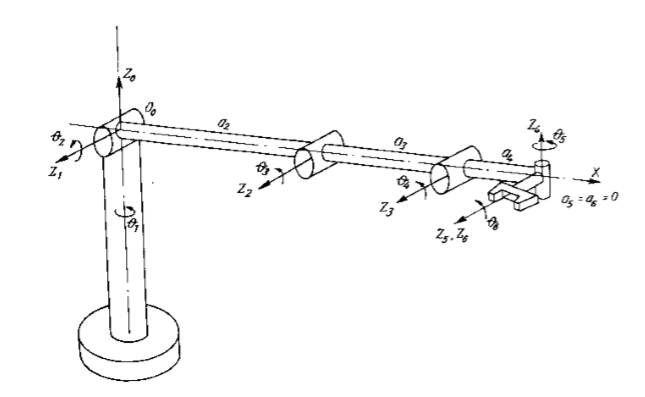
* + 1. **Lực suy rộng của các lực ko thế**



1. **Thiết kế**
   1. **Phương án thiết kế**



*Hình 2.10 Robot Stanford cấu hình RRTRRR Hình 2.11 Robot KUKA cấu hình RRRRRR*

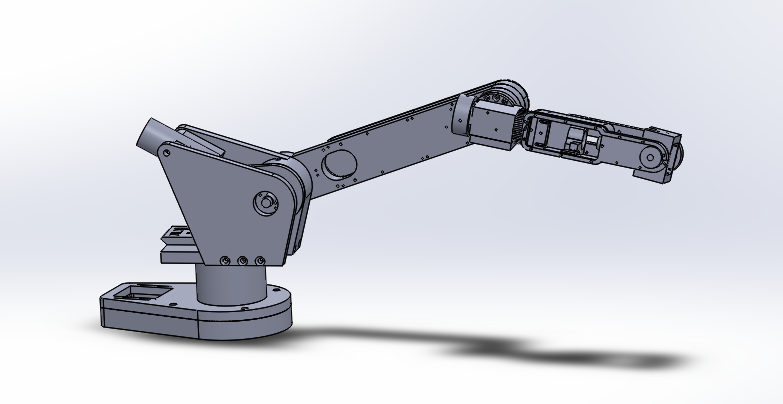
**

*Hình 2.12 Robot Elbow cấu hình RRRRRR*

Với mục tiêu là triển khai loại robot 6 bậc tự do sao cho việc chế tạo không quá khó khăn, phức tạp và giá thành hợp lý, trong phạm vi đồ án tốt nghiệp chúng em sẽ chọn Robot KUKA để phân tích, tính toán, thiết kế, chế tạo, điều khiển và lập trình.

* 1. **Xây dựng mô hình trên Solidworks**

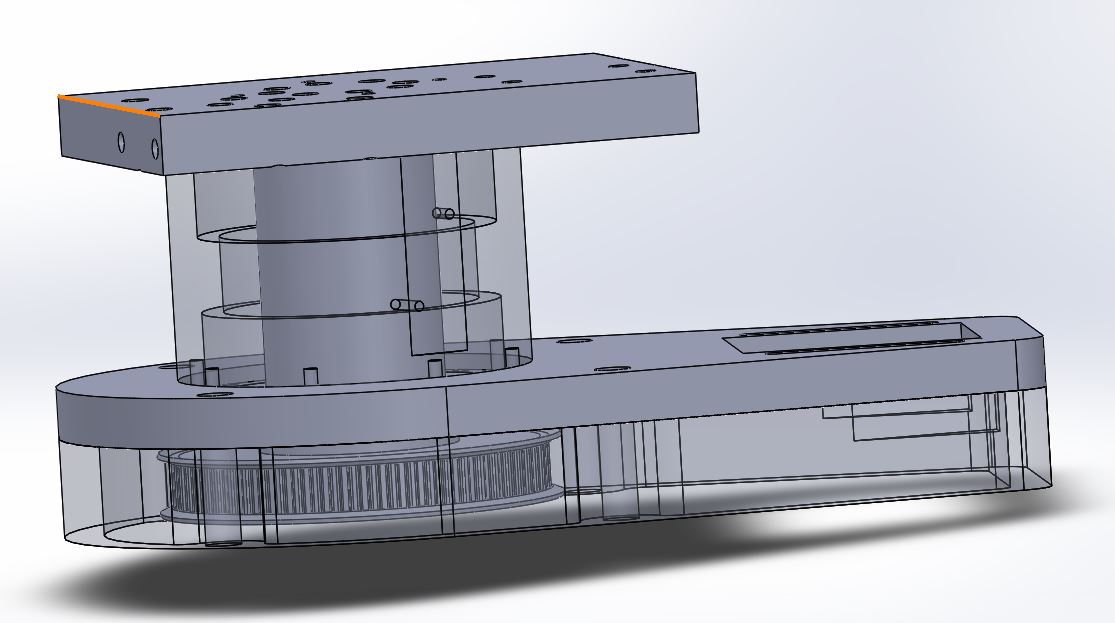
Robot gồm 6 khớp quay được mô phỏng theo thiết kế robot KUKA sử dụng bộ truyền xích và đai răng nhiều cấp làm tăng tỷ số truyền từ động cơ để có thể đáp ứng được các yêu cầu khi gắp vật.



*Hình 2.13 Mô hình 3D trên SolidWorks*

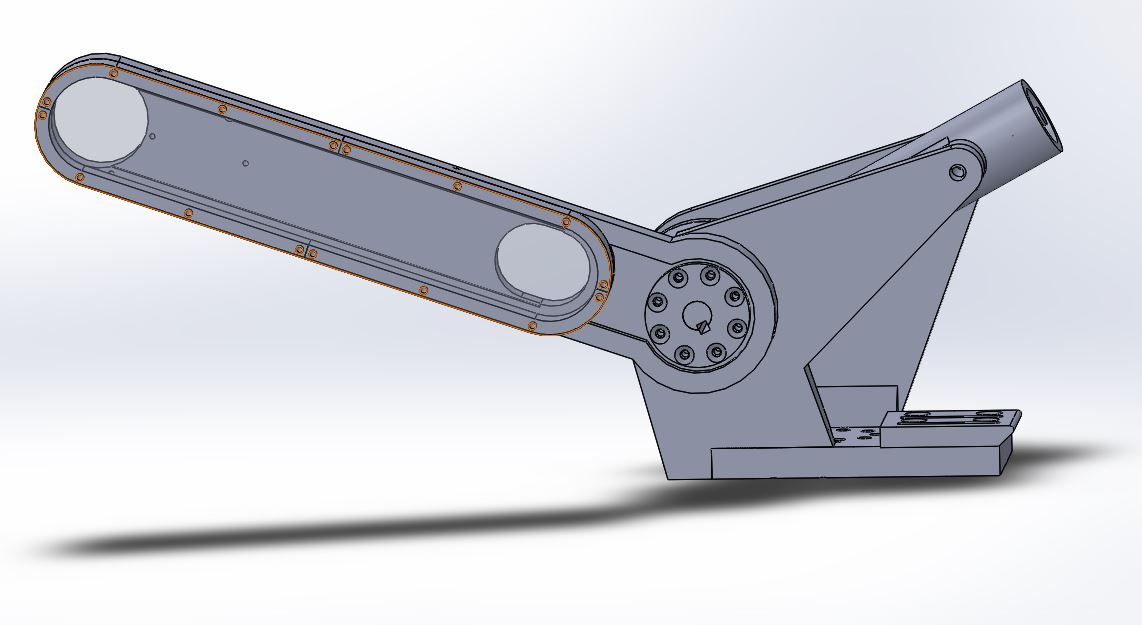
Robot được hình thành bởi 6 khâu quay. Các khâu được kết nối truyền động với nhau qua hệ thống động cơ – dây đai và động cơ - xích, mỗi khớp quay sẽ là một bậc tự do của robot.

Khâu 1 Sử dụng bộ truyền động đai răng đặt trên mặt phẳng nằm ngang với động cơ đặt ở đằng sau.Trên khâu 1 có gắn mount cho động cơ khâu 2.



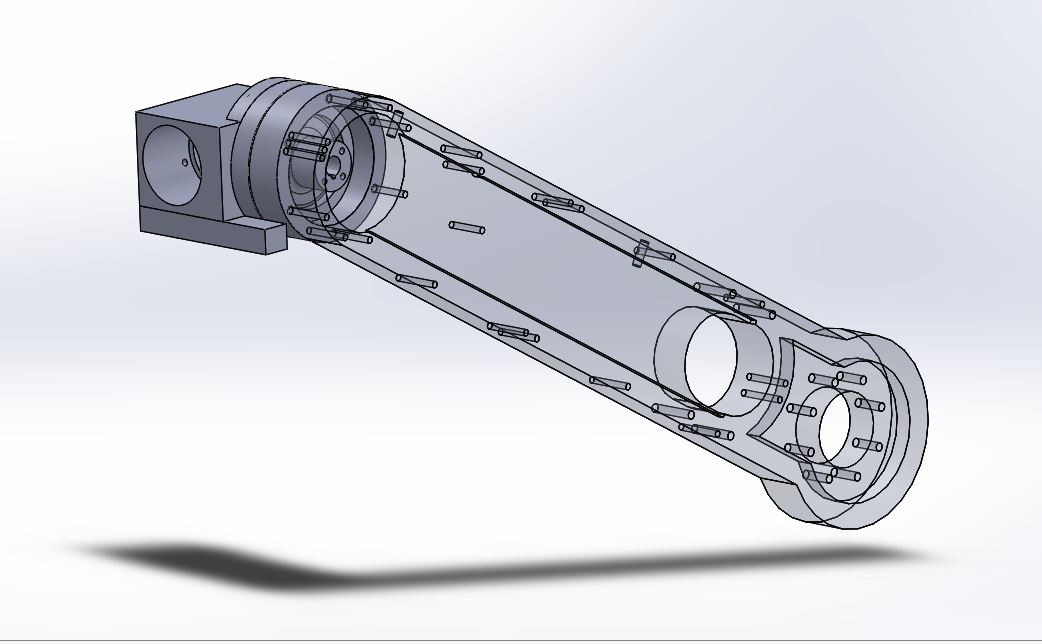
*Hình 2.14 Khâu 1*

Khâu 2 sử dụng bộ truyền xích trên mặt phẳng thẳng đứng.Đằng sau có hệ lò xo trợ lực.



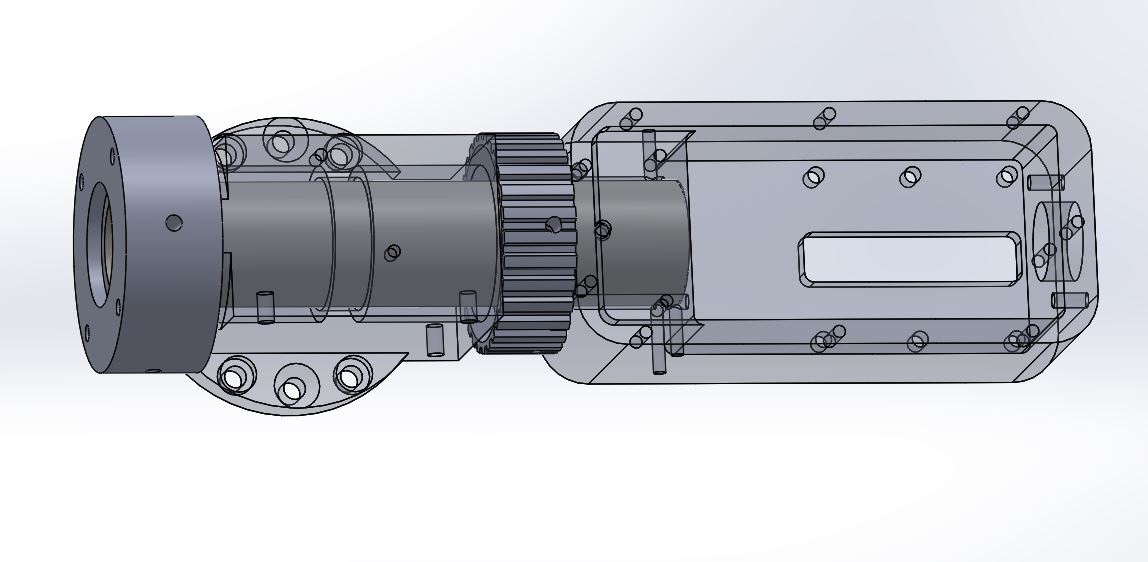
*Hình 2.15 Khâu 2*

Khâu 3 được thiết kế theo dạng hình hộp ở giữa có lỗ hình trụ để lắp trục khâu 4. Cả khâu 3 và khâu 4 đều sử dụng bộ truyền đai răng.

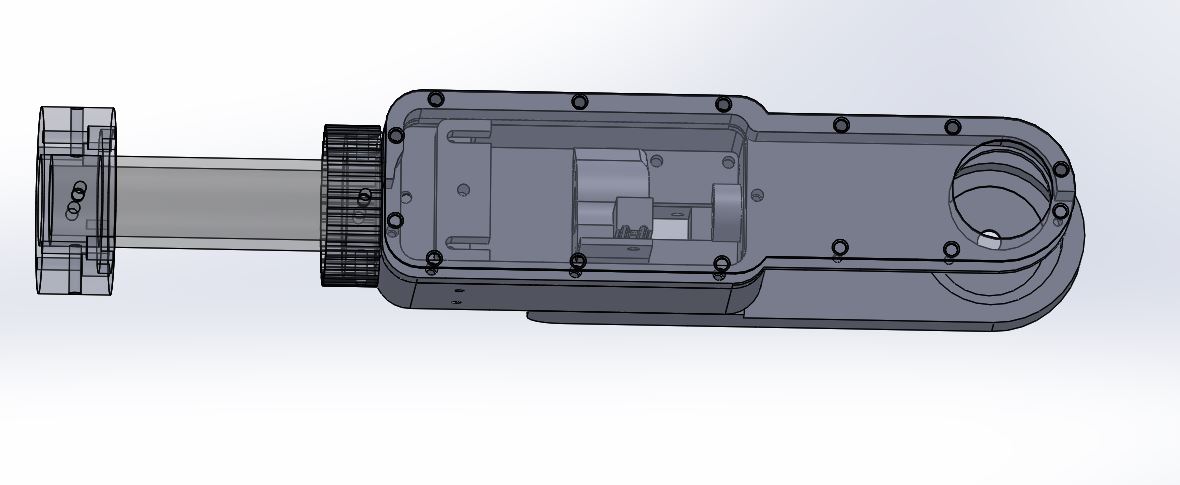


*Hình 2.16 Khâu 3*

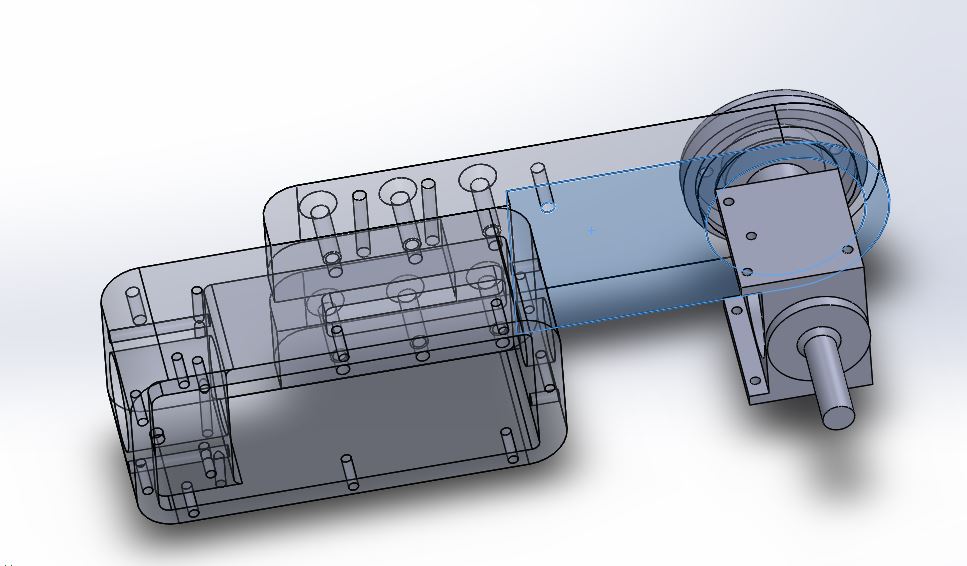
Động cơ khâu 5 được đặt ở cuối khâu 3,có trục 8mm dài 200mm, xỏ qua ống trục khâu 4.Đai ốc gắn trên trục 5 được thiết kế thêm bộ phận kẹp đai giúp truyền động sang khâu 6.



*Hình 2.17 Khâu 4*

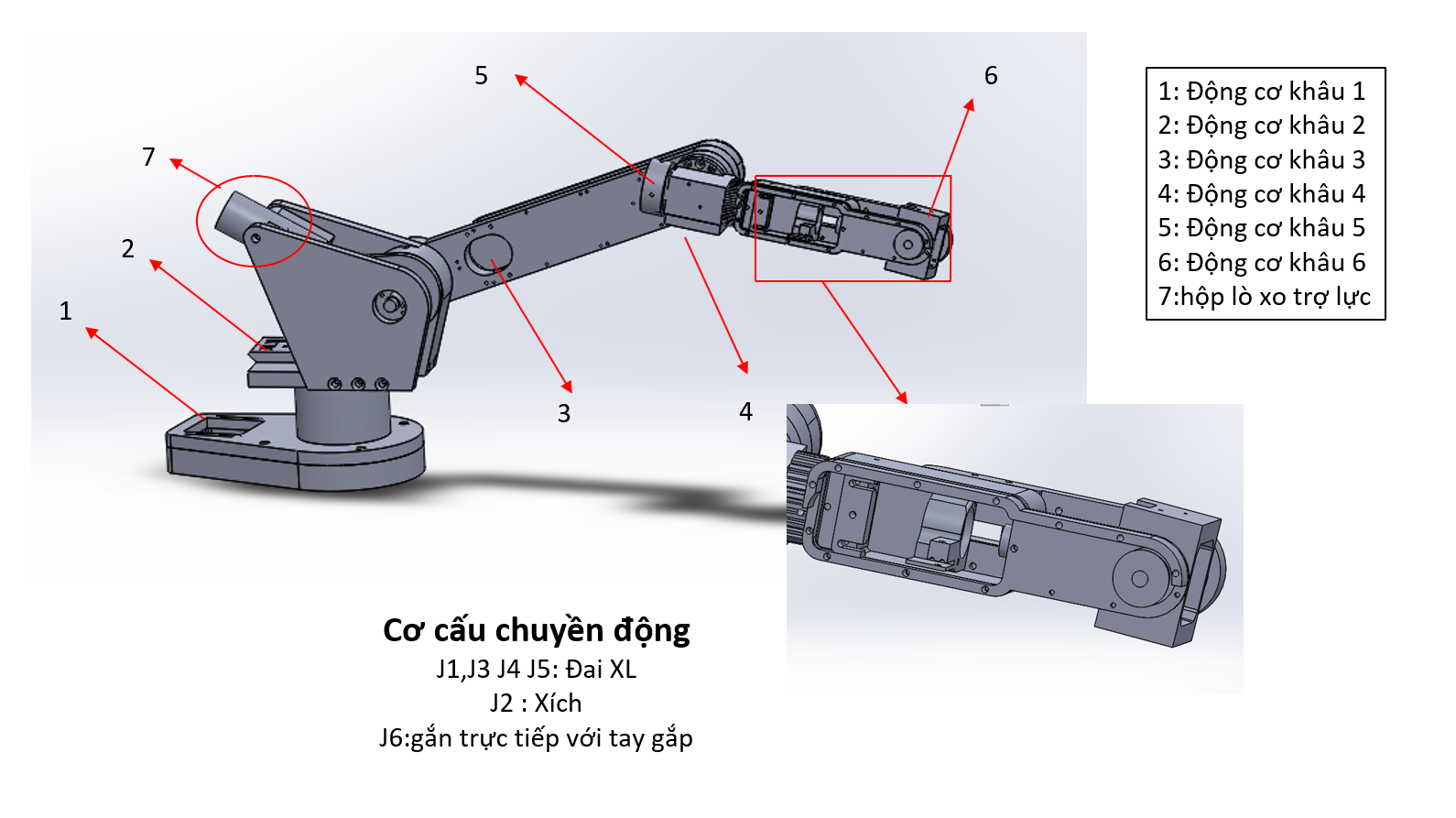


*Hình 2.18 Khâu 5*



*Hình 2.19 Khâu 6*

Tất cả các khâu được lắp vào với nhau thành 1 tay máy hoàn chỉnh.Tại các khớp động có ổ bi để chuyển ma sát trượt thành ma sát lăn giúp các khâu chuyển động mượt mà hơn.

**

*Hình 2.20 Robot sau khi liên kết*

1. **Chọn vật liệu**
   1. **Chọn phương pháp chế tạo**

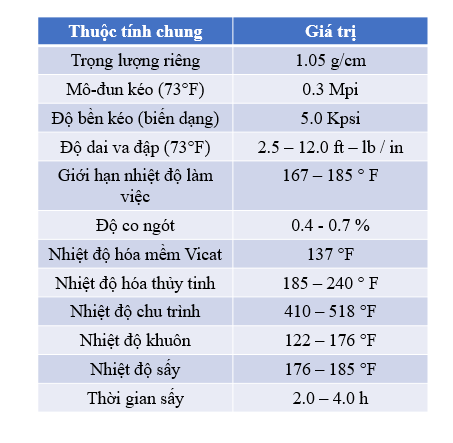
Thay vì sử dụng nhôm tấm, thanh nhôm định hình, nhựa mica cùng với phương pháp gia công cắt gọt thường được sử dụng để chế tạo các mô hình robot. Cánh tay robot 6 bậc của bọn em sử dụng phương pháp in 3D với nhựa ABS để in các chi tiết cho cánh tay robot.

Bằng phương pháp in 3D có thể thực hiện được ở tất cả các loại mẫu thiết kế, không bị hạn chế bởi độ phức tạp của bề mặt hay độ khó về cấu trúc thiết kế. Chỉ cần 1 lần thao tác là có thể in ra được toàn bộ sản phẩm dù có cấu trúc phức tạp đến đâu.

Có 2 loại nhựa phổ biến được sử dụng rộng rãi trong in 3D vì đặc tính vả giá thành hợp lý là nhựa ABS và nhựa PLA.

* Nhựa ABS cứng rắn nhưng không giòn, cân bằng tốt giữa độ bền kéo, va đập, độ cứng bề mặt, độ rắn, độ chịu nhiệt các tính chất ở nhiệt độ thấp và các đặc tính về điện. Có tính chất đặc trưng là độ bền cơ học cao, độ cứng cao, khả năng chịu va đập không giảm nhanh ở nhiệt độ thấp, độ ổn định dưới tải trọng rất tốt, ABS chịu nhiệt tương đương hoặc tốt hơn Acetal, PC…ở nhiệt độ phòng. Nhiệt độ in của nhựa ABS khá cao từ 230 độ C trở lên.
* Nhựa PLA tương đối dòn, cơ tính thấp, nhưng sản phẩm in ra khá đẹp và đầy đủ chi tiết. Việc sử dụng nhựa PLA khó khăn vì nó giòn khi nó đã hạ nhiệt. Ngưỡng nhiệt độ của nó là thấp hơn so với ABS. Nhiệt độ in của PLA cũng tương đối thấp, tầm dưới 190 độ C.
* Về chi phí thì nhựa ABS có giá rẻ hơn nhựa PLA ( 14-60$ / 1kg nhựa ABS so với 19-75$ / 1kg nhựa PLA ).

Việc lựa chọn nhựa ABS để in các chi tiết cho cánh tay Robot là phù hợp.



*Bảng 2.4: Thông số kỹ thuật nhựa ABS*.

* 1. **Chọn động cơ**

**-**  So sánh động cơ bước và động cơ servo

|  |  |  |
| --- | --- | --- |
|  | Động cơ bước | Động cơ servo |
| Mạch driver | Mạch đơn giản | Mạch phức tạp, thường phải mua driver từ nhà sản xuất |
| Nhiễu và rung động | Đáng kể | Rất ít |
| Tốc độ | Trung Bình( 2000-4000rpm) | Nhanh(3000-5000rpm) |
| Hiện tượng trượt bước | Xảy ra khi tải quá lớn | Khó xảy ra |
| Phương pháp điều khiển | Vòng hở (không có encoder) | Điều khiển vòng kín(có encoder) |
| Độ phân giải | 2 pha PM: 7.5° (48 ppr)  2 pha HB: 1.8° (200 ppr) hoặc 0.9° (400 ppr)  5 pha HB: 0.72° (500 ppr) hoặc 0.36° (1000 ppr) | Phụ thuộc độ phân giải của encoder.  Thông thường vào khoảng 0.36° (1000ppr) – 0.036° (10000ppr) |

*Bảng 2.5 So sánh động cơ bước và động cơ servo*

Trong đồ án này chúng em sử dụng động cơ step để làm động cơ dẫn dộng cho robot.

- Ưu, nhược điểm của động cơ bước

Ưu điểm :

Không chổi than : Không xảy ra hiện tượng đánh lửa chổi than làm tổn hao năng lượng, tại một số môi trường đặc biệt (hầm lò...) có thể gây nguy hiểm.

Tạo được mô men giữ :Một vấn đề khó trong điều khiển là điều khiển động cơ ở tốc độ thấp mà vẫn giữ được mômen tải lớn. Động cơ bước là thiết bị làm việc tốt trong vùng tốc độ nhỏ. Nó có thể giữ được mômen thậm chí cả vị trí nhừ vào tác dụng hãm lại của từ trường rotor.

Điều khiển vị trí theo vòng hở : Một lợi thế rất lớn của động cơ bước là ta có thể điều chỉnh vị trí quay của roto theo ý muốn mà không cần đến phản hồi vị trí như các động cơ khác, không phải dùng đến encoder hay máy phát tốc (khác với servo).

Độc lập với tải : Với các loại động cơ khác, đặc tính của tải rất ảnh hưởng tới chất lượng điều khiển. Với động cơ bước, tốc độ quay của rotor không phụ thuộc vào tải (khi vẫn nằm trong vùng momen có thể kéo được). Khi momen tải quá lớn gây ra hiện tượng trượt, do đó không thể kiểm soát được góc quay.

Giá thành thấp.

Nhược điểm :

Về cơ bản dòng từ driver tới cuộn dây động cơ không thể tăng hoặc giảm trong lúc hoạt động. Do đó , nếu bị quá tải động cơ sẽ bị trượt bước gây sai lệch điều khiển.

Động cơ bước gây ra nhiều nhiễu và rung động hơn động cơ servo.

Động cơ bước không thích hợp cho các ứng dụng cần tốc độ cao.



*Hình 2.21 Động cơ bước*

* Để đáp ứng khả năng chịu tải và tăng độ chính xác, bọn em sử dụng động cơ bước có tích hợp các bánh răng giảm tốc có tỷ số truyền lớn.

Dưới đây là danh sách các động cơ được sử dụng :

|  |  |  |  |
| --- | --- | --- | --- |
| **Khớp** | **Hình ảnh** | **Thông số kỹ thuật** | **Thông số kích thước** |
| **J1** |  | Góc/ bước: 1.8°  Moment xoắn max: 1.9Nm  Số pha : 2  Rated Current/phase: 2.8A  Voltage: 3.2V  Phase Resistance: 1.13ohms | Kích thước: 57 x 57mm  Chiều dài: 76mm  Đường kính trục: Φ6mm  Chiều dài trục: 21mm  Khối lượng: 1.2kg |
| **J2** |  | Góc/ bước: 0.039°  Tỷ số truyền: 46.656/ 1  Moment xoắn max: 60Nm  Số pha : 2  Rated Current/phase: 2.8A  Voltage: 3.2V  Phase Resistance: 1.13ohms | Kích thước: 60 x 60mm  Chiều dài: 76mm+72mm  Đường kính trục: Φ12mm  Chiều dài trục: 30mm  Chiều dài rãnh then: 20mm  Chiều rộng then : 4mm  Khối lượng: 2.3kg |
| **J3** |  | Góc/ bước: 0.094°  Tỷ số truyền: 19.19/ 1  Moment xoắn max: 5Nm  Số pha : 2  Rated Current/phase: 1.68A  Voltage: 2.8V  Phase Resistance: 1.65ohms | Kích thước: 42 x 42mm  Chiều dài: 48mm+35mm  Đường kính trục: Φ8mm  Chiều dài trục: 20mm  Chiều dài rãnh then:15mm  Chiều rộng then : 3mm  Khối lượng: 560g |
| **J4** |  | Góc/ bước: 0.35°  Tỷ số truyền: 5.18/ 1  Moment xoắn max: 4Nm  Số pha : 2  Rated Current/phase: 0.67A  Voltage: 6.2V  Phase Resistance: 9.2ohms | Kích thước: 30 x 30mm  Chiều dài: 50.5mm+27.2mm  Đường kính trục: Φ6mm  Chiều dài trục: 18mm  Chiều dài rãnh then:12mm  Chiều rộng then : 2mm  Khối lượng: 290g |
| **J5** |  | Góc/ bước: 1.8°  Moment xoắn max: 44Ncm  Số pha : 2  Rated Current/phase: 1.68A  Voltage: 2.8V  Phase Resistance: 1.65ohms | Kích thước: 42 x 42mm  Chiều dài: 48mm  Đường kính trục: Φ8mm  Chiều dài trục vitme: 200mm  Bước đai ốc vitme/step : 0.04mm  Khối lượng: 400g |
| **J6** |  | Góc/ bước: 1.8°  Moment xoắn max: 45Ncm  Số pha : 2  Rated Current/phase: 2A  Voltage: 2.2V  Phase Resistance: 1.1ohms | Kích thước: 42 x 42mm  Chiều dài: 40mm  Đường kính trục: Φ5mm  Chiều dài trục: 23m  Khối lượng: 310g |

*Bảng 2.6 Bảng thông số các động cơ của robot*

* 1. **Chọn hệ thống dẫn động**

Trong đồ án này chúng em sử dụng bộ truyền đai răng để làm bộ truyền dẫn động từ động cơ đến các khớp quay của robot. Bộ truyền đai răng - pulley được bọn em sử dụng cho các khớp 1,3,4,5.

* Bộ truyền đai :
* Ưu điểm :

Có khả năng truyền chuyển động giữa các trục xa nhau

Làm việc êm và không ồn

Giữ an toàn cho các chi tiết và động cơ khi bị quá tải nhờ hiện tượng trượt đai

Kết cấu đơn giản, dề dàng lắp ráp và thay thế, không yêu cầu bôi trơn định kỳ

* Nhược điểm :

Kích thước đai và pulley lớn

Hiệu suất thấp do trượt đàn hồi

Tuổi thọ của dây đai thấp

Lực tác dụng lên trục và ổ bi lớn do phải căng đai

Khớp 2 bọn em sử dụng bộ truyền xích để đảm bảo khả năng chịu tải và tránh hiện tượng giãn đai do mỏi gây sai số cho cơ cấu.

* Bộ truyền xích :
* Ưu điểm :

Có thể truyền chuyển động giữa các trục có khoảng cách lớn ( tối đa 8m )

Không có hiện tượng trượt

Khuôn khổ kích thước nhỏ hơn so với bộ truyền đai

Lực tác dụng lên trục nhỏ hơn so với bộ truyền đai vì không cần căng xích với lực căng ban đầu

* Nhược điểm :

Do có sự va đập giữa các khớp nên không thích hợp với truyền động tốc độ cao

Đòi hỏi chế tạo, lắp ráp với độ chính xác cao, yêu cầu bôi trơn thường xuyên

Tỷ số tuyền và vận tốc tức thời không ổn định

Chóng mòn nếu không được bôi trơn hoặc làm việc trong môi trường bụi

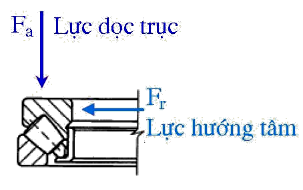
* 1. **Chọn ổ bi**

Tại các khớp động của cánh tay robot, bọn em sử dụng ổ bi để chuyển ma sát trượt giữa 2 khâu thành ma sát lăn, giúp giảm lực cản giữa các khâu, làm các khâu chuyển động linh hoạt hơn. Vì ổ bi yêu cầu phải chịu cả lực dọc trục (*F*a ) và lực hướng tâm (*F*r ) nên bọn em sử dụng 2 ổ đũa đỡ chặn một dãy ( ổ đũa côn ) trên một khớp.



*Hình 2.22 Ổ đũa côn*

Ổ đũa côn chịu lực dọc trục (*F*a ) và lực hướng tâm (*F*r ) trong đó có thể chịu lực hướng tâm (*F*r ) lớn hơn 70% so với ổ bi đỡ 1 dãy và chịu lực dọc trục (*F*a ) lớn hơn ổ bi đỡ 1 dãy. Ổ đũa côn dễ tháo lắp để bảo dưỡng bôi trơn, có thể điều chỉnh khe hở để bù lại lượng mòn.



*Hình 2.23 Khả năng tiếp nhận tải trọng của ổ đũa côn*

* 1. **Chọn tay gắp**

Tay gắp được sử dụng trên các cánh tay robot để gắp giữ vật thể và đặt vật thể đó ở vị trí mong muốn. Tùy theo mục đích sử dụng, mỗi robot sẽ được kết nối với loại tay gắp khác nhau. Các loại tay gắp phổ biến thường gặp trong công nghiệp :

*Hình 2.24 các loại tay gắp*

Trong đồ án này chúng em sử dụng tay tay gắp dùng động cơ servo. Tay gắp ứng dụng nguyên lý tay kẹp hình bình hành để có thể đóng mở má kẹp song song, tiếp xúc tốt hơn với vật thể cần gắp.

Tay kẹp được thiết kế theo tiêu chuẩn chung từ vật liệu nhôm cứng, thiết kế cân xứng giúp tay kẹp robot cứng chắc dễ dàng các thao tác gắp sản phẩm, có độ ổn định, và không bị trượt trong quá trình thao tác.



*Hình 2.25 Cơ cấu tay kẹp*



*Hình 2.26 Động cơ servo*

Kẹp mở tối đa là 55mm.

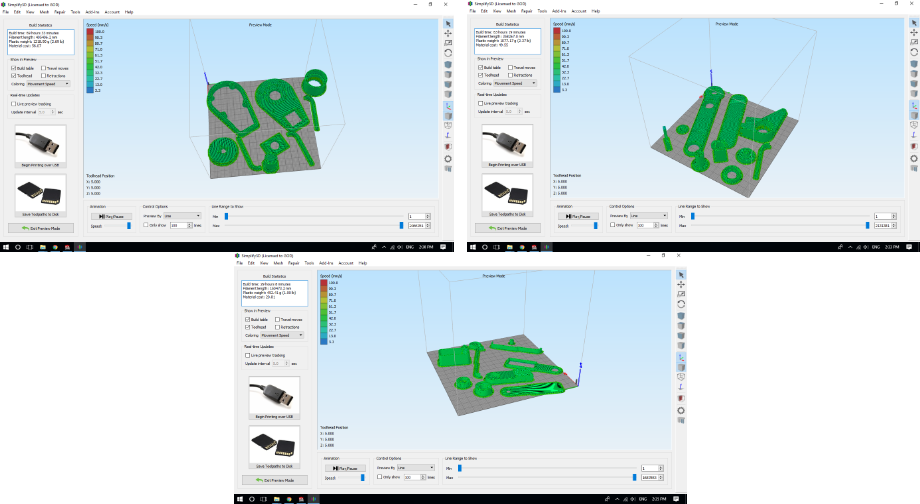
Kẹp tổng chiều dài: 108mm.

Tổng chiều rộng: 98mm( khi kẹp mở tối đa chiều rộng tổng thể).

Động cơ servo làm việc điện áp: 4,8-6v( khi kẹp mở).

Ván kit nhẹ​.

1. **Chế tạo và lắp ráp**
   1. **Chế tạo**

Để in các chi tiết cho cánh tay robot, bọn em sử dụng máy in 3D có đầu vòi phun 0.4mm, các chi tiết được in với fill density từ 60% ( cho các chi tiết không chịu tải trọng lớn vừa để giảm khối lượng của chi tiết ) tới 100% ( cho các chi tiết chịu tải trọng lớn ). Thời gian in cho mỗi chi tiết trung bình từ 3- 7h. Các chi tiết sau khi in có độ chính xác cao, đảm bảo thông số kích thước như trên bản vẽ

*Hình 2.27 Mô phỏng in 3D*

Hình ảnh một số chi tiết hoàn chỉnh sau khi in 3D:

*Hình 2.28 Một số chi tiết sau khi in 3D*

* 1. **Lắp ráp**

****

*Hình 2.29 Robot sau khi lắp ráp*

# **CHƯƠNG 3: THIẾT KẾ BỘ ĐIỀU KHIỂN**

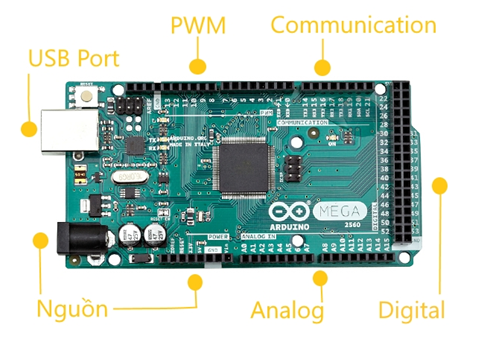
1. **Mạch Arduino**

Arduino được chọn làm bộ não xử lý của rất nhiều thiết bị từ đơn giản đến phức tạp. Trong số đó có một vài ứng dụng thực sự chứng tỏ khả năng vượt trội của Arduino do chúng có khả năng thực hiện nhiều nhiệm vụ rất phức tạp.

*Hình 3.1 Một số ứng dụng nổi bật của Arduino*

1. **Mạch Điều khiển robot**
   1. **Mạch Arduino AT mega 2560**

Board arduino mega 2560 sử dụng chip ATmega2560 của ATmel, bộ nhớ chương trình lên đến 256KB trong đó có 8KB được sử dụng bởi boeotloader. Với bộ nhớ chương trình lớn, bạn có thể viết nhiều chương trình phức tạp, điều khiển được nhiều thiết bị hơn. Dung lượng RAM là 8KB. 4KB EEPROM.



*Hình 3.2 Cấu trúc cơ bản của Mega 2560*

+ 4 cổng serial:

|  |  |  |
| --- | --- | --- |
| Cổng serial | Chân RX | ChânTX |
| Serial 0 | 0 | 1 |
| Serial 1 | 19 | 18 |
| Serial 2 | 17 | 16 |
| Serial 3 | 15 | 14 |

*Bảng 3.1 Cổng serial trên arduino mega*

Lưu ý: cổng serial 0 trên 2 chân 0 và 1 được sử dụng để upload chương trình từ Arduino IDE vào board Arduino Mega.

+ 6 ngắt ngoài

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Số thứ tự ngắt | 0 | 1 | 2 | 3 | 4 | 5 |
| Chân trên Arduino | 2 | 3 | 21 | 20 | 19 | 18 |

*Bảng 3.2 Cổng ngắt ngoài trên arduino mega*

+ 15 chân PWM

Các chân 2-13 và chân 44, 45, 46.

+ 16 chân analog

Thông số kĩ thuật :

|  |  |
| --- | --- |
| Chip xử lý | ATmega2560 |
| Điện áp hoạt động | 5V |
| Điện áp vào (đề nghị) | 7V-15V |
| Điện áp vào (giới hạn) | 6V-20V |
| Cường độ dòng điện trên mỗi 3.3V pin | 50 mA |
| Cường độ dòng điện trên mỗi I/O pin | 20 mA |
| Flash Memory | 256 KB |
| SRAM | 8 KB |
| EEPROM | 4 KB |
| Clock Speed | 16 MHz |

*Bảng 3.3: Thông số kĩ thuật của arduino mega*

* 1. **Driver điều khiển động cơ**

- Động cơ bước có nhiều cách điều khiển. Có thể điều khiển các dây trực tiếp qua 4 cổng qua MCU thông qua driver đệm công suất. Cách này hơi phức tạp một chút, cần phải hiểu rõ bên trong động cơ và thường chỉ điều khiển được full bước

- Cách thông dụng nhất là dùng các IC chuyên dụng điều khiển động cơ bước. Các IC hay gặp nhất là TB6560, TB6600, L297,L298, A4988, DRV8825, MA860H... Việc lựa chọn dùng loại driver nào phụ thuộc vào loại động cơ và công suất động cơ định điều khiển. Ví dụ như A4988, DRV8825 dùng để điều khiển các loại động cơ nhỏ có công suất bé ví dụ như trong máy photo hoặc máy in 3d, TB6560 hoặc TB6600 thì lại dùng để điều khiển các loại động cơ lớn hơn một chút ví dụ như trong các loại máy cnc mini.

- Nhìn chung cách giao tiếp với các module Driver này tương đối giống nhau. Chúng đều có 3 port cơ bản là DIR ( để điểu khiển hướng quay động cơ), EN ( để điều khiển bật tắt động cơ), CLK ( xuất xung để dịch chuyển từng step). Một số loại module như TB6560 TB6600 hoặc MA860H thì mỗi port đều có 2 pin. Ví dụ như EN+ EN- CW+ CW- CLK+ CLK-. vì thế người dùng có thể tùy chọn điều khiển theo mức 0 hoặc mức 1. Mình thường kéo 3 chân xuống thấp và điều khiển bằng 3 chân còn lại

- Trong đồ án này bọn em sử dụng module TB6560 để điều khiển động cơ stepper



*Hình 3.3 : Module TB6560 điều khiển động cơ bước*

- Module TB6560 có thể điều khiển được vi bước như 1/16 step, 1/8 step, 1/2 step và full step. nếu full step thì bạn cần 200 step để quay hết 1 vòng đối với loại 1.8 độ 1step. Còn nếu dùng chế độ vi bước 1/16 step thì một xung động cơ chỉ dịch chuyển 1,8/16 độ. Tức là bạn phải cần 200\*16=3200 xung để quay hết một vòng. điều này làm tăng độ phân giải và tăng độ chính xác cho step.

- Cách kết nối: với động cơ 2 pha 4 dây thì kênh A- A+ B- B+ được nối vào driver. với động cơ 5 dây 6 dây hoặc 8 dây thì cắt bỏ hết những dây chung đi và chỉ sử dụng 4 dây ở 2 đầu cuộn dây để điều khiển.

Thông số Module:

|  |  |
| --- | --- |
| Điện áp đầu vào | Từ 10V - 35VDC. Tốt nhất nên dùng 24VDC |
| Dòng tối đa đầu ra | 3.5A |
| Kích thước | 70mm x 50mm x 35mm |
| Chip | TB6560AHQ |
| Sử dụng thích hợp với động cơ bước | bước 3A, 3.5A loại 42, 57 với 2-phase/ 4- phase/ 4 dây/ 6 giây... |
| Chế độ quá tải | Tự động giảm 1/2 dòng điện khi quá tải. |
| Làm việc với các chế độ | Full step, half step, 1/8 step, 1/16 step, lên đế 16 chế độ |

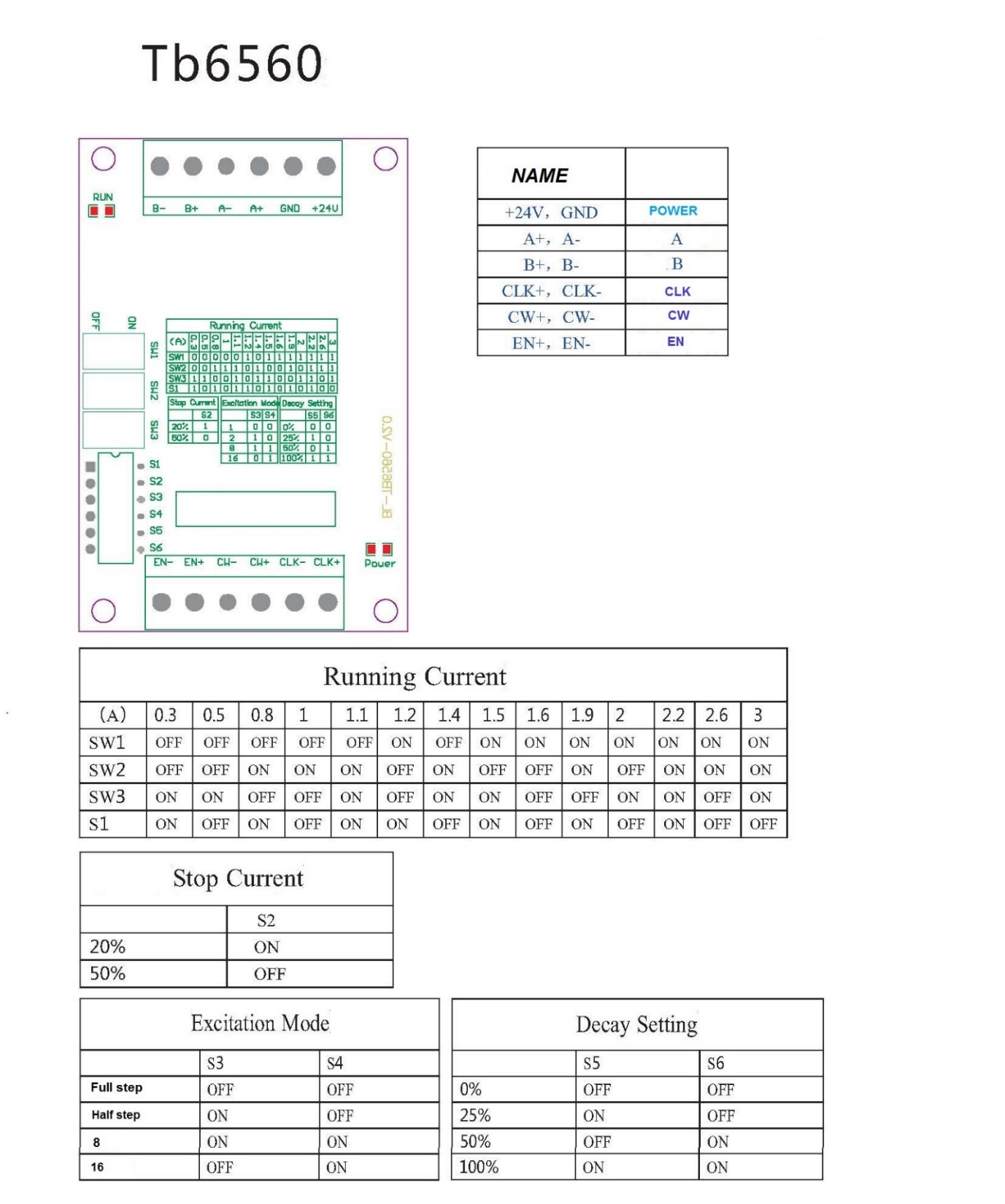
*Bảng 3.4 : Thông số kĩ thuật của drive TB6560*

+ Running Current(tùy chỉnh dòng ) : 0,3-3A(SW1,SW2,SW3,S1)

+Decay(Hạn dòng cho motor) : khi cấp điện cho cuộn dây motor qua cầu H bên trong IC thì sẽ có dòng điện chạy qua, ví dụ quy định dòng là 1A thì khi đến dòng 1A IC nó ngưng không cấp điện nữa, dòng điện tụt xuống, nó lại cấp tiếp... cứ vậy. Chu kỳ cấp rồi ngưng phụ thuộc vào dao động nội bên trong IC, dao động này thay đổi bằng cách thay đổi giá trị tụ C5, trong mạch dùng tụ 330pF, dao động là 130khz, thay tụ 100pF thì dao động là 400khz, tần số càng cao thì IC kiểm tra sự thay đổi dòng càng nhanh, hình dung đại khái thì càng nhanh càng mịn.

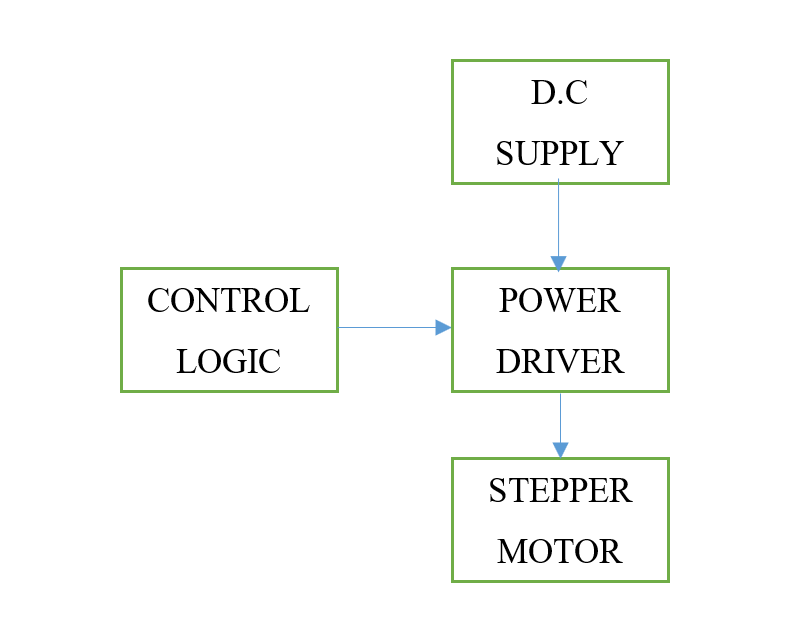
0%, 25%, 50%, 100% ( motor nhỏ hay quay chậm thì 0%, motor to hay quay nhanh thì 100%.Tùy chọn 25%, 50% với các loại động cơ trung bình ).

+Excitation: đặt vi bước cho mạch driver. Ví dụ motor 1,8 độ set (1) thì 1 xung ngoài vào chân step sẽ quay 1,8 độ, 200 xung sẽ quay 1 vòng. Set 1/2, 1/8, 1/16 thì 1 xung vào sẽ quay 1/2, 1/8, 1/16 của 1,8 độ.



*Hình 3.4 Thông số mạch AT 2560*

* 1. **Sơ đồ nối mạch**
* Sơ đồ khái quát về hệ thống sử dụng động cơ bước



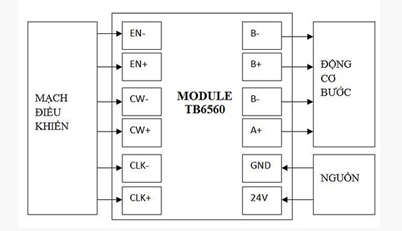
*Hình 3.5 Sơ đồ khái quát về hệ thống sử dụng động cơ bước*

**D.C.SUPPLY**: Có nhiệm vụ cung cấp nguồn một chiều cho hệ thống. Với các động cơ có công suất lớn có thể dùng nguồn điện được chỉnh lưu từ nguồn xoay chiều.

**CONTROL LOGIC**: Đây là khối điều khiển logic. Có nhiệm vụ tạo ra tin hiệu điều khiển động cơ. Khối logic này có thể la một nguồn xung, hoặc có thể là một hệ thống mạch điện tử. Nó tạo ra cac xung điều khiển. Động cơ bước có thể điều khiển theo cả bước hoặc theo nửa bước.

**POWER DRIVER**: Có nhiệm vụ cấp nguồn điện đa được điều chỉnh để đưa vào động cơ. Nó lấy điện từ nguồn cung cấp và xung điều khiển từ khối điều khiển để tạo ra dòng điện cấp cho động cơ hoạt động.

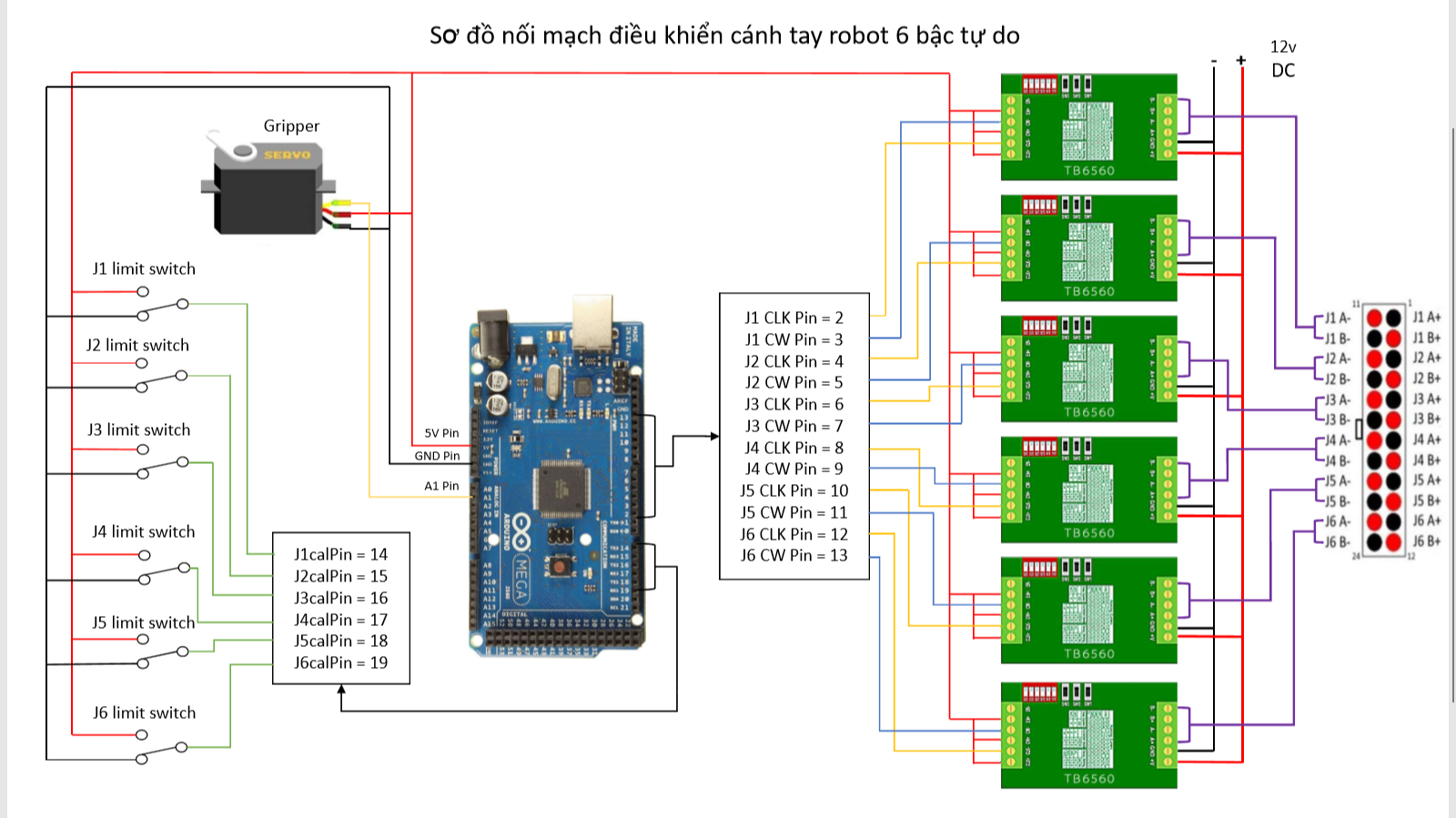
**STEPPER MOTOR**: Động cơ bước. Cac thông số của động cơ gồm có: Bước góc, sai số bước góc, momen kéo, momen hãm, momen làm việc.Đối với hệ điều khiển động cơ bước, ta thấy đó la một hệ thống khá đơn giản vì không hề có phần tử phản hồi. Điều này có được vì động cơ bước trong quá trình hoạt động không gây ra sai số tich lũy, sai số của động cơ do sai số trong khi chế tạo. Việc sử dụng động cơ bước tuy đem lại độ chính xác chưa cao nhưng ngày càng được sử dụng phổ biến. Vì công suất và độ chính xác của bước góc đang ngày càng được cải thiện.



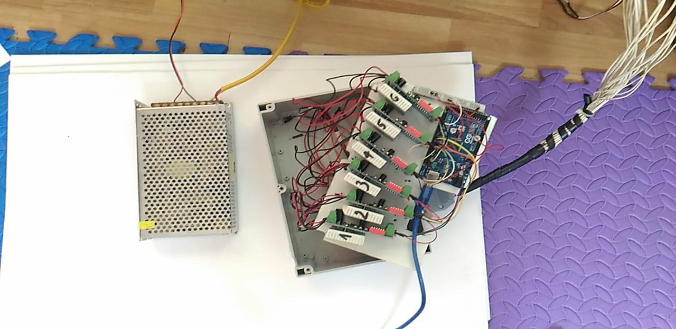
*Hình 3.6 Sơ đồ kết nối giữa TB6560, mạch điều khiển và động cơ bước*

Robot được thiết kế gồm 6 khớp xoay sử dụng 6 động cơ bước điều khiển chuyển động và sử dụng 1 động cơ servo để điều khiển tay gắp vật. Như vậy sẽ có 7 đối tượng được điều khiển.

Để ghép nối 6 động cơ bước với Arduino sẽ cần 6 module TB6560. Trên module TB6560 , các chân A+, A-, B+, B- sẽ được kết nối với các chân động cơ bước, các chân GND và 24V sẽ được nối với nguồn một chiều có dòng 3A,điện áp 12 VDC. Các chân CLK+, CLK-, CW+, CW-, EN+, EN- sẽ được kết nối tới các chân trên bo Arduino. Cụ thể , các chân CLK+, CW+, EN+ sẽ được nối chung vào chân 5V trên Arduino, các chân CLK-,CW- sẽ nối với các chân băm xung.

*Hình 3.7 Sơ đồ nối mạch điều khiển cánh tay robot*

* Trong sơ đồ trên ký hiệu 6 động cơ bước tương ứng là : J1, J2, J3, J4,J5,J6
* Các chân của động cơ lần lượt nối với các chân A+, A-, B+, B- trên TB6560.
* Các chân CLK- trên 6 module TB6560 sẽ nối lần lượt với các chân 2, 4, 6, 8, 10, 12 trên bo Arduino. Các chân này có tác dụng cấp xung điều khiển cho động cơ.
* Các chân CW- sẽ lần lượt nối với các chân 3, 5, 7, 9,11,13 trên Arduino. Các chân có tác dụng thay đổi chiều quay của động cơ.
* Các công tắc hành trình giúp tránh tình trạng chạy quá khớp và hỗ trợ cho việc calib cân chỉnh cánh tay sau này.
* Hình ảnh thực tế :



*Hình 3.8 Mạch điều khiển sau khi nối hoàn chỉnh*

# **CHƯƠNG 4: THIẾT KẾ CHƯƠNG TRÌNH ĐIỀU KHIỂN**

1. **Điều khiển Robot**

**-** Các bước thực hiện điều khiển robot

1. Tính toán động học : động học thuận, ngược.

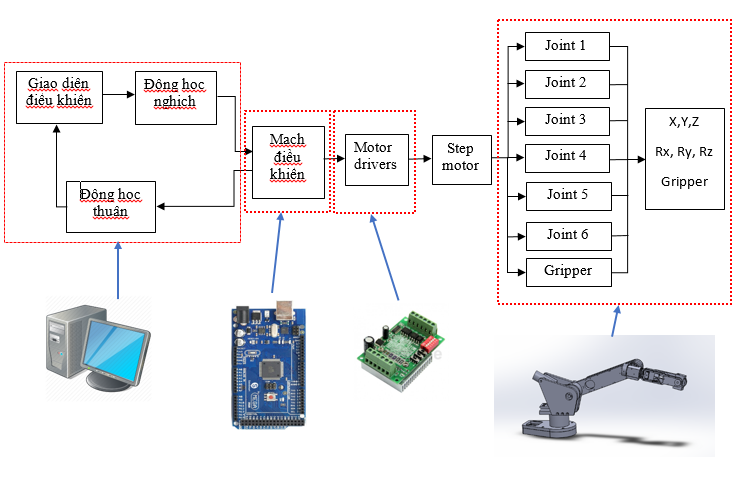
2. Quỹ đạo cần điều khiển cho robot.

3. Áp dụng bài toán động học ngược để xây dựng quỹ đạo chuyển động cho các biến khớp.

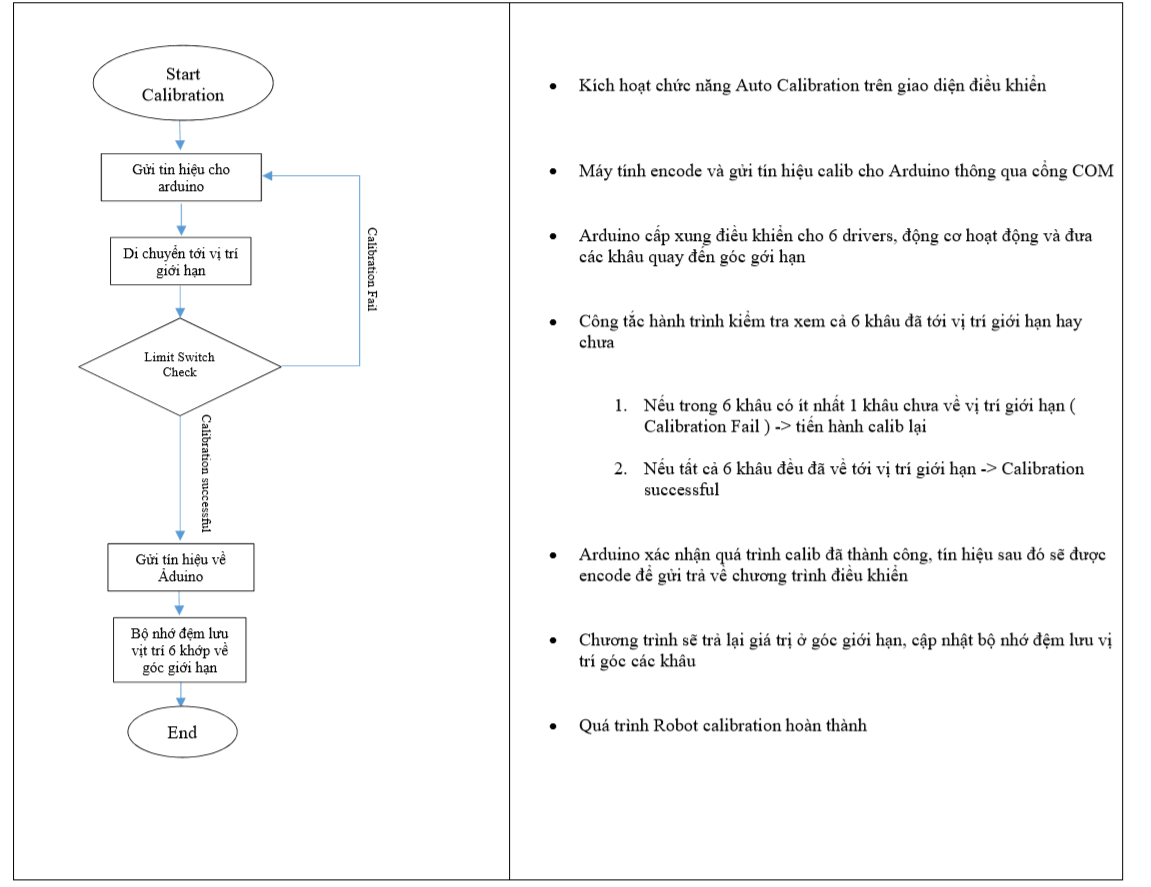
4. Xây dựng luật điều khiển các khớp theo quỹ đạo.

- Dựa vào quỹ đạo đã xây dựng được cho robot, ta điều khiển chuyển động của các biến khớp của robot đúng theo quy luật cần chuyển động để robot đạt được chuyển động mong muốn. Việc điều khiển các biến khớp xoay thông qua mô men, lực đã xác định từ phần động lực học. Mà các khớp lại được lấy nguồn động lực từ các động cơ, từ đó bài toán điều khiển sẽ được quy về bài toán điều khiển các động cơ(động cơ step).

Trong đồ án này, chúng em sử dụng động cơ bước và không sử dụng encoder để phản hồi vị trí nên hệ thống điều khiển sẽ là hệ thống điều khiển hở (open-loop system) và không có bù tác động ngoài (không có bù nhiễu). Sơ đồ điều khiển có dạng như sau :



*Hình 4.1 Sơ đồ điều khiển vòng hở cho robot.*

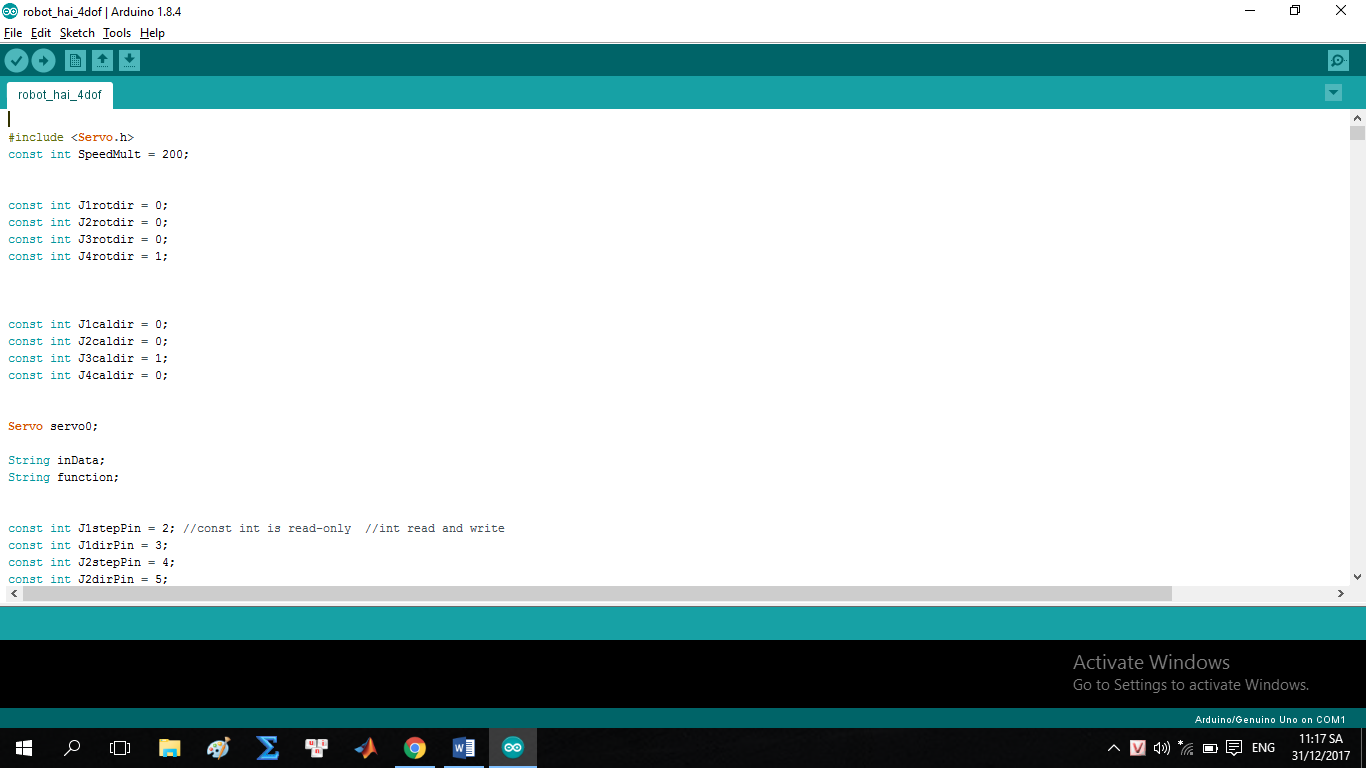
Để đảm bảo an toàn cho toàn bộ cơ hệ, bọn em có lắp đặt các công tắc hành trình tại mỗi khớp, những công tắc hành trình này còn giúp giảm thiểu sai số của cánh tay sau mỗi phiên làm việc nhờ đưa robot về vị trí HOME ( hay còn gọi là Robot calibration )

*Hình 4.2 Lưu đồ thuật toán quá trình Robot calibration*

1. **Lập trình điều khiển Robot**
   1. **Lập trình trên Andruino**

Chức năng của vi điều khiển Arduino ATmega trong hệ thống là tiếp nhận những tín hiệu điều khiển từ chương trình điều khiển chính trên máy tính thông qua giao tiếp Serial và chuyển những tín hiệu đó thành xung để điều khiển 6 drivers TB6560 sau đó arduino sẽ gửi trả lại chương trình điều khiển để xác nhận tín hiệu đó “pass” hay “fail”.

Để lập trình trên arduino ta sử dụng chương trình Arduino IDE



*Hình 4.3 Giao diện phần mềm Arduino IDE*

1 số đoạn code được nạp vào Arduino trông qua chương trình Arduino IDE( PHỤ LỤC 1.1 )

* 1. **Lập trình trên Python**
     1. **Giới thiệu ngôn ngữ Python**

Python là một ngôn ngữ lập trình hướng đối tượng, python hoàn toàn tạo kiểu động, cấp phát bộ nhớ động. Mục đích ra đời của Python là cung cấp một ngôn ngữ lập trình có cấu trúc rõ ràng, sáng sủa, thuận tiện cho người mới học lập trình. Python được phát triển bởi *Guido*và *Rossum.*Phiên bản đầu tiên được phát hành vào năm 1991. Python được lấy cảm hứng từ ABC, Haskell, Java, Lisp, Icon và Perl. Python là một ngôn ngữ thông dịch, đa nền tảng. Một trong những đặc điểm độc nhất của Python là ngôn ngữ này không dùng đến dấu chấm phẩy, dấu mở-đóng ngoặc {} để kết thúc câu lệnh hay khối lệnh, mà cách duy nhất để nó nhận biết một lệnh là dấu thụt đầu dòng.

* + 1. **Lập trình GUI bằng Tkinter**

*Tkinter* là một gói trong Python có chứa module Tk hỗ trợ cho việc lập trình GUI.

* Một vài ví dụ về lập trình giao diện bằng Python Tkinter:

Tạo cửa sổ mới :

**from** tkinter **import** Tk, Frame, BOTH

**class** Example(Frame):

**def** \_\_init\_\_(self, parent):

 Frame.\_\_init\_\_(self, parent, background="white")

 self.parent = parent

 self.initUI()

**def** initUI(self):

 self.parent.title("Simple")

 self.pack(fill=BOTH, expand=1)

root = Tk()

root.geometry("250x150+300+300")

app = Example(root)

root.mainloop()

Đoạn code trên sẽ tạo ra một cửa sổ mới với tên “Simple” như sau :



*Hình 4.4 : Cửa sổ đơn giản với Python Tkinter*

Để tạo ra một GUI bằng Python ta cần các Widget, các widget đơn giản gồm có :

*Nút (Button):*

Một widget, có chứa chữ hoặc hình, để thực hiện một việc khi được nhấn.

*Nền (Canvas):*

Một vùng trên đó có thể hiển thị đường nét, hình chữ nhật, hình tròn và các hình khác.

*Ô chữ (Entry):*

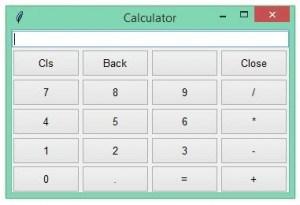
Một vùng mà người dùng có thể gõ chữ vào đó.

*Thanh trượt (Slider):*

Một widget có khả năng điều chỉnh phần hiển thị của một widget khác.

*Khung (Frame):*

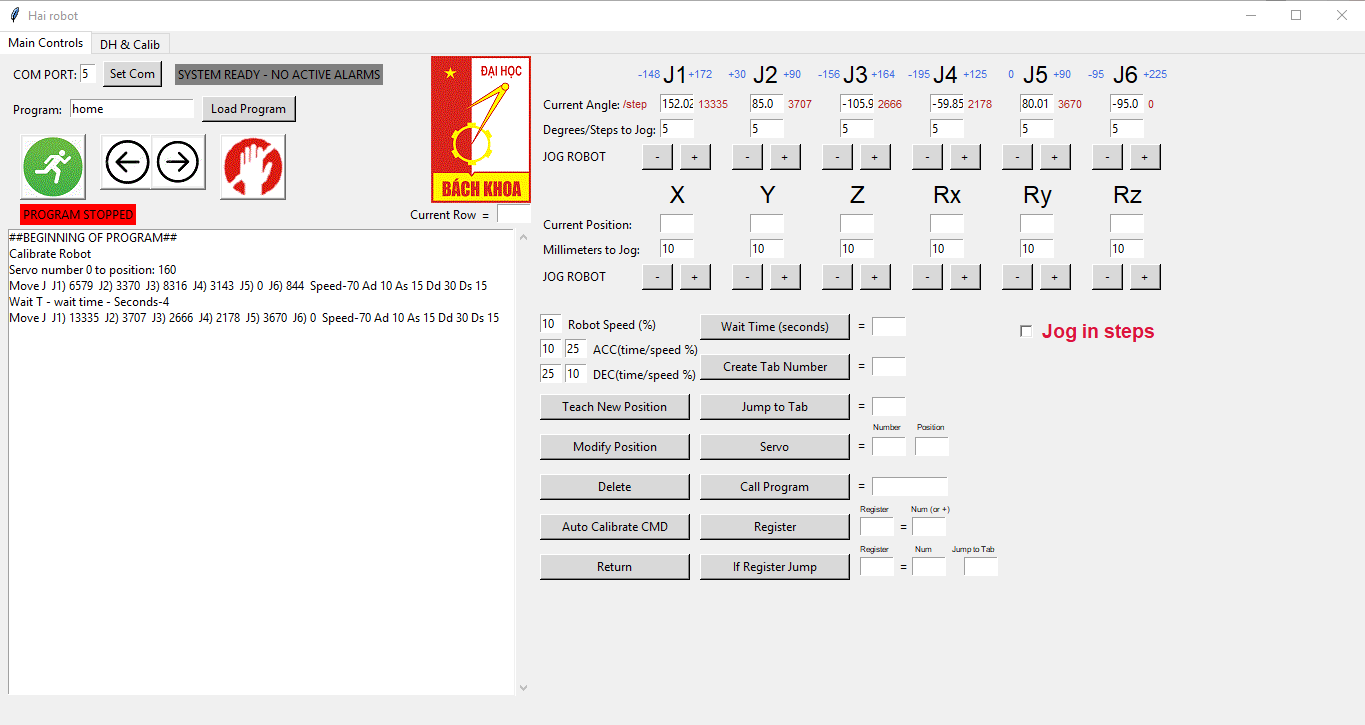
Một widget để chứa, thường luôn hiển thị, và đựng các widget khác ở trong.



*Hình 4.5 : Giao diện máy tính đơn giản với Python Tkinter*

* + 1. **Xây dựng giao diện chương trình**

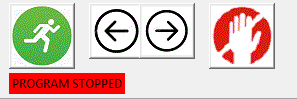
Sử dụng library Tkinter kết hợp với các chức năng được kèm theo như tạo label, text box, button … chúng em đã thiết kế được giao diện chương trình như hình 4.6:



*Hình 4.6 Giao diện điều khiển chính được viết trên Python*

Giao diện điều khiển gồm những khối chính như sau:

* Khối khởi động chương trình



*Hình 4.7 Khối chạy / dừng chương trình*

* Khối điều khiển góc khớp của robot



*Hình 4.8 Khối điều khiển góc khớp của robot*

Current Angle: thể hiện góc khớp hiện tại của robot

Degrees/Steps to Jog : Góc / số bước của động cơ muốn thay đổi trên từng khớp

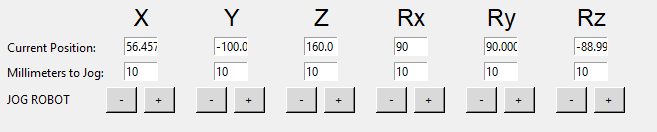
Jog Robot : tăng hoặc giảm giá trị được nhập vào ở “Degrees/Steps to Jog”

Ngoài ra còn có thể tùy chọn để điều khiển động cơ theo số lượng xung băm bằng cách kích hoạt “ Jog in steps “



*Hình 4.9 chức năng “Jog in steps”*

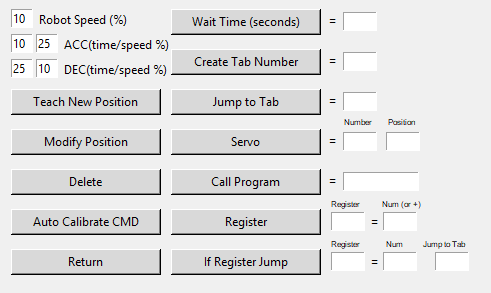
* Khối điều chỉnh vị trí khâu cuối của robot



*Hình 4.10 Khối điều khiển vị trí khâu cuối*

Current Position: vị trí hiện tại của robot theo hệ tọa độ Đề - các

Tương tự như điều khiển góc, ta có thể điều khiển vị trí của khâu cuối tương tự như trên

* Khối tùy chỉnh chương trình điều khiển robot
* 

*Hình 4.11 Khối tùy chỉnh chương trình*

Teach new position: lưu tọa độ hiện tại của robot ( giá trị trong “ Curent Angle “ )

Modify position: chỉnh sửa tọa độ đã lưu trong chương trình

Delete: xóa 1 dòng trong chương trình

Auto calibrate: đưa về vị trí HOME của robot ( quá trình robot calibration )

Return: kết thúc chương trình con và chuyển về chương trình chính

Wait time: đặt thời gian ( seconds ) delay cho robot

Create tab number: tạo điểm nhảy trên chương trình

Jump to tab: nhảy đến điểm mong muốn

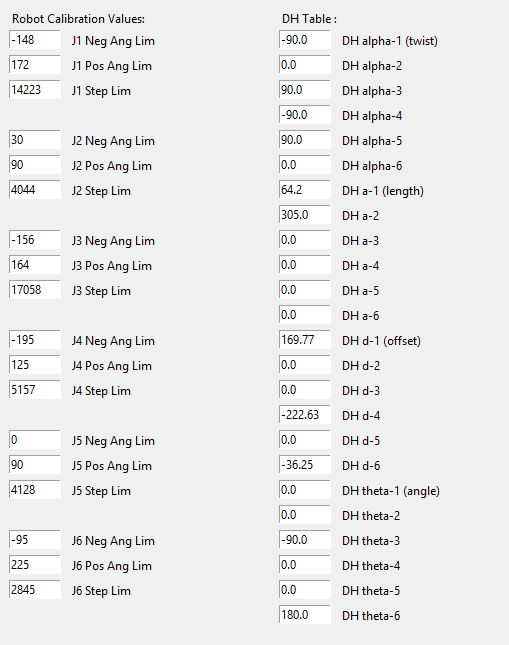
Servo: điều khiển tay gắp robot

Call program: gọi 1 chương trình con

Register: bộ đếm vòng loop

If register jump: nếu bộ đếm đếm đến 1 giá trị mong muốn thì sẽ nhảy đến tab mong muốn

* Khối điều chỉnh thông số của robot

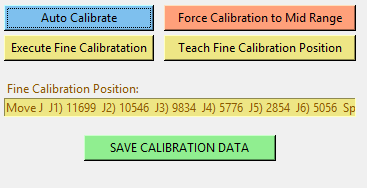


*Hình 4.12 Khối điều chỉnh thông số của robot*

Robot Calibration values : nhập giá trị góc lớn nhất, nhỏ nhất và số xung cần cấp để khớp có thể quay hết khoảng góc.

DH Table : nhập các giá trị an, αn, dn và θn trong bảng Denavit Hartenberg

* Khối calibration



*Hình 4.13 Khối calibraation*

Auto Calibrate : thực hiện calib robot

Force Calibration to mid range : đưa cánh tay robot về góc trung bình của góc lớn nhất và nhỏ nhất

Execute fine calibration : đưa robot về vị trí home được lưu trước đó

Teach fine calibration position : lưu vị trí home bằng tay

Save calibration data : lưu vị trí home vào bộ nhớ, nút này cũng giúp cập nhật giá trị trong khối điều chỉnh thông số robot.

# **KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN**

1. **Kết quả**

* Xây dựng thành công mô hình cánh tay robot 6 bậc tự do.
* Kết nối thành công giữa bộ điều khiển và cánh tay robot.
* Viết chương trình điều khiển Robot.
* Có thể điều khiển cánh tay robot hoạt động theo ý muốn.

1. **Phương hướng phát triển**

* Thiết kế bộ điều khiển kín có bù tác động ngoài
* Tích hợp camera để xử lý hình ảnh, scan 3D
* Tăng độ chính xác của hệ thống

**PHỤ LỤC**

1. **Code chương trình**
   1. **Code Arduino**

* Code điều khiển Servo trên tay gắp gripper :

*while (Serial.available() > 0) // đọc dữ liệu từ cổng COM*

*{*

*char recieved = Serial.read();*

*inData += recieved;*

*if (recieved == '\n')*

*{*

*String function = inData.substring(0, 2);*

*if (function == "SV")*

*{*

*int SVstart = inData.indexOf('V');*

*int POSstart = inData.indexOf('P');*

*int servoNum = inData.substring(SVstart + 1, POSstart).toInt();*

*int servoPOS = inData.substring(POSstart + 1).toInt();*

*if (servoNum == 0)*

*{*

*servo0.write(servoPOS);*

*}*

*Serial.print("Servo Done") }*

* Code cho quá trình Calibration

*if (function == "LL")*

*{*

*int J1start = inData.indexOf('A');*

*int J2start = inData.indexOf('B');*

*int J3start = inData.indexOf('C');*

*int J4start = inData.indexOf('D');*

*int J5start = inData.indexOf('E');*

*int J6start = inData.indexOf('F');*

*int J1step = (inData.substring(J1start + 1, J2start).toInt()) + 200;*

*int J2step = (inData.substring(J2start + 1, J3start).toInt()) + 200;*

*int J3step = (inData.substring(J3start + 1, J4start).toInt()) + 200;*

*int J4step = (inData.substring(J4start + 1, J5start).toInt()) + 200;*

*int J5step = (inData.substring(J5start + 1, J6start).toInt()) + 200;*

*int J6step = (inData.substring(J6start + 1).toInt()) + 200;*

*String J1calStat = "0";*

*// J1 //*

*if (J1rotdir == 1 && J1caldir == 1) {*

*digitalWrite(J1dirPin, HIGH);*

*}*

*else if (J1rotdir == 0 && J1caldir == 1) {*

*digitalWrite(J1dirPin, LOW);*

*}*

*else if (J1rotdir == 1 && J1caldir == 0) {*

*digitalWrite(J1dirPin, LOW);*

*}*

*else if (J1rotdir == 0 && J1caldir == 0) {*

*digitalWrite(J1dirPin, HIGH);*

*}*

*// J2 //*

*if (J2rotdir == 1 && J2caldir == 1) {*

*digitalWrite(J2dirPin, HIGH);*

*}*

*else if (J2rotdir == 0 && J2caldir == 1) {*

*digitalWrite(J2dirPin, LOW);*

*}*

*else if (J2rotdir == 1 && J2caldir == 0) {*

*digitalWrite(J2dirPin, LOW);*

*}*

*else if (J2rotdir == 0 && J2caldir == 0) {*

*digitalWrite(J2dirPin, HIGH);*

*}*

*// J3 //*

*if (J3rotdir == 1 && J3caldir == 1) {*

*digitalWrite(J3dirPin, HIGH);*

*}*

*else if (J3rotdir == 0 && J3caldir == 1) {*

*digitalWrite(J3dirPin, LOW);*

*}*

*else if (J3rotdir == 1 && J3caldir == 0) {*

*digitalWrite(J3dirPin, LOW);*

*}*

*else if (J3rotdir == 0 && J3caldir == 0) {*

*digitalWrite(J3dirPin, HIGH);*

*}*

*// J4 //*

*if (J4rotdir == 1 && J4caldir == 1) {*

*digitalWrite(J4dirPin, HIGH);*

*}*

*else if (J4rotdir == 0 && J4caldir == 1) {*

*digitalWrite(J4dirPin, LOW);*

*}*

*else if (J4rotdir == 1 && J4caldir == 0) {*

*digitalWrite(J4dirPin, LOW);*

*}*

*else if (J4rotdir == 0 && J4caldir == 0) {*

*digitalWrite(J4dirPin, HIGH);*

*}*

*// J5 //*

*if (J5rotdir == 1 && J5caldir == 1) {*

*digitalWrite(J5dirPin, HIGH);*

*}*

*else if (J5rotdir == 0 && J5caldir == 1) {*

*digitalWrite(J5dirPin, LOW);*

*}*

*else if (J5rotdir == 1 && J5caldir == 0) {*

*digitalWrite(J5dirPin, LOW);*

*}*

*else if (J5rotdir == 0 && J5caldir == 0) {*

*digitalWrite(J5dirPin, HIGH);*

*}*

*// J6 //*

*if (J6rotdir == 1 && J6caldir == 1) {*

*digitalWrite(J6dirPin, HIGH);*

*}*

*else if (J6rotdir == 0 && J6caldir == 1) {*

*digitalWrite(J6dirPin, LOW);*

*}*

*else if (J6rotdir == 1 && J6caldir == 0) {*

*digitalWrite(J6dirPin, LOW);*

*}*

*else if (J6rotdir == 0 && J6caldir == 0) {*

*digitalWrite(J6dirPin, HIGH);*

*}*

*int Speed = 1200;*

*while (digitalRead(J1calPin) == LOW && J1done < J1step || digitalRead(J2calPin) == LOW && J2done < J2step || digitalRead(J3calPin) == LOW && J3done < J3step || digitalRead(J4calPin) == LOW && J4done < J4step || digitalRead(J5calPin) == LOW && J5done < J5step || digitalRead(J6calPin) == LOW && J6done < J6step)*

*{*

*if (J1done < J1step && (digitalRead(J1calPin) == LOW))*

*{*

*digitalWrite(J1stepPin, LOW);*

*}*

*delayMicroseconds(5);*

*if (J1done < J1step && (digitalRead(J1calPin) == LOW))*

*{*

*digitalWrite(J1stepPin, HIGH);*

*J1done = ++J1done;*

*}*

*delayMicroseconds(5);*

*if (J2done < J2step && (digitalRead(J2calPin) == LOW))*

*{*

*digitalWrite(J2stepPin, LOW);*

*}*

*delayMicroseconds(5);*

*if (J2done < J2step && (digitalRead(J2calPin) == LOW))*

*{*

*digitalWrite(J2stepPin, HIGH);*

*J2done = ++J2done;*

*}*

*delayMicroseconds(5);*

*if (J3done < J3step && (digitalRead(J3calPin) == LOW))*

*{*

*digitalWrite(J3stepPin, LOW);*

*}*

*delayMicroseconds(5);*

*if (J3done < J3step && (digitalRead(J3calPin) == LOW))*

*{*

*digitalWrite(J3stepPin, HIGH);*

*J3done = ++J3done;*

*}*

*delayMicroseconds(5);*

*if (J4done < J4step && (digitalRead(J4calPin) == LOW))*

*{*

*digitalWrite(J4stepPin, LOW);*

*}*

*delayMicroseconds(5);*

*if (J4done < J4step && (digitalRead(J4calPin) == LOW))*

*{*

*digitalWrite(J4stepPin, HIGH);*

*J4done = ++J4done;*

*}*

*delayMicroseconds(5);*

*if (J5done < J5step && (digitalRead(J5calPin) == LOW))*

*{*

*digitalWrite(J5stepPin, LOW);*

*}*

*delayMicroseconds(5);*

*if (J5done < J5step && (digitalRead(J5calPin) == LOW))*

*{*

*digitalWrite(J5stepPin, HIGH);*

*J5done = ++J5done;;*

*}*

*delayMicroseconds(5);*

*if (J6done < J6step && (digitalRead(J6calPin) == LOW))*

*{*

*digitalWrite(J6stepPin, LOW);*

*}*

*delayMicroseconds(5);*

*if (J6done < J6step && (digitalRead(J6calPin) == LOW))*

*{*

*digitalWrite(J6stepPin, HIGH);*

*J6done = ++J6done;*

*}*

*}*

*delay(500);*

*if (digitalRead(J1calPin) == HIGH && digitalRead(J2calPin) == HIGH && digitalRead(J3calPin) == HIGH && digitalRead(J4calPin) == HIGH && digitalRead(J4calPin) == HIGH && digitalRead(J5calPin) == HIGH && digitalRead(J6calPin) == HIGH)*

*{*

*Serial.println("pass\n");*

*}*

*else*

*{*

*Serial.println("J1fail\n");*

*}*

*inData = ""; //reset bộ nhớ đệm*

*}*

* 1. **Lập trình giao diện**

*import tkinter*

*from tkinter import \**

*root = Tk()*

*root.wm\_title("Hai robot")*

*root.resizable(width=True, height=True)*

*root.geometry('1366x986+0+0')*

*root.runTrue = 0*

*nb = tkinter.ttk.Notebook(root, width=1366, height=698)*

*nb.place(x=0, y=0)*

*tab1 = tkinter.ttk.Frame(nb)*

*nb.add(tab1, text=' Main Controls ')*

*tab2 = tkinter.ttk.Frame(nb)*

*nb.add(tab2, text=' DH & Calib ')*

*curRowLab = Label(tab1, text = "Current Row = ")*

*curRowLab.place(x=407, y=150)*

*almStatusLab = Label(tab1, text = "SYSTEM READY - NO ACTIVE ALARMS", bg = "grey")*

*almStatusLab.place(x=175, y=10)*

*runStatusLab = Label(tab1, text = "PROGRAM STOPPED", bg = "red")*

*runStatusLab.place(x=20, y=150)*

*regEqLab = Label(tab1, font=("Arial", 6), text="Register Num (or +)")*

*regEqLab.place(x=857, y=447)*

*ifregTabJmpLab = Label(tab1, font=("Arial", 6), text="Register Num Jump to Tab")*

*ifregTabJmpLab.place(x=857, y=487)*

*servoLab = Label(tab1, font=("Arial", 6), text="Number Position")*

*servoLab.place(x=870, y=365)*

*ComPortLab = Label(tab1, text = "COM PORT:")*

*ComPortLab.place(x=10, y=10)*

*ProgLab = Label(tab1, text = "Program:")*

*ProgLab.place(x=10, y=45)*

*speedLab = Label(tab1, text = "Robot Speed (%)")*

*speedLab.place(x=565, y=260)*

*ACCLab = Label(tab1, text = "ACC(time/speed %)")*

*ACCLab.place(x=590, y=285)*

*DECLab = Label(tab1, text = "DEC(time/speed %)")*

*DECLab.place(x=590, y=310)*

*J1Lab = Label(tab1, font=("Arial", 18), text = "J1")*

*J1Lab.place(x=660, y=5)*

*J2Lab = Label(tab1, font=("Arial",18), text = "J2")*

*J2Lab.place(x=750, y=5)*

*J3Lab = Label(tab1, font=("Arial", 18), text = "J3")*

*J3Lab.place(x=840, y=5)*

*J4Lab = Label(tab1, font=("Arial", 18), text = "J4")*

*J4Lab.place(x=930, y=5)*

*J5Lab = Label(tab1, font=("Arial", 18), text = "J5")*

*J5Lab.place(x=1020, y=5)*

*J6Lab = Label(tab1, font=("Arial", 18), text = "J6")*

*J6Lab.place(x=1110, y=5)*

*teachInsBut = Button(tab1, bg="grey85", text="Teach New Position", height=1, width=20, command=teachInsertBelSelected)*

*teachInsBut.place(x=540, y=340)*

*teachReplaceBut = Button(tab1, bg="grey85", text="Modify Position", height=1, width=20, command=teachReplaceSelected)*

*teachReplaceBut.place(x=540, y=380)*

*waitTimeBut = Button(tab1, bg="grey85", text="Wait Time (seconds)", height=1, width=20, command=waitTime)*

*waitTimeBut.place(x=700, y=260)*

*tabNumBut = Button(tab1, bg="grey85", text="Create Tab Number", height=1, width=20, command=tabNumber)*

*tabNumBut.place(x=700, y=300)*

*jumpTabBut = Button(tab1, bg="grey85", text="Jump to Tab", height=1, width=20, command=jumpTab)*

*jumpTabBut.place(x=700, y=340)*

*servoBut = Button(tab1, bg="grey85", text="Servo", height=1, width=20, command=Servo)*

*servoBut.place(x=700, y=380)*

*callBut = Button(tab1, bg="grey85", text="Call Program", height=1, width=20, command=insertCallProg)*

*callBut.place(x=700, y=420)*

*returnBut = Button(tab1, bg="grey85", text="Return", height=1, width=20, command=insertReturn)*

*returnBut.place(x=540, y=500)*

*comPortBut = Button(tab1, bg="grey85", text="Set Com", height=0, width=7, command=setCom)*

*comPortBut.place(x=103, y=7)*

*ProgBut = Button(tab1, bg="grey85", text="Load Program", height=0, width=12, command=loadProg)*

*ProgBut.place(x=202, y=42)*

*deleteBut = Button(tab1, bg="grey85", text="Delete", height=1, width=20, command=deleteitem)*

*deleteBut.place(x=540, y=420)*

*runProgBut = Button(tab1, height=60, width=60, command=runProg)*

*playPhoto = PhotoImage(file="play.gif")*

*runProgBut.config(image=playPhoto, width="60", height="60")*

*runProgBut.place(x=20, y=80)*

*stopProgBut = Button(tab1, height=60, width=60, command=stopProg)*

*stopPhoto = PhotoImage(file="stop.gif")*

*stopProgBut.config(image=stopPhoto, width="60", height="60")*

*stopProgBut.place(x=220, y=80)*

*fwdBut = Button(tab1, height=50, width=50, command=stepFwd)*

*fwPhoto = PhotoImage(file="1.gif")*

*fwdBut.config(image=fwPhoto, width="50", height="50")*

*fwdBut.place(x=100, y=80)*

*revBut = Button(tab1, height=50, width=50, command=stepRev)*

*rePhoto = PhotoImage(file="2.gif")*

*revBut.config(image=rePhoto, width="50", height="50")*

*revBut.place(x=150, y=80)*

*RegNumBut = Button(tab1, bg="grey85", text="Register", height=1, width=20, command=insertRegister)*

*RegNumBut.place(x=700, y=460)*

*RegJmpBut = Button(tab1, bg="grey85", text="If Register Jump", height=1, width=20, command=IfRegjumpTab)*

*RegJmpBut.place(x=700, y=500)*

*CalibrateBut = Button(tab1, bg="grey85", text="Auto Calibrate CMD", height=1, width=20, command=insCalibrate)*

*CalibrateBut.place(x=540, y=460)*

* 1. **Code điều khiển chính**

def J1jogNeg():

global JogStepsStat

global J1StepCur

global J1AngCur

almStatusLab.config(text="SYSTEM READY", bg = "grey")

Speed = speedEntryField.get()

ACCdur = ACCdurField.get()

ACCspd = ACCspeedField.get()

DECdur = DECdurField.get()

DECspd = DECspeedField.get()

J1Degs = float(J1jogDegsEntryField.get())

if JogStepsStat.get() == 0:

J1jogSteps = int(J1Degs/J1DegPerStep)

else:

J1jogSteps = J1Degs

J1Degs = J1Degs\*J1DegPerStep

if (J1Degs <= -(J1NegAngLim - J1AngCur)):

ser.write(("MJA1"+str(J1jogSteps)+"S"+Speed+"G"+ACCdur+"H"+ACCspd+"I"+DECdur+"K"+DECspd+"\n").encode())

ser.flushInput()

time.sleep(.2)

ser.read()

J1StepCur = J1StepCur - int(J1jogSteps)

J1AngCur = round(J1NegAngLim + (J1StepCur \* J1DegPerStep),2)

J1curAngEntryField.delete(0, 'end')

J1curAngEntryField.insert(0,str(J1AngCur))

savePosData()

CalcFwdKin()

else:

almStatusLab.config(text="J1 AXIS LIMIT", bg = "red")

DisplaySteps()

def J1jogPos():

global JogStepsStat

global J1StepCur

global J1AngCur

almStatusLab.config(text="SYSTEM READY", bg = "grey")

Speed = speedEntryField.get()

ACCdur = ACCdurField.get()

ACCspd = ACCspeedField.get()

DECdur = DECdurField.get()

DECspd = DECspeedField.get()

J1Degs = float(J1jogDegsEntryField.get())

if JogStepsStat.get() == 0:

J1jogSteps = int(J1Degs/J1DegPerStep)

else:

#switch from degs to steps

J1jogSteps = J1Degs

J1Degs = J1Degs\*J1DegPerStep

if (J1Degs <= (J1PosAngLim - J1AngCur)):

ser.write(("MJA0"+str(J1jogSteps)+"S"+Speed+"G"+ACCdur+"H"+ACCspd+"I"+DECdur+"K"+DECspd+"\n").encode())

ser.flushInput()

time.sleep(.2)

ser.read()

J1StepCur = J1StepCur + int(J1jogSteps)

J1AngCur = round(J1NegAngLim + (J1StepCur \* J1DegPerStep),2)

J1curAngEntryField.delete(0, 'end')

J1curAngEntryField.insert(0,str(J1AngCur))

savePosData()

CalcFwdKin()

else:

almStatusLab.config(text="J1 AXIS LIMIT", bg = "red")

DisplaySteps()

def J2jogNeg():

global JogStepsStat

global J2StepCur

global J2AngCur

almStatusLab.config(text="SYSTEM READY", bg = "grey")

Speed = speedEntryField.get()

ACCdur = ACCdurField.get()

ACCspd = ACCspeedField.get()

DECdur = DECdurField.get()

DECspd = DECspeedField.get()

J2Degs = float(J2jogDegsEntryField.get())

if JogStepsStat.get() == 0:

J2jogSteps = int(J2Degs/J2DegPerStep)

else:

#switch from degs to steps

J2jogSteps = J2Degs

J2Degs = J2Degs\*J2DegPerStep

if (J2Degs <= -(J2NegAngLim - J2AngCur)):

ser.write(("MJB1"+str(J2jogSteps)+"S"+Speed+"G"+ACCdur+"H"+ACCspd+"I"+DECdur+"K"+DECspd+"\n").encode())

ser.flushInput()

time.sleep(.2)

ser.read()

J2StepCur = J2StepCur - int(J2jogSteps)

J2AngCur = round(J2NegAngLim + (J2StepCur \* J2DegPerStep),2)

J2curAngEntryField.delete(0, 'end')

J2curAngEntryField.insert(0,str(J2AngCur))

savePosData()

CalcFwdKin()

else:

almStatusLab.config(text="J2 AXIS LIMIT", bg = "red")

DisplaySteps()

def J2jogPos():

global JogStepsStat

global J2StepCur

global J2AngCur

almStatusLab.config(text="SYSTEM READY", bg = "grey")

Speed = speedEntryField.get()

ACCdur = ACCdurField.get()

ACCspd = ACCspeedField.get()

DECdur = DECdurField.get()

DECspd = DECspeedField.get()

J2Degs = float(J2jogDegsEntryField.get())

if JogStepsStat.get() == 0:

J2jogSteps = int(J2Degs/J2DegPerStep)

else:

#switch from degs to steps

J2jogSteps = J2Degs

J2Degs = J2Degs\*J2DegPerStep

if (J2Degs <= (J2PosAngLim - J2AngCur)):

ser.write(("MJB0"+str(J2jogSteps)+"S"+Speed+"G"+ACCdur+"H"+ACCspd+"I"+DECdur+"K"+DECspd+"\n").encode())

ser.flushInput()

time.sleep(.2)

ser.read()

J2StepCur = J2StepCur + int(J2jogSteps)

J2AngCur = round(J2NegAngLim + (J2StepCur \* J2DegPerStep),2)

J2curAngEntryField.delete(0, 'end')

J2curAngEntryField.insert(0,str(J2AngCur))

savePosData()

CalcFwdKin()

else:

almStatusLab.config(text="J2 AXIS LIMIT", bg = "red")

DisplaySteps()

def J3jogNeg():

global JogStepsStat

global J3StepCur

global J3AngCur

almStatusLab.config(text="SYSTEM READY", bg = "grey")

Speed = speedEntryField.get()

ACCdur = ACCdurField.get()

ACCspd = ACCspeedField.get()

DECdur = DECdurField.get()

DECspd = DECspeedField.get()

J3Degs = float(J3jogDegsEntryField.get())

if JogStepsStat.get() == 0:

J3jogSteps = int(J3Degs/J3DegPerStep)

else:

#switch from degs to steps

J3jogSteps = J3Degs

J3Degs = J3Degs\*J3DegPerStep

if (J3Degs <= -(J3NegAngLim - J3AngCur)):

ser.write(("MJC1"+str(J3jogSteps)+"S"+Speed+"G"+ACCdur+"H"+ACCspd+"I"+DECdur+"K"+DECspd+"\n").encode())

ser.flushInput()

time.sleep(.2)

ser.read()

J3StepCur = J3StepCur - int(J3jogSteps)

J3AngCur = round(J3NegAngLim + (J3StepCur \* J3DegPerStep),2)

J3curAngEntryField.delete(0, 'end')

J3curAngEntryField.insert(0,str(J3AngCur))

savePosData()

CalcFwdKin()

else:

almStatusLab.config(text="J3 AXIS LIMIT", bg = "red")

DisplaySteps()

def J3jogPos():

global JogStepsStat

global J3StepCur

global J3AngCur

almStatusLab.config(text="SYSTEM READY", bg = "grey")

Speed = speedEntryField.get()

ACCdur = ACCdurField.get()

ACCspd = ACCspeedField.get()

DECdur = DECdurField.get()

DECspd = DECspeedField.get()

J3Degs = float(J3jogDegsEntryField.get())

if JogStepsStat.get() == 0:

J3jogSteps = int(J3Degs/J3DegPerStep)

else:

#switch from degs to steps

J3jogSteps = J3Degs

J3Degs = J3Degs\*J3DegPerStep

if (J3Degs <= (J3PosAngLim - J3AngCur)):

ser.write(("MJC0"+str(J3jogSteps)+"S"+Speed+"G"+ACCdur+"H"+ACCspd+"I"+DECdur+"K"+DECspd+"\n").encode())

ser.flushInput()

time.sleep(.2)

ser.read()

J3StepCur = J3StepCur + int(J3jogSteps)

J3AngCur = round(J3NegAngLim + (J3StepCur \* J3DegPerStep),2)

J3curAngEntryField.delete(0, 'end')

J3curAngEntryField.insert(0,str(J3AngCur))

savePosData()

CalcFwdKin()

else:

almStatusLab.config(text="J3 AXIS LIMIT", bg = "red")

DisplaySteps()

def J4jogNeg():

global JogStepsStat

global J4StepCur

global J4AngCur

almStatusLab.config(text="SYSTEM READY", bg = "grey")

Speed = speedEntryField.get()

ACCdur = ACCdurField.get()

ACCspd = ACCspeedField.get()

DECdur = DECdurField.get()

DECspd = DECspeedField.get()

J4Degs = float(J4jogDegsEntryField.get())

if JogStepsStat.get() == 0:

J4jogSteps = int(J4Degs/J4DegPerStep)

else:

#switch from degs to steps

J4jogSteps = J4Degs

J4Degs = J4Degs\*J4DegPerStep

if (J4Degs <= -(J4NegAngLim - J4AngCur)):

ser.write(("MJD0"+str(J4jogSteps)+"S"+Speed+"G"+ACCdur+"H"+ACCspd+"I"+DECdur+"K"+DECspd+"\n").encode())

ser.flushInput()

time.sleep(.2)

ser.read()

J4StepCur = J4StepCur - int(J4jogSteps)

J4AngCur = round(J4NegAngLim + (J4StepCur \* J4DegPerStep),2)

J4curAngEntryField.delete(0, 'end')

J4curAngEntryField.insert(0,str(J4AngCur))

savePosData()

CalcFwdKin()

else:

almStatusLab.config(text="J4 AXIS LIMIT", bg = "red")

DisplaySteps()

def J4jogPos():

global JogStepsStat

global J4StepCur

global J4AngCur

almStatusLab.config(text="SYSTEM READY", bg = "grey")

Speed = speedEntryField.get()

ACCdur = ACCdurField.get()

ACCspd = ACCspeedField.get()

DECdur = DECdurField.get()

DECspd = DECspeedField.get()

J4Degs = float(J4jogDegsEntryField.get())

if JogStepsStat.get() == 0:

J4jogSteps = int(J4Degs/J4DegPerStep)

else:

#switch from degs to steps

J4jogSteps = J4Degs

J4Degs = J4Degs\*J4DegPerStep

if (J4Degs <= (J4PosAngLim - J4AngCur)):

ser.write(("MJD1"+str(J4jogSteps)+"S"+Speed+"G"+ACCdur+"H"+ACCspd+"I"+DECdur+"K"+DECspd+"\n").encode())

ser.flushInput()

time.sleep(.2)

ser.read()

J4StepCur = J4StepCur + int(J4jogSteps)

J4AngCur = round(J4NegAngLim + (J4StepCur \* J4DegPerStep),2)

J4curAngEntryField.delete(0, 'end')

J4curAngEntryField.insert(0,str(J4AngCur))

savePosData()

CalcFwdKin()

else:

almStatusLab.config(text="J4 AXIS LIMIT", bg = "red")

DisplaySteps()

def J5jogNeg():

global JogStepsStat

global J5StepCur

global J5AngCur

almStatusLab.config(text="SYSTEM READY", bg = "grey")

Speed = speedEntryField.get()

ACCdur = ACCdurField.get()

ACCspd = ACCspeedField.get()

DECdur = DECdurField.get()

DECspd = DECspeedField.get()

J5Degs = float(J5jogDegsEntryField.get())

if JogStepsStat.get() == 0:

J5jogSteps = int(J5Degs/J5DegPerStep)

else:

#switch from degs to steps

J5jogSteps = J5Degs

J5Degs = J5Degs\*J5DegPerStep

if (J5Degs <= -(J5NegAngLim - J5AngCur)):

ser.write(("MJE1"+str(J5jogSteps)+"S"+Speed+"G"+ACCdur+"H"+ACCspd+"I"+DECdur+"K"+DECspd+"\n").encode())

ser.flushInput()

time.sleep(.2)

ser.read()

J5StepCur = J5StepCur - int(J5jogSteps)

J5AngCur = round(J5NegAngLim + (J5StepCur \* J5DegPerStep),2)

J5curAngEntryField.delete(0, 'end')

J5curAngEntryField.insert(0,str(J5AngCur))

savePosData()

CalcFwdKin()

else:

almStatusLab.config(text="J5 AXIS LIMIT", bg = "red")

DisplaySteps()

def J5jogPos():

global JogStepsStat

global J5StepCur

global J5AngCur

almStatusLab.config(text="SYSTEM READY", bg = "grey")

Speed = speedEntryField.get()

ACCdur = ACCdurField.get()

ACCspd = ACCspeedField.get()

DECdur = DECdurField.get()

DECspd = DECspeedField.get()

J5Degs = float(J5jogDegsEntryField.get())

if JogStepsStat.get() == 0:

J5jogSteps = int(J5Degs/J5DegPerStep)

else:

#switch from degs to steps

J5jogSteps = J5Degs

J5Degs = J5Degs\*J5DegPerStep

if (J5Degs <= (J5PosAngLim - J5AngCur)):

ser.write(("MJE0"+str(J5jogSteps)+"S"+Speed+"G"+ACCdur+"H"+ACCspd+"I"+DECdur+"K"+DECspd+"\n").encode())

ser.flushInput()

time.sleep(.2)

ser.read()

J5StepCur = J5StepCur + int(J5jogSteps)

J5AngCur = round(J5NegAngLim + (J5StepCur \* J5DegPerStep),2)

J5curAngEntryField.delete(0, 'end')

J5curAngEntryField.insert(0,str(J5AngCur))

savePosData()

CalcFwdKin()

else:

almStatusLab.config(text="J5 AXIS LIMIT", bg = "red")

DisplaySteps()

def J6jogNeg():

global JogStepsStat

global J6StepCur

global J6AngCur

almStatusLab.config(text="SYSTEM READY", bg = "grey")

Speed = speedEntryField.get()

ACCdur = ACCdurField.get()

ACCspd = ACCspeedField.get()

DECdur = DECdurField.get()

DECspd = DECspeedField.get()

J6Degs = float(J6jogDegsEntryField.get())

if JogStepsStat.get() == 0:

J6jogSteps = int(J6Degs/J6DegPerStep)

else:

#switch from degs to steps

J6jogSteps = J6Degs

J6Degs = J6Degs\*J6DegPerStep

if (J6Degs <= -(J6NegAngLim - J6AngCur)):

ser.write(("MJF0"+str(J6jogSteps)+"S"+Speed+"G"+ACCdur+"H"+ACCspd+"I"+DECdur+"K"+DECspd+"\n").encode())

ser.flushInput()

time.sleep(.2)

ser.read()

J6StepCur = J6StepCur - int(J6jogSteps)

J6AngCur = round(J6NegAngLim + (J6StepCur \* J6DegPerStep),2)

J6curAngEntryField.delete(0, 'end')

J6curAngEntryField.insert(0,str(J6AngCur))

savePosData()

CalcFwdKin()

else:

almStatusLab.config(text="J6 AXIS LIMIT", bg = "red")

DisplaySteps()

def J6jogPos():

global JogStepsStat

global J6StepCur

global J6AngCur

almStatusLab.config(text="SYSTEM READY", bg = "grey")

Speed = speedEntryField.get()

ACCdur = ACCdurField.get()

ACCspd = ACCspeedField.get()

DECdur = DECdurField.get()

DECspd = DECspeedField.get()

J6Degs = float(J6jogDegsEntryField.get())

if JogStepsStat.get() == 0:

J6jogSteps = int(J6Degs/J6DegPerStep)

else:

#switch from degs to steps

J6jogSteps = J6Degs

J6Degs = J6Degs\*J6DegPerStep

if (J6Degs <= (J6PosAngLim - J6AngCur)):

ser.write(("MJF1"+str(J6jogSteps)+"S"+Speed+"G"+ACCdur+"H"+ACCspd+"I"+DECdur+"K"+DECspd+"\n").encode())

ser.flushInput()

time.sleep(.2)

ser.read()

J6StepCur = J6StepCur + int(J6jogSteps)

J6AngCur = round(J6NegAngLim + (J6StepCur \* J6DegPerStep),2)

J6curAngEntryField.delete(0, 'end')

J6curAngEntryField.insert(0,str(J6AngCur))

savePosData()

CalcFwdKin()

else:

almStatusLab.config(text="J6 AXIS LIMIT", bg = "red")

DisplaySteps()

# **TÀI LIỆU THAM KHẢO**

1. Bài giảng “Chi tiết máy”, TS. Vũ Lê Huy
2. Bài giảng “Robotics”, PGS.TS. Phan Bùi Khôi
3. Bài giảng “Tính toán và thiết kế robot”, PGS.TS. Phan Bùi Khôi
4. Robot công nghiệp, GS.TSKH Nguyễn Thiện Phúc
5. Tkinter wiki, <https://wiki.python.org/moin/TkInter>
6. PySerial wiki, <https://wiki.python.org/moin/PySerial>
7. Python math wiki, <https://wiki.python.org/moin/BeginnersGuide/Mathematics>
8. Stepper motor, <https://www.omc-stepperonline.com>
9. Driver TB6560, <http://mualinhkien.vn/san-pham/64/module-dieu-khien-dong-co-tb6560.html>
10. GUI controller, <https://github.com/Chris-Annin/AR2>
11. Kinematical and Dynamical Models of 6DOF KUKA robot, <https://cdn.intechweb.org/pdfs/10640.pdf>