BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM

KHOA CƠ KHÍ CHẾ TẠO MÁY

🙠🙟🕮🙝🙢

MÔN HỌC: KỸ THUẬT ROBOT

THIẾT KẾ VÀ CHẾ TẠO

ROBOT ĐÁNH CỜ CARO

GVHD:Nguyễn Trường Thịnh

SVTH MSSV

Nhóm

Trần Hữu Chính 20146480

Thái Việt Cường 20146482

Huỳnh Trọng Trí 20146209

Lớp thứ 6: Tiết 1-4

Mã số lớp HP: ROBO331129\_22\_1\_08

Tp. Hồ Chí Minh, Tháng 12 năm 2022

DANH SÁCH NHÓM

*HỌC KÌ 1, NĂM HỌC 2022-2023*

Đề tài: THIẾT KẾ VÀ CHẾ TẠO ROBOT ĐÁNH CỜ CARO

|  |  |  |  |
| --- | --- | --- | --- |
| STT | **HỌ VÀ TÊN SINH VIÊN** | **MÃ SỐ SINH VIÊN** | **TỶ LỆ %**  **HOÀN THÀNH** |
| 1 | Thái Việt Cường | 20146482 | 100% |
| 2 | Trần Hữu Chính | 20146480 | 100% |
| 3 | Huỳnh Trọng Trí | 20146209 | 100% |

Ghi chú:

* Tỷ lệ % = 100%
* Trưởng nhóm: Thái Việt Cường

Nhận xét của giáo viên:

*Ngày 1 tháng 1 năm 2023*

*Giáo viên chấm điểm*

[LỜI NÓI ĐẦU 1](#_Toc123848183)

[Chương 1: TỔNG QUAN 2](#_Toc123848184)

[1.1.Lịch sử phát triển robot 2](#_Toc123848185)

[1.1.1.Sơ lược quá trình phát triển robot 2](#_Toc123848186)

[1.1.2.Khái niệm robotic 3](#_Toc123848187)

[1.2.Các định nghĩa robot 3](#_Toc123848188)

[1.2.1.Bậc tự do của robot 3](#_Toc123848189)

[1.2.2.Hệ tọa độ động (coordinate frames) 3](#_Toc123848190)

[1.2.3.Trường công tác của robot( workspace or range of motion) 4](#_Toc123848191)

[1.3.Phân loại Robot 4](#_Toc123848192)

[1.3.1.Phân loại theo kết cấu 4](#_Toc123848193)

[1.3.2.Phân loại theo điều khiển 4](#_Toc123848194)

[1.3.3.Phân loại theo ứng dụng 5](#_Toc123848195)

[1.4.Ý tưởng đề tài 5](#_Toc123848196)

[1.5.Mục tiêu đề tài và nội dung nhóm thực hiện 5](#_Toc123848197)

[1.5.1.Mục tiêu 5](#_Toc123848198)

[1.5.2.Giới hạn của đề tài 5](#_Toc123848199)

[1.5.3.Phương pháp nghiên cứu 5](#_Toc123848200)

[1.5.4.Nội dung nhóm thực hiện 5](#_Toc123848201)

[Chương 2: PHÂN TÍCH, LỰA CHỌN PHƯƠNG ÁN THIẾT KẾ 7](#_Toc123848202)

[2.1 Phân tích đề bài 7](#_Toc123848203)

[2.2 Lựa chọn phương án thiết kế 7](#_Toc123848204)

[2.2.1.Robot Cartesian 7](#_Toc123848205)

[2.2.2 Robot tọa độ cầu 7](#_Toc123848206)

[2.2.3 Robot Delta 8](#_Toc123848207)

[2.2.4 Robot Scara 8](#_Toc123848208)

[2.2.5 Ưu điểm và nhược điểm phương án lựa chọn 9](#_Toc123848209)

[Chương 3: CƠ SỞ LÝ THUYẾT VÀ TÍNH TOÁN ĐỘNG HỌC 10](#_Toc123848210)

[3.1.Cơ sở lý thuyết 10](#_Toc123848211)

[3.1.1.Các ma trận quay cơ bản thuần nhất và tịnh tiến thuần nhất 10](#_Toc123848212)

[3.1.2.Phép quay thuần nhất tổng hợp 11](#_Toc123848213)

[3.2.Một số phép quay đặc biệt và ma trận biến đổi thuần nhất 12](#_Toc123848214)

[3.2.1.Các góc Euler và ma trận quay thuần nhất 12](#_Toc123848215)

[3.2.2.Các góc R-P-Y(Roll-Pitch-Yaw) và ma trận quay thuần nhất 14](#_Toc123848216)

[3.3.Bộ thông số Denavit\_Hartenberg và ma trận Denavit\_Hartenberg 15](#_Toc123848217)

[3.3.1.Các tham số động học Denavit-Hartenberg 15](#_Toc123848218)

[3.3.2.Ma trận Denavit-Hartenberg 17](#_Toc123848219)

[3.3.3.Ma trận quan hệ 18](#_Toc123848220)

[3.3.4.Phương trình xác định vị trí khâu thao tác (bàn kẹp) của robot 18](#_Toc123848221)

[3.4.Động học thuận 19](#_Toc123848222)

[3.5.Động học ngược 20](#_Toc123848223)

[3.6.Tính toán động học 21](#_Toc123848224)

[3.6.1.Thông số đầu vào và hệ tọa độ đặt lên robot 21](#_Toc123848225)

[3.6.2.Bảng tham số D-H 21](#_Toc123848226)

[3.6.3.Giải bài toán động học thuận 22](#_Toc123848227)

[3.6.4.Giải bài toán động học nghịch 25](#_Toc123848228)

[Chương 4: ĐIỀU KHIỂN ROBOT SCARA 27](#_Toc123848229)

[4.1 Sơ lược thiết bị và hệ thống điều khiển 27](#_Toc123848230)

[4.2.Sơ đồ kết nối mạch điều khiển 28](#_Toc123848231)

[4.3.Lưu đồ thuật toán điều khiển vị trí trên Arduino 28](#_Toc123848232)

[4.4.Code arduino 29](#_Toc123848233)

[Chương 5: KẾT QUẢ THIẾT KẾ VÀ CHẾ TẠO 47](#_Toc123848234)

[Chương 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 49](#_Toc123848235)

[6.1. Kết luận 49](#_Toc123848236)

[6.2. Kết quả thực nghiệm đánh cờ: 49](#_Toc123848237)

[6.3. Kết quả khi đánh với các nhóm khác: 49](#_Toc123848238)

[6.4. Hướng phát triển 49](#_Toc123848239)

[TÀI LIỆU THAM KHẢO 50](#_Toc123848240)

LỜI NÓI ĐẦU

Bill Gates đã từng viết: “ A Robot in Every Home: The Robotic Future” . Nhìn vào thực tế của sự phát triển công nghệ và kỹ thuật như hiện nay thì đây là một điều hoàn toàn có cơ sở. Thuật ngữ Robot đã và đang tiếp cận một cách dễ dàng với tất cả mọi người trên thế giới, giúp mọi người có thể hình dung ra một cách rõ ràng về một ngày nào đó không xa trong tương lai robot có thể ở khắp mọi nơi đi lại như một công dân giữa chúng ta. Robot được nghiên cứu, chế tạo, phát triển để phục vụ những nhu cầu của con người, nên những chủng loại và kiểu dáng tương đối phong phú và đa dạng. Công nghệ robot trên thế giới đã rất phát triển nhưng con người không muốn dừng lại ở đó và vẫn muốn đẩy mạnh phát triển hơn nữa cho robot, tiến đến giải phóng hoàn toàn sức lao động của con người.

Ứng dụng của robot là rất nhiều, chúng cần thiết và quan trọng đối với nhân loại. Cùng với sự phát triển không ngừng của công nghệ khoa học như hiện nay thì viễn cảnh mà robot có mặt ở mọi nhà là một điều hoàn toàn có thể xảy ra trong tương lai.

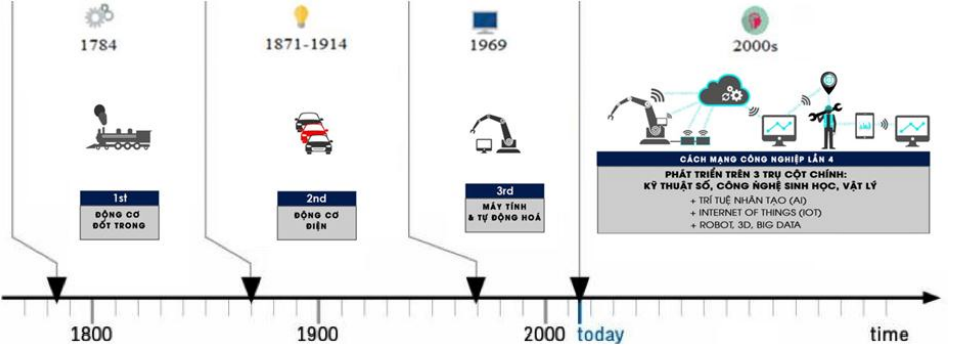
Ở cuộc thi Robot đánh cờ Caro, cảm ơn thầy Nguyễn Trường Thịnh đã tạo điều kiện để chúng em ứng dụng kiến thức về robot vào thực tế, giúp nhóm học hỏi được nhiều kinh nghiệm, thiết kế, chế tạo hơn trong sân chơi này.

Chương 1: TỔNG QUAN

1.1.Lịch sử phát triển robot

1.1.1.Sơ lược quá trình phát triển robot

Trở về lịch sử của quá trình phát triển ngành công nghiệp có thể thấy sự phát triển công nghiệp đã trải qua bốn giai đoạn chính, từ những máy móc đơn giản nhất cho đến những máy móc tinh vi, thông minh và phức tạp nhất.



Hình 1.1 Thời gian các cuộc cách mạng công nghiệp

Giai đoạn I: Diễn ra vào khoảng cuối thế kỉ 18 và đầu thế kỉ 19. Cuộc cách mạng công nghiệp bắt đầu bằng sự phát triển sản xuất hàng hóa của ngành công nghiệp dệt. Sau đó, thương mại phát triển thuận lợi cho sự ra đời của kênh đào giao thông và đường sắt. Động cơ hơi nước đưa đến gia tăng năng suất lao động đột biến. Sự phát triển của máy móc trong hai thập kỉ đầu của thế kỉ 19 đưa đến sự chế tạo máy móc phục vụ cho các ngành sản xuất khác.

Giai đoạn II: Bắt đầu khoảng thập kỉ 1850 và kéo dài đến đầu thập kỉ 1900. Đến cuối thế kỉ 19 nổi bật của cách mạng công nghiệp là động cơ đốt trong và máy móc sử dụng điện.

Giai đoạn III: Bắt đầu khoảng năm 1969 khi các tiến bộ về điện tử xuất hiện. Quá trình này cơ bản hoàn thành nhở những thành tựu khoa học công nghệ cao. Năm 1997 khi cuộc khủng hoảng tài chính Á Châu nổ ra thì đồng thời cũng kết thúc gia đoạn thứ ba.

Giai đoạn IV: Bắt đầu vào đầu thế kỉ 21. Nó được hình thành trên căn bản của những công nghệ mới như ROBOT, NANO, Trí tuệ nhân tạo,... Hiện tại thế giới đang ở trong giai đoạn đầu của cách mạng cộng nghiệp thứ tư. Đây là cơ hội quan trọng để các nước đang phát triển tiến lên để theo kịp thế giới.

1.1.2.Khái niệm robotic

Robotics là một khoa học liên ngành, gồm cơ khí, điện tử, kỹ thuật điều khiển và công nghệ thông tin. Được hiểu là một ngành khoa học có nhiệm vụ nghiên cứu, thiết kế, chế tạo các robot và ứng dụng chúng trong các lĩnh vực khoa học, kỹ thuật, kinh tế,nông nghiệp, quốc phòng và dân sinh.

1.2.Các định nghĩa robot

1.2.1.Bậc tự do của robot

Bậc tự do là số khả năng chuyển động của một cơ cấu (chuyển động quay hoặc tịnh tiến). Để dịch chuyển được một vật thể trong không gian, cơ cấu chấp hành của robot phải đạt được một số bậc tự do. Nói chung cơ hệ của robot là một cơ cấu hở, do đó bậc tự do của nó có thể tính theo công thức:

w = 6n -

Trong đó:

n: Số khâu động.

Pi:Số khớp loại i (i = 1, 2,...,5: Số bậc tự do bị hạn chế). Đối với các cơ cấu có các khâu được nối với nhau bằng khớp quay hoặc tịnh.

1.2.2.Hệ tọa độ động (coordinate frames)

Mỗi robot thường bao gồm nhiều khâu (links) liên kết với nhau qua các khớp (joints), tạo thành một xích động học xuất phát từ một khâu cơ bản (base) đứng yên. Hệ toạ độ gắn với khâu cơ bản gọi là hệ toạ độ cơ bản (hay hệ toạ độ chuẩn). Các hệ toạ độ trung gian khác gắn với các khâu động gọi là hệ toạ độ suy rộng. Trong từng thời điểm hoạt động, các toạ độ suy rộng xác định cấu hình của robot bằng các chuyển dịch dài hoặc các chuyển dịch góc của các khớp tịnh tiến hoặc khớp quay. Các toạ độ suy rộng còn được gọi là biến khớp.

1.2.3.Trường công tác của robot( workspace or range of motion)

Trường công tác (hay vùng làm việc, không gian công tác) của robot là toàn bộ thể tích được quét bởi khâu chấp hành cuối khi robot thực hiện tất cả các chuyển động có thể. Trường công tác bị ràng buộc bởi các thông số hình học của robot cũng như các ràng buộc cơ học của các khớp; ví dụ, một khớp quay có chuyển động nhỏ hơn một góc 360 . Người ta thường dùng hai hình chiếu để mô tả trường công tác của một robot.

1.3.Phân loại Robot

Thế giới robot hiện nay đã rất phong phú và đa dạng, vì vậy phân loại chúng không đơn giản. Có rất nhiều quan điểm phân loại khác nhau. Mỗi quan điểm phục vụ một mục đích riêng. Tuy nhiên, có thể nêu ra đây 3 cách phân loại cơ bản: theo kết cấu, theo điều khiển và theo phạm vi ứng dụng của robot.

1.3.1.Phân loại theo kết cấu

Theo kết cấu (hay theo hình học), người ta phân robot thành các loại: đề các, trụ, cầu, SCARA, kiểu tay người và các dạng khác.

1.3.2.Phân loại theo điều khiển

Có 2 kiểu điều khiển robot: điểu khiển hở và điều khiển kín.

Điều khiển hở, dùng truyền động bước (động cơ điện hoặc động cơ thủy lực, khí nén,… ) mà quãng đường hoặc góc dịch chuyển tỷ lệ với số xung điều khiển. Kiểu điều khiển này đơn giản, nhưng đạt độ chính xác thấp.

Điều khiển kín (hay điều khiển servo), sử dụng tín hiệu phản hồi vị trí để tãng độ chính xác điều khiển. Có 2 kiểu điều khiển servo: điều khiển điểm – điểm và điều khiển theo đường (contour).

Với kiểu điều khiển điểm – điểm, phần công tác dịch chuyển từ điểm này đến điểm kia theo đường thẳng với tốc độ cao (không làm việc). Nó chỉ làm việc tại các điểm dừng. Kiểu điều khiển này được dùng trên các robot hàn điểm, vận chuyển, tán đinh, bắn đinh,…

Điều khiển contour đảm bảo cho phần công tác dịch chuyển theo quỹ đạo bất kỳ, với tốc độ có thể điều khiển được. Có thể gặp kiểu điểu khiển này trên các robot hàn hồ quang, phun sơn.

1.3.3.Phân loại theo ứng dụng

Cách phân loại này dựa vào ứng dụng của robot. Ví dụ, có robot công nghiệp, robot dùng trong nghiên cứu khoa học, robot dùng trong kỹ thuật vũ trụ, robot dùng trong quân sự,…

1.4.Ý tưởng đề tài

Để tạo ra một robot có thể đánh được cờ Caro, với không gian bàn cờ 300x300mm, robot di chuyển được các vị trí trong bàn cờ thì nhóm suy nghĩ đến cánh tay robot để giải quyết vấn đề này.

1.5.Mục tiêu đề tài và nội dung nhóm thực hiện

1.5.1.Mục tiêu

Đề tài nghiên cứu, thiết kế và chế tạo robot có khả năng di chuyển tới các vị trí mà ta muốn và vẽ được hình tròn hoặc dấu X.

Thiết kế phần cơ khí:

- Nghiên cứu, thiết kế hệ thống truyền động.

- Tính toán động lực học nhằm chọn động cơ, cơ cấu truyền động…

- Lựa chọn vật liệu mô hình.

1.5.2.Giới hạn của đề tài

Robot chỉ di chuyển trong vùng không gian của bàn cờ và theo điều khiển chỉ vẽ được hình tròn và dấu X.

1.5.3.Phương pháp nghiên cứu

Tham khảo đọc các lại tài liệu lên quan đến lĩnh vực robot công nghiệp.

Phương pháp thu thập số liệu thực tiễn.

Phương pháp phân tích tổng kết kinh nghiệm

1.5.4.Nội dung nhóm thực hiện

Phân tích lựa chọn và thiết kế cấu trúc động học cơ cấu phù hợp với ứng dụng.

Tính toán động học cho Robot.

Tính toán thiết kế 3D của robot

Tính toán thiết kế phần điện của robot

Chế tạo cơ khí cho robot.

Lắp ráp phần điện

Lập trình điều khiển

Kiểm tra hoạt động robot

Thực nghiệm

Chương 2: PHÂN TÍCH, LỰA CHỌN PHƯƠNG ÁN THIẾT KẾ

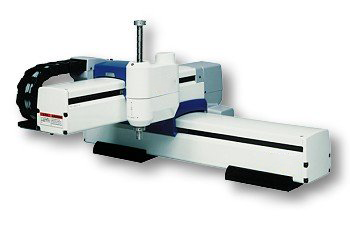
2.1 Phân tích đề bài

Đề bài yêu cầu tính toán thiết kế Robot đánh cờ CARO trên mặt phẳng ngang với tùy ý trong không gian làm việc 30×30cm, từ đó ta có thể hình dung cần 3 bậc tự do để xác định tọa độ một điểm trên một mặt phẳng và có thể vẽ được đường tròn và đoạn thẳng , một hoặc hai bậc tự do để xác định chiều cao trong không gian( tùy thuộc vào loại robot thiết kế), và một bậc tự do để di chuyển bút trên mặt phẳng xy, do đó số bậc tự do tối thiểu mà Robot cần có là 3 bậc tự do.

2.2 Lựa chọn phương án thiết kế

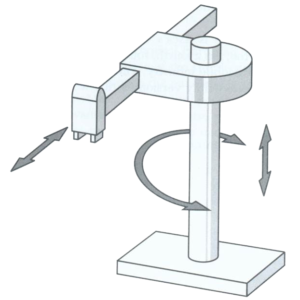
2.2.1.Robot Cartesian

Robot Cartesian, còn được gọi là robot tuyến tính hoặc robot giàn, là các robot công nghiệp hoạt động trên ba trục tuyến tính sử dụng hệ tọa độ Cartesian (X, Y và Z), có nghĩa là chúng di chuyển theo các đường thẳng trên 3 trục (lên xuống, vào và ra và từ bên này sang bên kia). Robot Cartesian là một lựa chọn phổ biến do có tính linh hoạt cao trong cấu hình của chúng, mang đến cho người dùng khả năng điều chỉnh tốc độ, độ chính xác, chiều dài hành trình và kích thước của robot. Robot Cartesian là một trong những loại robot được sử dụng phổ biến nhất cho các ứng dụng công nghiệp và thường được sử dụng cho máy CNC và in 3D.



Hình 2.1 Robot tọa độ Cartesian

2.2.2 Robot tọa độ cầu



Hình 2.2 Cấu hình robot tọa độ cầu

Robot cực, hay robot hình cầu, có một cánh tay với hai khớp quay và một khớp tuyến tính được kết nối với một đế có khớp xoắn. Các trục của robot hoạt động cùng nhau để tạo thành tọa độ cực, cho phép robot có một phong bì làm việc hình cầu. Robot Polar được ghi nhận là một trong những loại robot công nghiệp đầu tiên từng được phát triển. Robot vùng cực thường được sử dụng để đúc khuôn, ép phun, hàn và xử lý vật liệu.

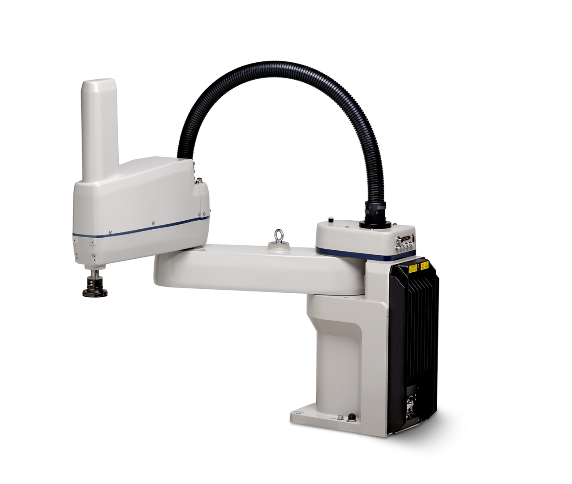
2.2.3 Robot Delta



Hình 2.3 Cấu hình robot Delta

Delta Robots, hay robot song song, sở hữu ba cánh tay được kết nối với một căn cứ duy nhất, được gắn phía trên không gian làm việc. Robot Delta hoạt động theo hình vòm và có thể di chuyển cả tinh tế và chính xác ở tốc độ cao do mỗi khớp của hiệu ứng cuối được điều khiển trực tiếp bởi cả ba cánh tay. Delta Robots thường được sử dụng cho các ứng dụng chọn và đặt nhanh trong các ngành công nghiệp thực phẩm, dược phẩm và điện tử.

2.2.4 Robot Scara

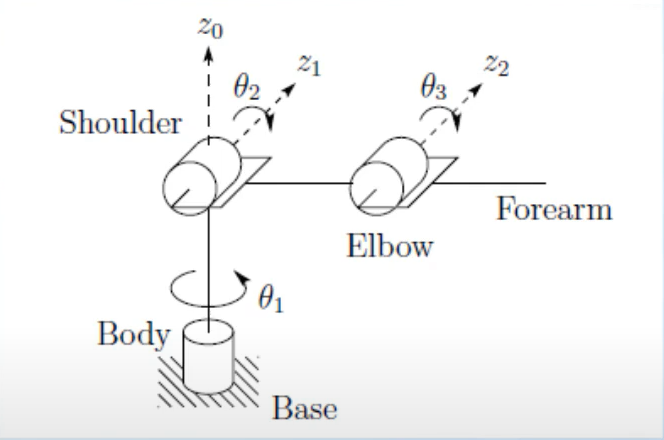


Hình 2.4 Cấu hình robot Scara

SCARA là từ viết tắt của Selective Compliance Assembly Robot Arm hoặc Selective Compliance Articulated Robot Arm. Robot SCARA hoạt động trên 3 trục (X, Y và Z) và cũng có chuyển động quay. Robot SCARA vượt trội trong các chuyển động bên và thường di chuyển nhanh hơn và tích hợp dễ dàng hơn Robot Cartesian. Thông thường, robot SCARA được sử dụng để lắp ráp và xếp pallet, cũng như ứng dụng bio-med.

2.2.5 Ưu điểm và nhược điểm phương án lựa chọn

Để tiết kiệm về mặt kinh tế nhưng vẫn đảm bảo được các yêu cầu của bài toán đặt ra, ta lựa chọn phương án thiết kế Robot 3 bậc tự do. Có thể thấy Robot Scada là loại robot phù hợp nhất với đề bài trên, nhưng kiểu robot này đã được các nhóm đi trước cũng như các nhóm năm nay làm quá nhiều, nên để có một sự mới mẻ và cùng với một tinh thần ham học hỏi thích thử thách không ngại vượt khó, nhóm chúng em đã quyết định chọn loại hình robot khớp nối RRR để thực hiện đề bài mà thầy đã đưa ra.



Hình 2.5 Phương án được lựa chọn cho

Ưu điểm: Robot khớp nối có không gian hoạt động linh hoạt, khả năng đa ứng dụng, độ tin cậy cao và dễ bảo trì.

Nhược điểm: Chi phí chế tạo tốn kém hơn, khi hoạt động ở điểm xa trung tâm cần lực tải lớn nên nếu step không đủ lớn sẽ bị quá tả, và khi viết khó thể luôn song song với mặt phẳng Oxy.

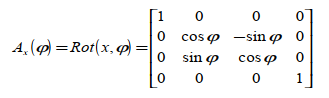
Chương 3: CƠ SỞ LÝ THUYẾT VÀ TÍNH TOÁN ĐỘNG HỌC

3.1.Cơ sở lý thuyết

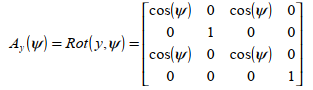
Bài toán động học của robot bao gồm các bài toán về vị trí, bài toán về  
vận tốc, về gia tốc. Trong bài toán về vị trí thì việc xác định vị trí và hướng của điểm tác động cuối tại những thời điểm khác nhau là vấn đề cốt lõi. Để có thể giải quyết được bài toán, thì như ta đã biết robot là một hệ nhiều vật rắn ghép nối với nhau bằng các khớp, chủ yếu là khớp quay và khớp tịnh tiến, do vậy cần phải xác định được các hệ toạ độ gắn với các khâu của robot.

3.1.1.Các ma trận quay cơ bản thuần nhất và tịnh tiến thuần nhất

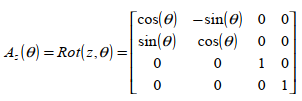
Phép quay quanh trục x một góc



Phép quay quanh trục y một góc

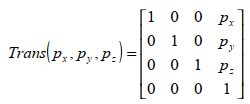


Phép quay quanh trục z một góc



Ma trận tịnh tiến thuần nhất cơ bản

Ngoài ra ta còn có phép tịnh tiến dọc trục x một đoạn là px, trục y một đoạn là py, trục z một đoạn là pz.



3.1.2.Phép quay thuần nhất tổng hợp

Nói chung ma trận biến đổi thuần nhất có thể biểu diễn cả phép quay và phép tịnh tiến của một hệ toạ độ động so với một hệ toạ độ cố định. Một trình tự phép quay và phép tịnh tiến có thể biểu diễn bằng một phép nhân các ma trận thuần nhất cơ bản. Tuy nhiên vì phép nhân ma trận không có tính giao hoỏn nờn trình tự thực hiện phép quay và phép tịnh tiến rất quan trọng. Hơn nữa hệ toạ độ động có thể quay hay tịnh tiến theo vector đơn vị của hệ trục cố định hay của chính nó.

Để giải quyết vấn đề này người ta thường sử dụng thuật toán sau:

a.Khởi gán ma trận biến đổi T=I, trong đó I là ma trận đơn vị. Điều này tương ứng với hai hệ toạ độ trực chuẩn F và M trùng nhau.

b.Biểu diễn các phép quay, tịnh tiến bằng các ma trận biến đổi thuần nhất riêng rẽ.

c.Biểu diễn các phép quay tổng hợp bằng các ma trận thuần nhất cơ bản

d.Nếu hệ toạ độ động M quay hay tịnh tiến dọc theo vector đơn vị của hệ trục cố định F thỡ nhõn trước ma trận T bởi ma trận quay hay tịnh tiến tương ứng.

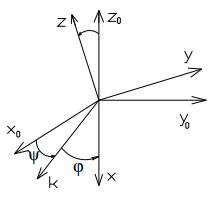
e.Nếu hệ toạ độ động M quay hay tịnh tiến dọc theo vector đơn vị của chính nó thỡ nhõn trước ma trận T bởi ma trận quay hay tịnh tiến tương ứng.

f.Tiếp bước d nếu có thêm nhiều phép quay hay tịnh tiến.

3.2.Một số phép quay đặc biệt và ma trận biến đổi thuần nhất

3.2.1.Các góc Euler và ma trận quay thuần nhất

Ta cs hình vẽ biểu thị ba góc quay Euler như sau



Vị trí của vật rắn B quay quanh điểm O cố định được xác định bởi vị trí của hệ qui chiếu động 0xyz (gắn chặt vào vật rắn B) đối với hệ qui chiếu cố định 0x0y0z0. Giả sử giao của mặt phẳng 0x0y0 và mặt phẳng 0xy là trục OK. Trục OK này được gọi là đường nót.

Ta đưa vào các ký hiệu sau:

+ Góc giữa trục Ox0 và OK là ψ

+ Góc giữa trục OK và Ox là

+ Góc giữa trục Oz0 và Oz là θ

Ba góc ψ, được gọi là ba góc Euler. Như vậy vị trí của điểm B đối với hệ qui chiếu cố định được xác định bởi ba toạ độ suy rộng ψ, .

Phương trình chuyển động của vật rắn quay quanh một điểm cố định có dạng:

ψ =ψ (t);ϕ=ϕ(t); θ=θ(t)

Từ đó ta suy ra, vật rắn quay quanh một điểm cố định có ba bậc tự do. Khi xác định vị trí của vật rắn bằng các góc Euler, ta có thể quay hệ qui chiếu cố định Ox0y0z0 sang hệ qui chiếu động Oxyz bằng ba phép Euler như sau:



















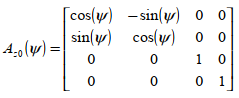
Quay hệ qui chiếu R0  Ox0y0z0 quanh trục Oz0 một góc  để trục Oz0 chuyển tới đường nót OK. Với phép quay này hệ Ox0y0z0 chuyển sang hệ Ox1y1z1 với Ox0  Oz1.

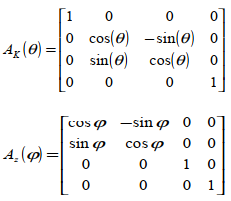
Quay hệ qui chiếu R1 Ox1y1z1 quanh trục Ox1  OK một góc  để Oz0  Oz1 chuyển tới trục Oz2  Oz. Như thế hệ qui chiếu Ox1y1z1 chuyển sang hệ qui chiếu Ox2y2z2 với Ox1  Ox2  OK.

Quay hệ qui chiếu R2  Ox2y2z2 quanh trục Ox2  OK một góc để trục Oz2  OK chuyển tới trục Ox. Như thế hệ qui chiếu Ox2y2z2 chuyển sang hệ qui chiếu Oxyz với Oz2  Oz.

Như thế bằng ba phép quay Euler quanh trục Oz0 một góc y , quanh trục OK một gúc q, quanh trục Oz một góc , hệ qui chiếu Ox0y0z0 chuyển sang hệ qui chiếu Oxyz.

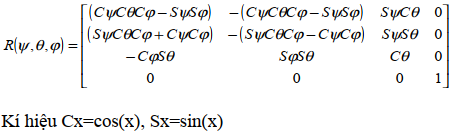
Các ma trận quay ứng với các phép quay Euler có dạng





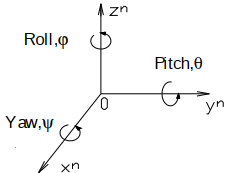
Bây giờ ta xác định ma trận quay hệ qui chiếu Ox0y0z0 sang hệ qui chiếu Oxyz (cũng là ma trận cosin chỉ hướng của hệ qui chiếu Oxyz đối với hệ qui chiếu Ox0y0z0). Ta có:





3.2.2.Các góc R-P-Y(Roll-Pitch-Yaw) và ma trận quay thuần nhất

Việc định hướng khâu cuối có thể thực hiện theo các phép quay Roll- Pitch-Yaw như sau:

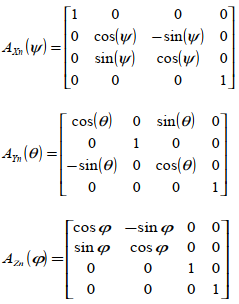


Quay quanh trục Zn một góc

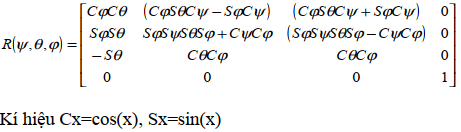
Quay quanh trục Yn một góc θ

Quay quanh trục Xn một góc ψ

Như vậy ta các góc ma trận quay ứng với cỏc gúc quay Roll-Pitch-Yaw:



Tương tự như đối với phép quay Euler, ta cũng có ma trận chuyển hệ qui chiếu như sau:



3.3.Bộ thông số Denavit\_Hartenberg và ma trận Denavit\_Hartenberg

3.3.1.Các tham số động học Denavit-Hartenberg

khâu i

khâu i

i

khâu i-1

Xét hệ các vật rắn nối ghép với nhau bằng các khớp quay và các khớp tịnh tiến. Khi đó quan hệ vị trí giữa hai khâu kế tiếp có thể xác định bởi tham khớp . Trên hình khâu thứ i-1 được nối với khâu thứ i bằng khớp.Trục zi-1 được gọi là trục khớp của khâu thứ i. Tham số thứ nhất i , được gọi là gúc khớp là góc quay của trục xi-1 quanh trục zi-1 đến trục xi’// x.Tham số thứ hai là di, là khoảng cách giữa trục xi’ và trục xi . Nếu khớp i là khớp quay thì i là biến, còn di là hằng số. Nếu khớp i là khớp tịnh tiến thì i là hằng số còn di là biến.

+ Trục zi-1 được chon dọc theo hướng của trục khớp động thứ i.

+ Trục xi-1 được chọn theo đường vuông góc chung của hai trục zi-1 hướng đi từ trục zi-2 sang trục zi-1. Nếu trục zi-1 cắt trục zi-2 thì hướng của trục zi-2 được chọn tuỳ ý.

+ Gốc toạ độ Oi-1 được chọn ttại giao điểm của trục xi-1 và trục zi-1

+ Trục yi-1 được chọn sao cho hệ (0xyz)0 là hệ quy chiếu thuận.

+ Với cách chọn như trên đôi khi các hệ toạ độ khác (0xyz)i-1 không được xác định một cách duy nhất. Vì vậy ta cần có một số bổ sung như sau:

Đối với hệ toạ độ (0xyz)0 được quy ước trên ta mới chỉ chọn được trục z0, còn trục x0 chưa có trong qui ước trên. Ta có thể chọn trục x0 một cách tuỳ ý.

Đối với hệ toạ độ (0xyz)n do không có khớp n+1, nên theo qui ước trên ta không xác định được trục zn. Trục zn không được xác định duy nhất, trong khi trục xn lại được chọn theo phương pháp tuyến của trục zi-1. Trong trường hợp này, nếu khớp n là khớp quay ta nên chọn trục zn song song với trục zn-1. Ngoài ra ta có thể chọn tuỳ ý sao cho hợp lý.

Khi hai trục zn-2 và trục zn-1 song song với nhau, giữa hai trục này có nhiều đường pháp tuyến chung, ta có thể chọn trục xn theo hướng pháp tuyến chung nào cũng được.

Khi khớp quay thứ i là khớp tịnh tiến, về nguyên tắc ta có thể chọn trục zn-1 một cách tuỳ ý. Tuy nhiên trong nhiều trường hợp người ta thường chọn trục xn-1 dọc theo trục của khớp tịnh tiến này.

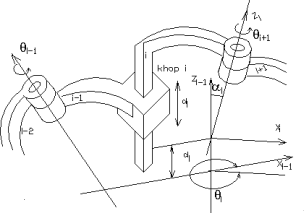
Cách chọn các hệ toạ độ khâu theo ý tưởng của Denavit-Hartenberg đối với trường hợp khớp quay và khớp tịnh tiến.

+ Vị trí của hệ toạ độ khâu (0xyz)i đối với hệ toạ độ khâu (0xyz)i-1, được xác định bởi bốn tham sè Denavit-Hartenberg i ,di,ai và  i như sau

+ Dịch chuyển tịnh tiến dọc theo trục zi-1để gốc toạ độ 0i-1 chuyển đến gốc 0i-1’, giao điểm của trục xi và trục zi-1.

+ Dịch chuyển tịnh tiến dọc theo trục xi để điểm 0i’ chuyển đến điểm 0i

Quay quanh trục xi một góc để trục zi’ chuyển đến trục zi’



3.3.2.Ma trận Denavit-Hartenberg

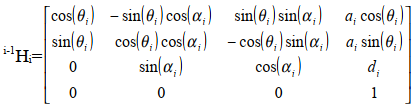
Ta có ma tận DH chuyển từ hệ toạ độ (0xyz)i-1 sang hệ toạ độ thứ (0xyz)i như sau:

* Quay quanh trục zi-1 một góc i với ma trận quay là i-1Ai(zi—1, i).
* Dịch chuyển tịnh tiến dọc theo trục zi-1một đoại di với ma trận tịnh tiến là i- 1Ti(0,,0,di).
* Dịch chuyển tịnh tiến dọc theo trục xi1một đoại ai với ma trận tịnh tiến là i- 1Ti(ai,0,0).
* Quay quanh trục xi một góc  i với ma trận quay là i-1Ai(xi,  i).

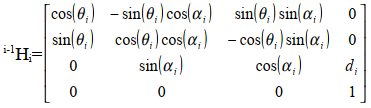
Ma trận của phép biến đổi, ký hiệu là i-1Hi , là tích của bốn ma trận :

i-1Hi = i-1Ai(zi—1, i). 1Ti(0,,0,di). i-1Ti(ai,0,0). i-1Ai(xi,  i)

Đối với khớp quay:



Đối với khớp tịnh tiến:



Định nghĩa: Ma trận được xác định bởi đối với khớp quay hoặc đối với khớp tịnh tiến được gọi là ma trận Denavit-Hartenberg.

3.3.3.Ma trận quan hệ

Chọn hệ toạ độ gắn liền với giá đỡ và các hệ toạ độ động gắn liền với từng khâu động Các hệ toạ độ này được ký hiệu từ 0 đến n.

Một điểm bất kỳ nào đó trong không gian được xác định trong hệ toạ  
độ thứ i bằng bán kính vec tơ ri và trong hệ toạ độ cố định x0y0z0 được xác  
định bằng bán kính vec tơ r0:

r0=0H1. 1H2. 2H3... 0Hiri

Đặt 0Ti=0H1. 1H2. 2H3... 0Hi

Trong đó ma trận 0H1 mô tả vị trí và hướng của khâu đầu tiến với hệ toạ  
độ cố định, ma trận 1H2 mô tả vị trí và hướng của khâu thứ hai so với khâu thứ  
nhất, ma trận i-1Hi mô tả vị trí và hướng của khâu thứ i so với khâu thứ i-1.  
 Như vậy, ma trận tích 0Hi là ma trận mô tả vị trí và hướng của hệ toạ độ thứ i  
so với hệ toạ độ cố định.

3.3.4.Phương trình xác định vị trí khâu thao tác (bàn kẹp) của robot

Áp dụng liên tục phép biến đổi đối với hai khâu liên tiếp đối với robot n khâu ta được phương trình xác định vị trí khâu thao tác (bàn kẹp) của robot.

An =Tn(0)= T1(0) T2(1) T3(2)… Tn(n-1)=T1T2T3…Tn =TE

3.4.Động học thuận

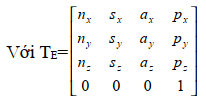
Cơ cấu chấp hành của robot thường là một cơ cấu hở, gồm một chuỗi các khâu nối với nhau bằng các khớp . Để robot có thể thao tác linh hoạt, cơ cấu chấp hành của nó phải cấu tạo sao cho điểm mót của khâu cuối cùng đảm bảo dễ dàng di chuyển theo một quĩ đạo nào đó, đồng thời khâu này có một định hướng nhất định nào đó theo yêu cầu. Khâu cuối cùng thường là bàn kẹp hoặc là khâu gắn liền với dụng cụ làm việc. Điểm mót của khâu cuối cùng là điểm đáng quan tâm nhất vì đó là điểm tác động cuối của robot lên đối tác và được gọi là “điểm tác động cuối”.Ta cần quan tâm đến vị trí của điểm tác động cuối và hướng của khâu cuối trong không gian làm việc của robot.

Gắn vào điểm tác động cuối này một hệ toạ độ động thứ n, gắn với mỗi khâu động một hệ toạ độ động khác và gắn lên giá đỡ một hệ toạ độ cố định. Đánh số kí hiệu các hệ này từ 0 đến n bắt đầu từ giá cố định. Khi khảo sát chuyển động của robot cần biết định vị của điểm tác động cuối và định hướng của khâu cuối trong mọi thời điểm. Nhiều khi lại cần biết cả vận tốc và gia tốc của điểm tác động cuối còng như các điểm khác của robot. Đó là nội dung quan trọng của bài toán động học robot, chúng được xây dựng trên cơ sở thiết lập các mối quan hệ toạ độ động nói trên so với hệ toạ độ cố định.

Ma trận Ti ( xem 4.1) là ma trận mô tả vị trí và hướng của hệ toạ độ thứ i gắn với khâu thứ i so với hệ toạ độ cố định. Trong trường hợp i=n, với n là số hiệu chỉ hệ toạ độ gắn liền với khâu làm việc. Ta có :

0Tn=0H1. 1H2. 2H3... n-1Hn=TE

Là ma trận chỉ hướng và vị trí của điểm tác động cuối.



Là ma trận chỉ hướng và vị trí của điểm tác động cuối.

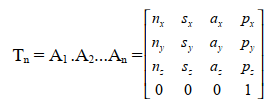
Bài toán động học thuận là bài toán tìm 9 tham sè nx, ny, nx, ax, ay, az, px,  
py, pz theo các biến khớp.

Từ hệ phương trình (5.1) 0Tn= TE ta tìm được nx, ny, nx, ax, ay, az, px, py,  
pz theo các biến khớp bằng cách so sánh các phần tử của hai ma trận tương  
ứng 0Tn và TE.

3.5.Động học ngược

Bài toán động học ngược được đặc biệt quan tâm vì lời giải của nó là cơ sở chủ yếu để xây dựng chương trình điều khiển chuyển động của robot bám theo quĩ đạo cho trước. Các lời giải tìm được cho líp bài toán này hầu như chỉ cho trường hợp riờng, cỏc đặc điểm động học riêng biệt được tận dụng để thiết lập các quan hệ cần thiết khi thiết lập lời giải.

Xuất phát từ phương trình động học cơ bản ta có:



Hai ma trận ở hai vế của phương trình đều là ma trận thuần nhất 4x4. So sánh các phần tử tương ứng của hai ma trận trên ta có 6 phương trình độc lập với các Èn sè qi, (i=1...n). Có 3 trường hợp có thể xảy ra:

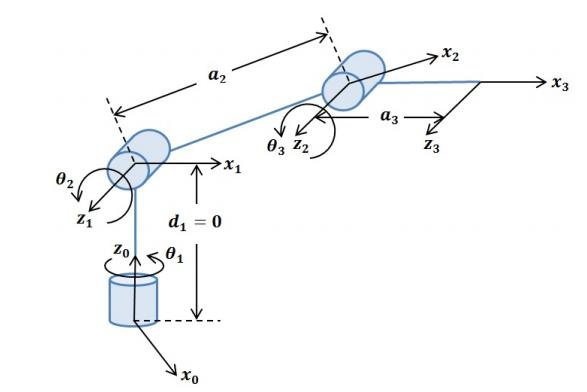
Nếu sè Èn sè ( thường cũng là số bậc tự do của cơ cấu robot) n<6 thì cơ cấu robot chỉ đưa bàn kẹp tới những vị trí và hướng hạn chế mà không đưa bàn kẹp tới vị trí và định hướng bất kì. Trường hợp này được áp dụng khi không có yêu cầu thay đổi một số thông số định vị và định hướng của bàn kẹp.

Nếu n= 6 , tức là số Èn bằng số phương trình thì bộ biến khớp q1...q6 hoàn toàn xác định. Tuy nhiên, lời giải không phải lúc nào cũng dễ dàng tìm ra. Bởi vì nói chung hệ phương trình tìm được thường là siêu việt và việc tìm nghiệm của hệ phương trình này không phải lúc nào cũng hội tụ.

Nếu n>6 , tức là số Èn lớn hơn sè phương trình thì có khả năng có nhiều lời giải, tức là cùng đạt tới một vị trí và định hướng của bàn kẹp có thể có nhiều bộ thông số biến khớp qi.

3.6.Tính toán động học

3.6.1.Thông số đầu vào và hệ tọa độ đặt lên robot



Hình 3.1 Hệ tọa độ đặt vào robot

3.6.2.Bảng tham số D-H

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # |  | l |  | d | Var |
| 1 |  | 0 |  |  |  |
| 2 |  |  | 0 | 0 |  |
| 3 |  |  | 0 | 0 |  |

3.6.3.Giải bài toán động học thuận

Ma trận chuyển trục Ai

Ai =

Thay các thông số trên bảng D-H vào ma trận tiêu chuẩn và nhân chúng lại với nhau.

Sử dụng phần mềm matlab để tìm ra Px, Py hay nói cách khác là tìm ra vị trí tay gắp nằm ở đâu.

Thay thông số của từng khâu trên bảng D-H ta tìm được các ma trận chuyển tọa độ như sau:

Khâu 0 sang khâu 1:

A1 A1 = 0T1 =

Khâu 1 sang khâu 2:

A2 = 1T2 =

Khâu 2 sang khâu 3:

A3 = 1T3 =

Suy ra ma trận chuyển vị từ hệ trục 0 đến hệ trục số 3:

0T3 = 0T1.1T2 2T3

0T3=

Tiếp tục, ta suy ra được tọa độ đầu công tác:

P = =

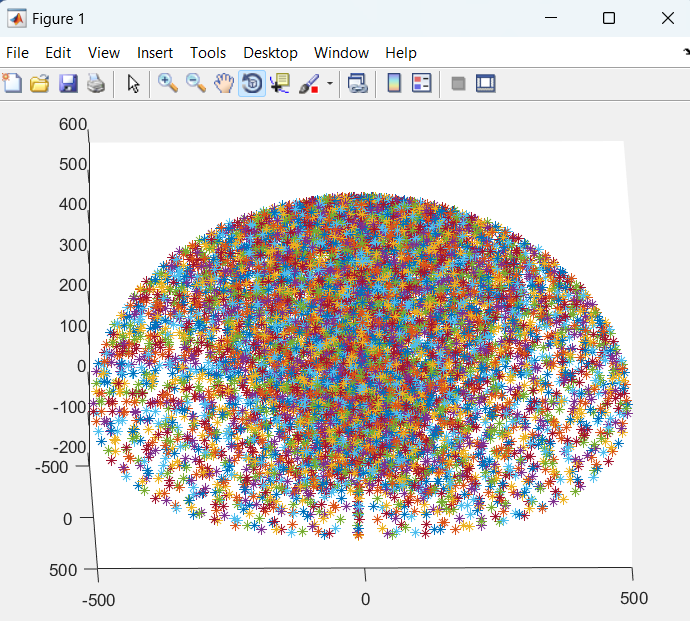
Từ bảng thông số DH ta tính được tọa độ điểm cuối của robot:

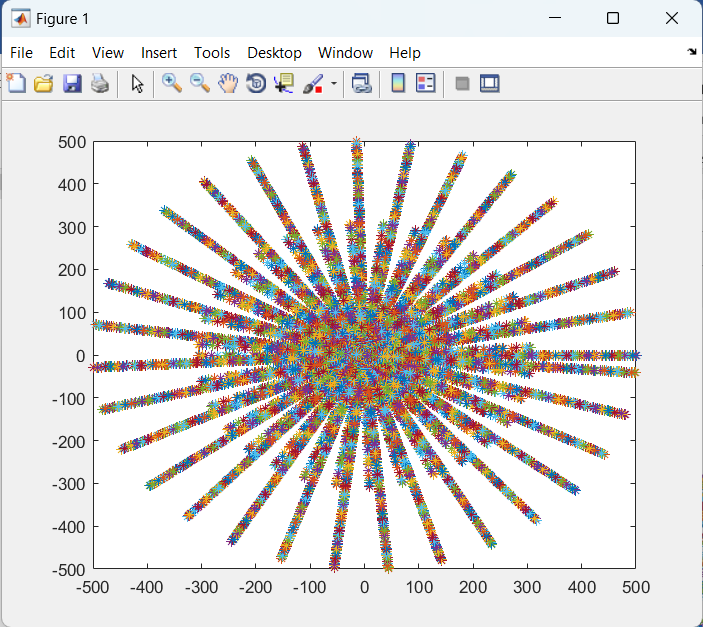
Px =

Py =

Pz =

Mô phỏng quỹ đạo hoạt động trên phần mềm matlab





Hình 3.2 Vùng quỹ đạo chuyển động của robot

*Với code như sau:*

clc

clear all

l1=75;l2=250;l3=250;

syms t1 t2 t3 pi

for t1=0:0.2:2\*pi

for t2=0:0.2:0.58\*pi

for t3=0:0.2:2\*pi

Px= cos(t1)\*(l3\*cos(t2+t3)+cos(t2)\*l2);

Py= sin(t1)\*(l3\*cos(t2+t3)+cos(t2)\*l2);

Pz= l1+l3\*sin(t2+t3)+sin(t2)\*l2;

plot(Px,Py,"\*");

hold on

end

end

end

3.6.4.Giải bài toán động học nghịch

Quy ước: sin() = s

cos() = c

* Góc

tan() = Py / Px

* = atan2(Py,Px)
* Góc

Ta đặt: Py/s1 = l3.c23 + l2.c2 = A

Pz – l1= l3.s23 + l2.s2 = B

Ta lấy : A2 + B2 = l22 + l32 + 2\* l2\*l3\*c3

* c3 = ( A2 + B2 – l22 - l32 ) / (2.l3.l2)
* s3= . Vậy = atan2(s3,c3)
* Góc

A = (l3.c3 + l2 ) .c2 – l3.s3.s2

B= (l3.c3+l2).s2 + l3.s3.c2

Sử dụng phương pháp Crammer

=

=

=

Vậy = atan2(s2,c2)

Chương 4: ĐIỀU KHIỂN ROBOT SCARA

4.1 Sơ lược thiết bị và hệ thống điều khiển

- Nguồn phát 24V DC

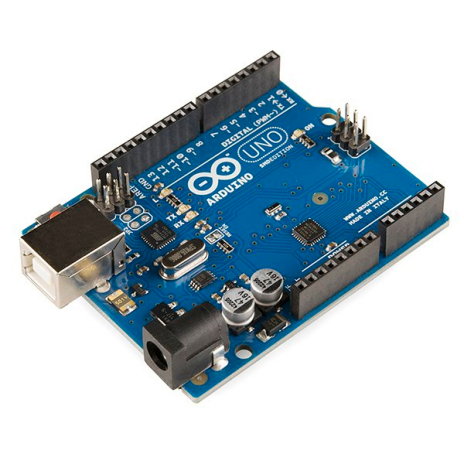
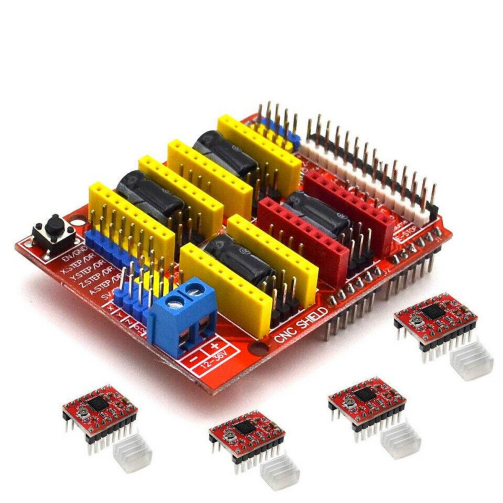
- Drive điều khiển động cơ bước 4988 và DRV8825 và Arduino CNC Shield V3

- Động cơ bước size 42x42x42 ( 2 cái )

- Động cơ bước size 56x56x56 ( 1 cái )

- Khung robot gắn bút để truyền động

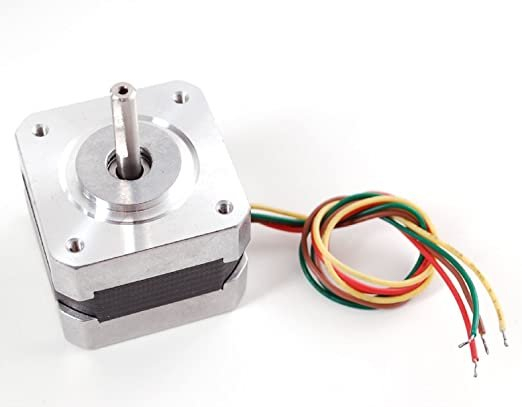
CNC Shield Board + A4988 + DVR 8825



Arduino UNO R3

Nguồn 24V-5V

A



Step motor





*Hình 4-1: Sơ lược thiết bị*

4.2.Sơ đồ kết nối mạch điều khiển



X Axis

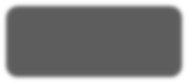
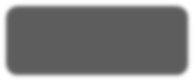
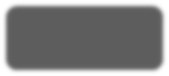
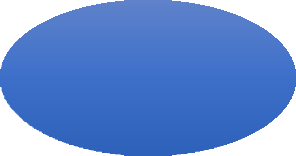
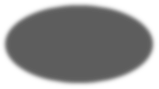
Z Axis

Y Axis

Nguồn tổ ong

*Sơ đồ 4-2: Sơ đồ kết nối dây*

4.3.Lưu đồ thuật toán điều khiển vị trí trên Arduino



Bắt đầu

Nhận thông số tọa độ

Tính toán động học nghịch

Chuyển đổi thông số

góc thành số xung

Điều xung đến Drive động cơ bước

Điều khiển robot đánh cờ

4.4.Code arduino

#include <AccelStepper.h>

#include <MultiStepper.h>

#include <math.h>

#define dirX 5 // pin D5 Uno arduino

#define stepX 2 //pin D2 Uno arduino

#define dirY 6

#define stepY 3

#define dirZ 7

#define stepZ 4

#define en 8

long speed1,speed2,speed3;

long positions[3]; // Array of desired stepper positions

float dem,x,y,L=300;

float l1=75.0;

float l2=250.0;

float l3=250.0;

int x0=90,y0=-84,z0= 138;

float lech\_but=40;

String input=" " ;

String siz=" ";

String hang=" ";

String cot=" ";

char ki\_tu;

int col=0,row=0,sizechess=0;

//float Px, Py, Pz, x, y;

AccelStepper StepperX(1, stepX, dirX); //1 là chế độ dùng Driver

AccelStepper StepperY(1, stepY, dirY); //1 là chế độ dùng Driver

AccelStepper StepperZ(1, stepZ, dirZ); //1 là chế độ dùng Driver

MultiStepper steppers;

////////////////////////////////////////////////////////////////////////////////////////////////////////////

void set\_begin\_position() {

StepperX.setCurrentPosition(0); //Set vị trí hiện tại của động cơ Y là 0

//Cho động cơ X chạy ra đúng vị trí ban đầu

StepperX.moveTo(0\*2\*8.888888889); // 866 cần thực nghiệm lại để xác định chính xác

StepperX.runToPosition(); //chạy đến vị trí đã set

delay(1000);

StepperY.setCurrentPosition(0); //Set vị trí hiện tại của động cơ Y là 0

//Cho động cơ Y chạy ra đúng vị trí ban đầu

StepperY.moveTo(26\*3\*8.888888889); // 866 cần thực nghiệm lại để xác định chính xác

StepperY.runToPosition(); //chạy đến vị trí đã set

delay(1000);

StepperZ.setCurrentPosition(0); //Set vị trí hiện tại của động cơ Y là 0

//Cho động cơ Z chạy ra đúng vị trí ban đầu

StepperZ.moveTo(5\*3\*8.888888889); // -267 cần thực nghiệm lại để xác định chính xác

StepperZ.runToPosition(); //chạy đến vị trí đã set

delay(100);

/\* StepperY.setCurrentPosition(0); //Set vị trí hiện tại của động cơ Y là 0

StepperZ.setCurrentPosition(0); //Set vị trí hiện tại của động cơ Z là 0

positions[0] = 0;

positions[1] = 31\*8.8888888889\*3.75;

positions[2] = 31\*8.8888888889\*3.75;

steppers.moveTo(positions);

steppers.runSpeedToPosition(); // Blocks until all are in position

delay(1000);

positions[2] = 31\*8.8888888889\*3.75;

steppers.moveTo(positions);

steppers.runSpeedToPosition(); // Blocks until all are in position

delay(1000);\*/

}

/////////////////////////////////////////////////////////////////////////////////////////////////

void setup()

{

pinMode(en, OUTPUT);

set\_vantoc\_3\_Step(500,500,500);

set\_giatoc\_3\_Step(1000,1500,1500);

/////////////////////////////////////////////////////

//Start

Serial.begin(9600);

Serial.print("Nhập <1> nếu bạn muốn ROBOT bắt đầu chạy= ");

while (Serial.available() == 0) {} //wait for data available

String str5 = Serial.readString(); //read until timeout

str5.trim(); // remove any \r \n whitespace at the end of the String

Serial.println(str5);

if(str5=="1"){

set\_begin\_position();

}

///////////////////////////////////////////////////

// Give stepX,Y,Z to MultiStepper to manage

steppers.addStepper(StepperX);

steppers.addStepper(StepperY);

steppers.addStepper(StepperZ);

digitalWrite(en, LOW);

////////////////////////////////////////////////////

/\*// Nhap speed stepX

Serial.print("Nhập vận tốc bàn quay(stepX) = ");

while (Serial.available() == 0) {} //wait for data available

String str6 = Serial.readString(); //read until timeout

str6.trim(); // remove any \r \n whitespace at the end of the String

speed1 = atoi(str6.c\_str());

Serial.println(speed1);\*/

}

void loop() // Driver: 1/16 step , Motor 1.8' => 3200 bước = 1 vòng 360'

{

Serial.println(" NHAP : SIZ\_POS\_X(O) (EX: 3\_11\_X) ");

while (Serial.available()==0);

input=Serial.readString();

siz[0]=input[0];

hang[0]=input[2];

cot[0]=input[3];

ki\_tu= input[5];

sizechess=siz.toInt();

col=cot.toInt();

row=hang.toInt();

Serial.println(sizechess);

Serial.println(col);

Serial.println(row);

delay(500);

if (ki\_tu=='X')

{

draw\_X();

}

if (ki\_tu=='O')

{

draw\_O(sizechess,col,row);

}

delay(500);

}

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void draw\_O(int siz,int col,int row)

{

int tamX= ((L/siz)\*col -L/(2\*siz) )-150 ;

int tamY= (L/siz)\*row -L/(2\*siz)+120 ;

float R= 0.50\*float(L/(2\*siz));

float px=0,py=0,pz=75,c1=0,s1=0,c2=0,s2=0,c3=0,s3=0,A,B,goc\_lech,l\_lech,them;

float t1=0,t2=0,t3=0;

StepperX.setCurrentPosition(0); //Set vị trí hiện tại của động cơ X là 0

StepperY.setCurrentPosition(0); //Set vị trí hiện tại của động cơ Y là 0

StepperZ.setCurrentPosition(0); //Set vị trí hiện tại của động cơ Z là 0

for (float t = 0 ; t <= 2\*PI + 0.1; t = t + 0.08)

{

if(t==0){

set\_vantoc\_3\_Step(300,300,300);

set\_giatoc\_3\_Step(2000,3500,3500);

}

if(t>0){

pz=111;

set\_vantoc\_3\_Step(100,150,150);

set\_giatoc\_3\_Step(800,1000,1000);

}

if ( row <5 ){

them=5;

}

if ( row >5 ){

them=-5;

}

px = tamX +R\*cos(t);

py = tamY +R\*sin(t);

l\_lech= sqrtf((px\*px)+(py\*py));

goc\_lech= atan2(lech\_but,l\_lech);

goc\_lech= goc\_lech \*180/PI;

Serial.print("Goc lech = ");

Serial.println(goc\_lech);

t1 = atan2(py,px);

c1= cos(t1);

s1= sin(t1);

A= px /c1;

B= pz-l1;

c3= (sq(A) +sq(B) -sq(l2) - sq(l3))/(2\*l2\*l3);

s3 = sqrtf(abs(1-sq(c3)));

t3= atan2(s3,c3);

c2= (A\*(l2 +l3\*c3 )+ B\*l3\*s3)/(sq(l2+l3\*c3) +sq(l3\*s3));

s2 = (B\*(l2+l3\*c3)- A\*l3\*s3)/(sq(l2+l3\*c3) +sq(l3\*s3));

t2= atan2(s2,c2);

Serial.println(t2);

t1=(t1\*180/PI)-x0+goc\_lech;

t2=(t2\*180/PI)-y0;

t3=(t3\*180/PI)-z0-them;

Serial.print("Goc theta 1 = ");

Serial.println(t1);

Serial.print("Goc theta 2 = ");

Serial.println(t2);

Serial.print("Goc theta 3 = ");

Serial.println(t3);

//tinh toan goc quay//

positions[0] = t1\*8.8888888889\*2;

positions[1] = t2\*8.8888888889\*3;

positions[2] = t3\*8.8888888889\*3;

steppers.moveTo(positions);

steppers.runSpeedToPosition(); // Blocks until all are in position

if ( row <5 ){

pz=107;

}

if ( row >5 ){

pz=107;

}

if (t==0) delay(1000);

}

delay(1000);

Go\_back();

}

/\*

Nhap\_toa\_do\_matrix();

//Robot hạ bút để viết

int angle\_pen=5;

StepperY.moveTo( (t2+angle\_pen)\*8.8888888889\*3); //<cần thực nghiệm lại để xác định chính xác>

StepperY.runToPosition(); //chạy đến vị trí đã set

delay(1000);

for(float t=0;t<=2\*PI;t=t+0.01)

{

/// Code đánh O|X

Danh\_O(t);

}

//Back

Go\_back();

}

\*/

///////////////////////////////////////////////////////////////////////////////////////////////////////////

void draw\_X()

{

StepperX.setCurrentPosition(0); //Set vị trí hiện tại của động cơ X là 0

StepperY.setCurrentPosition(0); //Set vị trí hiện tại của động cơ Y là 0

StepperZ.setCurrentPosition(0); //Set vị trí hiện tại của động cơ Z là 0

draw\_phai(sizechess,col,row);

Serial.print("XONG1/////////////////////////////////////////////////////////////////////////////////////////");

draw\_trai(sizechess,col,row);

}

///////////////////////////////////////////////////////////////////////////////////////////////////////////

void draw\_phai(int siz,int col,int row)

{

float d= 0.4\*float(L/(2\*siz)); //giam kich thuoc chu x

float xa= (L/siz)\*(col-1)+d;

float xb= (L/siz)\*col-d;

float ya=(L/siz)\*(row-1)+d;

float yb= (L/siz)\*row -d ;

Serial.println(xa);

Serial.println(xb);

Serial.println(ya);

Serial.println(yb);

float px=0,py=0,pz=75,c1=0,s1=0,c2=0,s2=0,c3=0,s3=0,A,B,goc\_lech,l\_lech,them;

float t1=0,t2=0,t3=0;

for (float t = 0 ; t <= 1; t = t + 0.09)

{

if(t==0){

set\_vantoc\_3\_Step(300,300,300);

set\_giatoc\_3\_Step(2000,3500,3500);

}

if(t!=0){

pz=113;

set\_vantoc\_3\_Step(100,150,150);

set\_giatoc\_3\_Step(800,1000,1000);

}

if ( row ==1 ){

them=9;

pz=115;

}

if (row!=1) {

them=6;

}

px =(xa) +(xb-xa)\*t-150;

py =(ya) +(yb-ya)\*t+120;

l\_lech= sqrtf((px\*px)+(py\*py));

goc\_lech= atan2(lech\_but,l\_lech);

goc\_lech= goc\_lech \*180/PI;

Serial.print("Goc lech = ");

Serial.println(goc\_lech);

t1 = atan2(py,px);

c1= cos(t1);

s1= sin(t1);

A= px /c1;

B= pz-l1;

c3= (sq(A) +sq(B) -sq(l2) - sq(l3))/(2\*l2\*l3);

s3 = sqrtf(abs(1-sq(c3)));

t3= atan2(s3,c3);

c2= (A\*(l2 +l3\*c3 )+ B\*l3\*s3)/(sq(l2+l3\*c3) +sq(l3\*s3));

s2 = (B\*(l2+l3\*c3)- A\*l3\*s3)/(sq(l2+l3\*c3) +sq(l3\*s3));

t2= atan2(s2,c2);

t1=(t1\*180/PI)-x0+goc\_lech;

t2=(t2\*180/PI)-y0;

t3=(t3\*180/PI)-z0-them;

Serial.print("Goc theta 1 = ");

Serial.println(t1);

Serial.print("Goc theta 2 = ");

Serial.println(t2);

Serial.print("Goc theta 3 = ");

Serial.println(t3);

//tinh toan goc quay//

positions[0] = t1\*8.8888888889\*2;

positions[1] = t2\*8.8888888889\*3;

positions[2] = t3\*8.8888888889\*3;

steppers.moveTo(positions);

steppers.runSpeedToPosition(); // Blocks until all are in position

if(t==0){

delay(1000);

//Ha\_but

}

if (row!=1) {

pz=115;

}

}

Nang\_ha\_but(t2);

}

////////////////////////////////////////////////////////////////////////////////////////////////////////

void draw\_trai(int siz,int col,int row)

{

float d= 0.4\*float(L/(2\*siz)); //giam kich thuoc chu x

float xa= (L/siz)\*(col-1)+d;

float xb= (L/siz)\*col-d;

float ya=(L/siz)\*(row-1)+d;

float yb= (L/siz)\*row -d ;

float px=0,py=0, pz=90,c1=0,s1=0,c2=0,s2=0,c3=0,s3=0,A,B,goc\_lech,l\_lech,them;

float t1=0,t2=0,t3=0;

for (float t = 0 ; t <= 1; t = t + 0.09)

{

if(t==0){

set\_vantoc\_3\_Step(200,200,200);

set\_giatoc\_3\_Step(2000,3500,3500);

}

if(t!=0){

pz=113;

set\_vantoc\_3\_Step(100,150,150);

set\_giatoc\_3\_Step(800,1000,1000);

}

if ( row ==1 ){

them=6;

pz=115;

}

if (row!=1) {

them=6;

}

px =(xa) +(xb-xa)\*t-150;

py =(yb) +(ya-yb)\*t+120;

l\_lech= sqrtf((px\*px)+(py\*py));

goc\_lech= atan2(lech\_but,l\_lech);

goc\_lech= goc\_lech \*180/PI;

Serial.print("Goc lech = ");

Serial.println(goc\_lech);

t1 = atan2(py,px);

c1= cos(t1);

s1= sin(t1);

A= px /c1;

B= pz-l1;

c3= (sq(A) +sq(B) -sq(l2) - sq(l3))/(2\*l2\*l3);

s3 = sqrtf(abs(1-sq(c3)));

t3= atan2(s3,c3);

c2= (A\*(l2 +l3\*c3 )+ B\*l3\*s3)/(sq(l2+l3\*c3) +sq(l3\*s3));

s2 = (B\*(l2+l3\*c3)- A\*l3\*s3)/(sq(l2+l3\*c3) +sq(l3\*s3));

t2= atan2(s2,c2);

t1=(t1\*180/PI)-x0+goc\_lech;

t2=(t2\*180/PI)-y0;

t3=(t3\*180/PI)-z0-them;

Serial.print("Goc theta 1 = ");

Serial.println(t1);

Serial.print("Goc theta 2 = ");

Serial.println(t2);

Serial.print("Goc theta 3 = ");

Serial.println(t3);

//tinh toan goc quay//

positions[0] = t1\*8.8888888889\*2;

positions[1] = t2\*8.8888888889\*3;

positions[2] = t3\*8.8888888889\*3;

steppers.moveTo(positions);

steppers.runSpeedToPosition(); // Blocks until all are in position

if(t==0){

Serial.print("BAT\_DAU/////////////////////////////////////////////////////////////////////////////////////////");

delay(2000);

//Ha\_but

}

if (row!=1) {

pz=115;

}

}

Nang\_ha\_but(t2);

Go\_back() ;

}

///////////////////////////////////////////////////////////////////////////////////////////////////////

void Go\_back(){

set\_vantoc\_3\_Step(400,400,400);

set\_giatoc\_3\_Step(3500,3500,2500);

positions[0] = 0;

positions[1] = 0;

positions[2] = 0;

steppers.moveTo(positions);

steppers.runSpeedToPosition(); // Blocks until all are in position

delay(1000);

}

void Nang\_ha\_but(int h){

h=h-5;

positions[1] = h\*8.8888888889\*3;

steppers.moveTo(positions);

steppers.runSpeedToPosition(); // Blocks until all are in position

delay(100);

}

void set\_vantoc\_3\_Step(int v\_stepX,int v\_stepY,int v\_stepZ ){

StepperX.setMaxSpeed(v\_stepX);

StepperY.setMaxSpeed(v\_stepY);

StepperZ.setMaxSpeed(v\_stepZ);

}

void set\_giatoc\_3\_Step(int a\_stepX,int a\_stepY,int a\_stepZ ){

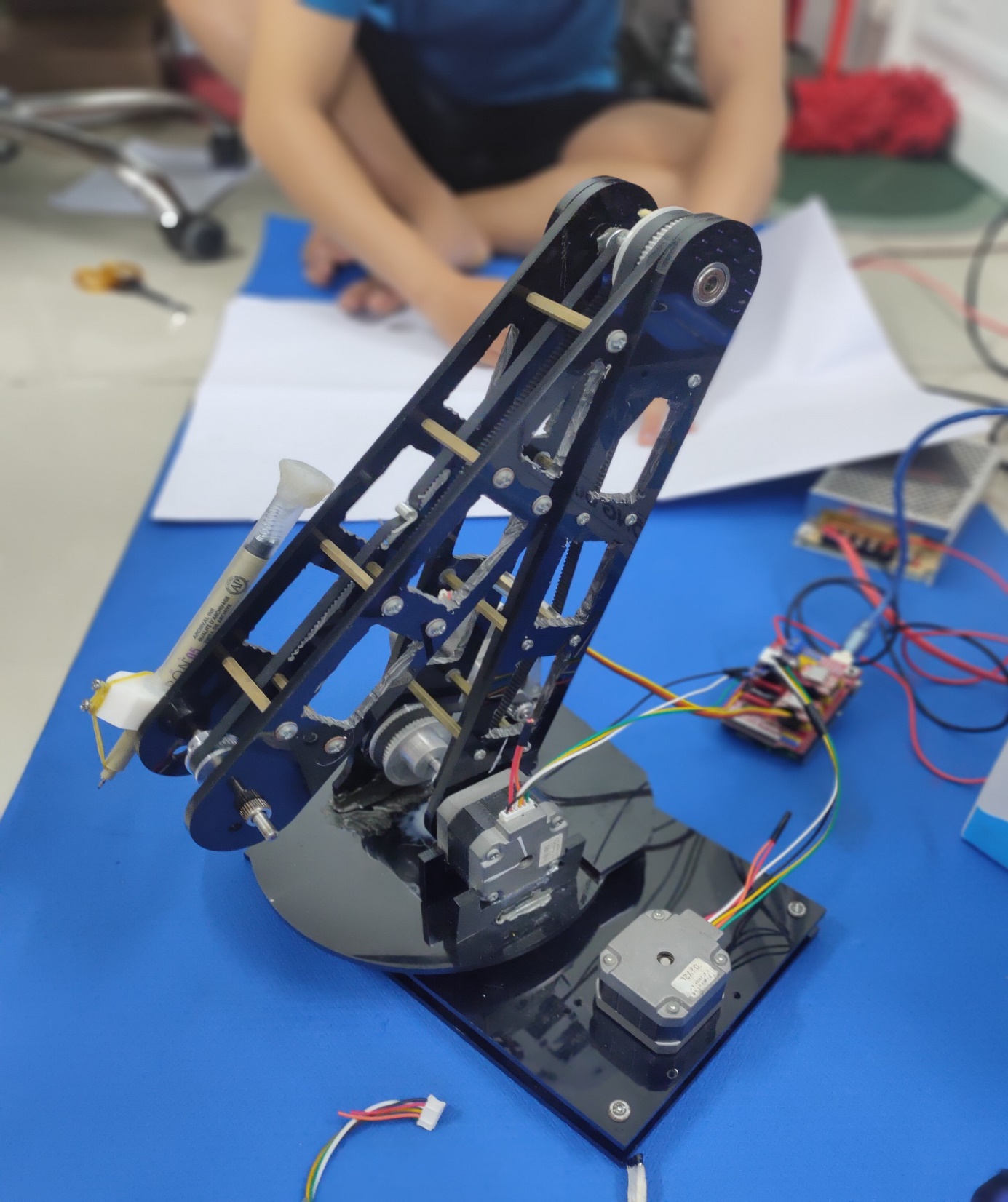
StepperX.setAcceleration(a\_stepX);

StepperY.setAcceleration(a\_stepY);

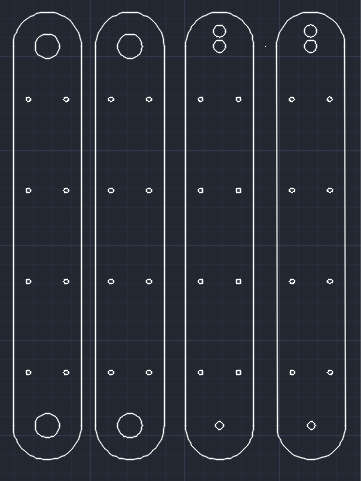
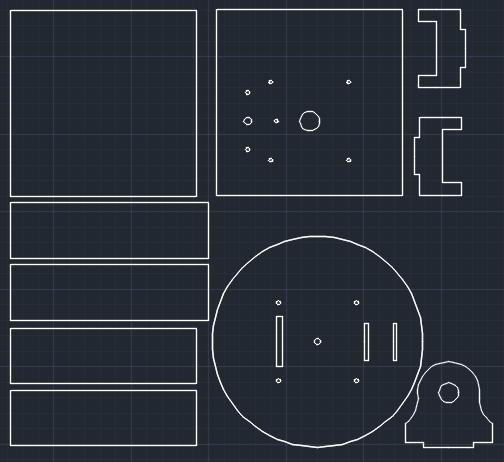
StepperZ.setAcceleration(a\_stepZ);

}

Chương 5: KẾT QUẢ THIẾT KẾ VÀ CHẾ TẠO



*Hình 5-1: Sản phầm thực tế*



*Hình 5-2: Vẽ autocad khung robot*

Chương 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1. Kết luận

Trong thời gian thực hiện Thiết kế và chế tạo robot đánh cờ Caro, nhóm đã nghiên cứu, tìm hiểu và thực hiện các vấn đề:

- Nghiên cứu thiết kế và tính toán lựa chọn cơ cấu truyền động, cũng như thiết kế phần cơ khí và chế tạo robot.

- Đã đạt được những mục tiêu:

- Thiết kế bản vẽ mô hình 3D của Robot.

- Tiến hành chế tạo gia công robot.

- Thiết kế phần điện và lập trình điều khiển robot.

6.2. Kết quả thực nghiệm đánh cờ:

Việc kiểm nghiệm với thực tế, nhóm thực hiện bằng việc kiểm tra độ chính xác theo tọa độ ô trên bàn cờ của đầu công tác.

Thực hiện kiểm tra khả năng làm việc của Robot ở các bàn cờ cấp độ khác nhau và liên tục. Từ đó rút các sai lệch.

6.3. Kết quả khi đánh với các nhóm khác:

Với nhiều cấp độ bàn cờ, robot di chuyển đánh cờ nhanh, chính xác, không vướng vào các robot của nhóm khác.

Kết quả chung cuộc: vượt qua được 2 vòng loại 3x3, 5x5, 7x7, 9x9.

Link video báo cáo nhóm: <https://youtu.be/aY8h3Haz8Bk>

6.4. Hướng phát triển

Cải tiến cho Robot gọn nhẹ hơn và linh hoạt, ít chi phí, cơ cấu cánh tay vững chắc hơn.

Tích hợp thêm các tính năng khác như: Điều khiển đánh caro bằng giọng nói, điều khiển qua app trên điện thoại , tạo giao diện điều khiển để dễ sử dụng hơn,…..

TÀI LIỆU THAM KHẢO

1.PGS.TS. Nguyễn Trường Thịnh (2014), *Giáo trình Kĩ thuật robot*, Nhà xuất bản đại học quốc gia TP.HCM.