**ĐẠI HỌC THÁI NGUYÊN**

**TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP**

-----🙞🙜🕮🙞🙜-----



**LUẬN VĂN THẠC SĨ KỸ THUẬT**

**CHUYÊN NGÀNH: KỸ THUẬT ĐIỀU KHIỂN VÀ TỰ ĐỘNG HÓA**

**ĐỀ TÀI: ĐIỀU KHIỂN ROBOT 5 BẬC TỰ DO BẰNG**

**NHẬN DẠNG GIỌNG NÓI**

**Người hướng dẫn khoa học : PGS.TS. LẠI KHẮC LÃI**

**Người thực hiện : NGÔ THANH HẢI**

**Thái Nguyên - 2017**

|  |  |
| --- | --- |
| **ĐẠI HỌC THÁI NGUYÊN**  **TRƯỜNG ĐẠI HỌC**  **KỸ THUẬT CÔNG NGHIỆP**  --------------------------------- | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập - Tự do - Hạnh phúc**  **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*** |

**LUẬN VĂN THẠC SĨ KỸ THUẬT**

*Người hướng dẫn* : PGS.TS Lại Khắc Lãi

*Họ và tên học viên* : Ngô Thanh Hải

*Đơn vị công tác* : Khoa Điện - Trường Đại học Kỹ thuật Công nghiệp

*Cơ sở đào tạo* :Trường Đại học Kỹ thuật Công nghiệp - Đại học Thái Nguyên

*Ngành đào tạo* : Kỹ thuật điều khiển và tự động hóa

*Khoá học* : 2015 - 2017

**Tên đề tài:**

**“Điều khiển robot 5 bậc tự do bằng nhận dạng giọng nói’’**

|  |  |  |
| --- | --- | --- |
| **PHÒNG ĐÀO TẠO**  **TS. Đặng Danh Hoằng** | **KHOA CHUYÊN MÔN** | **NGƯỜI HƯỚNG DẪN**  **PGS.TS Lại Khắc Lãi** |

**LỜI MỞ ĐẦU**

Trong thời gian thực hiện luận văn, tác giả đã nhận được sự quan tâm rất lớn của nhà trường, các khoa, phòng ban chức năng, các thầy cô giáo và đồng nghiệp.

Tác giả xin chân thành cảm ơn Ban giám hiệu, phòng sau đại học, các giảng viên đã tạo điều kiện cho tôi hoàn thành luận văn này.

Tác giả xin bày tỏ lời cảm ơn chân thành nhất đến thầy hướng dẫn khoa học PGS.TS Lại Khắc Lãi về những chỉ dẫn khoa học, định hướng nghiên cứu và tận tình hướng dẫn tôi trong suốt quá trình làm luận văn.

Mặc dù đã rất cố gắng, song do trình độ và kinh nghiệm còn hạn chế nên có thể luận văn còn những thiếu sót. Tác giả rất mong nhận được những ý kiến đóng góp từ các thầy cô giáo và các bạn đồng nghiệp để luận văn được hoàn thiện và có ý nghĩa ứng dụng trong thực tế.

Cuối cùng, tôi xin cảm ơn gia đình và bạn bè, những người đã luôn ủng hộ và động viên tôi để tôi yên tâm nghiên cứu luận văn này.

*Thái nguyên, tháng 11 năm 2017*

**Tác giả**

**Ngô Thanh Hải**

**MỤC LỤC**

[ĐẶT VẤN ĐỀ 1](#_Toc498784767)

[CHƯƠNG 1. TỔNG QUAN VỀ ROBOT 2](#_Toc498784768)

[1.1. KHÁI QUÁT CHUNG VỀ ROBOT CÔNG NGHIỆP (RBCN) 2](#_Toc498784769)

[1.2. TỰ ĐỘNG HÓA VÀ ROBOT CÔNG NGHIỆP 3](#_Toc498784770)

[1.3. SƠ LƯỢC VỀ LỊCH SỬ PHÁT TRIỂN CỦA ROBOT CÔNG NGHIỆP 4](#_Toc498784771)

[1.4. CÁC ĐẶC TÍNH CỦA ROBOT CÔNG NGHIỆP 6](#_Toc498784772)

[1.4.1. Tải trọng 6](#_Toc498784773)

[1.4.2. Tầm với 7](#_Toc498784774)

[1.4.3. Độ phân giải không gian 7](#_Toc498784775)

[1.4.4. Độ chính xác 7](#_Toc498784776)

[1.4.5. Độ lặp lại 7](#_Toc498784777)

[1.4.6. Độ nhún 8](#_Toc498784778)

[1.5. HỆ THỐNG ROBOT TRONG CÔNG NGHIỆP 8](#_Toc498784779)

[1.5.1. Hệ thống chuyển động robot 8](#_Toc498784780)

[1.5.2. Hệ thống truyền động robot 12](#_Toc498784781)

[1.5.3. Hệ thống điều khiển robot 14](#_Toc498784782)

[1.5.4. Hệ thống cảm biến 15](#_Toc498784783)

[1.6. ỨNG DỤNG CỦA ROBOT CÔNG NGHIỆP 16](#_Toc498784784)

[CHƯƠNG 2. 18](#_Toc498784785)

[TỔNG HỢP HỆ THỐNG ĐIỀU KHIỂN CHO ROBOT 5 BẬC TỰ DO 18](#_Toc498784786)

[2.1. MÔ TẢ TOÁN HỌC ROBOT 5 BẬC TỰ DO 18](#_Toc498784787)

[2.1.1. Động học thuận robot 18](#_Toc498784788)

[2.1.2. Động học ngược robot 22](#_Toc498784789)

[2.2. ĐỘNG LỰC HỌC ROBOT 23](#_Toc498784790)

[2.2.1. Hàm Lagrange 23](#_Toc498784791)

[2.2.2 Phương trình động lực học robot 25](#_Toc498784792)

[2.2.3 Phương trình động lực học robot 5 bậc tự do 28](#_Toc498784793)

[2.3. THIẾT KẾ BỘ ĐIỀU KHIỂN CÁC KHỚP 36](#_Toc498784794)

[2.3.1. Cấu trúc hệ thống điều khiển 36](#_Toc498784795)

[2.3.2 Thiết kế bộ điều khiển pid điều khiển đối tượng 37](#_Toc498784796)

[2.3.3. Kết quả mô phỏng dùng PID 45](#_Toc498784797)

[CHƯƠNG 3. 47](#_Toc498784798)

[ĐIỀU KHIỂN HOẠT ĐỘNG CỦA ROBOT BẰNG GIỌNG NÓI 47](#_Toc498784799)

[3.1. TỔNG QUAN VỀ NHẬN DẠNG GIỌNG NÓI 47](#_Toc498784800)

[3.2. NHẬN DẠNG GIỌNG NÓI TIẾNG VIỆT 49](#_Toc498784801)

[3.2.1. Tổng quan về tiếng Việt 49](#_Toc498784802)

[3.2.2. Mô hình nhận dạng tiếng Việt 52](#_Toc498784803)

[3.2.3. Hệ thống nhận dạng cơ sở (Baseline) 52](#_Toc498784804)

[3.2.4. Đặc trưng thanh điệu và vấn đề không liên tục của dữ liệu 54](#_Toc498784805)

[3.3. KẾT NỐI MODULE NHẬN DẠNG GIỌNG NÓI VỚI MODULE ĐIỀU KHIỂN ROBOT 56](#_Toc498784806)

[3.3.1. cánh tay robot 5 bậc tự do 56](#_Toc498784807)

[3.3.2. linh kiện và module kèm theo 57](#_Toc498784808)

[KẾT LUẬN VÀ KIẾN NGHỊ 81](#_Toc498784810)

[TÀI LIỆU THAM KHẢO 82](#_Toc498784811)

**DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT**

|  |  |
| --- | --- |
| **Ký hiệu** | **Chú thích** |
| RBCN | Robot công nghiệp |
| DOF | Degree of Freedom |
| CMM | Coordiante Measureent Machine |
| D-H | Denavit-Hartenberg |
| DC | Direct current |
| PID | Proportional Integral Derivative |
| NCC | Normalized Cross Correlation |
| AMDF | Average Magnitude Difference |

**DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ**

|  |  |  |
| --- | --- | --- |
| **Hình** | **Tên hình** | **Trang** |
| Hình 1.1. | Robot công nghiệp IRB – 7600 | 3 |
| Hình 1.2. | Hình dạng cơ khí của 1 RBCN | 9 |
| Hình 1.3. | (1)Bàn tay robot truyền động thủy lực có 4 ngón tay đối xứng, (2)Bàn tay robot có 3 ngón tay không đối xứng. | 11 |
| Hình 1.4. | Sơ đồ khối hệ truyền động thủy lực. | 13 |
| Hình 2.1. | Sơ đồ khối động học | 18 |
| Hình 2.2. | Minh họa phương pháp DH | 19 |
| Hình 2.3. | Khảo sát tốc độ của vi khối lượng dm | 24 |
| Hình 2.4. | Hệ tọa độ của robot 5 DOF | 28 |
| Hình 2.5. | Sơ đồ khối hệ thống điều khiển ở không gian khớp | 36 |
| Hình 2.6. | Sơ đồ khối hệ thống điều khiển ở không gian làm việc | 37 |
| Hình 2.7. | Sơ đồ hệ thống điều khiển vòng kín | 38 |
| Hình 2.8. | Sơ đồ hệ chấp hành có điều khiển | 38 |
| Hình 2.9. | Vòng điều chỉnh dòng của động cơ một chiều | 39 |
| Hình 2.10. | Sơ đồ khâu điều chỉnh dòng phần ứng | 39 |
| Hình 2.11. | Cấu trúc hệ thống điều chỉnh tốc độ quay của động cơ một chiều | 40 |
| Hình 2.12. | Sơ đồ điều khiển độc lập một khớp nối | 41 |
| Hình 2.13. | Sơ đồ mô phỏng hệ điều khiển robot 5 DOF bằng PID | 44 |
| Hình 2.14. | Đặc tính quá độ của các khớp khi điều khiển bằng PID | 45 |
| Hình 3.1. | Sơ đồ khối tổng quan của một hệ thống nhận dạng tiếng nói | 48 |
| Hình 3.2. | Các đường đặc tính của 6 thanh điệu tiếng Việt | 51 |
| Hình 3.3. | Đường pitch của câu nói “Nhận dạng tiếng Việt” | 54 |
| Hình 3.4. | Mô Hình cánh tay robot 5 bậc tự do | 56 |
| Hình 3.5. | Tay kẹp robot | 57 |
| Hình 3.6. | Board arduino mega 2560 | 59 |
| Hình 3.7. | Sơ đồ chân Board arduino mega 2560 | 60 |
| Hình 3.8. | Sơ đồ linh kiện Board arduino mega 2560 | 61 |
| Hình 3.9. | Sơ đồ chân linh kiện Board arduino mega 2560 | 62 |
| Hình 3.10. | Một số ứng dụng của arduino mega 2560 | 62 |
| Hình 3.11. | động cơ RC Servo Digital RC FR1501 | 64 |

# **ĐẶT VẤN ĐỀ**

Hiện  nay robot được điều khiển bằng công nghệ nhận dạng giọng nói (*automatic speech recognition)* có một vai trò quan trọng trong nhiều lĩnh vực và được ứng dụng rộng rãi trong sản xuất công nghiệp, y học, dịch vụ và quốc phòng. Thực tế, hầu hết các robot hiện nay hoạt động thông qua các hệ thống điều khiển bằng bàn phím hay các thiết bị xuất nhập khác, nên đòi hỏi những người tương tác với chúng phải có hiểu biết và kiến thức chuyên sâu về robot và lập trình. Nên việc giao tiếp giữa người và robot cần phải đơn giản và trực quan để ai cũng có thể sử dụng. Do đó, giao tiếp giữa người với robot bằng giọng nói sẽ là phương thức giao tiếp hiện đại và có ý nghĩa quan trọng.  Ví dụ trong sản xuất, một người công nhân bình thường có thể điều khiển một hệ thống phức tạp bằng giọng nói, hay con người sẽ không dùng bàn phím hoặc chuột để giao tiếp với máy tính, mà thay vào đó sẽ điều khiển máy tính theo mệnh lệnh bằng ngôn ngữ.

Trên thế giới đã có nhiều hệ thống nhận dạng giọng nói (tiếng Anh) đã và đang được ứng dụng rất hiệu quả như: ViaVoice, Dragon Naturally Speaking, Spoken Toolket nhưng do sự khác biệt về ngôn ngữ nên chúng ta không thể áp dụng chương trình trên để nhận dạng tiếng Việt. Do đó, một hệ thống nhận dạng giọng nói tiếng Việt cần phải được xây dựng. Đề tài này xin được thiết kế module nhận dạng giọng nói tiếng Việt để điều khiển robot 5 bậc tự do. Bên cạnh đó, đề tài cũng tìm hiểu và nghiên cứu về robot 5 bậc tự do ứng dụng trong công nghiệp gắp sản phẩm.

Đề tài “điều khiển robot 5 bậc tự do bằng giọng nói tiếng việt’’ là một đề tài có tính ứng dụng thực tế cao, có thể áp dụng trong sản xuất công nghiệp và nghiên cứu trong các trường đại học.

**CHƯƠNG 1. TỔNG QUAN VỀ ROBOT**

**1.1. KHÁI QUÁT CHUNG VỀ ROBOT CÔNG NGHIỆP (RBCN)**

Robot công nghiệp là thuật ngữ có nhiều quan điểm khác nhau. Có thể định nghĩa là: Robot công nghiệp là một cơ cấu cơ khí có thể lập trình được và có thể thực hiện những công việc có ích một cách tự động không cần sự giúp đỡ trực tiếp của con người. Theo ISO thì “Robot công nghiệp là một tay máy đa mục tiêu, có một số bậc tự do, dễ dàng lập trình, điều khiển tự động, dùng để tháo lắp phôi, dụng cụ và các vật dụng khác”.

Do chương trình thao tác có thể thay đổi, thực hiện nhiều nhiệm vụ đa dạng nên có thể nói robot công nghiệp được hiểu là những thiết bị tự động, linh hoạt, bắt chước được các chức năng lao động của con người. Theo đó, robot công nghiệp cũng là một hệ thống tự động hóa lập trình được, giống như NC, CNC, DNC và AC. Điểm khác biệt giữa robot và NC là NC điều khiển các chuyển động trên bề mặt, theo các trục của hệ tọa độ thì robot điều khiển các chuyển động trong không gian.

Yếu tố đa chức năng nhấn mạnh robot có khả năng thực hiện nhiều chức năng, phụ thuộc vào chương trình và công cụ làm việc. Ví dụ trong dây chuyền sản xuất ô tô, một robot có thể được gắn mỏ hàn để thực hiện công nghệ hàn trong một phân xưởng. Tại phân xưởng khác, robot có cấu hình tương tự với khâu tác động cuối thay thế mỏ hàn bằng các bàn kẹp có thể được điều khiển để vận chuyển các chi tiết và lắp ráp nó vào các vị trí yêu cầu. Ứng với mỗi chức năng khác nhau, chương trình điều khiển của robot sẽ được lập trình lại cho phù hợp. Yếu tố đa chức năng là một trong những điểm chính để phân biệt robot với các máy tự động đang sử dụng trong sản xuất hiện nay.

  
*Hình 1.1. Robot công nghiệp IRB – 7600*

**1.2. TỰ ĐỘNG HÓA VÀ ROBOT CÔNG NGHIỆP**

Hai lĩnh vực tự động hóa (Automation) và kỹ thuật robot (Robotics) có nhiều liên quan mật thiết với nhau. Về phương diện công nghiệp, tự động hóa là một công nghệ liên kết với sử dụng các hệ thống cơ khí, điện tử và hệ thống máy tính trong vận hành và điều khiển sản xuất. Ví dụ, dây chuyền vận chuyển, các máy lắp ráp cơ khí, các hệ thống điều khiển phản hồi, các máy công cụ điều khiển chương trình số và robot. Như vậy, có thể coi robot là một dạng của thiết bị tự động hóa công nghiệp.

* Có ba loại hệ thống tự động hóa công nghiệp: Tự động hóa cố định, tự động hóa lập trình được và tự động hóa linh hoạt.
* Tự động hóa cố định là những hệ thống sản xuất mà trình tự hoạt động là cố định, được xác lập sẵn bởi thiết bị. Mỗi một hoạt động trong quá trình tuần tự thường là rất đơn giản. Các máy móc kết hợp các hoạt động này lại trong một hệ thống phức tạp.
* Tự động hóa lập trình được đặc trưng bởi khả năng thay đổi được trình tự sản xuất theo từng loại sản phẩm. Trình tự sản xuất được điều khiển bởi chương trình.
* Tự động hóa linh hoạt là bước phát triển cao hơn của tự động hóa lập trình được, trong đó hệ thống có thể đáp ứng các yêu cầu thay đổi sản phẩm mà không mất thời gian để thiết lập lại trình tự hoạt động, do đó hệ thống có thể sản xuất ra được các loại sản phẩm khác nhau theo các lịch trình khác nhau.

Robot có liên quan mật thiết với tự động hóa lập trình được. Robot là một máy có khả năng lập trình và có một số đặc tính giống con người. Robot có thể được lập trình để di chuyển cách tay thông qua các trình tự chuyển động có tính chu kỳ để thực hiện nhiệm vụ khác nhau. Ví dụ, các máy bốc dỡ hàng, robot hàn, sơn…Robot cũng được sử dụng rộng rãi trong hệ thống sản xuất linh hoạt hoặc trong hệ thống tự động hóa cố định. Hệ thống này gồm một số máy, hoặc các robot làm việc cùng nhau được điều khiển bằng máy tính hoặc bộ điều khiển lập trình. Ví dụ, dây chuyền hàn vỏ ô tô gồm nhiều cánh tay robot có nhiệm vụ hàn các bộ phận khác nhau. Chương trình lưu trữ trong máy tính được nạp cho từng robot làm việc ở mỗi bộ phận của dây chuyền hàn ô tô. Như vậy đây là một dây chuyền sản xuất linh hoạt với mức độ tự động hóa cao.

## **1.3. SƠ LƯỢC VỀ LỊCH SỬ PHÁT TRIỂN CỦA ROBOT CÔNG NGHIỆP**

Thuật ngữ “Robot” xuất phát từ tiếng Sec (Czech) là “Robota” (có nghĩa là công việc tạp dịch) trong vở kịch Rossum’s Universal Robots của Karel Capek vào năm 1920. Trong vở kịch này, nhân vật Rossum và con trai của ông ta đã chế tạo ra những chiếc máy có thể ứng xử như con người, có khả năng làm việc khỏe gấp đôi con người, nhưng không có cảm tính, cảm giác như con người.

Về mặt kỹ thuật, những robot công nghiệp ngày nay có nguồn gốc từ hai lĩnh vực kỹ thuật ra đời sớm hơn đó là các cơ cấu điều khiển từ xa (Teleoperators) và các máy công cụ điều khiển số (NC-Numerically Controlled machine tool). Năm 1952, mẫu máy điều khiển số đầu tiên được trưng bày ở Viện Công Nghệ Massachuasetts sau một vài năm nghiên cứu chế tạo.

Các cơ cấu điều khiển từ xa (hay các thiết bị kiểu chủ-tớ) đã phát triển mạnh trong Chiến tranh thế giới lần thứ II nhằm nghiên cứu các vật liệu phóng xạ. Người thao tác được tách biệt khỏi khu vực phóng xạ bởi một bức tường có một hoặc vài cửa quan sát để có thể nhìn thấy được công việc bên trong. Các cơ cấu điều khiển từ xa thay thế cho cánh tay của người thao tác; nó gồm có một bộ kẹp ở bên trong (tớ) và hai tay cầm ở bên ngoài (chủ). Cả hai, tay cầm và bộ kẹp, được nối với nhau bằng một cơ cấu sáu bậc tự do để tạo ra các vị trí và hướng tùy ý của tay cầm và bộ kẹp. Cơ cấu dùng để điều khiển bộ kẹp theo chuyển động của tay cầm.

Vào khoảng năm 1949, các máy công cụ điều khiển số ra đời, nhằm đáp ứng yêu cầu gia công các chi tiết trong nghành chế tạo máy bay. Những robot đầu tiên thực chất là sự nối kết giữa các khâu cơ khí của cơ cấu điều khiển từ xa với khả năng lập trình của máy công cụ điều khiển số.

Một trong những robot công nghiệp đầu tiên được chế tạo là robot Versatran của công ty AMF của Mỹ vào năm 1960. Cũng vào khoảng thời gian này ở Mỹ xuất hiện loại robot Unimate được dùng đầu tiên trong kỹ nghệ ô tô.

Tiếp theo Mỹ, các nước khác bắt đầu sản xuất robot công nghiệp: Anh-1967, Thụy Điển và Nhật-1968 theo bản quyền của Mỹ; CHLB Đức-1971; Pháp-1972; Ý-1973…

Tính năng làm việc của robot ngày càng được nâng cao, nhất là khả năng nhận biết và xử lý. Năm 1968, trường đại học tổng hợp Stanford (Mỹ) đã chế tạo ra mẫu robot hoạt động theo mô hình “mắt-tay”, có khả năng nhận biết và định hướng bàn kẹp theo vị trí vật kẹp nhờ các cảm biến.

Năm 1974, Công ty Cincinnati của Mỹ đã đưa ra loại robot được điều khiển bằng máy vi tính, gọi là robot T3 (The Tomorrow Tool: Công cụ của tương lai). Robot này có thể nâng được vật có khối lượng đến 40kg.

Năm 1976, cánh tay robot đầu tiên trong không gian trên tàu thám hiểm Viking của cơ quan không gian Nasa Hoa Kỳ để lấy mẫu đất trên sao Hỏa.

Từ những năm 70, việc nghiên cứu nâng cao tính năng của robot đã chú ý nhiều đến sự lắp đặt thêm các cảm biến để nhận biết môi trường làm việc. Tại trường đại học tổng hợp Stanford người ta đã tạo ra loại robot dùng để lắp ráp tự động và được điều khiển bằng máy vi tính trên cơ sở xử lý thông tin từ các cảm biến lực và thị giác. Cũng vào thời gian này công ty IBM đã chế tạo loại robot có các cảm biến xúc giác và cảm biến lực, điều khiển bằng máy tính để lắp ráp các máy in gồm 20 cụm chi tiết.

Năm 1990, có hơn 40 công ty của Nhật Bản trong đó có những công ty khổng lồ như công ty Hitachi và công ty Mitsubishi đã đưa ra thị trường quốc tế nhiều loại robot nổi tiếng.

Trong những năm sau này, việc nâng cao tính năng hoạt động của robot không ngừng phát triển. Các robot được trang bị thêm các loại cảm biến khác nhau để nhận biết môi trường xung quanh, cùng với những thành tựu to lớn trong lĩnh vực Tin học-Điện tử đã tạo ra các thế hệ robot với nhiều tính năng đặc biệt. Có thể nói robot là sự tổ hợp khả năng hoạt động linh hoạt của các cơ cấu điều khiển từ xa với mức độ “tri thức” ngày càng phong phú của hệ thống điều khiển theo chương trình số cũng như kỹ thuật chế tạo các bộ cảm biến, công nghệ lập trình và các phát triển của trí không nhân tạo, hệ chuyên gia…

Số lượng robot ngày càng gia tăng, giá thành ngày càng giảm. Nhờ vậy, robot công nghiệp đã có vị trí quan trọng trong các dây chuyền sản xuất hiện đại.

**1.4. CÁC ĐẶC TÍNH CỦA ROBOT CÔNG NGHIỆP**

**1.4.1. Tải trọng**

Tải trọng là trọng lượng robot có thể mang và giữ trong khi vẫn đảm bảo một số đặc tính nào đó. Tải trọng lớn nhất lớn hơn tải trọng định mức nhiều nhưng robot không thể mang tải trọng lớn hơn định mức vì khi đó robot không đảm bảo được độ chính xác di chuyển. Tải trọng robot thông thường nhỏ hơn trọng lượng robot.

### **1.4.2. Tầm với**

Tầm với là khoảng cách lớn nhất robot có thể vươn tới trong phạm vi làm việc. Tầm với là một hàm phụ thuộc vào cấu trúc của robot.

**1.4.3. Độ phân giải không gian**

Độ phân giải không gian là lượng gia tăng nhỏ nhất robot có thể thực hiện khi di chuyển trong không gian, phụ thuộc vào độ phân giải điều khiển và sai số cơ khí.

Độ phân giải điều khiển, kí hiệu là CR, xác định bởi độ phân giải hệ thống điều khiển vị trí và hệ thống phản hồi: là tỷ số của phạm vi di chuyển và số bước di chuyển của khớp được địa chỉ hóa trong bộ điều khiển của robot:

CR=(dải chuyển động)/2n

Trong đó n là số bit để biểu diễn một số trong hệ thống điều khiển

Sai số cơ khí phụ thuộc vào khe hở trong hộp truyền, sự rò rỉ của hệ thống thủy lực, tải trọng trên tay robot, tốc độ di chuyển, điều khiện bảo dưỡng robot. Nói chung sai số cơ khí tuân theo phân bố xác suất chuẩn.

Độ phân giải không gian, kí hiệu là SR, được xác định như sau:

SR=CR+6. (độ lệch chuẩn của phân bố sai số cơ khí)

**1.4.4. Độ chính xác**

Độ chính xác đặc trưng cho khả năng của robot điều chỉnh điểm cuối của tay máy đến một điểm bất kỳ trong không gian hoạt động của nó.

Độ chính xác=CR/2+3. (độ lệch chuẩn của phân bố sai số cơ khí)

Độ chính xác=SR/2

**1.4.5. Độ lặp lại**

Độ lặp lại đánh giá độ chính xác khi robot di chuyển để với tới một điểm trong nhiều lần hoạt động. Do sai số cơ khí mà robot không thể với tới cùng một điểm trong nhiều lần hoạt động, mà các điểm với của robot nằm trong một vòng tròn với tâm là điểm đích mong muốn. Bán kính của đường tròn đó là độ lặp lại.

Độ lặp lại = (+/-) 3.(độ lệch chuẩn của phân bố sai số cơ khí)

**1.4.6. Độ nhún**

Độ nhún biểu thị sự dịch chuyển của điểm cuối cổ tay robot đáp ứng lại lực hoặc momen tác dụng. Độ nhún lớn có nghĩa là tay robot dịch chuyển nhiều khi lực tác dụng nhỏ và ngược lại. Độ nhún có ý nghĩa quan trọng vì nó làm giảm độ chính xác dịch chuyển khi robot mang tải trọng. Nếu robot mang tải trọng nặng, trọng lượng tải trọng sẽ làm cho cánh tay robot bị dịch chuyển.

**1.5. HỆ THỐNG ROBOT TRONG CÔNG NGHIỆP**

* Một hệ thống RBCN điển hình gồm các bộ phận sau:
* Hệ thống chuyển động
* Hệ thống truyền động
* Hệ thống điều khiển
* Hệ thống cảm biến

**1.5.1. Hệ thống chuyển động robot**

Các robot công nghiệp ngày nay thường được đặt trên đế và gắn chặt trên sàn. Cơ thể được gắn với đế, tổ hợp cánh tay được gắn với cơ thể. Cuối cánh tay là cổ tay. Hệ thống chuyển động RBCN đảm bảo cho robot có thể thực hiện các nhiệm vụ trong không gian làm việc bao gồm các chuyển động của thân, cánh tay, cổ tay giữa các vị trí hoặc chuyển động theo một quỹ đạo đặt trước.

Bộ phận cơ bản của robot là cánh tay (arm) , cánh tay được cấu thành bởi các thanh nối liên kết với nhau qua các khớp nối mềm (joint), nhờ có khớp nối mà có sự chuyển động tương đối giữa hai thanh nối liền nhau. Cánh tay robot được gắn lên thân (bệ – base), cổ tay (wrist) được gắn ở thanh nối cuối cùng của cánh tay robot, bàn tay (hand – còn được gọi là cơ cấu tác động cuối(end effector)) được gắn lên cổ tay để thực hiện các nhiệm vụ theo yêu cầu công nghệ: cầm nắm hoặc gia công.



*Hình 1.2. Hình dạng cơ khí của 1 RBCN*

***a. Bậc tự do của robot***

Bậc tự do của robot là số tọa độ cần thiết để biểu diễn vị trí và hướng của vật thể ở tay robot trong không gian làm việc. Để biểu diễn hoàn chỉnh một đối tượng trong không gian cần 6 tham số: 3 tọa độ xác định vị trí đối tượng trong không gian và 3 tọa độ biểu diễn hướng của đối tượng. Như vậy một robot công nghiệp điển hình có số bậc tự do là 6. Nếu số bậc tự do nhỏ hơn 6 thì không gian chuyển động của tay robot sẽ bị hạn chế. Với một robot 3 bậc tự do, tay robot chỉ có thể chuyển động dọc theo các trục x,y,z và hướng của tay không xác định.

Số bậc tự do của RBCN sẽ tương ứng với số khớp hoặc số thanh nối của robot. Robot trên hình 1.2 là robot 3 bậc tự do.

***b. Khớp robot***

Khớp là khâu liên kết hai thanh nối có chức năng truyền chuyển động để thực hiện di chyển của robot. Thanh nối gần với robot là thanh nối vào, thanh nối ra sẽ chuyển động tương đối so với thanh nối vào.

* Khớp robot được sử dụng trong thiết kế là khớp tịnh tiến và khớp quay.
* Khớp tịnh tiến thực hiện chuyển động tịnh tiến hoặc trượt thanh nối đầu ra. Các dạng cơ cấu khớp tịnh tiến là cơ cấu xilanh-piston, cơ cấu kính viễn vọng.
* Khớp quay có 3 dạng: R, T, V. Khớp quay dạng R có trục xoay vuông góc với trục hai thanh nối. Khớp quay dạng T có trục quay trùng với hai thanh nối. Khớp quay dạng V có trục quay trùng với trục thanh nối vào và vuông góc với trục thanh nối ra.

***c. Cổ tay robot***

Cổ tay robot có nhiệm vụ định hướng chính xác bàn tay robot (cơ cấu tác động cuối) trong không gian làm việc. Thông thường cơ cấu cổ tay robot có 3 bậc tự do tương ứng với 3 chuyển động: cổ tay xoay xung quanh trục thanh nối cuối cùng (Roll), cổ tay xoay xung quanh trục nằm ngang tạo ra chuyển động lên xuống của bàn tay (Pitch), cổ tay quay xung quanh trục thẳng đứng tạo chuyển động lắc phải, trái của bàn tay (Yaw).

***d. Bàn tay robot (cơ cấu tác động cuối)***

Bàn tay được gắn lên cổ tay robot đảm bảo cho robot thực hiện các nhiệm vụ khác nhau trong không gian làm việc. Cơ cấu bàn tay có hai dạng khác nhau tùy theo chức năng của robot trong dây chuyền sản xuất: cơ cấu bàn kẹp (gripper) và cơ cấu dụng cụ (tool).

– Cơ cấu kẹp

* Cơ cấu kẹp được sử dụng để cầm giữ một vật thể hoặc chi tiết ở các robot làm việc trong dây chuyền lắp ráp khi gắp một chi tiết và lắp ráp một bộ phận của một máy; robot ở dây chuyền đóng gói hoặc ở robot có chức năng vận chuyển như gắp một chi tiết đặt lên một băng tải hoặc vận chuyển một chi tiết từ vị trí này sang vị trí khác. Các chi tiết cũng có các loại và hình dạng khác nhau: chai, hộp, vật liệu thô hoặc một dụng cụ.
* Cơ cấu kẹp thông thường gồm hai hay nhiều ngón tay (finger). Các ngón tay có chức năng biến đổi một dạng năng lượng (điện, cơ khí, khí nén hoặc thủy lực) nhờ một cơ cấu chấp hành thành lực để nắm giữ một vật thể. Cơ cấu có khả năng mở ra và nắm lại các ngón tay và sinh lực đủ lớn để giữ một vật thể trong tay.
* Cơ cấu dụng cụ

Trong nhiều dây chuyền sản xuất, robot thực hiện nhiệm vụ như một dụng cụ để gia công kim loại hoặc một công nghệ đặc biệt như sơn, hàn. Để thực hiện các công nghệ đó, dụng cụ có thể được kẹp trên bàn tay robot (cơ cấu kẹp) hoặc một dụng cụ được gắn cố định trên cổ tay của robot. Các dụng cụ là: mũi khoan, dụng cụ cắt, đá mài, một bình sơn, cơ cấu hàn điểm, hàn hồ quang. Khi bàn tay robot là một dụng cụ, robot cần được điều khiển chuyển động của dụng cụ tương tự như điều khiển cơ cấu bàn tay kiểu kẹp.



*Hình 1.3. Hai ví dụ về bàn tay robot:(1)\_Bàn tay robot truyền động thủy lực có 4 ngón tay đối xứng, (2)\_Bàn tay robot có 3 ngón tay không đối xứng.*

***e. Các dạng cơ cấu hình học và không gian làm việc của RBCN***

Cấu hình robot thông thường được định nghĩa theo các khung tọa độ không gian làm việc của tay robot. Có 5 dạng cơ cấu hình học điển hình: cơ cấu kiểu tọa độ Đề các, tọa độ trụ, tọa độ cầu, SCARA, kiểu tay người:

Cơ cấu kiểu tọa độ Đề các: dùng 3 khớp trượt, cho phép phần công tác thực hiện một cách độc lập các chuyển động thẳng, song song với 3 trục. Không gian làm việc của tay máy có dạng hình hộp chữ nhật. Kết cấu tay máy đơn giản nên có độ cứng vững cao, độ chính xác được đảm bảo đồng đều trong toàn bộ không làm việc, nhưng ít khéo léo. Tay máy kiểu này dùng để vận chuyển và lắp ráp.

Cơ cấu kiểu tọa độ trụ: dùng 1 khớp quay và 2 khớp trượt. Không gian làm việc của nó có dạng hình trụ rỗng. Khớp trượt nằm ngang cho phép tay máy “thò” được vào khoang rỗng nằm ngang. 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.

Cơ cấu kiểu tọa độ cầu: dùng 2 khớp quay và 1 khớp trượt. Không gian làm việc của nó là khối cầu rỗng. Độ chính xác định vị phụ thuộc và tầm với.

Cơ cấu kiểu SCARA: đây là kiểu tay máy có cấu tạo đặc biệt, gồm 2 khớp quay và 1 khớp trượt nhưng cả ba khớp đều có trục song song với nhau. Tên gọi SCARA là viết tắt của “Selective Compliant Articulated Robot Arm”: tay máy mềm dẻo tùy ý. Loại tay máy này thường dùng trong công việc lắp ráp.

Cơ cấu kiểu tay người: cả 3 khớp đều là khớp quay, trong đó trục quay thứ nhất vuông góc với hai trục quay còn lại.

**1.5.2. Hệ thống truyền động robot**

Các khớp có thể thực hiện chuyển động nhờ vào các cơ cấu chấp hành được truyền động bởi các hệ truyền động khác nhau như truyền động điện, thủy lực, khí nén.

***a. Truyền động thủy lực***

Cơ cấu chấp hành thủy lực có hai dạng cơ bản đơn giản nhất: cơ cấu xilanh piston sử dụng cho các khớp tịnh tiến và cơ cấu van quay truyền động cho các khớp quay.



*Hình 1.4. Sơ đồ khối hệ truyền động thủy lực.*

Trên hình 1.4 là sơ đồ khối hệ truyền động thủy lực điển hình. Cơ cấu xilanh hai chiều được cấp dầu và điều khiển bằng một cơ cấu van secvo. Van secvo được truyền động từ một động cơ secvo. Động cơ secvo được điều khiển bởi hệ thống truyền động điều khiển vị trí với tín hiệu phản hồi là vị trí của piston được đo nhờ cảm biến vị trí. Động cơ chỉ cần sinh một lực nhỏ để di chuyển piston của van secvo, từ đó điều chỉnh được lưu lượng và hướng của đường dầu cung cấp cho xilanh và điều khiển được tốc độ và hướng dịch chuyển của xilanh.

Ưu điểm của các chấp hành thủy lực là công suất lớn và cho phép chịu được tải lớn. Tuy nhiên hệ truyền động thủy lực lại có nhiều nhược điểm như: hiện tượng rò rỉ dầu gây ảnh hưởng tới môi trường, có thể gây cháy khi ứng dụng cho hàn hồ quang, cần nhiều cơ cấu phụ trợ, độ ồn lớn, phải kiểm tra chất lượng dầu thường xuyên.

***b. Truyền động khí nén***

Nguyên lý làm việc của cơ cấu khí nén tương tự như cơ cấu thủy lực nhưng dầu áp suất cao được thay bằng khí nén. Cơ cấu khí nén cũng chia làm hai loại tuyến tính và quay.

Ưu điểm của cơ cấu khí nén: nguồn khí nén sẵn có, giá thành cơ cấu khí nén thấp, không làm ảnh hưởng tới môi trường, chuyển động nhanh. Nhược điểm của cơ cấu khí nén là khó áp dụng luật điều khiển phản hồi.

Cơ cấu khí nén chỉ được dùng cho công suất nhỏ và cho các ứng dụng đơn giản như trong các cơ cấu vận chuyển, bàn kẹp.

***c. Truyền động điện***

Hệ thống truyền động điện bao gồm bộ biến đổi, nguồn điện và động cơ điện. Các dạng động cơ điện sử dụng trong hệ thống truyền động robot là: động cơ secvo một chiều, động cơ secvo xoay chiều, động cơ secvo một chiều không chổi than và động cơ bước.

Hệ thống truyền động điện thường được chia làm hai loại: truyền động trực tiếp và gián tiếp qua bộ truyền động cơ khí. Động cơ điện sẽ cung cấp mômen cần thiết để định vị góc quay chính xác cho các khớp trực tiếp hoặc gián tiếp thông qua bộ truyền động cơ khí như các hệ thống puli và đai truyền, các hộp giảm tốc và các hệ thống truyền động điều hòa.

Ưu điểm của động cơ điện là các hệ thống truyền động trực tiếp cho chuyển động nhanh và chính xác, dễ dàng thực hiện luật điều khiển phản hồi, dễ dàng phối hợp với máy tính trong hệ thống điều khiển. Do đó dây là loại cơ cấu chấp hành phổ biến nhất trong các hệ thống robot.

**1.5.3. Hệ thống điều khiển robot**

Bộ điều khiển có thể được thiết kế từ các vi xử lý, các vi điều khiển, bộ điều khiển logic khả trình PLC hoặc máy tính.

* Liên quan đến đặc điểm làm việc của robot có thể chia bài toán điều khiển robot thành hai loại: điều khiển thô và điều khiển tinh.
* Điều khiển thô còn gọi là điều khiển chuyển động hay điều khiển quỹ đạo, được áp dụng cho robot chuyển động tự do trong không gian làm việc của robot nghĩa là không tương tác với môi trường làm việc. Khi đó cần phải xác định luật điều khiển thích hợp để tốc độ, vị trí do đó chuyển động của các khớp bám sát quỹ đạo thiết kế trong thời gian quá trình quá độ nhỏ nhất. Điều khiển chuyển động có thể thực hiện ở hệ tọa độ khớp hay tọa độ Đề các tùy thuộc quỹ đạo thiết kế cho tọa độ khớp hay tọa độ Đề các. Đồ án này sẽ tập trung nghiên cứu các phương pháp và các luật điều khiển chuyển động.
* Điều khiển tinh còn gọi là điều khiển lực, được áp dụng cho robot có tương tác với môi trường làm việc. Khi đó yêu cầu điều khiển cả lực và chuyển động.

Hai phương pháp điều khiển lực là: điều khiển trở kháng (điều khiển độ nhún) và điều khiển hỗn hợp.

**1.5.4. Hệ thống cảm biến**

* Các cảm biến trong robot có thể chia làm hai loại:
* Cảm biến ngoại tuyến tăng khả năng nhận thức cho robot về môi trường xung quanh.
* Cảm biến nội tuyến cung cấp các thông tin về đặc tính của bản thân robot.

***a. Cảm biến nội tuyến***

Gắn trực tiếp trên trục khớp hoặc động cơ, thường là các encodor, chiết áp đo vị trí, các cảm biến lực, thiết bị đo lực.

***b. Cảm biến ngoại tuyến***

Cung cấp các thông tin về đối tượng hoặc môi trường tương tác. Các cảm biến ngoại tuyến có chức năng như các giác quan chính của con người.

+ Cảm biến hình ảnh (Camera)

Camera có cấu tạo bao gồm thấu kính, tế bào quang học, màng chắn. Các tín hiệu về ánh sáng sẽ được chuyển thành tín hiệu điện

+ Cảm biến thính giác (Micro phone)

Chuyển các âm thanh trong không gian thành tín hiệu điện.

Ngoài ra còn có cảm biến về mùi vị, cảm biến nhiệt độ cao bằng tia hồng ngoại, cảm biến khoảng cách bằng phát siêu âm.

**1.6. ỨNG DỤNG CỦA ROBOT CÔNG NGHIỆP**

Robot được sử dụng rộng rãi trong nhiều ngành công nghiệp khác nhau bởi những lợi ích kinh tế mà nó mang lại là rất lớn. Nhìn chung robot có thể nâng các vật nặng, làm việc với các nguyên liệu không an toàn, các hoạt động nguy hiểm hoặc môi trường không thích hợp với con người hoặc những công việc nhàm chán lặp đi lặp lại. Có thể phân loại ứng dụng công nghiệp của robot gồm các lĩnh vực chính: vận chuyển, bốc dỡ vật liệu, gia công, lắp ráp thăm dò và các ứng dụng khác.

***a. Ứng dụng robot công nghiệp trong vận chuyển, bốc dỡ vật liệu***

Ứng dụng vào vận chuyển, robot có nhiệm vụ di chuyển đối tượng từ vị trí này đến vị trí khác. Nhiệm vụ này của robot thực hiện bởi các thao tác nhặt và đặt vật thể. Robot nhặt chi tiết ở một vị trí và chuyển dời đến một vị trí khác. Robot có thể gắp một chi tiết ở một vị trí cố định hoặc trên một băng tải đang chuyển động và đặt trên một băng tải khác đang chuyển động với định hướng chi tiết.

***b. Ứng dụng robot trong lĩnh vực gia công vật liệu***

Trong công nghiệp gia công vật liệu, robot thực hiện nhiệm vụ như một máy gia công. Do đó tay robot sẽ gắn một dụng cụ thay cho một cơ cấu kẹp.

Ứng dụng của robot trong công nghiệp gia công vật liệu bao gồm các công nghệ sau: hàn điểm, hàn hồ quang liên tục, sơn phủ, công nghệ gia công kim loại.

***c. Ứng dụng robot trong lắp ráp và kiểm ta sản phẩm***

* Công nghệ lắp ráp là lắp một chi tiết vào một bộ phận khác. Robot được sử dụng trong dây chuyền lắp ráp thông thường ở bốn dạng sau: lắp chi tiết vào lỗ, lắp lỗ vào chi tiết, lắp chi tiết nhiều chân vào lỗ và lắp ngăn xếp.
* Trong công nghiệp lắp ráp, robot có thể hoạt động riêng lẻ để lắp hoàn thiện một thiết bị hoặc làm việc trong một dây chuyền, trong đó mỗi robot sẽ có nhiệm vụ lắp một chi tiết trong một thiết bị máy.
* Robot cũng được sử dụng trong công đoạn thử nghiệm và kiểm tra. Một trong những ứng dụng của robot trong lĩnh vực đo và kiểm tra sản phẩm là các máy đo tọa độ (Coordiante Measureent Machine - CMM). Máy đo tọa độ được sử dụng rộng rãi để kiểm tra kích thước, vị trí và hình dạng của các chi tiết máy hoặc các bộ phận cơ khí.

**KẾT LUẬN CHƯƠNG 1**

Chương 1: *"tổng quan về robot"* đã giải quyết được một số vấn đề cơ bản sau:

- Khái quát chung về sự phát triển và lịch sử của robot công nghiệp

- Tóm tắt các đặc tính của robot công nghiệp

- Nêu ứng dụng và sơ lược các hệ thống của robot công nghiệp

# **CHƯƠNG 2.**

# **TỔNG HỢP HỆ THỐNG ĐIỀU KHIỂN CHO ROBOT 5 BẬC TỰ DO**

## **2.1. MÔ TẢ TOÁN HỌC ROBOT 5 BẬC TỰ DO**

Trong mô phỏng robot, phân tích hệ thống là cần thiết như là phân tích động học, mục đích là mang lại những hiểu biết của những chuyển động của từng phần cơ khí robot và mối quan hệ giữa chúng. Phân tích động học được chia thành động học thuận và động học ngược. Động học thuận bao gồm tìm ra vị trí cuối trong không gian chuyển động của các khớp là  và động học ngược bao gồm xác định thông số các biến khớp để có được vị trí cuối và hướng mong muốn . Hình 2.1 biểu diễn sơ đồ khối đơn giản của động học robot.

Động học thuận

Thông số hình học

Động học ngược

Chuyển động các khớp

Vị trí và hướng của điểm cuối

*Hình 2.1. Sơ đồ khối động học*

### **2.1.1. Động học thuận robot**

Một phương pháp phổ biến sử dụng thuận tiện cho việc chọn lựa khung tham chiếu trong ứng dụng robot là phương pháp Denavit-Hartenberg hoặc phương pháp D-H biểu diễn như hình 2.2

ai



di   





*Hình 2.2. Minh họa phương pháp DH*

- ai: khoảng cách theo phương xi từ Oi đến giao điểm của các trục xi và zi-1.

- di: khoảng cách theo phương zi-1 từ Oi-1 đến giao điểm của các trục xi và zi-1 ,di thay đổi khi khớp i là khớp trượt.

- : là góc quay quanh trục xi từ zi-1đến zi.

- : là góc quay quanh trụczi-1 từ xi-1 đến xi.

***a. Bảng thông số DH***

**Bước 1:** Chọn hệ tọa độ cơ sở, gắn các hệ tọa độ mở rộng lên các khâu:

- Giả định vị trí ban đầu của robot.

- Xác định các trục khớp và đặt tên tương ứng z … z.

- Xác định hệ tọa độ nền. Đặt gốc của hệ tọa độ này tại bất kỳ điểm nào trên trục . Các trục và được chọn thỏa qui tắc tam diện thuận.

- Chọn gốc tọa độ O là giao điểm của đường vuông góc chung giữa z và z với z. Nếu z giao với z, đặt O tại điểm này. Nếu z song song với z, đặt O tại bất kỳ vị trí nào trên z sao cho thuận tiện.

- Xác định x đi qua Ovà dọc theo đường vuông góc chung giữa z và z. Trong trường hợp các trục khớp cắt nhau thì trục x chọn theo hướng vuông góc với mặt phẳng tạo bởi z và z.

- Xác định y thỏa quy tắc tam diện thuận.

**Bước 2**: Lập bảng thông số Denavit – Hartenberg (D-H) cho các khâu trên robot.

**Bước 3:** Dựa vào bảng thông số D-H xác định các ma trận A bằng cách thay các thông số ở bước 2.

Trường hợp đang xét ở đây là khớp quay thì là biến, còn . Bảng thông số DH như sau:



Bảng 2.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Khâu | Khớp nối |  |  |  |  |
| 1 | 0-1 |  |  |  |  |
| 2 | 1-2 |  |  |  |  |
| ….. | ….. | ….. | ….. | ….. | ….. |
| j | (j-1)-j |  |  |  |  |
| ….. | ….. | ….. | ….. | ….. | ….. |
| n | (n-1)-n |  |  |  |  |

Dựa vào bảng thông số DH, mỗi ma trận chuyển đổi thuần nhất được trình bày như là tích của 4 chuyển đổi cơ bản sử dụng thông số của khâu i và khớp nối (i-1 và i).



(2.1)



Ở đây ký hiệu viết tắt quay quanh trục một góc . là dịch chuyển theo trục một khoảng . Giải thích tương tự ký hiệu và (2.2)



Đối với một khâu đi theo một khớp quay thì là hằng số. Như vậy ma trận của khớp quay là một hàm số của biến khớp



***b. Tính toán ma trận mô tả quan hệ khâu i đối với hệ tọa độ gốc***



Như ta đã trình bày trong mục 1.2.1 ma trận mô tả vi trí và hướng của khâu thứ i so với khâu thứ i-1. Như vậy tích của các ma trận là ma trận mô tả vị trí và hướng của khâu thứ i so với phần đế cố định.



(2.3)



Nếu một robot có n khâu ta có ma trận T của khâu chấp hành cuối:

(2.4)



mô tả mối quan hệ về hướng và vị trí của khâu chấp hành cuối đối với hệ tọa độ gốc, có kích thước 4x4, có thể biểu diễn như sau:



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| = | Ma trận định hướng R | | | Vecto vị trí p |
| 0 | 0 | 0 | 1 |

Ma trận R có kích thước 3x3, là ma trận biểu diễn hướng của khâu chấp hành cuối.

Vecto có kích thước 3x1, biểu diễn mối quan hệ tọa độ vị trí của gốc tọa độ gắn trên khâu chấp hành cuối đối với hệ tọa độ gốc.



### **2.1.2. Động học ngược robot**

Ở bài toán động học thuận, vị trí và hướng của tay được xác định từ các biến khớp (góc quay hoặc độ dịch chuyển ở khớp tịnh tiến) đã biết.

Để điều khiển robot di chuyển theo các vị trí mong muốn của tay trong không gian, cần xác định các giá trị biến khớp tương ứng với vị trí và hướng của tay robot mong muốn. Đây là nội dung của bài toán động học ngược.

Theo động học thuận của robot, xây dựng được phương trình

 (2.5)

Ma trậnđã biết, tức là vị trí và hướng của khung tọa độ tay robot đã biết cần xác định giá trị các biến khớp. Nhân hai vế phương trình (2.5) ma trận nghịch đảo của ma trận: nhận được phương trình sau:

 (2.6)

Thay các ma trậnvào (2.6) ta được:

**** (2.7a)

**** (2.7b)

Trong đó: 





Với x, y, z tương ứng là các thành phần của vectơ v

Ma trận sử dụng các ma trận có dạng:



## **2.2. ĐỘNG LỰC HỌC ROBOT**

Nghiên cứu động lực học robot là công việc cần thiết khi phân tích cũng như tổng hợp quá trình điều khiển chuyển động. Việc nghiên cứu động lực học robot thường giải quyết hai nhiệm vụ sau đây:

-Xác định momen và lực động xuất hiện trong quá trình chuyển động. Khi đó qui luật biến đổi của biến khớp  coi như đã biết. Việc tính toán lực trong cơ cấu tay máy là rất cần thiết để chọn công suất động cơ, kiểm tra độ bền, độ cứng vững, đảm bảo độ tin cậy của robot.

-Xác định các sai số động tức là sai lệch so với qui luật chuyển động theo chương trình. Lúc này cần khảo sát Phương trình chuyển động của robot có tính đến đặc tính động lực của động cơ và các khâu.

Có nhiều phương pháp nghiên cứu động lực học robot, nhưng thường gặp hơn cả là phương pháp cơ học Lagrange, cụ thể là dùng phương trình Lagrange - Euler. Đối với các khâu khớp của robot, với các nguồn động lực và kênh điều khiển riêng biệt, không thể bỏ qua các hiệu ứng trọng trường (gravity effect), quán tính (initial), tương hổ (Coriolis), ly tâm (centripetal)... mà những khía cạnh này chưa được xét đầy đủ trong cơ học cổ điển; Cơ học Lagrange nghiên cứu các vấn đề nêu trên như một hệ thống khép kín nên đây là nguyên lý cơ học thích hợp đối với các bài toán động lực học robot.

Xét khâu thứ i của robot có n khâu

### **2.2.1. Hàm Lagrange**

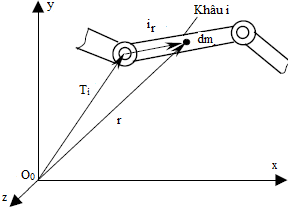
Một điểm trên khâu thứ i được mô tả trong hệ tọa độ cơ bản là:

 ;  (2.8)

Tốc độ của vi khối lượng dm được tính bởi công thức:

 (2.9)  
 Khi tính bình phương của vận tốc này ta có:

 (2.10)



*Hình 2.3. Khảo sát tốc độ của vi khối lượng dm*

Với



Do vậy

 (2.11)

Động năng của vi khối lượng dm đặt tại vị trí trên khâu thứ i

 (2.12)

Động năng của khâu thứ i sẽ là:

 (2.13)

Với  là ma trận quán tính được tính:



Động năng của robot có n khâu được tính:

 (2.14)

Thế năng của khâu i có khối lượng dm, trọng tâm được xác định bởi vecto 

 (2.15)

Trong đó vecto gia tốc trọng trường được biểu diễn dưới dạng:



Thế năng của robot có n khâu được tính:

 (2.16)

Hàm Lagrange của một hệ thống năng lượng:

 (2.17)

Sau khi xác định động năng và thế năng của toàn cơ cấu, ta có hàm Lagrange của robot có 5 bậc tự do:

 (2.18)

### **2.2.2 Phương trình động lực học robot**

Ta có lực tác dụng lên khâu thứ i (lực tổng quát) được xác định bởi phương trình Lagrange:

 (2.19)

Với hàm Lagrange đã xác định ở trên ta tính được:

 (2.20)

Đạo hàm của ma trận  đối với biến khớp  có thể dễ dàng xác định theo công thức sau:

 (2.21)

Trong đó đối với khớp quay:



Trong trường hợp  ta có





Đặt  và đơn giản hóa cách viết như sau:

 (2.22)

Tương tự ta có

 (2.23)

Cuối cùng ta có

 (2.24)

Hay  (2.25)

Hoặc viết dưới dạng ma trận

 (2.26)

Trong đó:

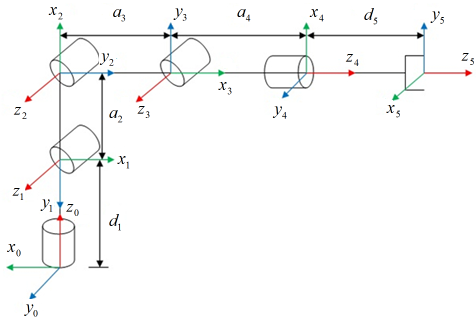
D thể hiện tác dụng quán tính, là một ma trận đối xứng (nxn)

V thể hiện tác dụng của lực ly tâm và Cariolis, là một vecto (nx1)

C thể hiện tác dụng của lực trọng trường, cũng là một vecto (nx1)

### **2.2.3 Phương trình động lực học robot 5 bậc tự do**

Ta xem xét một cơ cấu tay máy 5 DOF như sơ đồ hình 2.4



*Hình 2.4 Hệ tọa độ của robot 5 DOF*

Thông số DH của robot 5 DOF như trong bảng 2.2

Bảng 2.2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Khâu | Khớp nối |  |  |  |  |
| 1 | 0-1 |  |  |  | 0 |
| 2 | 1-2 |  | 0 | 0 |  |
| 3 | 2-3 |  | 0 | 0 |  |
| 4 | 3-4 |  |  | 0 |  |
| 5 | 4-5 |  | 0 |  | 0 |

Ma trận chuyển đổi vị trí giữa hệ tọa độ (i) và (i-1)

; ; 

; 

Ma trận chuyển đổi vị trí giữa hệ tọa độ i và hệ tọa độ gốc



; 







Phương trình động lực học

 ;  (2.27)

 (2.28)

Với robot có 5 bậc tự do thì:

****

****

****

**Ma trận **

 (2.29)



; 

; 

; 

; 



; 

; 

; 



; 

; 



; 



**Vecto **- vecto (5x1) lực ly tâm và Coriolit



****

 ; 

****

Như vậy vec tơ được biểu diễn dưới dạng sau:

 (2.30)

Với 

; 

; 

; 

; 



; 

; 

; 

;

; 

; 

;

; 



Với 

; 

; 

; 

; 

;

; 

; 

; 

;

; 

; 

;

; 



Với 

; 

; 

; 

; 

;

; 

; 

; 

;

; 

; 

;

; 



Với 

;

; 

; 

; 

; 

;

; 

; 

; 

;

; 

; 

;

; 



Với 

;

; 

; 

; 

; 

;

; 

; 

; 

;

; 

; 

;

; 



**Vecto :**

**** (2.31)

****

****

****

****

****

**Tính toán trên máy tính ta thu được**



 (2.32)

## **2.3. THIẾT KẾ BỘ ĐIỀU KHIỂN CÁC KHỚP**

### **2.3.1. Cấu trúc hệ thống điều khiển**

Chức năng của hệ thống điều khiển chuyển động là đảm bảo tay robot chuyển động bám theo quỹ đạo đặt trước trong môi trường làm việc. Chuyển động tay của robot được thực hiện nhờ các hệ thống truyền động khớp robot. Trên cơ sở đó, có hai dạng hệ thống điều khiển chuyển động: hệ thống điều khiển ở không gian khớp và hệ thống điều khiển ở không gian làm việc.

Ở hệ thống điều khiển khớp, đại lượng điều khiển là vị trí của khớp robot: góc quay đối với khớp quay; độ dịch chuyển thẳng đối với khớp tịnh tiến. Bộ điều khiển được thiết kế đảm bảo vị trí khớp luôn bám theo vị trí đặt, tức là sai lệch vị trí khớp hội tụ về không với thời gian nhỏ nhất. Vị trí đặt của khớp được tính toán từ lượng đặt của vị trí tay robot trong không gian làm việc thông qua khâu tính toán động học ngược. Ưu điểm của phương pháp điều khiển ở không gian khớp là bộ điều khiển tác động trực tiếp đến hệ thống truyền động của khớp. Tuy nhiên, hệ thống điều khiển này khó đảm bảo độ chính xác vị trí của tay khi tồn tại các sai lệch trong cơ cấu cơ khí hoặc thiếu thông tin về quan hệ vị trí giữa tay robot và đối tượng

Cảm biến

Động học ngược

Bộđiều khiển

BBĐ

Động cơ

Cơ cấu robot











*Hình 2.5. Sơ đồ khối hệ thống điều khiển ở không gian khớp*

Hệ thống điều khiển không gian làm việc có chức năng duy trì trực tiếp sai lệch vị trí của tay robot trong không gian làm việc bằng không. Lượng đặt của hệ thống điều khiển là vị trí đặt tay trong không gian làm việc và lượng phản hồi là vị trí thực của tay. Khâu tính toán động học ngược sẽ thuộc mạch vòng điều khiển phản hồi. Ưu điểm của phương pháp điều khiển này là tác động trực tiếp các biến không gian làm việc. Nhược điểm là khối lượng tính toán lớn do tồn tại khâu tính toán động học ngược trong mạch vòng điều khiển.

Cảm biến

Bộđiều khiển

BBĐ

Động cơ

Cơ cấu robot









*Hình 2.6. Sơ đồ khối hệ thống điều khiển ở không gian làm việc*

### **2.3.2 Thiết kế bộ điều khiển pid điều khiển đối tượng**

Phương trình Lagrange của robot 5 DOF là phương trình vi phân phi tuyến bậc hai, biến điều khiển đầu vào F lực tác động lên từng khớp của robot là chưa biết, (ta giả thiết rằng cơ cấu chấp hành robot được xem là nguồn tạo ra lực và momen tác động). Bài toán đặt ra là điều khiển lực tác động lên từng khớp ra sao để đạt được vị trí cuối mong muốn. Có nhiều kỹ thuật và phương pháp điều khiển, loại đơn giản nhất ta xem xét ở đây là điều khiển khớp nối độc lập như sơ đồ hình 2.7 Mỗi khớp nối của tay máy robot được xem như là hệ thống một vào một ra.

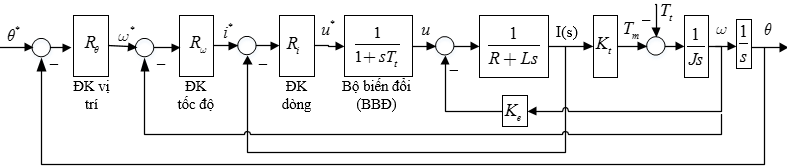


*Hình 2.7. Sơ đồ hệ thống điều khiển vòng kín*

**a. Tổng hợp vòng điều chỉnh động cơ**

Hệ thống điều chỉnh tốc độ quay cho động cơ một chiều có nhiều vòng phân cấp với một khâu điều chỉnh tốc độ quay  ở vòng ngoài và khâu điều chỉnh dòng phần ứng  ở vòng trong. Vòng điều chỉnh  có tác dụng áp nhanh momen quay  (do từ thông là hằng, momen chỉ phụ thuộc , ).

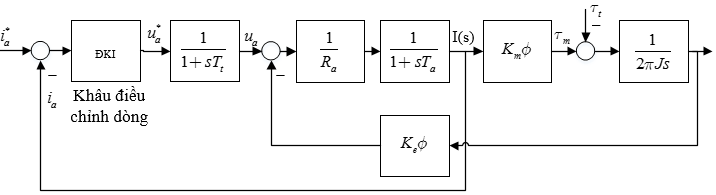
Mô hình cơ cấu chấp hành có 3 vòng điều khiển (dòng điện, tốc độ, vị trí) như Hình 2.8



*Hình 2.8 Sơ đồ hệ chấp hành có điều khiển*

***Điều chỉnh dòng phần ứng***

Để thiết kế khâu điều chỉnh dòng, ta sử dụng hàm truyền đạt của mạch điện phần ứng và bỏ qua sức điện động cảm ứng . Vòng điều chỉnh dòng có cấu trúc như hình 2.9



*Hình 2.9. Vòng điều chỉnh dòng của động cơ một chiều*

Nếu coi gần đúng thiết bị chỉnh lưu có điều khiển là khâu tỷ lệ với quán tính bậc nhất hàm truyền đạt của mạch phần ứng là

 (2.33)

Hằng số thời gian của mạch phần ứng và hệ số khuếch đại ta đã biết

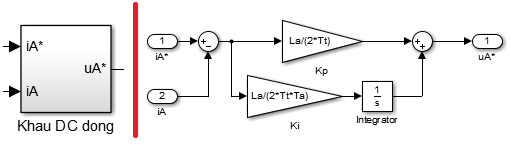


Khâu điều chỉnh dòng chọn là khâu PI, thiết kế theo chuẩn tối ưu module ta có

 (2.34)

Từ đó ta thu được hàm truyền đạt của khâu điều chỉnh dòng kiểu PI như sau

 (2.35)



*Hình 2.10. Sơ đồ khâu điều chỉnh dòng phần ứng*

***Điều chỉnh tốc độ quay***

Đối với vòng điều chỉnh tốc độ quay, vòng điều chỉnh dòng vừa thiết kế là vòng cấp dưới, có thể được gom lại thành một phần của đối tượng điều khiển với hàm truyền đạt như sau:

 (2.36)

Cùng với hàm truyền đạt của phần cơ, đối tượng điều khiển tổng quát lúc này sẽ có mô hình hàm truyền đạt như sau:

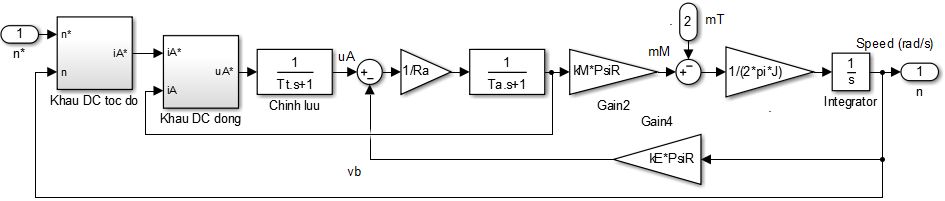
 (2.37)

Với hai tham số của đối tượng là hằng số thời gian cơ  và hằng số thời gian thay thế của vòng trong  ta áp dụng phương pháp thiết kế theo tiêu chuẩn tối ưu đối xứng và thu được các tham số của khâu điều chỉnh

 (2.38)

Với các tham số trên ta thu được hàm truyền đạt tổng quát thay thế của vòng điều chỉnh tốc độ quay thiết kế theo tiêu chuẩn tối ưu đối xứng:

 (2.40)



*Hình 2.11. Cấu trúc hệ thống điều chỉnh tốc độ quay của động cơ một chiều*

**b. Thiết kế bộ điều khiển PID điều khiển đối tượng**

Phương trình Lagrange của robot 5 DOF là phương trình vi phân phi tuyến bậc hai, biến điều khiển đầu vào F lực tác động lên từng khớp của robot là chưa biết, (ta giả thiết rằng cơ cấu chấp hành robot được xem là nguồn tạo ra momen chủ động). Bài toán đặt ra là điều khiển lực tác động lên từng khớp ra sao để đạt được vị trí cuối mong muốn. Có nhiều kỹ thuật và phương pháp điều khiển, loại đơn giản nhất ta xem xét ở đây là điều khiển khớp nối độc lập. Mỗi khớp nối của tay máy robot được xem như là hệ thống một vào một ra, sự ràng buộc giữa các khớp được coi là thành phần nhiễu.

****

*Hình 2.12 Sơ đồ điều khiển độc lập một khớp nối*

Bộ điều khiển PID thử nghiệm biểu diễn như sau:

. Với là vị trí đặt của khớp được tính toán từ lượng đặt vị trí của tay robot trong không gian làm việc (S) thông qua khâu tính toán động học ngược; e là sai số giữa giá trị đặt và giá trị đầu ra; là hệ số tỷ lệ, tích phân, vi phân của bộ điều khiển PID.

Hàm truyền của bộ điều khiển PID viết dưới dạng:



Với  là hằng số thời gian tích phân và vi phân.

Gắn các động cơ vào robot ta xây dựng được sơ đồ mô phỏng hệ thống điều khiển vị trí các khớp của tay máy bằng PID kinh điển. Hàm truyền đạt của cơ cấu chấp hành bao gồm khâu điều chỉnh dòng, khâu điều chỉnh tốc độ quay.

Bảng 2.3 Tham số và giá trị của động cơ DC lựa chọn để mô phỏng

|  |  |
| --- | --- |
| Tham số | Giá trị |
| Điện trở phần ứng |  |
| Điện cảm phần ứng |  |
| Momen quán tính |  |
| Từ thông danh định |  |
| Hằng số động cơ |  |

Bảng 2.4 Bảng thông số DH của robot 5 DOF lựa chọn để mô phỏng

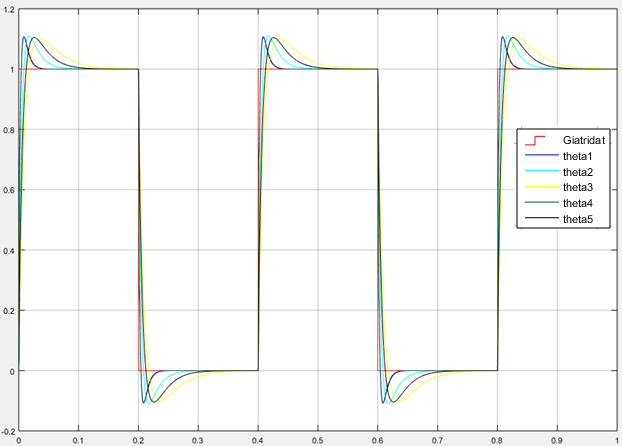
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Khâu | Khớp nối |  |  |  |  |
| 1 | 0-1 |  |  | 1 | 0 |
| 2 | 1-2 |  | 0 | 0 | 1 |
| 3 | 2-3 |  | 0 | 0 | 1 |
| 4 | 3-4 |  |  | 0 | 1 |
| 5 | 4-5 |  | 0 | 1 | 0 |

Với thông số động cơ lựa chọn như bảng 2.3, hằng số thời gian của khâu chỉnh lưu  và thông số robot 5 DOF lựa chọn như bảng 2.4 thì từ phần mềm tối ưu của Matlab ta xác định được các thông số tối ưu của bộ điều khiển PID kinh điển: 



*Hình 2.13. Sơ đồ mô phỏng hệ điều khiển robot 5 DOF bằng PID*

### **2.3.3. Kết quả mô phỏng dùng PID**

****

*Hình 2.14 Đặc tính quá độ của các khớp khi điều khiển bằng PID*

Kết quả mô phỏng cho thấy đáp ứng đầu ra bám giá trị đặt cho trước, chỉ tiêu chất lượng thu được như bảng 2.5. Tuy nhiên kết quả đạt được còn hạn chế về chất lượng, cụ thể là độ quá điều chỉnh lớn (trên dưới 10%). Để khắc phục điều này ta sử dụng các bộ điều khiển thích nghi phi tuyến được xây dựng trên cơ sở lý thuyết điều khiển hiện đại.

Bảng 2.5

|  |  |  |  |
| --- | --- | --- | --- |
| Theta | Độ quá điều chỉnh % | Thời gian xác lập (s) | Sai lệch tĩnh |
| Theta1 | 10.81 | 0.023 | 0 |
| Theta2 | 10.96 | 0.0494 | 0 |
| Theta3 | 12.56 | 0.024 | 0 |
| Theta4 | 10.73 | 0.024 | 0 |
| Theta5 | 10.06 | 0.024 | 0 |

**KẾT LUẬN CHƯƠNG 2**

Chương 2: *"tổng hợp hệ thống điều khiển cho robot 5 bậc tự do"* đã giải quyết được một số vấn đề cơ bản sau:

- Thành lập được phương trình động học cho tay máy 5 bậc tự do

- Thành lập được phương trình động lực học cho tay máy 5 bậc tự do

- Mô tả toán học hệ điều khiển chuyển động bằng phương trình vi phân

- Mô phỏng robot 5 bậc tự do, cơ cấu chấp hành trên Matlab; xây dựng hệ điều khiển các chuyển động của robot bằng các qui luật điều khiển kinh điển.

# **CHƯƠNG 3.**

# **ĐIỀU KHIỂN HOẠT ĐỘNG CỦA ROBOT BẰNG GIỌNG NÓI**

## **3.1. TỔNG QUAN VỀ NHẬN DẠNG GIỌNG NÓI**

***Nhận dạng giọng nói***

Nhận dạng giọng nói là quá trình biến đổi tín hiệu âm thanh thu được của người nói thành một chuỗi các từ có nội dung tưng ứng dưới dạng văn bản. Nếu gọi tín hiệu giọng nói thu được trên miền thời gian là *s(t)* thì *s(t)* đầu tiên sẽ được rời rạc hóa để xử lý và trích chọn ra các thông tin quan trọng. Kết quả thu được là một chuỗi các vector đặc trưng tương ứng. Sau đó nhiệm vụ của hệ thống nhận dạng giọng nói là tìm ra một chuỗi các từ, có nội dung tương ứng với *X* về mặt ngữ nghĩa. Công thứ 3.1 mô tả mô hình toán học của một hệ thống nhận dạng giọng nói theo nguyên lý xác suất của Bayes. Hầu hết các hệ thống nhận dạng giọng nói thống kê ngày nay đều dựa trên mô hình này.

 (3.1)

Trong đó *P(W)* là xác suất của chuỗi *W,* giá trị này có thể được tính toán thông qua một mô hình ngôn ngữ n-gram và nó hoàn toàn độc lập với tín hiệu tiếng nói *X, P(X\W)* là xác suất để *X* là *W* được xác định thông qua mô hình âm học (acoustic model). Giá trị *P(X)* có thể được bỏ qua do giá trị của nó không thay đổi trong một bộ dữ liệu cụ thể với tất cả các chuỗi dự đoán *W.*

Các hệ thống nhận dạng giọng nói hiện nay có thể được phân loại theo các cách như sau:

* Nhận dạng các từ phát âm rời rạc hoặc liên tục.
* Nhận dạng giọng nói phụ thuộc hoặc không phụ thuộc người nói.
* Nhận dạng với hệ thống từ vựng nhỏ (vài trăm từ) hoặc từ vựng lớn (hàng nghìn từ).
* Nhận dạng giọng nói trong môi trường nhiễu cao hoặc thấp

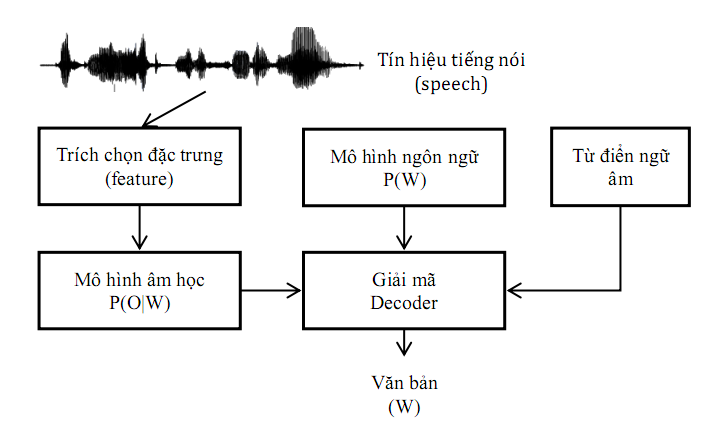
***Ứng dụng***

Cùng với sự phát triển nhanh chóng của các thiết bị tính toán tốc độ cao như máy tính, điện thoại thông minh, vi xử lý – vi điều khiển, ngày nay nhận dạng giọng nói được ứng dụng cho rất nhiều các lĩnh vực trong cuộc sống. Có thể kể đến một số ứng dụng thông dụng như hệ thống dịch tiếng nói tự động, giao tiếp robot, tóm tắt tiếng nói, …

***Các vấn đề trong nhận dạng giọng nói***

Nhận dạng giọng nói là một dạng bài toán trong lĩnh vực nhận dạng mẫu, vì vậy cũng tồn tại những khó khăn tương tự như các bài toán nhận dạng khác. Ngoài ra còn tồn tại một số vấn đề khác do đặc tính biến đổi ngẫu nhiên của tín hiệu tiếng nói. Các vấn đề chính ảnh hưởng đến độ chính xác và hiệu suất làm việc của một hệ thống nhận dạng giọng nói có thể kể đến như nhiễu và sự khác nhau trong tiếng nói: Vấn đề phụ thuộc người nói; Vấn đề về tốc độ phát âm, hiện tượng đồng phát âm; Vấn đề về kích thước của bộ từ vựng (từ điển); Vấn đề nhiễu; Vấn đề về ngôn ngữ;

***Các thành phần chính của một hệ thống nhận dạng tiếng nói***

****

*Hình 3.1. Sơ đồ khối tổng quan của một hệ thống nhận dạng tiếng nói*

## **3.2. NHẬN DẠNG GIỌNG NÓI TIẾNG VIỆT**

Nhìn chung tính đến hiện nay các nghiên cứu về nhận dạng giọng nói tiếng Việt vẫn còn rất hạn chế. Phần lớn các nghiên cứu mới chỉ dừng lại ở nhận dạng số hoặc nhận dạng các từ phát âm rời rạc. Tiếng Việt là một ngôn ngữ có thanh điệu, như vậy một hệ thống nhận dạng đầy đủ sẽ phải bao gồm 2 thành phần là nhận dạng âm vị và nhận dạng thanh điệu. Đã có một só nghiên cứu về nhận dạng thanh điệu cho tiếng Việt, tuy nhiên các nghiên cứu này mới chủ yếu tập trung vào việc phân tích đặc tính và tìm ra mô hình phù hợp trong việc mô hình hóa và nhận dạng thanh điệu đơn lẻ. Hầu hết chưa tích hợp việc nhận dạng thanh điệu với nhận dạng âm vị để tạo thành một hệ thống hoàn chỉnh.

### **3.2.1. Tổng quan về tiếng Việt**

Tiếng việt là một ngôn ngữ đơn âm tiết (Monosyllable), nghĩa là mỗi một âm tiết được thể hiện bởi một từ và cũng là đơn vị cơ bản trong phát âm. Các đặc tính chính của âm tiết tiếng Việt như sau:

1. **Âm tiết tiếng Việt có tính độc lập cao**

Âm tiết là đơn vị cơ bản trong hệ thống các đơn vị ngôn ngữ. Mỗi âm tiết đều có khả năng được thể hiện bởi một từ không biến hình, hay nói cách khác một âm tiết cũng đồng thời là một hình vị. Về mặt ý nghĩa và ngữ pháp trong tiếng Việt được thể hiện chủ yếu bằng trật tự giữa các từ. Như vậy tiếng Việt khác với một số ngôn ngữ khác như tiếng Anh, Pháp,… là luôn có ranh giới rõ ràng giữa hai âm tiết.

1. **Âm tiết tiếng Việt có khả năng biểu hiện ý nghĩa**

Hầu hết các âm tiết tiếng Việt khi đứng một mình đều có khả năng biểu hiện một ý nghĩa xác định. Như vậy âm tiết tiếng Việt ngoài vai trò là một đơn vị ngữ âm nó còn có vai trò về từ vựng và ngữ pháp.

1. **Âm tiết tiếng Việt có cấu trúc chặt chẽ**

Tất cả các âm tiết tiếng Việt đều có thể phân tích thành một cấu trúc gồm năm thành phần như bảng 3-1

Bảng 3-1: Cấu trúc âm tiết tiếng Việt

|  |  |  |  |
| --- | --- | --- | --- |
| Thanh điệu | | | |
| Âm đầu | Vần | | |
| Âm đệm | Âm chính | Âm cuối |

Ví dụ cấu trúc của âm tiết từ “chuyển” có thể được phân tích thành 5 thành phần như sau:

Bảng 3-2: Ví dụ cấu trúc ngữ âm của âm tiết “chuyển”

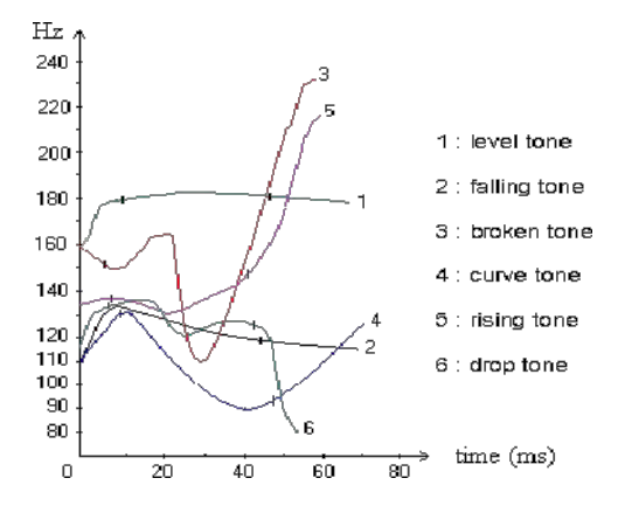
|  |  |  |  |
| --- | --- | --- | --- |
| Thanh điệu (Thanh hỏi) | | | |
| Âm đầu  (Ch) | Vần (uyên) | | |
| Âm đệm  (u) | Âm chính  (yê) | Âm cuối  (n) |

**3.2.1.1. Âm vị tiếng Việt**

Âm vị (phoneme) là đơn vị trừu tượng nhỏ nhất của một ngôn ngữ. Mọi âm tiết trong một ngôn ngữ đều được tạo ra bằng tổ hợp của các âm vị. Trong nhận dạng tiếng nói từ vựng lớn thì âm vị thường được chọn như là đơn vị của hệ thống nhận dạng với mục đích giảm số lượng đơn vị trong hệ thống. Trong khoảng thời gian ngắn từ 10ms đến 40ms có thể coi tín hiệu âm thanh là ổn định và đây cũng là khoảng thời gian phổ biển cho một âm vị. Vì vậy chọn âm vị là đơn vị nhận dạng còn là để giảm ảnh hưởng của sự biến đổi của tín hiệu tiếng nói. Dựa theo cấu trúc của âm tiết tiếng Việt thì hệ thống âm vị của tiếng Việt bao gồm 21 âm đầu, 1 âm đệm, 16 âm chính và 8 âm cuối

**3.2.1.2. Thanh điệu tiếng Việt**

Về mặt hình thức nếu không xét đến sự biến đổi thanh điệu trên các phụ âm dừng ở cuối âm tiết thì tiếng Việt có 6 thanh điệu. Bao gồm thanh huyền, ngã, hỏi, sắc, nặng và thanh bằng (thể hiện trong chữ viết là không dấu).



*Hình 3.2. Các đường đặc tính của 6 thanh điệu tiếng Việt*

* **Thanh bằng:** Là thanh điệu cao, có đường đặc tính bằng phẳng như đường số 1 trong hình 3.2;
* **Thanh huyền:** Là thanh điệu thấp, đường đặc tính có dạng bằng phẳng tương tự thanh bằng nhưng phần cuối có phần đi xuống thấp hơn như thể hiện ở đường số 2 trong hình 3.2;
* **Thanh ngã:** Đường số 3 hình 3.2, đường đặc tính của thanh ngã biến đổi từ ngang, thấp rồi cao;
* **Thanh hỏi:** Đường số 4 hình 3.2, là thanh thấp và có đường đặc tính gãy ở giữa;
* **Thanh sắc:** Đường số 5 hình 3.2, đường đặc tính của thanh sắc có hướng đi lên;
* **Thanh nặng:** Đường số 6 hình 3.2, là thanh thấp có đường đặc tính đi xuống.

Đối với các âm tiết kết thúc bởi các phụ âm đóng “p, k, t” thì các âm tiết này có xu hướng kết thúc nhanh hơn so với các âm tiết khác, chính vì thế hai thanh sắc và thanh nặng (trong tiếng Việt chỉ có hai thanh này tồn tại với các âm tiết kết thúc bằng các phụ âm đóng “p, k, t”) cũng có xu hướng kết thúc nhanh hơn khi cùng với các âm tiết khác. Trong trường hợp này có thể coi tiếng Việt có 8 thanh điệu.

### **3.2.2. Mô hình nhận dạng tiếng Việt**

Xét một hệ thống nhận dạng có bộ từ vựng kích thước *N.* Xét một ngôn ngữ *L* có tập từ vựng là  . Mục tiêu của đề tài là xây dựng mô hình hệ thống có thể nhận dạng từ trong *L* (trong phạm vi đề tài này *L* là tiếng Việt). Khi đó ta sẽ có  . Nếu mô hình hóa mỗi một từ  bởi một mô hình  thì kích thước của hệ thống sẽ là *N.* Trong thực tế thì *N* thường rất lớn, và rất khó để có thể liệt kê hết tất cả các từ của . Nguyên nhân là do hạn chế của người liệt kê, tính chất vùng miền, tính chất thế hệ hoặc theo sự phát triển của văn hóa, công nghệ thì vẫn luôn có những từ mới được bổ sung vào . Như vậy nhược điểm của phương pháp này là kích thước hệ thống vô cùng lớn và hệ thống không có khả năng nhận dạng  nếu nhưng  . Để giải quyết nhược điểm này thì  sẽ được phân tích thành một chuỗi các âm vị với *M* là số âm vị tạo ra *Wi*  trong đó . Trong đó *K* là kích thước tập âm vị  . là bộ âm vị theo cấu trúc ngữ âm của ngôn ngữ L. Cụ thể với tiếng Việt thì  chính là bộ 45 âm vị (*N*=45) . Với cách tiếp cận này thì các âm vị  sẽ được chọn là đơn vị nhận dạng của hệ thống. Như vậy một cách tổng quát thì kích thước của hệ thống nhận dạng luôn là 45 và không phụ thuộc vào kích thước của  . Đồng thời do  được chọn từ  là tập tất cả các âm vị của *L* nên mọi từ có trong *L* đều có thể nhận dạng được bằng cách nhận dạng các âm vị cấu tạo nên nó. Mô hình xác suất để đoán nhận vector đặc trưng đầu vào tại thời điểm *k, xk* (hoặc một chuỗi vector *xk*) là được xác định theo công thức (3.2).

 (3.2)

### **3.2.3. Hệ thống nhận dạng cơ sở (Baseline)**

Các nghiên cứu đã có về nhận dạng tiếng Việt hiện mới chỉ áp dụng mô hình HMM trên các loại đặc trưng phổ biến là MFCC hoặc PLP trên bộ từ vựng kích thước nhỏ cỡ vài trăm từ hoặc trên tiếng nói phát âm rời rạc. Tính đến hiện nay vẫn chưa có một công bố nào về bộ dữ liệu chuẩn cho huấn luyện và đánh giá chất lượng hệ thống chung cho cộng đồng nghiên cứu nhận dạng tiếng Việt. Nghĩa là các thử nghiệm của các nghiên cứu đã khó có thể so sánh với nhau do không cùng các điều kiện tiêu chuẩn như dữ liệu thử nghiệm, đầu ra của hệ thống nhận dạng. Vì thế để có thể so sánh và đánh giá chất lượng của các phương pháp mới trong đề tài thì một hệ thống nhận dạng cơ sở (Baseline) ban đầu cần được xây dựng. Hệ thống cơ sở này được xây dựng dựa trên mô hình không có thanh điệu. Tức là bộ âm vị không có thanh điệu sẽ được sử dụng làm đơn vị nhận dạng và được mô hình hóa bởi mô hình HMM truyền thống với các tham số chính sau:

Đặc trưng đầu vào: MFCC/PLP. Trong đó kích thước của mỗi vector đặc trưng MFCC/PLP là 39 bao gồm 13 thành phần MFCC/PLP, 13 thành phần Delta và 13 thành phần Acceleration của MFCC/PLP.

* Mô hình HMM: Được huấn luyện ở mức triphone với 2179 âm buộc (tied-states).
* Từ điển: Từ điển sử dụng tập âm vị không có thông tin thanh điệu có 45 âm vị.
* Mô hình ngôn ngữ: 2-gram được xây dựng từ dữ liệu phiên âm của VOV.
* Kết quả đánh giá theo tham số độ chính xác theo từ ACC (Word Accuracy) đạt 77.7% với đặc trưng MFCC.

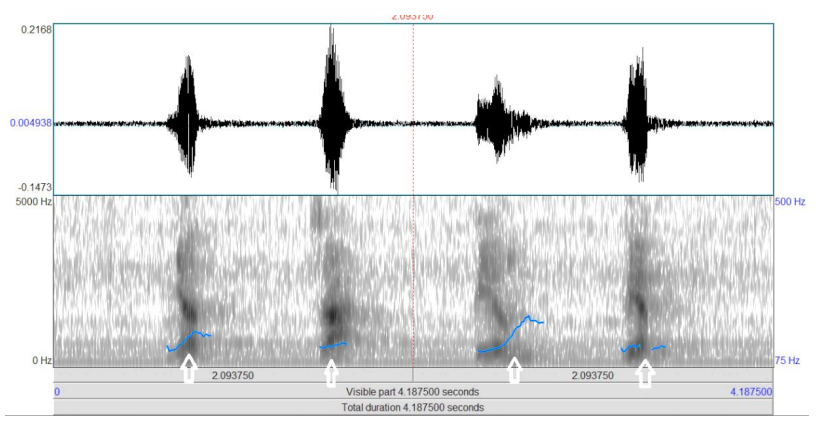
Kết quả nhận dạng trên dữ liệu kiểm thử VOV – test được đánh giá theo tham số độ chính xác theo từ ACC (Word Accuracy) trên hai loại đặc trưng PLP và MFCC được trình bàyởbảng 3-3

Bảng 3-3: Kết quả nhận dạng của hệ thống cơ sở

|  |  |  |
| --- | --- | --- |
| Hệ thống | Đặc trưng | ACC (%) |
| Sys1 (Baseline) | MFCC | 77.70 |
| Sys2 | PLP | 76.77 |

Như vậy đặc trưng MFCC cho chất lượng nhận dạng tốt hơn PLP là 0.93% theo ACC

### **3.2.4. Đặc trưng thanh điệu và vấn đề không liên tục của dữ liệu**



*Hình 3.3. Đường pitch của câu nói “Nhận dạng tiếng Việt”*

Thanh điệu được tạo ra do dao động của dây thanh. Tuy nhiên dây thanh chỉ dao động đối với các âm hữu thanh vì vậy mà trong vùng âm vô thanh không tồn tại thanh điệu. Nếu xét trong cả một câu phát âm thì đường đặc trưng của thanh điệu sẽ bị đứt gãy tại các vùng vô thanh. Hình 3.3 mô tả đường đặt trưng thanh điệu không liên tục của câu nói “nhận dạng tiếng Việt”. Để có thể mô hình hóa đặc trưng thanh điệu sử dụng mô hình HMM hoặc mạng nơron thì đặc trưng này cần phải đượcáp dụng một kỹ thuật tiền xử lý trước để bổ sung các giá trị cho các vùng đứt gãy. Biện pháp đơn giản nhất là thay thế các vùng đứt gãy bằng giá trị 0. Hoặc có thể áp dụng một số kỹ thuật là trơn khác. Tuy nhiên việc áp dụng các kỹ thuật khác để bổ sung giá trị vào vùng mà thanh điệu không tồn tại sẽ làm biến đổi đặc trưng này và đặc trưng mới không còn thể hiện đúng đắn đặc tính đó. Đối với các ngôn ngữ không có thanh điệu như tiếng Anh, Pháp đặc trưng thanh điệu chỉ làm tăng thông tin về ngữ điệu, người nói, giới tính,… do nó không làm thay đổi ngữ nghĩa âm tiết. Vì thế việc thay đổi đặc trưng thanh điệu bằng việc bổ sung các giá trị ”nhân tạo” cũng có thể chấp nhận được hoặc thậm chí có thể bỏ qua đặc trưng này khi xây dựng các hệ thống nhận dạng giọng nói. Đối với tiếng Việt do thanh điệu còn ảnh hưởng trực tiếp đến ngữ nghĩa của từ, vì vậy việc thay đổi nó có thể làm giảm chất lượng nhận dạng. Như vậy cần phải có một phương pháp mô hình hóa sao cho có thể mô hình hóa được đặc tính thanh điệu bị đứt gãy để mô tả đúng nhất đặc tính của nó trong việc góp phần thay đổi ngữ nghĩa trong tiếng Việt.

Tính đến hiện nay có rất nhiều cách nghiên cứu đề xuất các kỹ thuật để trích chọn đặc trưng thanh điệu thông qua việc tính toán tần số cơ bản (F0) từ tín hiệu tiếng nói. Mục tiêu là đề xuất một mô hình có khả năng mô hình hóa loại đặc tính đứt gãy hay nói cách khác là mô hình được loại đặc trưng đầu vào chứa cả giá trị liên tục và giá trị rời rạc. Sử dụng hai phương pháp trích chọn đặc trưng thanh điệu được sử dụng phổ biến là đặc trưng về độ lệch biên độ trung bình (AMDF- Average Magnitude Difference Function) và đặc trưng giá trị tương quan chéo đã chuẩn hóa (NCC – Normalized Cross Correlation). NCC tính toán đặc trưng thanh điệu bằng hương pháp tương quan chéo. Cả NCC và AMDF đều thay thế các giá trị ở vùng vô thanh bằng giá trị 0. Mục đích việc sử dụng 2 loại đặc trưng này là muốn kiểm chứng chất lượng của mô hình và xác định loại đặc trưng nào trong hai phương pháp NCC và AMDF phù hợp với loại mô hình này.

## **3.3. KẾT NỐI MODULE NHẬN DẠNG GIỌNG NÓI VỚI MODULE ĐIỀU KHIỂN ROBOT**

### **3.3.1. cánh tay robot 5 bậc tự do**

**Cánh tay robot** là một sản phẩm công nghệ hiện nay đang sử dụng rộng rãi trong lĩnh vực công nghiệp, bạn có thể nhìn thấy nó trong sản xuất công nghiệp, điều trị y tế : có thể thay thế con người thực hiện những ca mổ khó, nội soi… và thấy nó trong lĩnh vực vui chơi giải trí, các ứng dụng trong giảng dạy, ứng dụng trong quân sự. Chúng ta thường thấy nó trên phương tiện truyền thông.

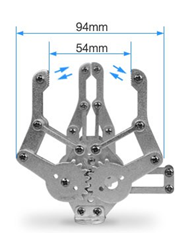
Cánh tay robot 5 bậc được thiết kế bằng kim loại. Có khối lượng nhẹ , nhỏ gọn hoạt động rất linh hoạt đầy đủ chức năng của 1 mô hình cánh tay robot thông thường, là sản phẩm ứng dụng cho lập trình cánh tay robot. Sử dụng 6 động cơ [**RC Servo Digital RC FR1501**](http://mohinhrobot.com/robot/dong-co-servo-mg996r)bánh răng đồng sức nâng 13kg.

Vì nó là một cánh tay cơ khí giống như hành động của cánh tay con người: Khớp vai, khửu tay và khớp cổ tay cộng với nhiều ngón tay sẽ làm cho cánh tay  hoạt động rất linh hoạt. Cánh tay robot 6 bậc với thiết kế thông minh sử dụng 6 động cơ servo kết hợp với hệ thống điều khiển của nó, nó có thể thao tác được 6 mức độ trong không gian.



*Hình 3.4. Mô Hình cánh tay robot 5 bậc tự do*

* Góc quay của servo đáy: 180 độ
* Bán kính của chuyển động gập: 275mm
* Chiều cao tối đa: 420mm
* Độ mở rộng nhất của tay kẹp: 55mm



*Hình 3.5. Tay kẹp robot*

* + 1. **linh kiện và module kèm theo**

**3.3.2.1. Board arduino mega 2560**

Arduino là một board mạch vi xử lý, nhằm xây dựng các ứng dụng tương tác với nhau hoặc với môi trường được thuận lợi hơn. Phần cứng bao gồm một board mạch nguồn mở được thiết kế trên nền tảng vi xử lý AVR Atmel 8bit, hoặc ARM Atmel 32-bit.

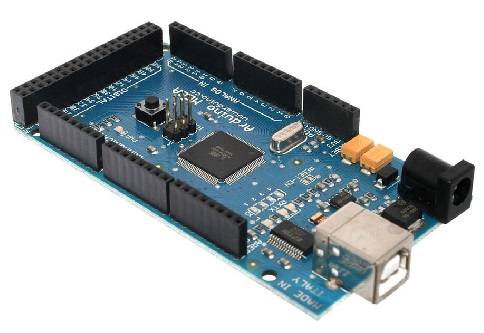
Một mạch Arduino bao gồm một vi điều khiển AVR với nhiều linh kiện bổ sung giúp dễ dàng lập trình và có thể mở rộng với các mạch khác. Một khía cạnh quan trọng của Arduino là các kết nối tiêu chuẩn của nó, cho phép người dùng kết nối với CPU của board với các module thêm vào có thể dễ dàng chuyển đổi, được gọi là shield. Vài shield truyền thông với board Arduino trực tiếp thông qua các chân khách nhau, nhưng nhiều shield được định địa chỉ thông qua serial bus I²C-nhiều shield có thể được xếp chồng và sử dụng dưới dạng song song. Arduino chính thức thường sử dụng các dòng chip megaAVR, đặc biệt là ATmega8, ATmega168, ATmega328, ATmega1280, và ATmega2560. Một vài các bộ vi xử lý khác cũng được sử dụng bởi các mạch Aquino tương thích. Hầu hết các mạch gồm một bộ điều chỉnh tuyến tính 5V và một thạch anh dao động 16 MHz (hoặc bộ cộng hưởng ceramic trong một vài biến thể), mặc dù một vài thiết kế như LilyPad chạy tại 8 MHz và bỏ qua bộ điều chỉnh điện áp onboard do hạn chế về kích cỡ thiết bị. Một vi điều khiển Arduino cũng có thể được lập trình sẵn với một boot loader cho phép đơn giản là upload chương trình vào bộ nhớ flash on-chip, so với các thiết bị khác thường phải cần một bộ nạp bên ngoài. Điều này giúp cho việc sử dụng Arduino được trực tiếp hơn bằng cách cho phép sử dụng 1 máy tính gốc như là một bộ nạp chương trình.

Theo nguyên tắc, khi sử dụng ngăn xếp phần mềm Arduino, tất cả các board được lập trình thông qua một kết nối RS-232, nhưng cách thức thực hiện lại tùy thuộc vào đời phần cứng. Các board Serial Arduino có chứa một mạch chuyển đổi giữa RS232 sang TTL. Các board Arduino hiện tại được lập trình thông qua cổng USB, thực hiện thông qua chip chuyển đổi USB-to-serial như là FTDI FT232. Vài biến thể, như Arduino Mini và Boarduino không chính thức, sử dụng một board adapter hoặc cáp nối USB-to-serial có thể tháo rời được, Bluetooth hoặc các phương thức khác. (Khi sử dụng một công cụ lập trình vi điều khiển truyền thống thay vì ArduinoIDE, công cụ lập trình AVR ISP tiêu chuẩn sẽ được sử dụng.)

Board Arduino sẽ đưa ra hầu hết các chân I/O của vi điều khiển để sử dụng cho những mạch ngoài. Diecimila, Duemilanove, và bây giờ là Uno đưa ra 14 chân I/O kỹ thuật số, 6 trong số đó có thể tạo xung PWM (điều chế độ rộng xung) và 6 chân input analog, có thể được sử dụng như là 6 chân I/O số. Những chân này được thiết kế nằm phía trên mặt board, thông qua các header cái 0.10-inch (2.5 mm). Nhiều shield ứng dụng plug-in cũng được thương mại hóa. Các board Arduino Nano, và Arduino-compatible Bare Bones Board và Boarduino có thể cung cấp các chân header đực ở mặt trên của board dùng để cắm vào các breadboard.

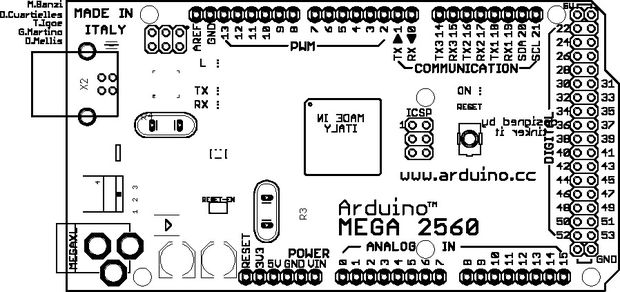
Có nhiều biến thể như Arduino-compatible và Arduino-derived. Một vài trong số đó có chức năng tương đương với Arduino và có thể sử dụng để thay thế qua lại. Nhiều mở rộng cho Arduino được thực thiện bằng cách thêm vào các driver đầu ra, thường sử dụng trong các trường học để đơn giản hóa các cấu trúc của các 'con rệp' và các robot nhỏ. Những board khác thường tương đương về điện nhưng có thay đổi về hình dạng-đôi khi còn duy trì độ tương thích với các shield, đôi khi không. Vài biến thể sử dụng bộ vi xử lý hoàn toàn khác biệt, với các mức độ tương thích khác nhau.

Board arduino mega 2560 sử dụng chip ATmega2560 của [ATmel](http://machtudong.vn/manufacturer-atmel), bộ nhớ chương trình 256KB trong đó có 8KB được sử dụng bởi bootloader. Với bộ nhớ chương trình lớn, có thể viết nhiều chương trình phức tạp, điều khiển được nhiều thiết bị hơn. Dung lượng RAM là 8KB. 4KB EEPROM.



*Hình 3.6. Board arduino mega 2560*

***a. Thành phần Arduino Mega 2560***

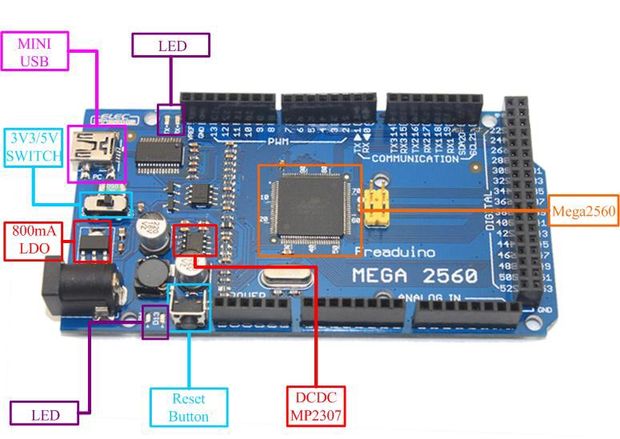


*Hình 3.7. Sơ đồ chân Board arduino mega 2560*

Arduino Mega2560 là một vi điều khiển bằng cách sử dụng ATmega2560.

**Bao gồm:**

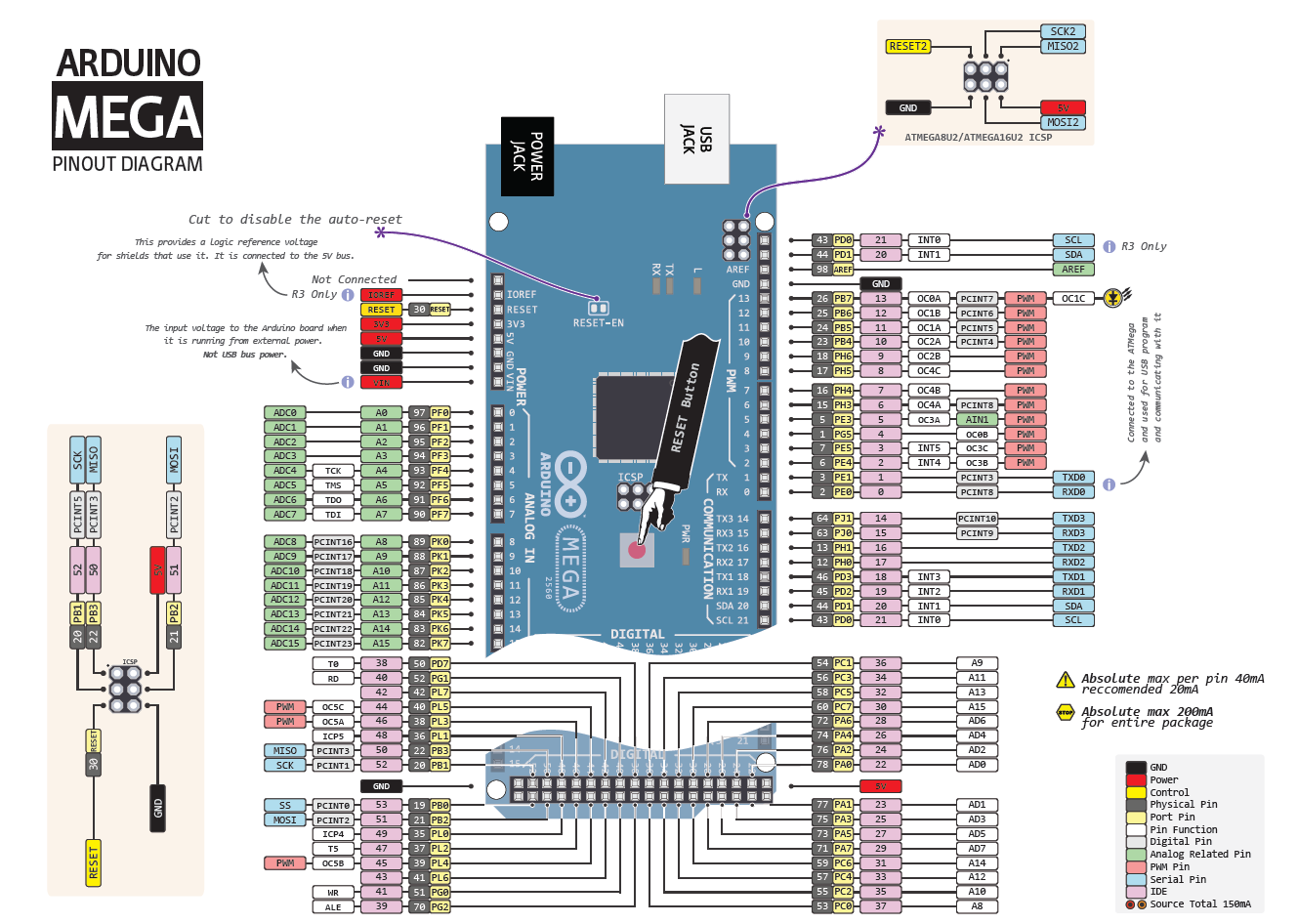
* 54 chân digital (15 có thể được sử dụng như các chân PWM)
* 16 đầu vào analog,
* 4 UARTs (cổng nối tiếp phần cứng),
* 1 thạch anh 16 MHz,
* 1 cổng kết nối USB,
* 1 jack cắm điện,
* 1 đầu ICSP,
* 1 nút reset.
  1. ***Sơ đồ các linh kiện của Arduino Mega 2560***



*Hình 3.8. Sơ đồ linh kiện Board arduino mega 2560*

Arduino sử dụng cáp USB để giao tiếp với máy tính. Thông qua cáp USB chúng ta có thể Upload chương trình cho Arduino hoạt động, ngoài ra USB còn là nguồn cho Arduino.

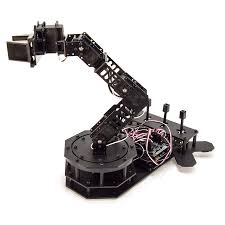
Khi không sử dụng USB làm nguồn thì chúng ta có thể sử dụng nguồn ngoài thông qua jack cắm 2.1mm ( cực dương ở giửa ) hoặc có thể sử dụng 2 chân Vin và GND để cấp nguồn cho Arduino.



*Hình 3.9. Sơ đồ chân linh kiện Board arduino mega 2560*

***c.Một số ứng dụng của Arduino Mega 2560***

Một số ứng dụng của Arduino Mega như: ma trận LED 8x8x8 dùng để hiển thị hình ảnh, máy gắp thú bông, các robot điều khiển từ xa, máy in 3D, cánh tay robot, ….rất nhiều ứng dụng mới.



*Hình 3.10. Một số ứng dụng của arduino mega 2560*

* + - 1. **Động cơ RC Servo Digital RC FR1501**

***a.Mô tả động cơ RC Servo Digital RC FR1501***

Servo là một dạng động cơ điện đặc biệt. Không giống như động cơ thông thường cứ cắm điện vào là quay liên tục, servo chỉ quay khi được điều khiển (bằng [xung PPM](http://arduino.vn/reference/xung-ppm)) với góc quay nằm trong khoảng bất kì từ 0o -180o. Mỗi loại servo có kích thước, khối lượng và cấu tạo khác nhau. Có loại thì nặng chỉ 9g (chủ yếu dùng trên máy bay mô mình), có loại thì sở hữu một momen lực bá đạo (vài chục Newton/m), hoặc có loại thì khỏe và nhông sắc chắc chắn,...

Động cơ servo được thiết kế những hệ thống hồi tiếp vòng kín. Tín hiệu ra của động cơ được nối với một mạch điều khiển. Khi động cơ quay, vận tốc và vị trí sẽ được hồi tiếp về mạch điều khiển này. Nếu có bầt kỳ lý do nào ngăn cản chuyển động quay của động cơ, cơ cấu hồi tiếp sẽ nhận thấy tín hiệu ra chưa đạt được vị trí mong muốn. Mạch điều khiển tiếp tục chỉnh sai lệch cho động cơ đạt được điểm chính xác. Các động cơ servo điều khiển bằng liên lạc vô tuyến được gọi là động cơ servo RC (radio-controlled). Trong thực tế, bản thân động cơ servo không phải được điều khiển bằng vô tuyến, nó chỉ nối với máy thu vô tuyến trên máy bay hay xe hơi. Động cơ servo nhận tín hiệu từ máy thu này.

Động cơ RC Servo Digital là thích hợp nhất cho việc làm các loại robot vì nó luôn giữ trạng thái xung gần nhất, tránh trường hợp khung robot bị đổ sập bất ngờ ở trạng thái không điều khiển, động cơ RC Servo Digital FR1501 có cấu tạo một trục xoay giống như Servo truyền thống giúp dễ ứng dụng cho các thiết kế robot, ngoài ra chất lượng của loại động cơ này rất tốt, động cơ có bánh răng kim loại, lực kéo mạnh, xoay êm, không rung, giữ vị trí rất tốt, là một sự lựa chọn tốt cho thiết kế robot.

***b. Thông só kỹ thuật***



*Hình3.11. động cơRC Servo Digital RC FR1501*

* Trọng lượng: 60g
* Kích thước: 40\*20\*40.5mm
* Tốc độ: 0.16sec / tại 4.8 VDC đến 7.2 VDC
* Lực kéo: 17kg / cm tại 4.8 VDC đến 7.2 VDC
* Điện áp làm việc: 4.8 VDC đến 7.2 VDC
* Dòng điện tiêu thụ: >600mA
* Chiều dài cáp: 32cm

**Thứ tự dây:**

* Nâu: GND (0V)
* Đỏ: VCC (5 - 7.2V)
* Cam: Dây cấp xung
  + - 1. **Bluetooth module kết nối Serial (SLAVE) HC-06**

Bluetooth module SLAVE cho phép vi điều khiển kết nối với các thiết bị ngoại vi: martphone, laptop, usb bluetooth... thông qua giao tiếp Serial gửi và nhận tín hiệu 2 chiều.

Module bluetooth được tích hợp trên board cho phép sử dụng nguồn  từ 3.5V đến 6V cung cấp cho board mà không cần lo lắng về chênh lệch điện áp 3V - 5V gây hỏng board.

Bluetooth module gồm 4 chân theo thứ tự: VCC, GND, TX, RX.

Đây là module bluetooth SLAVE nghĩa là không thể chủ động kết nối bằng vi điều khiển, mà cần sử dụng smartphone, laptop, bluetooth usb... để dò tín hiệu và kết nối (pair) từ smartphone, laptop, bluetooth usb... Sau khi pair thành công, có thể gửi và nhận tín hiệu từ vi điều khiển đến các thiết bị này.

Cách nối dây:

|  |  |
| --- | --- |
| **Bluetooth module** | [**Arduino**](http://machtudong.vn/arduino) |
| VCC | 5V |
| GND | GND |
| TX | RX |
| RX | TX |

Tần số làm việc: 2.4GHz

Kích thước module: 44mm x 16mm x 7mm.

**Phụ lục**

#include <stdlib.h>

#include <math.h>

#include <SoftwareSerial.h>

#include <TaskScheduler.h>

#include <Servo.h>

#include <Encoder.h>

SoftwareSerial mySerial(10, 11); // RX, TX

/\*NOTE\*/

/\*servo 5 from 35 - 105, 105 is claimed, 35 is released\*/

/\*-------------DEFINE-------------------------------------------------------------------\*/

#define NUM\_PART 5

#define VELO\_RADIAN (0.5)

#define TIME\_STEPS (0.01f)

#define ROBOT\_MOVE\_STEP 10

#define SERVO\_0 2

#define SERVO\_1 3

#define SERVO\_2 4

#define SERVO\_3 5

#define SERVO\_4 6

#define SERVO\_5 7

struct Point

{

double X;

double Y;

};

struct ControlServo{

double Ti;

double T1i;

double T2i;

double theta1i;

double theta2i;

double ai;

double t;

double qi;

double qdes;

};

/\*-------------DRIVER PARAMETER-----------------------------------------------------------\*/

Servo qservo[NUM\_PART+1];

/\*-------------ROBOT PARAMETER------------------------------------------------------------\*/

static double q[NUM\_PART];

static bool moveEnable[NUM\_PART];

static double q\_offset[NUM\_PART];

static double lrParam = 90;

static double d0 = 10; // offset from amid to origin

static double l1 = 105;

static double l2 = 105.45;

static double l3 = 60;

static double l4 = 50;

static double d3 = 25;

static double EOF\_X = 150;

static double EOF\_Y = 0;

static double EOF\_Z = 5;

static double Ts = 2.5f; // Time to roll 1 radian angle is 2.5 second

static double Scale = 0.15f; // 15 %

//static double TimeStep = 0.01f; // 10ms

static struct ControlServo ctrlServo[NUM\_PART];

void errorReport(){

Serial.println("check!");

}

/\*-------------COMMON FUNCTION------------------------------------------------------------\*/

// Callback methods prototypes

void t3Callback();

//Control Tasks

Task t3(20 , TASK\_FOREVER, &t3Callback);

//Servo task

Task s0(TIME\_STEPS\*1000 , TASK\_FOREVER, &s0Callback);

Task s1(TIME\_STEPS\*1000 , TASK\_FOREVER, &s1Callback);

Task s2(TIME\_STEPS\*1000 , TASK\_FOREVER, &s2Callback);

Task s3(TIME\_STEPS\*1000 , TASK\_FOREVER, &s3Callback);

Task s4(TIME\_STEPS\*1000 , TASK\_FOREVER, &s4Callback);

Scheduler runner;

void setup() {

Serial.begin(115200);

mySerial.begin(9600);

qservo[0].attach(SERVO\_0);

qservo[1].attach(SERVO\_1);

qservo[2].attach(SERVO\_2);

qservo[3].attach(SERVO\_3);

qservo[4].attach(SERVO\_4);

qservo[5].attach(SERVO\_5);

while (!Serial) {

; // wait for serial port to connect. Needed for native USB port only

}

q\_offset[0] = 0;

q\_offset[1] = PI / 2;

q\_offset[2] = PI / 2;

q\_offset[3] = 0;

q\_offset[4] = 0;

for (int i = 2; i< NUM\_PART; i++){

q[i] = 0;

moveEnable[i] = false;

qservo[i].write(getPWMValue(captureQiValue(i), i));

}

/\*Init task\*/

runner.init();

Serial.println("Initialized scheduler");

runner.addTask(t3);

Serial.println("added t3");

runner.addTask(s0);

runner.addTask(s1);

runner.addTask(s2);

runner.addTask(s3);

runner.addTask(s4);

delay(200);

t3.enable();

Serial.println("Enabled t3");

s0.enable();

s1.enable();

s2.enable();

s3.enable();

s4.enable();

Serial.println("OK! Ready!");

}

void loop() {

runner.execute();

}

void t3Callback() {

// Serial.print("t3: ");

// Serial.println(millis());

if (mySerial.available()) {

char c = ((byte)mySerial.read());

Serial.println(c);

if (c == '1') {

robot\_turn\_left();

Serial.println("Led turned off!");

}

else if (c == '2') {

robot\_turn\_right();

Serial.println("Led turned on!");

}

else if (c == '3'){

robot\_go\_up();

}

else if (c == '4'){

robot\_go\_down();

}

else if (c == '7'){

robot\_move\_forward();

}

else if (c == '8'){

robot\_move\_backward();

}

else if (c == '5'){

robot\_claim(1);

}

else if(c == '6')

{

robot\_claim(0);

}

}

}

/\*Servo control task< -parallel move>\*/

void s0Callback(){

if(moveEnable[0]){

qservo[0].write((int)lrParam);

//qservo[0].write(8\*(int)getPWMValue(captureQiValue(0), 0));

}

}

void s1Callback(){

if(moveEnable[1]){

qservo[1].write((int)getPWMValue(captureQiValue(1), 1));

}

}

void s2Callback(){

if(moveEnable[2]){

qservo[2].write((int)getPWMValue(captureQiValue(2), 2));

}

}

void s3Callback(){

if(moveEnable[3]){

qservo[3].write((int)getPWMValue(captureQiValue(3), 3));

}

}

void s4Callback(){

if(moveEnable[4]){

qservo[4].write(90);

}

}

/\*-------------ARM ROBOT FUNCTION---------------------------------------------------------\*/

void robot\_claim(int idx)

{

if(idx == 1)

{

for (int pos = 35; pos <= 135; pos += 1) { // goes from 180 degrees to 0 degrees

qservo[5].write(pos); // tell servo to go to position in variable 'pos'

delay(15); // waits 15ms for the servo to reach the position

}

}

else

{

for (int pos = 135; pos >= 35; pos -= 1) { // goes from 180 degrees to 0 degrees

qservo[5].write(pos); // tell servo to go to position in variable 'pos'

delay(15); // waits 15ms for the servo to reach the position

}

}

}

void robot\_go\_up(){

// for(int i = 0; i< NUM\_PART; i++){

// if(moveEnable[i]){

// return;

// }

// }

EOF\_Z += ROBOT\_MOVE\_STEP;

if( !calculate()){

EOF\_Z -= ROBOT\_MOVE\_STEP;

}

}

void robot\_go\_down(){

// for(int i = 0; i< NUM\_PART; i++){

// if(moveEnable[i]){

// return;

// }

// }

EOF\_Z -= ROBOT\_MOVE\_STEP;

if( !calculate()){

EOF\_Z += ROBOT\_MOVE\_STEP;

}

}

void robot\_turn\_left(){

// for(int i = 0; i< NUM\_PART; i++){

// if(moveEnable[i]){

// return;

// }

// }

// EOF\_Y += ROBOT\_MOVE\_STEP;

// if( !calculate()){

// EOF\_Y -= ROBOT\_MOVE\_STEP;

// }

if(lrParam < 180)

lrParam += 5;

moveEnable[0] = true;

Serial.println(lrParam);

}

void robot\_turn\_right(){

// for(int i = 0; i< NUM\_PART; i++){

// if(moveEnable[i]){

// return;

// }

// }

// EOF\_Y -= ROBOT\_MOVE\_STEP;

// if( !calculate()){

// EOF\_Y += ROBOT\_MOVE\_STEP;

// Serial.println("T\_R fail");

// }

if(lrParam > 5)

lrParam -= 5;

moveEnable[0] = true;

Serial.println(lrParam);

}

void robot\_move\_forward(){

// for(int i = 0; i< NUM\_PART; i++){

// if(moveEnable[i]){

// errorReport();

// return;

// }

// }

EOF\_X += ROBOT\_MOVE\_STEP;

if( !calculate()){

EOF\_X -= ROBOT\_MOVE\_STEP;

}

}

void robot\_move\_backward(){

// for(int i = 0; i< NUM\_PART; i++){

// if(moveEnable[i]){

// return;

// }

// }

EOF\_X -= ROBOT\_MOVE\_STEP;

if( !calculate()){

EOF\_X += ROBOT\_MOVE\_STEP;

}

}

bool calculate(){

double alpha0 = 0;

double alpha1 = 0;

double Temp1,Temp2;

double q\_new[NUM\_PART];

/\*Check valid input\*/

if(EOF\_X == 0){

errorReport();

return false;

}

/\*calculate qi data\*/

q\_new[0] = atan(EOF\_Y / EOF\_X);

alpha0 = atan((EOF\_Z + l3 + l4)\*cos(q\_new[0]) / EOF\_X);

Temp1 = (2 \* l1\*sqrt(EOF\_X\*EOF\_X\*cos(q\_new[0])\*cos(q\_new[0]) + (EOF\_Z + l3 + l4)\*(EOF\_Z + l3 + l4)));

if(Temp1 == 0){

errorReport();

return false;

}

Temp2 = (l1\*l1 + EOF\_X\*EOF\_X\*cos(q\_new[0])\*cos(q\_new[0]) + (EOF\_Z + l3 + l4)\*(EOF\_Z + l3 + l4) - l2\*l2) / Temp1;

if(Temp2 >1 || Temp2 < -1){

errorReport();

return false;

}

alpha1 = acos(Temp2);

q\_new[1] = PI / 2 - alpha0 - alpha1;

Temp1 = (EOF\_Z + l3 + l4 - l1\*cos(q\_new[1]));

if(Temp1 == 0 || cos(q\_new[0]) == 0){

errorReport();

return false;

}

q\_new[2] = atan((EOF\_X / cos(q\_new[0]) + d3 - l1\*sin(q\_new[1])) / Temp1) - q\_new[1];

if (q\_new[2] <-PI/2){

q\_new[2] += PI;

}

else if (q\_new[2] > PI/2){

q\_new[2] -= PI;

}

else{

errorReport();

}

q\_new[3] = PI - q\_new[2] - q\_new[1];

Serial.println(q\_new[0], 4);

Serial.println(q\_new[1], 4);

Serial.println(q\_new[2], 4);

Serial.println(q\_new[3], 4);

for(int i = 0 ; i< NUM\_PART; i++){

startQiLink(i, q\_new[i]);

Serial.println("Start Q : ");

Serial.println(i, 2);

}

return true;

}

void startQiLink(int link, double qi){

if (q[link] == qi){

Serial.println(link);

return;

}

if (qi - q[link] > 0){

ctrlServo[link].qdes = qi;

ctrlServo[link].t = 0;

ctrlServo[link].Ti = (qi - q[link])\*Ts;

ctrlServo[link].T1i = Scale\*ctrlServo[link].Ti;

ctrlServo[link].T2i = ctrlServo[link].Ti - ctrlServo[link].T1i;

ctrlServo[link].ai = (qi - q[link])/(ctrlServo[link].T1i\*ctrlServo[link].T2i);

ctrlServo[link].theta1i = ctrlServo[link].ai\*ctrlServo[link].T1i\*ctrlServo[link].T1i / 2;

ctrlServo[link].theta2i = ctrlServo[link].theta1i + ctrlServo[link].ai\*ctrlServo[link].T1i\*(ctrlServo[link].T2i - ctrlServo[link].T1i);

}

else{

ctrlServo[link].qdes = qi;

ctrlServo[link].t = 0;

ctrlServo[link].Ti = (q[link] - qi)\*Ts;

ctrlServo[link].T1i = Scale\*ctrlServo[link].Ti;

ctrlServo[link].T2i = ctrlServo[link].Ti - ctrlServo[link].T1i;

ctrlServo[link].ai = -(q[link] - qi)/(ctrlServo[link].T1i\*ctrlServo[link].T2i);

ctrlServo[link].theta1i = ctrlServo[link].ai\*ctrlServo[link].T1i\*ctrlServo[link].T1i / 2;

ctrlServo[link].theta2i = ctrlServo[link].theta1i + ctrlServo[link].ai\*ctrlServo[link].T1i\*(ctrlServo[link].T2i - ctrlServo[link].T1i);

}

moveEnable[link] = true;

}

double captureQiValue(int link){

if (!moveEnable[link]){

// if(link == 2)

// Serial.println(ctrlServo[link].qi,4);

return q[link];

}

if (ctrlServo[link].t > ctrlServo[link].Ti){ // done

q[link] = ctrlServo[link].qdes;

moveEnable[link] = false;

// if(link == 2)

// Serial.println(ctrlServo[link].qi,4);

return q[link];

}

ctrlServo[link].t += TIME\_STEPS;

if (ctrlServo[link].t <= ctrlServo[link].T1i){

ctrlServo[link].qi = q[link]/\*q[i]0\*/ + ctrlServo[link].ai \* ctrlServo[link].t \* ctrlServo[link].t / 2;

}

else if (ctrlServo[link].t > ctrlServo[link].T1i && ctrlServo[link].t <= ctrlServo[link].T2i){

ctrlServo[link].qi = q[link]/\*q[i]0\*/ + ctrlServo[link].theta1i + ctrlServo[link].ai\*ctrlServo[link].T1i\*(ctrlServo[link].t - ctrlServo[link].T1i);

}

else if(ctrlServo[link].t <= ctrlServo[link].Ti){

ctrlServo[link].qi = q[link]/\*q[i]0\*/ + ctrlServo[link].theta2i + ctrlServo[link].ai\*ctrlServo[link].T1i\*(ctrlServo[link].t - ctrlServo[link].T2i) - ctrlServo[link].ai\*(ctrlServo[link].t - ctrlServo[link].T2i)\*(ctrlServo[link].t - ctrlServo[link].T2i) / 2;

}

else{

q[link] = ctrlServo[link].qdes;

moveEnable[link] = false;

// if(link == 2)

// Serial.println(ctrlServo[link].qi,4);

return q[link];

}

// if(link == 2)

// Serial.println(ctrlServo[link].qi,4);

return ctrlServo[link].qi;

}

double getPWMValue(double radian, int link){

return(200 \* (radian + q\_offset[link]) / PI);

}

/\*End of file\*/

**KẾT LUẬN CHƯƠNG 3**

Chương 3: *"điều khiển hoạt động của robot bằng giọng nói"* đã giải quyết một số vấn đề cơ bản sau:

- Tổng quan về nhận dạng giọng nói.

- Khái quát chung về nhận dạng giọng nói tiếng Việt về thanh điệu âm vị và hệ thống nhận dạng cơ sở.

- Xây dựng hệ thống điều khiển robot 5 bậc tự do bằng nhận dạng giọng nói thông qua mô hình thật bằng lập trình trên Android.

# **KẾT LUẬN VÀ KIẾN NGHỊ**

Đề tài: *"****điều khiển robot 5 bậc tự do bằng nhận dạng giọng nói****"* đã hoàn thành yêu cầu đặt ra và thu được một số kết quả sau:

- Khái quát chung về lịch sử, sự phát triển, hệ thống và ứng dụng của robot công nghiệp

- Xây dựng mô hình toán học robot 5 bậc tự do toàn khớp quay; mô phỏng robot trên Matlab/Simulink; xây dựng hệ điều khiển các chuyển động của robot bằng các qui luật điều khiển kinh điển, đồng thời tiến hành mô phỏng với các quĩ đạo chuyển động khác nhau để thấy rõ mối quan hệ và tác động ảnh hưởng qua lại giữa các chuyển động đến chất lượng và độ chính xác điều khiển vị trí robot;

- Tìm hiểu được sơ lược về nhận dạng giọng nói tiếng việt;

- Xây dựng được bộ điều khiển robot 5 bậc tự do bằng nhận dạng giọng nói với mô hình thật thông qua lập trình trên Android

Do thời gian và trình độ có hạn nên việc xây dựng bộ điều khiển cho robot còn chưa thật chính xác, chủ yếu mới chỉ lập trình được cho robot 5 bậc tự do nhận dạng giọng nói thông qua các khẩu lệnh đơn âm, vẫn chưa nhận dạng được giọng nói một cách hoàn chỉnh nguyên một câu. Robot hoạt động vẫn chưa thật chính xác và còn xảy ra hiện tượng nhiễu quá điều chỉnh.

* Kết cấu cơ khí còn hạn chế chưa áp dụng được những nghiên cứu ở chương 2 vào robot này nên việc điều khiển còn chưa được chính xác.

Kiến nghị:

- Lập trình chính xác để robot hoạt động ổn định và không bị nhiễu;

- Xây dựng tập cơ sở dữ liệu cho nhận dạng giọng nói phong phú hơn;

- Xây dựng hệ thống điều khiển để nhận dạng giọng nói cả câu không bị giới hạn bởi các âm đơn.

# **TÀI LIỆU THAM KHẢO**

1. Nguyễn Mạnh Tiến, “Điều khiển robot công nghiệp”,Nhà xuất bản Khoa học và Kỹ thuật, Hà Nội.
2. Nguyễn Thiện Phúc, “Robot công nghiệp”,Nhà xuất bản Khoa học và Kỹ thuật, Hà Nội.
3. Phạm Đăng Thức “Robot công nghiệp” Nhà xuất bản Khoa học và Kỹ thuật.
4. Nguyễn Văn Huy, “Nghiên cứu mô hình thanh điệu cho nhận dạng tiếng nói tiếng Việt từ vựng lớn phátâm liên tục”, đề tài khoa học và công nghệ cấpđại học Thái Nguyên.
5. Saeed B. Niku, “Introduction to Robotics: Analysis, Systems, Application”, Prentice Hall, Inc 2000.
6. Tsuneo Yoshikawa, “Foundation of Robotics: Analysis and Control”, MIT 1990.
7. Lorenzo Sciavicco, Bruno Siviliano,” Modeling and control of Robot Manipulators”, McGraw-Hill companies, Inc 1996.
8. K. H. Davis, R. Biddulph, and S. Balashek, “Automatic Recognition of Spoken Digits”, J. Acoust. Soc. Am., Vol 24, No. 6, pp. 627-642, 1952.
9. B.H. Juang, Lawrence R. Rabiner, "Automatic Speech Recognition – A Brief History of the Technology".
10. Website: arduino.vn ; google.com ; vi.Wikipedia.org và một sốwebsite vềđiện tử khác