ỨNG DỤNG SIMCAPE MUTIBODY CỦA SIMULINK CHO MÔ HÌNH HÓA VÀ MÔ PHỎNG CÁNH TAY ROBOT 4 BẬC TỰ DO

**DANH MỤC HÌNH VẼ**

**DANH MỤC BẢNG**

**DANH MỤC CÁC TỪ VIẾT TẮT**

**TÓM TẮT CÔNG TRÌNH NGHIÊN CỨU**

1. **ĐẶT VẤN ĐỀ**
2. **Đối tượng và mục đích nghiên cứu**

Hiện nay, cách mạng công nghiệp 4.0 đang mang đến cho nhân loại cơ hội để thay đổi bộ mặt của các nền kinh tế. Tự động hóa trong công nghiệp bằng Robot chính là chìa khóa chiến lược để nâng cao hơn nữa khả năng cạnh tranh cho tất cả các doanh nghiệp và phát triển kinh tế. Công nghiệp robot phát triển mạnh, mang lại những lợi ích to lớn cho con người và xã hội… Ở nước ta hiện nay cũng đang trong thời kì công nghiệp hóa, hiện đại hóa đất nước, vì vậy việc đẩy mạnh sản xuất, tăng năng suất lao động bằng việc ứng dụng công nghệ tự động hóa và robot. Ngành công nghiệp robot hiện nay đang phát triển rất mạnh mẽ với rất nhiều chủng loại robot làm việc trong nhiều môi trường khác nhau như trong công xưởng, ngoài vũ trụ, dưới biển hay thậm chí là trong các đường ống nước thải. Phổ biến nhất trong số đó phải kể tới robot công nghiệp. Cùng với đó, vấn đề sản xuất robot để phục vụ sản xuất đang là một vấn đề cấp bách mà phải trải qua rất nhiều công đoạn phức tạp khác nhau.

Chính vì vậy, việc nghiên cứu và phát triền robot công nghiệp cũng cần được đẩy mạnh để có thể đáp ứng được nhu cầu của thị trường và những yêu cầu cụ thể trong từng môi trường làm việc khác nhau. Việc liên kết, kết nối từng công đoạn nhỏ như thiết kế, mô phỏng chuyển động, điều khiển,… cũng sẽ làm việc nghiên cứu và phát triển trở nên đơn giản hơn mà vẫn đàm bảo được các vấn đề cơ bản về kĩ thuật. Nhận thức được vấn đề đó, chúng em đã lựa chọn đề tài “Thiết kế, mô phỏng và điều khiển cánh tay robot 4 bậc tự do” với hi vọng có mở ra một hướng đi mới trong việc nghiên cứu và phát triển robot từ những kiến thức đã được học tập tại giảng đường đại học.

1. **Những vẫn đề cần được nghiên cứu**

* Tìm hiểu về cấu trúc, hoạt động cánh tay robot 4 bậc tự do.
* Tìm hiểu phương pháp giải bài toán động học tay máy.
* Thiết kế mô hình robot công nghiệp 4 bậc tự do.
* Xây dựng mô phỏng chuyển động cánh tay robot 4 bậc tự do.
* Xây dựng mô hình cánh tay robot.
* Xây dựng phương pháp, thuật toán điều khiển đồng thời mô hình và mô phỏng.
* Xây dựng giao diện trên máy tính để để điều khiển, hiển thị các thông số quan trọng của cánh tay robot.

1. **GIẢI QUYẾT VẤN ĐỀ**
2. **Cách tiếp cận và phương pháp nghiên cứu**
   1. **Cách tiếp cận**

* Dựa vào các kiến thức đã được học và tích lũy tại trường lớp: Ở môi trường giáo dục của trường Đại học Công nghệ, chúng em đã được học các kiến thức thiết kế, kiến thức về lập trình lập trình C, các lý thuyết chuyên ngành về điện tử, đo lường, điều khiển tự động,… Đó là công cụ cho chúng em tìm tòi và phát triển các hướng giải quyết các vấn đề đã đặt ra.
* Tiếp cận qua Internet, các thông tin từ các nguồn khác nhau: Qua các bài báo khoa học, các thông tin tìm hiểu được qua các trang công nghệ và các kênh thông tin thời sự, chúng em đã tích lũy được một lượng kiến thức nhất định về robot công nghiệp, hướng nghiên cứu và phát triển robot.
  1. **Phương pháp nghiên cứu**
* Tham khảo các tài liệu giới thiệu về robot, các cơ cấu robot và đặc biệt là các mô hình cấu cánh tay robot 4 bậc tự do. Dựa trên đó để thiết kế được hệ thống robot 4 bậc tự do làm cơ sở để mô hình hóa và mô phỏng.
* Nghiên cứu và áp dụng các giải thuật điều khiển đã được học trên giảng đường để điều khiển mô phỏng trên máy tính và mô hình thực tế.
* Nghiên cứu xây dựng giao điện người dùng đồ họa GUI để thiết kế bảng điều khiển, điều khiển song song cả mô hình và mô phỏng, đồng thời hiển thị các thông số quan trọng của robot như tọa độ, vị trí, góc quay của các khớp,…

1. **Quá trình thực hiện**
   1. **Tổng quan về cánh tay robot (Arm Robot)**
      1. **Giới thiệu**

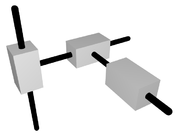
Cánh tay robot là một cỗ máy phục vụ trong quy trình sản xuất công nghiệp được vận hành bởi con người. Ưu điểm nổi bật nhất của nó là thiết kế linh hoạt, thao tác nhanh nhẹn, tỉ mỉ và có khả năng hoàn thiện cả những phần chi tiết sản phẩm nhỏ nhất.

[**Cánh tay robot**](https://carnovn.com/canh-tay-robot/) hoạt động rất linh hoạt với cơ chế khớp ngón hệt như một con người. Ngày nay, nhờ những ưu điểm vượt trội, các cỗ máy cơ khí này đang ngày càng được ứng dụng nhiều vào nhiều lĩnh vực công nghiệp khác nhau, đặc biệt là sản xuất nhựa.

Theo đó, những cỗ máy này sẽ thay thế bàn tay của con người trong các công đoạn lắp ráp, sửa chữa, thay thế linh kiện siêu nhỏ trong nhà máy. Cơ chế hoạt động của chúng dựa trên liên kết giữa bộ điều khiển và các khớp nối chuyển động. Theo đó, bộ phận điều khiển chủ động truyền hiệu lệnh đến các khớp nối cho phép cánh tay chuyển động quay (như trong một robot có khớp nối ) hoặc chuyển dịch (tuyến tính).

Các liên kết được thao túng theo nhu cầu sản xuất để tạo thành một chuỗi động học. Điểm cuối của chuỗi hành động này được gọi là hiệu ứng kết thúc và quy luật hoạt động của nó mô phỏng hoàn hảo chuyển động của bàn tay con người.

* + 1. **Phân loại**
* **Robot trục tọa độ Descartes(còn gọi là robot tuyến tính):** là một [robot công nghiệp](https://vi.wikipedia.org/w/index.php?title=Industrial_robot&action=edit&redlink=1) có ba [trục điều khiển](https://vi.wikipedia.org/wiki/M%C3%B4_men_qu%C3%A1n_t%C3%ADnh) chính là tuyến tính (tức là chúng di chuyển theo đường thẳng chứ không phải xoay) và [vuông góc](https://vi.wikipedia.org/w/index.php?title=Right_angle&action=edit&redlink=1) với nhau. Ba khớp trượt tương ứng với việc di chuyển [cổ tay](https://vi.wikipedia.org/wiki/C%E1%BB%95_tay) lên xuống, vào-ra, tới-lùi. Trong số các ưu điểm khác, cách bố trí cơ khí này đơn giản hóa [giải pháp cánh tay](https://vi.wikipedia.org/w/index.php?title=Arm_solution&action=edit&redlink=1) [điều khiển Robot](https://vi.wikipedia.org/w/index.php?title=Robot_control&action=edit&redlink=1).

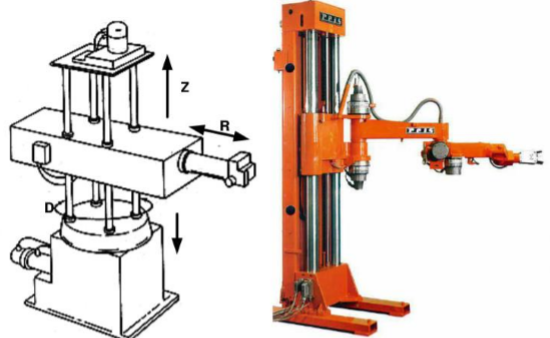


*Hình 1: Sơ đồ động học của robot Descartes*



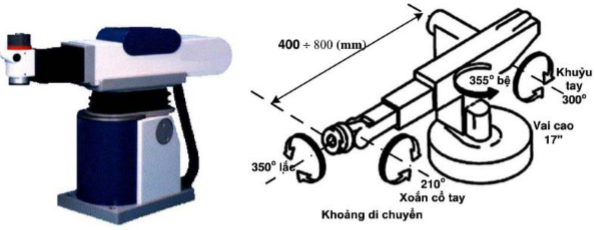
*Hình 2: Robot Descartes*

* **Robot hình trụ** : Được sử dụng cho các hoạt động lắp ráp, xử lý tại các máy công cụ, hàn điểm và xử lý tại các máy diecasting. Đó là một robot có trục tạo thành một hệ tọa độ hình trụ.



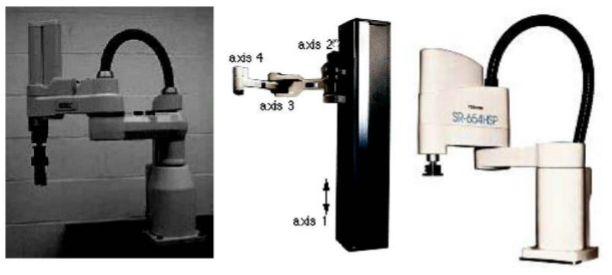
*Hình 3: Robot hệ tọa độ trụ*

* **Robot hình cầu / Robot cực** Được sử dụng để xử lý các công cụ máy móc, hàn điểm, diecasting, máy làm ướt, hàn khí và hàn hồ quang. Đó là một robot có trục tạo thành một hệ tọa độ cực.



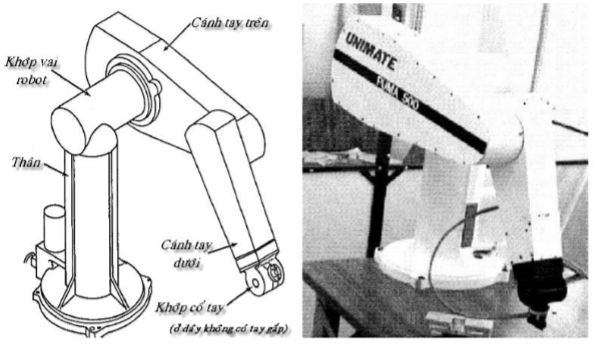
*Hình 4: Robot hệ tọa cầu*

* [**Robot SCARA**](https://translate.googleusercontent.com/translate_c?depth=1&hl=vi&prev=search&rurl=translate.google.com&sl=en&sp=nmt4&u=https://en.m.wikipedia.org/wiki/SCARA_robot&usg=ALkJrhiIQzDSx0ESfhra3zwWvLEipudnDQ): Được sử dụng để chọn và đặt công việc, ứng dụng chất trám, hoạt động lắp ráp và xử lý các công cụ máy móc. Robot này có hai khớp quay song song để cung cấp sự tuân thủ trong một mặt phẳng.



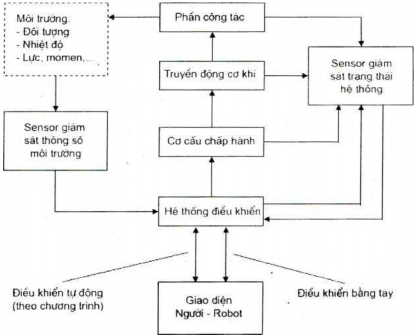
*Hình 5: Robot Scara*

* [**Robot khớp**](https://translate.googleusercontent.com/translate_c?depth=1&hl=vi&prev=search&rurl=translate.google.com&sl=en&sp=nmt4&u=https://en.m.wikipedia.org/wiki/Articulated_robot&usg=ALkJrhhx4fRKaptA3_jcAPz6zNlz-NHO4A) **bản lề** : Được sử dụng cho các hoạt động lắp ráp, diecasting, máy làm ướt, hàn khí, hàn hồ quang và phun sơn. Đó là một robot có cánh tay có ít nhất ba khớp quay.



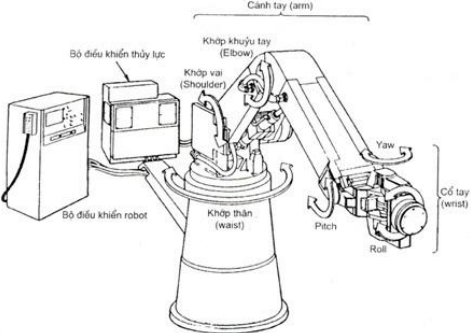
*Hình 2.6: Robot khớp bản lề*

* + 1. **Cấu trúc cơ bản tay máy robot**
  1. **Các thành phần cơ bản**
* Tay máy (Manipulator) là cơ cấu cơ khí gồm các khâu, khớp. Chúng hình thành cánh tay để tạo các chuyển động cơ bản, cổ tay tạo nên sự khéo léo, linh hoạt và bàn tay để trực tiếp hoàn thành các thao tác trên đối tượng.

****

*Hình 2.7: Sơ đồ khối cấu trúc cơ bản của robot công nghiệp*

* Cơ cấu chấp hành tạo chuyển động cho các khâu của tay máy. Nguồn động lực của các cơ cấu chấp hành là động cơ các loại: điện, thuỷ lực, khí nén hoặc kết hợp giữa chúng.
* Hệ thống cảm biến gồm các sensor và thiết bị chuyển đổi tín hiệu cần thiết khác. Các robot cần hệ thống sensor trong để nhận biết trạng thái của bản thân các cơ cấu của robot và các sensor ngoài để nhận biết trạng thái của môi trường.
* Hệ thống điều khiển (Controller) hiện nay thường là máy tính để giám sát và điều khiển hoạt động của robot.



*Hình 2.8: Sơ đồ khối cấu trúc cơ bản của robot công nghiệp*

* + 1. **Ứng dụng**
* Ứng dụng của [Arm robot](https://baoanjsc.com.vn/du-an/ung-dung-4-0-trong--quan-ly-giam-sat-trao-doi-thong-tin-nguoi---may-trong-san-xuat_5_44_21173_vn.aspx) trong công nghiệp rất đa đạng, tùy vào những nghành nghề, công việc khác nhau mà ta có thể áp dụng những robot công nghiệp riêng biệt. Dưới đây là một số nghành trong hệ thống sản xuất mà áp dụng robot công nghiệp.
* Công nghiệp đúc: Arm robot làm nhiệm vụ rót kim loại nóng chảy vào khuôn, cắt mép thừa, làm sạch vật đúc hoặc làm tăng bền vật đúc bằng cách phun cát.
* Ngành gia công áp lực: các quá trình hàn và nhiệt luyện thường bao gồm nhiều công việc độc hại và ở nhiệt độ cao, điều kiện làm việc khá nặng nề, dễ gây mệt mỏi nhất là ở trong các phân xưởng rèn dập.
* Ngành gia công và lắp ráp: [Arm robot](https://baoanjsc.com.vn/du-an/day-chuyen-ma-tu-dong_5_44_21097_vn.aspx) thường được sử dụng vào những việc như tháo lắp phôi và sản phẩm cho các máy ra công bánh răng, máy khoan, máy tiện bán tự động.

****

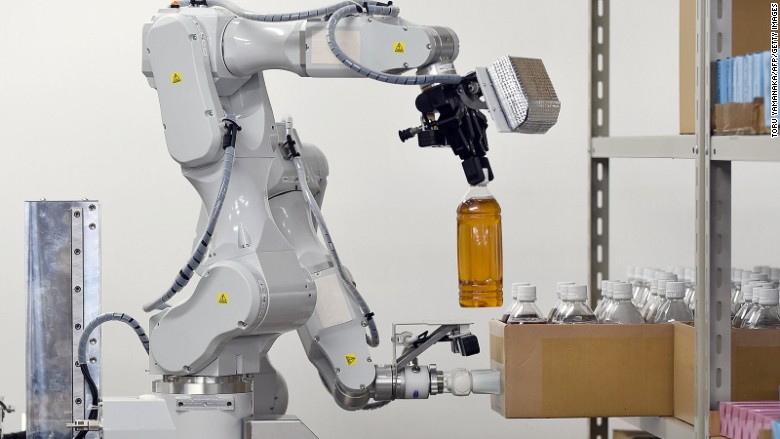
*Hình 2.9: Ứng dụng trong sản xuất cơ khí*

****

*Hình 2.10: Ứng dụng trong sản xuất ô tô*

****

*Hình 2.11: Ứng dụng trong xếp hàng Pallet*

****

*Hình 2.12: Ứng dụng trong công nghiệp thực phẩm*

* 1. **Xây dựng bài toán động học tay máy**
     1. **Các khái niêm ban đầu**

1. **Hệ tọa độ**

* Phương pháp sử dụng: Phương pháp hệ tọa độ tham chiếu
* Hệ tọa độ thuận
* Hệ tọa độ tuyệt đối
* Hệ tọa độ tương đối

1. **Quỹ đạo**

* Tọa độ suy rộng: có thể là chuyển vị góc ở các khớp quay hoặc chuyển vị dài ở các khớp tịnh tiến của các khâu thành viên: q1, q2,… qn

q = q(t)

xM = (q1, q2,… qn) = xM(t) (2 – 1)

* Quỹ đạo điểm M:

1. **Bậc tự do của robot**

* Bậc tự do(DOF – Degree Of Freedom): là thông số độc lập cần thiết để xác định hoàn toàn vị trí của cơ cấu, nó cũng là số khả năng chuyển động tương đối độc lập của cơ cấu đó.
* Công thức tính bậc tự do:

F = (n – 1) – F = (n – j – 1) +

|  |  |
| --- | --- |
| Với: | * ci số chuyển động bị ràng buộc khớp i * fi số chuyển động tương đối cho phép bởi khớp i * j: Tổng số khớp trong cơ cấu * n: Tổng số khâu trong cơ cấu(kể cả khâu cố định) * : Số bậc tự do của không gian mà cơ cấu hoạt động |

* + 1. **Đặt vấn đề**

Cơ cấu chấp hành của robot thường là một cơ cấu hở, gồm một chuỗi các khâu ( links ) nối với nhau bằng các khớp (joint ). Các khớp động này là khớp quay (R) hoặc khớp tịnh tiến (T). Để robot có thể thao tác linh hoạt, cơ cấu chấp hành của nó phải cấu tạo sao cho điểm mút của khâu cuối cùng đảm bảo dễ dàng di chuyển theo một quỹ đạo nào đó, đồng thời khâu này có một định hướng nhất định theo yêu cầu. Khâu cuối thường là bàn kẹp (gripper) hoặc là khâu gắn liền với dụng cụ làm việc (tools). Điểm mút của khâu cuối cùng là điểm đáng quan tâm nhất vì đó là điểm tác động của robot lên đối tác và được gọi là “điểm tác động cuối”(end effector)

Gắn vào “điểm tác động cuối” này một hệ tọa độ thứ n và gắn mỗi khâu động một hệ tọa độ động khác, còn gắn liền giá đỡ với một hệ tọa độ cố định. Đánh số ký 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 và định hướng ” tại “điểm tác động cuối” trong mọi thời điểm.

* + 1. **Xác định trạng thái của robot tại “điểm tác động cuối”**

Như trên đã nhấn mạnh tại các “điểm tác động cuối” của robot không những cần biết vị trí của điểm mút của khâu cuối cùng , tức là điểm tác động của robot lên đối tác, mà còn phải xác định hướng tác động của khâu cuối cùng đó.

Như vậy trạng thái của robot tại “điểm tác động cuối” hoàn toàn xác định bằng sự định vị và định hướng tại điểm tác động cuối đó.

Sự biểu thị định vị và định hướng đó bằng ma trận cuối A. Ma trận này được viết lại như sau:

A=

Trong đó tọa độ px , py,pz là tọa độ của điểm cuối.

a: vector có hướng tiếp cận (approach) với đối tác

s: vector có hướng đường trượt (sliding) đóng mở bàn kẹp

n: vector pháp tuyến (normal)

* + 1. **Mô hình động học**

1. **Ma trận quan hệ**

Chọn hệ tọa độ cố định gắn liền với giá đỡ các hệ tọa độ động gắn với từng khâu động. Ký hiệu các hệ tọa độ này từ 0 đến n, kể từ giá cố định trở đi.

Một điểm bất kỳ nào đó trong không gian được xác định trong hệ tọa độ thứ i bằng bán kính véctơ r­I và trong hệ tọa độ cố định x0, y0, z0 được xác định bằng bán kính véctơ r0:

r0= A1A2…Airi

hoặc r0 = Ti ri

với Ti = A1A2…Ai

i=1,…,n

Trong đó ma trận A1 mô tả vị trí hướng của khâu đầu tiên; ma trận A2 mô tả vị trí và hướng của khâu thứ 2 so vs khâu đầu; ma trận A­I mô tả vị trí và hướng của khâu thứ i so với khâu thứ i-1.

Như vậy, tích của các ma trân Ai là ma trận Ti mô tả vị trí và hướng khâu của khâu thứ i so với giá cố định. Thường ký hiệu ma trận T với 2 chỉ số trên và dưới. Chỉ số dưới để chỉ khâu đang xét còn chỉ số trên để chỉ tọa đọ được dùng để đối chiếu.Ví dụ, biểu thức (4.7) có thể viết lại là:

Ti = 0Ti = AiTi

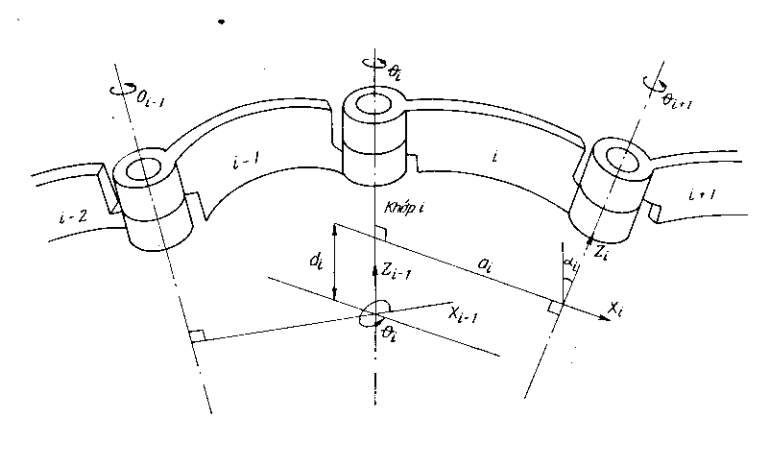
với 1Ti = A1A2A3...Ai

là ma trận mô tả vị trí và hướng của khâu thứ i so với khâu thứ nhất. Trong ký hiệu thường bỏ qua chỉ số trên nếu chỉ số đó bằng 0.

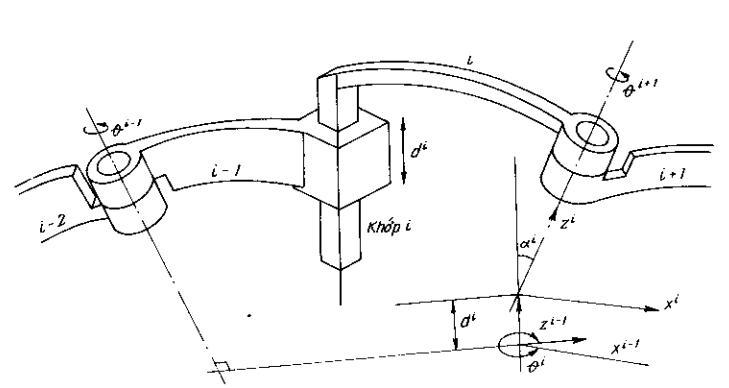
Denavit L.& Hartenbeg R.S. [5] đã đề xuất dùng ma trận thuẩn nhất 4x4 để mô tả quan hệ giữa 2 khâu liên tiếp trong cơ cấu không gian. Pieper D.L. [12] đã đầu tiên áp dụng ma trận thuần nhất 4x4 trong nghiên cứu robot. Litvin F.L.[54] đã dùng ma trận thuần nhất 4x4 trong nghiên cứu lý thuyết ăn khớp với bánh rang.

1. **Bộ thông số DH**

Dưới đây trình bày cách xây dựng các hệ tọa độ đối với 2 khâu động liên tiếp i và i+1. Hình 4.4 là trường hợp 2 khớp động liên tiếp là khớp quay. Hình 4.5 khớp i là khớp tịnh tiến.

****

*Hình 2.13: Các hệ tọa độ đối với 2 khâu động liên tiếp*



*Hình 2.14: Các hệ tọa độ đối với 2 khâu động liên tiếp có khớp tịnh tiến*

Trước hết xác định bộ thông số cơ bản giữa 2 trục quay của khớp động i+1 và i:

* ai  là độ dài đường vuông góc chung giữa 2 trục khớp động i+1 và i.
* αi là góc chéo giữa 2 trục khớp động i+1 và i.
* di  là khoảng cách đo dọc trục khớp động i từ đường vuông góc chung giữa trục khớp động i và trục khớp động i-1.
* θi là góc giữa 2 đường vuông góc chung nói trên.

Bộ thông số này được gọi là bộ thông số Denavit – Hartenberg, hoặc viết tắt là *bộ thông số DH*.

* *Biến khớp* ( joint variable):

+ Nếu khớp động i là khớp quay thì θi là biến khớp.

+ Nếu khớp động i là tịnh tiến thì di là biến khớp.

Để ký hiệu biến khớp dùng thêm dấu \* và trong trường hợp khớp tịnh tiến thì ai­ được xem là bằng 0.

1. **Thiết lập hệ tọa độ**

Gốc của hệ tọa độ gắn liền với khâu thứ I (gọi là hệ tọa độ thứ i) đặt tại giao điểm giữa đường vuông góc chung (ai) và trục khớp động i+1. Trường hợp hai trục giao nhau thì gốc hệ tọa độ lấy trùng với giao điểm đó. Nếu hai trục song song với nhau thì chọn gốc tọa độ là điểm bất kỳ trên trục khớp động i+1.

Trục zi của hệ tọa độ thứ i nằm dọc theo trục khớp động i+1.

Trục xi của hệ tọa độ thứ i nằm dọc theo đường vuông góc chung hướng từ khớp động i đến khớp động i+1. Trường hợp hai trục giao nhau , hướng trục xi trùng với hướng vector tích zi \* zi-1, tức là vuông góc với mặt phẳng chứa zi và zi-1.

1. **Mô hình biến đổi**

Trên cơ sở đã xây dựng các hệ tọa độ với 2 khâu động liên tiếp như trên , có thể thiết lập mối quan hệ giữa 2 hệ tọa độ liên tiếp theo 4 bước sau:

1.Quay quanh trục zi-1 một góc .

2 Tịnh tiến dọc trục zi-1 một quãng d1.

3. Tịnh tiến dọc trục xx-1 ( đã trùng với xi ) một đoạn ai.

4. Quay quanh trục xi một góc .

Bốn bước biến đổi này được biểu hiện này bằng tích các ma trận thuần nhất sau:

Ai  =

Biểu thức trên là quan hệ giữa hệ tọa độ i so với hệ tọa độ i-1 và được gọi là mô hình DH ( DH - model ). Cách thiết lập mô hình động học theo kiểu mô hình DH tỏ ra thuận tiện trong khi giải quyết các vấn đề cơ học robot, vì thế được dùng khá rộng rãi. Ngoài ra còn tồn tại một vài cách khác để thiết lập mô hình động học như Universal – model, S- model [26].

Ta có:

T =

R(x,) =

R(z,) =

Qua đó Ai  =

* + 1. **Trình tự thiết lập hệ phương trình động học của robot**

Để thiết lập phương trình động học của robot, có thể tiến hành các bước sau:

1. Xác định các hệ tọa độ

2. Lập bảng thông số DH.

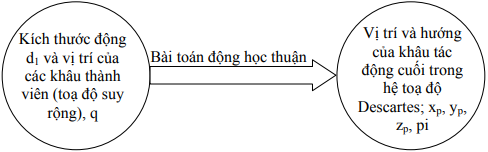
3. Xác định các ma trận Ai theo các thông số DH.

4. Lập phương trình động học cơ bản Ai  = 0A1 1A2 2A3 ... i-1Ai

* + 1. **Động học thuận tay máy**

1. **Tổng quan về bài toán động học tay máy**

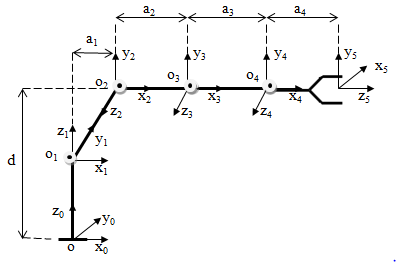
Cho trước cơ cấu và quy luật của các yếu tố chuyển động thể hiện bằng các toạ độ suy rộng ta phải xác định quy luật chuyển động của điểm trên khâu tác động cuối nói riêng hoặc của điểm bất kỳ trên một khâu nào đó của tay máy nói chung trong hệ trục toạ độ vuông góc (hệ trục toạ độ Descartes). Hay nói cách khác là xác định vị trí trong hệ tọa độ Descartes(X, Y, Z) khi biết được vị trí của các góc khớp.

****

*Hình 2.15: Sơ đồ mô tả tổng quan bài toán động học thuận tay máy*

* 1. **Giải bài toán động học thuận sử dụng phương pháp Denavit – Hartenberg**
* Đặt hệ trục tọa độ: Theo Denavit – Hartenberg (1955) đã quy ước hệ tọa độ Decard gắn vào mỗi khâu của một tay máy robot như sau:
* Chọn trục Zi : trục Zi nằm dọc theo trục khớp thứ i+1. Hướng của phép quay và phép tịnh tiến được chọn tùy ý.
* Chọn trục Xi :được xác định dọc theo đường vuông góc chung giữa trục khớp động thứ i và (i+1), hướng từ khớp động thứ i tới trục (i+1).
* Trục Yi  được xác định theo qui tắc bàn tay phải.

Từ quy tắc trên ta xây dựng các tọa độ khảo sát: O0X0Y0Z0, O1X1Y1Z1, O2X2Y2Z2, O3X3Y3Z3, O4X4Y4Z4.

****

*Hình 2.17: Hệ tọa độ của cánh tay robot 4 bậc tự do*

* Xác định các thông số động học của bảng Denavit – Hartenberg (DH)

Vị trí của hệ tọa độ khớp (Oxyz)i đối với hệ tọa độ khớp (Oxyz)i-1 được xác định bởi 4 tham số như sau:

* : khoảng cách giữa hai trục z
* : zi-1 xoay quanh x1 một gócđể sao cho zi-1 trùng với zi.
* : khoảng cách giữa hai trục x
* : góc xoay quanh trục z
* Thiết lập bộ thông số động học DH

Với cách thành lập hệ tọa độ như trên ta có thể xác định các tham số động học của robot như sau:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Khâu |  |  |  |  |
| 1 | a1 |  | d |  |
| 2 | a2 |  | 0 |  |
| 3 | a3 |  | 0 |  |
| 4 | a4 |  | 0 |  |

*Bảng …: Tham số động học của Robot*

* Dạng tổng quát của ma trận Denavit – Hartenberg cho các khâu

i-1Ai=

* Ma trận DH cho từng khâu
* Ma trận Denavit- Hartenberg của khâu 1:

0A1 =

* Ma trận Denavit- Hartenberg của khâu 2:

1A2 =

* Ma trận Denavit- Hartenberg của khâu 3:

2A3 =

* Ma trận Denavit- Hartenberg của khâu 4:

3A4 =

Từ đó ta có thể tính được ma trận chuyển đổi từ tọa độ ban đầu tới tọa độ điểm cuối là:

0A4=0A1\*1A2\*2A3\*3A4

orientation

position

0A4 =

Ma trận chuyển đổi 0A4 giúp ta có thể tính được tọa độ các điểm px, py, pz. Từ phần mềm Matlab ta có thể tính được giá trị px, py, pz cụ thể như sau:

px = cos()\*(a3\*cos() + a2\*cos() + a4\*cos())

py = sin()\*(a3\*cos() + a2\*cos() + a4\*cos())

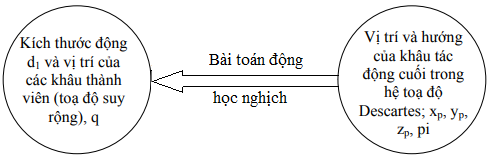
pz = a1 + a3\*sin() + a2\*sin() + a4\*sin()

Có quỹ đạo điểm P(Px, Py, Pz) là tọa độ điểm cuối của tay gắp tương ứng với tọa độ Ox5y5z5

* + 1. **Bài toán động học nghịch(động học ngược) tay máy.**

1. **Tổng quan về bài toán động học nghịch tay máy**

Cho trước cơ cấu và quy luật chuyển động của điểm trong khâu tác động cuối (hoặc quy luật chuyển động của khâu cuối bao gồm vị trí và hướng của nó) được biểu diễn trong hệ tọa độ vuông góc, ta phải xác định quy luật chuyển động của các khâu thành viên thể hiện qua các hệ tọa độ suy rộng. Hay nói một cánh đơn giản là cho tọa độ điểm P(Px, Py, Pz) bất kì trong không gian hoạt động của tay máy, ta phải tính toán chiều, và góc quay của từng khớp sao cho khâu cuối đạt đên tọa độ của điểm P.



*Hình …: Sơ đồ mô tả tổng quan bài toán động học nghịch tay máy*

1. **Giải bài toán động học nghịch**

* Bài toán động học vị trí nghịch là xác định giá trị các biến khớp () khi biết được vị trí và hướng của dụng cụ (so với hệ tọa độ gốc)

A4 = 0A11A22A33A4

⬄ 0A1-1 A4 = 0A1-10A11A22A33A4

⬄ =

⬄

Ký hiệu: ,,,

,,,

,

,

Có:

n1 =

n2  =

n3 =

m1 =

m2 =

m3 = 0

\*

⬄⬄

⬄

Lại có:

⬄

⬄

⬄

Đặt

⬄ (1)

Tính nx, ny:

mà

* tan

Tính :

⬄

⬄

⬄ (\*)

Thay (3) vào (2):

⬄C2  =

⬄C2  =

⬄C2  =

⬄=

⬄ =

⬄ =

Trong đó nx =

ny =

Thay C2 vào (\*):

S2 =

tan =

* = arctan2(S2 ,C2)

Lại có: + +



***Kết luận:*** Từ tọa độ điểm P(x,y,z) ta có thể tính toán được giá trị các góc , , , .

* 1. **Thiết kế cánh tay robot 4 bậc tự do sử dụng phần mềm Autodesk Inventor**
     1. **Giới thiệu phần mềm Autodesk Inventor**

1. **Inventor là gì?**

Autodesk Inventor là phần mềm xây dựng mô hình 3D, thiết kế, hình mẫu và kiểm tra ý tưởng các sản phầm. Inventor tạo ra các nguyên mẫu mô phỏng chuẩn xác khối lượng, áp lực, độ ma sát, tải trọng,… của các đối tượng sản phẩm trong môi trường 3D. Các công cụ mô phỏng, phân tích được tích hợp trong Inventor cho phép người dùng thiết kế từ khuôn đúc cơ bản đến nâng cao như thiết kế chi tiết máy, trực quan hóa sản phẩm. Inventor còn được tích hợp CAD và các công cụ giao tiếp thiết kế nhằm nâng cao năng suất làm việc của CAD và giảm thiếu phát sinh lỗi, tiết kiệm thời gian.



*Hình …: Phần mềm Autodesk Inventor Professional 2018*

1. **Lợi ích của Inventor**

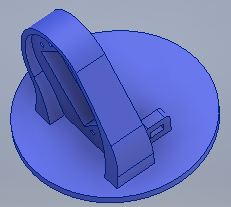
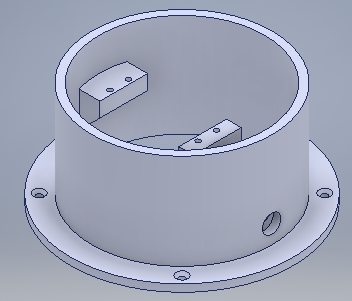
Những người đang hoặc sẽ sử dụng AutoCAD sẽ thừa hưởng được nhiều lợi ích của Inventor. Inventor cũng cấp một môi trường thiết kế và phím tắt tương tự với AutoCAD, hỗ trợ tập tin DWG, cho phép người dùng chuyển từ vẽ 2D hiện hành sang xây dựng mô hình 3D. Inventor được sử dụng phổ biến trong tạo nguyên mẫu kỹ thuật số, các mẫu được tạo ra từ bản vẽ 2D AutoCAD được tích hợp và các dữ liệu 3D, hình thành nên sản phẩm ảo. Bằng cách này, các kỹ sư có thể thiết kế, mô phỏng sản phẩm mà không phải tạo ra các mẫu vật lý. Người dùng có thể sử dụng các công cụ thiết kế 3D cơ khí trong Inventor để nghiên cứu và đánh giá mô hình thuận tiện và hiệu quả hơn. AutoDesk Inventor còn đưa ra các công cụ và tính năng khác nhằm nâng cao năng suất làm việc như:  Integrated Data Management, Design Automation, Automatic Drawing Updates and Views, Automatic Bill of Materials,…

1. **Ứng dụng của Inventor**

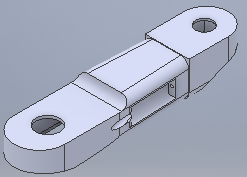
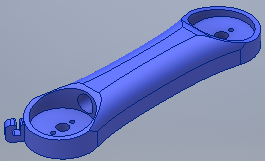
Autodesk Inventor được sử dụng để rút ngắn khoảng cách giữa thiết kế, kỹ thuật và sản xuất. Ví dụ: ngành sản xuất xe đạp leo núi sử dụng Inventor tạo ra các mẫu kỹ thuật số và thành phẩm để tối ưu các tác động giữa các bộ phận lắp ráp, đảm bảo chính xác dung sai và các khoảng hở. Trong sản xuất thuyền buồm, Inventor được sử dụng để xây dựng mô hình và tạo mẫu đột phá, chạy các thử nghiệm áp lực nhằm xác định các bộ phận cần cắt giảm khối lượng và cải thiện hiệu suất. Một ứng dụng khác của Inventor là trong công nghiệp khai khoáng, nhằm thực hiện các phân tích ứng suất, mô phỏng chuyển động máy tìm ra các va chạm ngoài ý muốn và các lỗi phát sinh. Inventor giúp cắt giảm chi phí sản xuất bằng việc tạo ra các nguyên mẫu và thử nghiệm ảo, hạn chế lỗi và lao động thủ công, tăng chu kỳ sản xuất và đưa sản phẩm đến thị trường nhanh hơn.

* + 1. **Thiết kế cánh tay robot 4 bậc tự do**

1. **Phần thân cánh tay robot 4 bậc tự do**

****

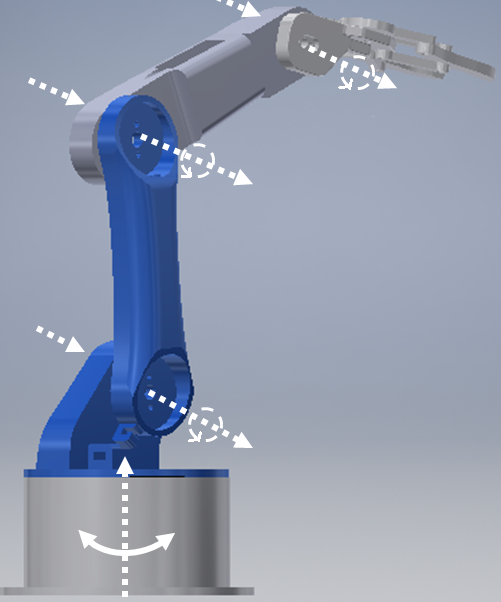
|  |  |
| --- | --- |
| ***(a)*** | ***(b)*** |

****

|  |  |
| --- | --- |
| ***(c)*** | ***(d)*** |

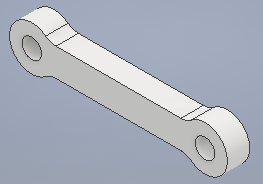
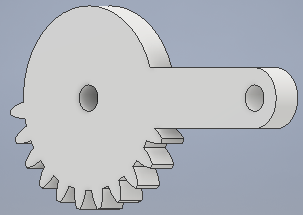
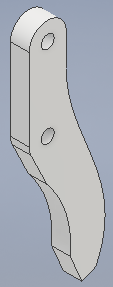
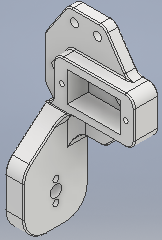
*Hình …: Các khâu của robot*

* + - * 1. *base; (b) waist; (c) link 1 ;(d) link 2*

****

*Hình …: 4 khớp xoay của robot khi lắp ghép các phần của robot lại với nhau*

1. **Phần tay gắp**

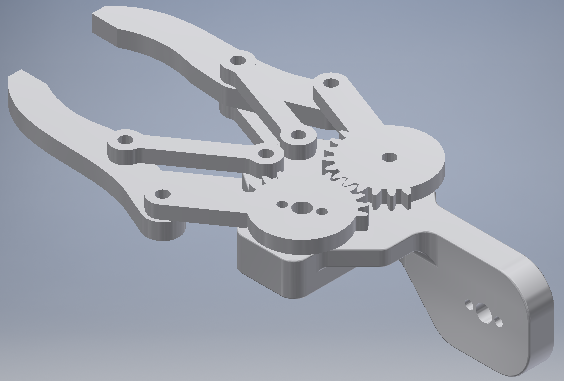


***(b)***

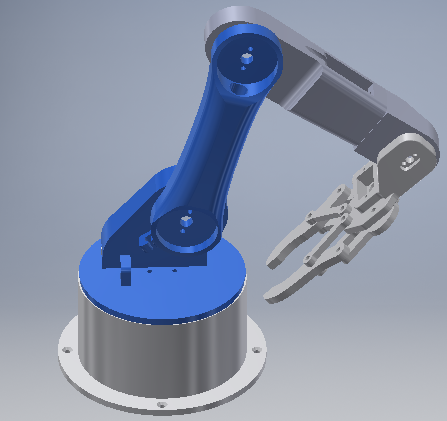
***(a)***

|  |  |
| --- | --- |
| ***(c)*** | ***(d)*** |
| *Hình …: Các khâu của robot*   1. *Gripper base; (b)Gipper; (c)Gear;(d) Grip link* | |
|  |  |

Để lắp ghép thành 1 tay gắp hoàn chỉnh cần 01 Gripper base, 02 Gripper, 02 Gear, 04 Gripper link. Động cơ servo sẽ được gắn ở Gripper base sau đó chuyền động tới các gear, thông qua các grip link tới Gripper.

****

*Hình …: Hình ảnh gripper hoàn chỉnh*

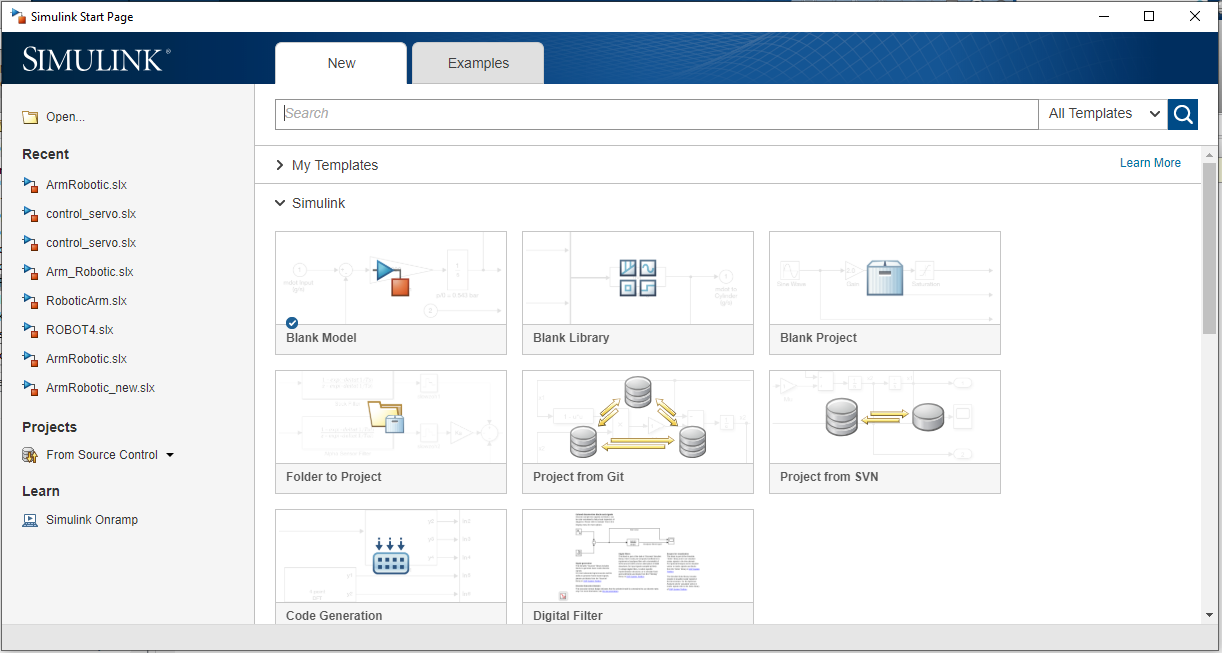
****

*Hình …: Mô hình cánh tay robot 4 bậc tự do hoàn chỉnh*

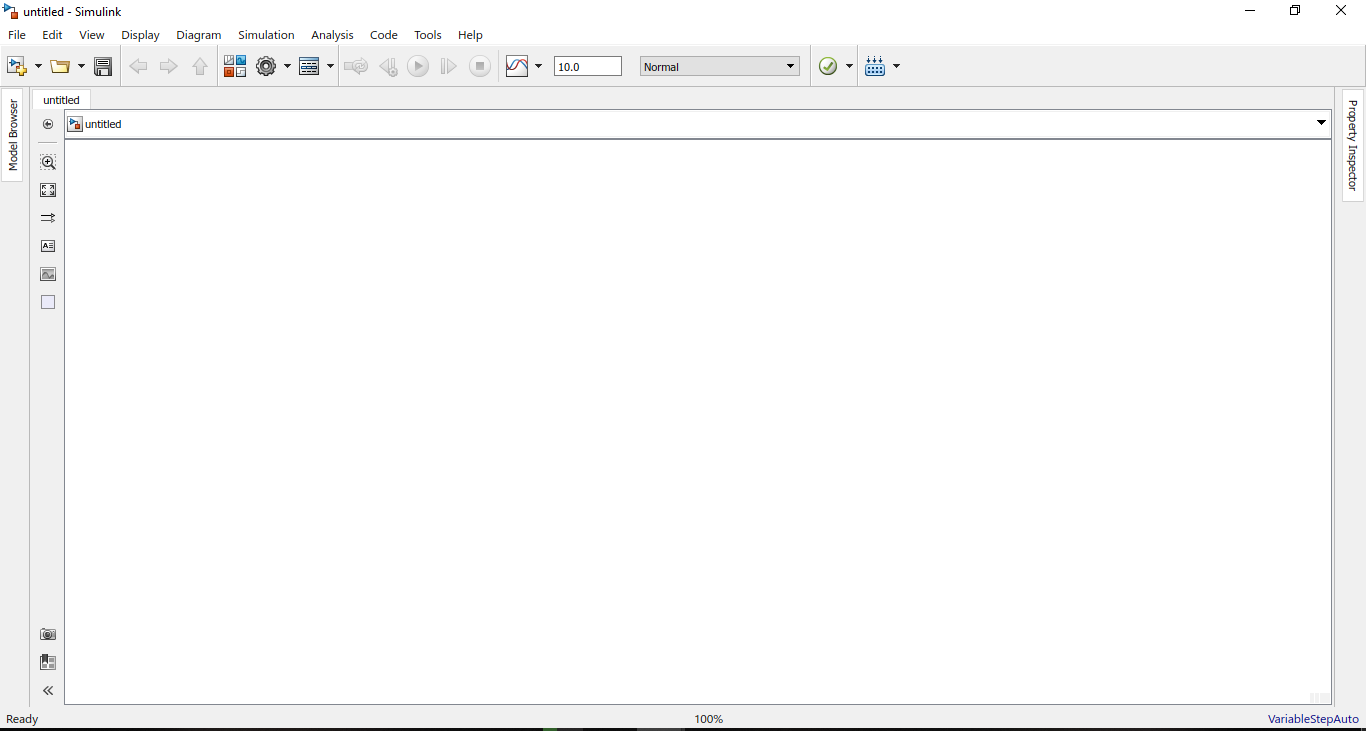
* 1. **Mô phỏng với thời gian liên tục trên nền tảng Matlab/Simulink**
     1. **Giới thiệu về Simulink và Simcape Mutibody**

1. **Simulink**

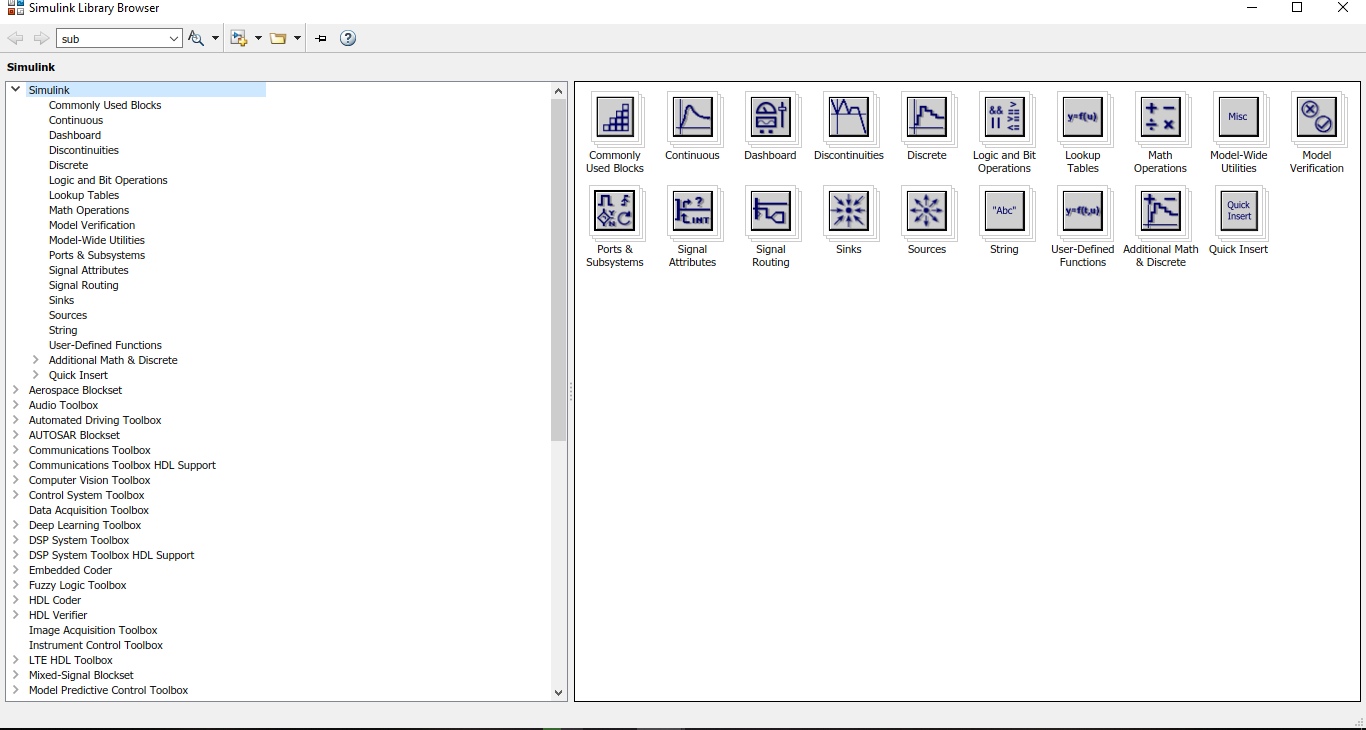
Simulink là phần chương trình mở rộng của Maltab nhắm mục đích mô hình hóa, mô phỏng và khảo sát các hệ thống động học. Giao diện đồ họa trên màn hình của Simulink cho phép thể hiện hệ thống dưới dạng sơ đồ tín hiệu với các khối chức năng quen thuôn. Simulink cung cấp cho người sử dụng một thư viện rất phong phú, có sẵn với số lượng lớn các khối chức năng cho các hệ tuyến tính, phi tuyến, gián đoạn. Hơn thế, người sử dụng cũng có thể tạo nên các khối riêng mình.



*Hình …: Giao diện của simulink*



*Hình …: Cửa sổ làm việc của simulink*

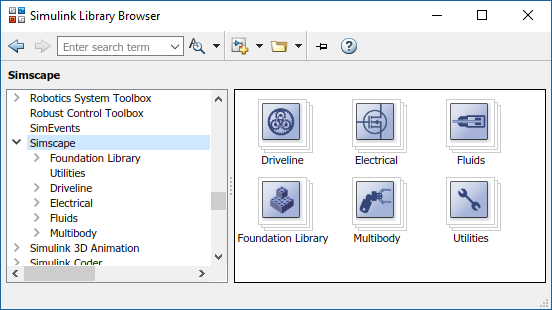


*Hình …: Các thư viện của simulink*

1. **Simcape Mutibody(trước đây là SimMechanics)**

Simscape Multibody ™ (trước đây là SimMechanics ™) cung cấp môi trường mô phỏng đa khung cho các hệ thống cơ khí 3D, chẳng hạn như robot, hệ thống treo xe, thiết bị xây dựng và thiết bị hạ cánh của máy bay. Người dùng có thể mô hình hóa các hệ thống đa khung bằng cách sử dụng các khối đại diện cho các khớp, các ràng buộc, các yếu tố lực và cảm biến. Simscape Multibody xây dựng và giải các phương trình chuyển động cho hệ thống cơ hoàn chỉnh. Người dùng có thể tạo hệ CAD hoàn chỉnh, bao gồm tất cả khối lượng, quán tính, khớp, ràng buộc và hình học 3D, vào mô hình của bạn. Hoạt hình 3D được tạo tự động cho phép người hình dung động lực học của hệ thống.

Simscape Multibody giúp người phát triển các hệ thống điều khiển và kiểm tra hiệu năng cấp hệ thống. Người dùng có thể tham số hóa các mô hình của mình bằng các biến và biểu thức MATLAB ® và hệ thống kiểm soát thiết kế cho hệ thống đa khung của người dùng trong Simulink ®. Người dùng có thể tích hợp hệ thống thủy lực, điện, khí nén và các hệ thống vật lý khác vào mô hình của mình bằng các bộ phận từ dòng sản phẩm Simscape. Để triển khai các mô hình của người dùng đến các môi trường mô phỏng khác, bao gồm các hệ thống vòng lặp phần cứng (HIL), bên canh đó Simscape Multibody còn hỗ trợ lập trình C – ngôn ngữ lập trình cơ bản, để người dùng dễ thao tác.



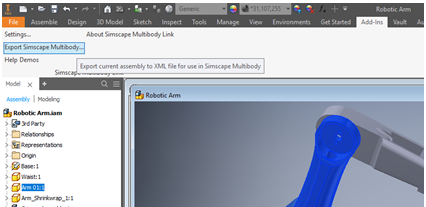
*Hình …: Tập hợp các loại khối Simcape*

* + 1. **Mô hình hóa và mô phỏng sử dụng Simcape Mutibody trong Simulink.**

1. **Liên kết phần mềm thiết kế Autodesk Inventor với phần mềm mô phỏng Matlab/Simulink sử dụng Simscape Multibody**

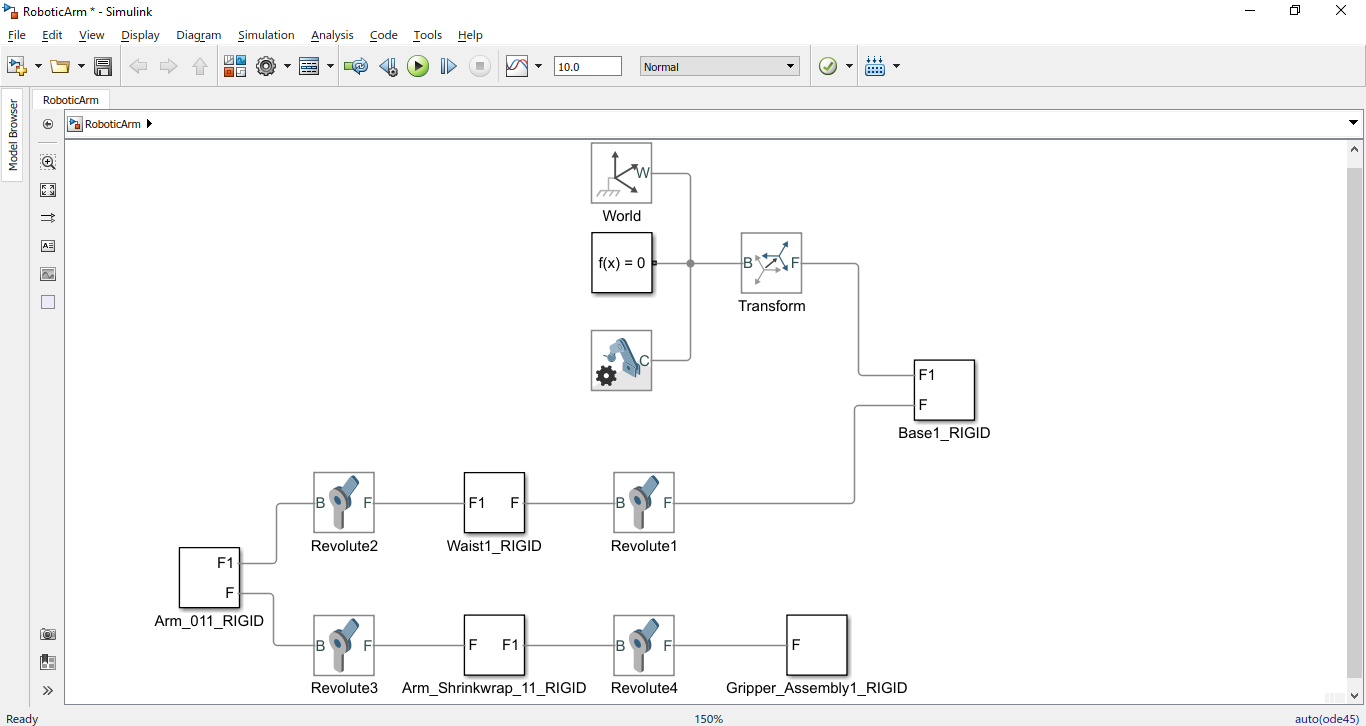
Matlab hỗ trợ chuyển đổi các phần trong phần mềm thiết kế như Soliwork hay Autodesk Inventor thành các khối trong Matlab/Simulink thông qua thư viện “smlink.r2019a.win64” với phiên bản Matlab 2019a.

Sau khi cài đặt thư viện và kết nối Matlab với Autodesk Inventor tiến hành chuyển đổi các bước: Chọn Add-Ins> Chọn Export Simscape Multibody.



*Hình …: Chuyển đổi bản thiết kế từ Autodesk Inventor sang các khối Simulink sử dụng chức năng Export Simscape Multibody*

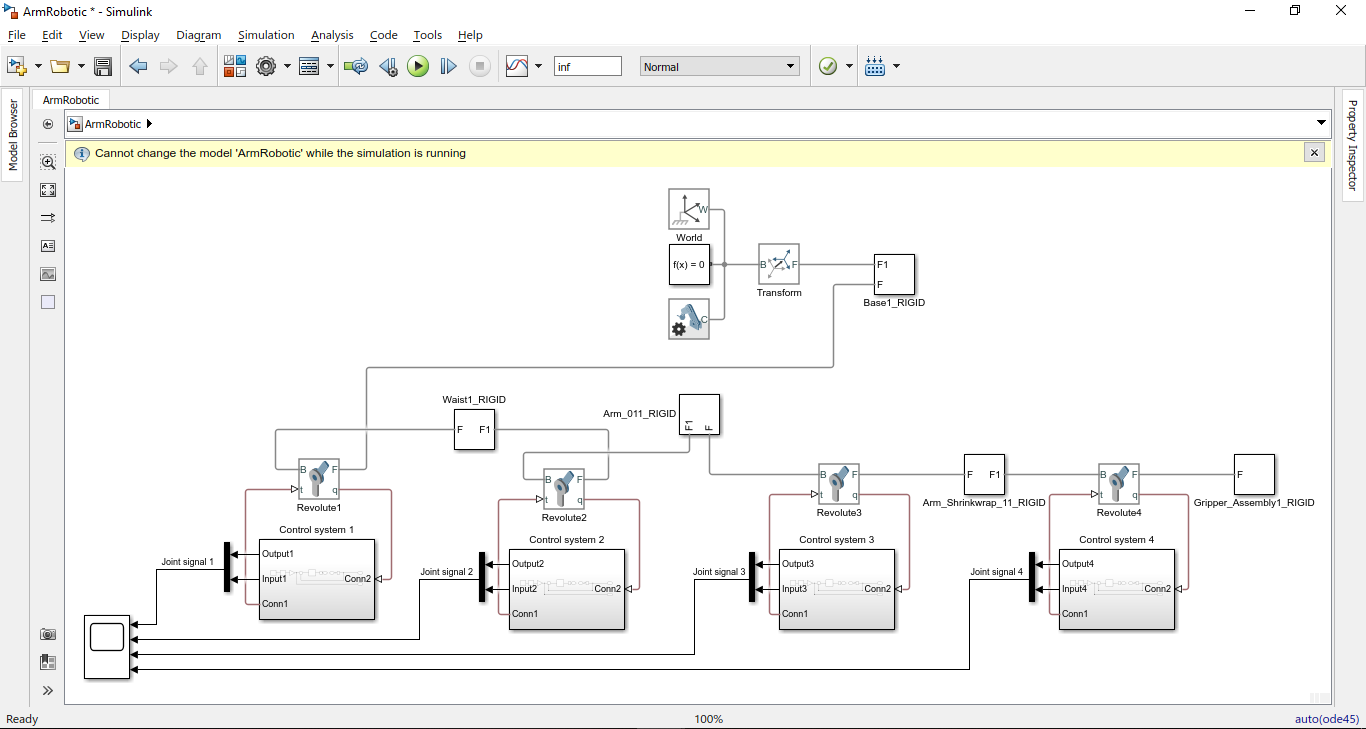
Sau khi chuyển thành công sẽ có một file .xml được tạo ra. File này là file mà Simulink có thể đọc để chuyển đổi nó thành các khối liên kết với nhau. Để đọc file .xml từ Matlab thông qua lệnh smimport(‘’) được các khối trong nền tảng simulink như ảnh dưới đây:



*Hình …: Sơ đồ khối sau khi chuyển sang Simulink*

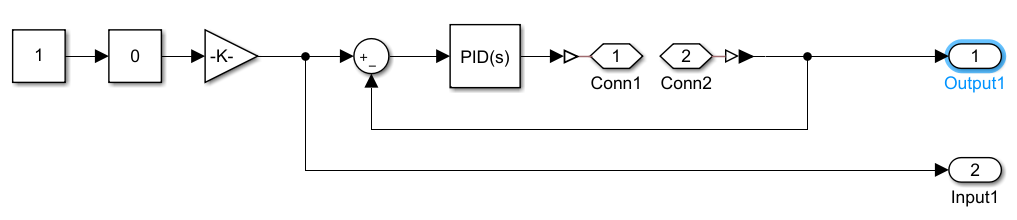
1. **Mô phỏng thời gian thực sử dụng Simcape trong Matlab/Simulink**

Trong Matlab, mô phỏng này là ở thế hệ thứ hai(2nd Generation) hay vì thế thệ thứ nhất(1st Generation). Ở các phiên bản Matlab trước đó người dùng có thể tùy chọn một trong hai thế hệ ngay từ khi chuyển từ các phần mềm thiết kế sang Matlab nhưng với phiên bản Maltab 2019a thì mặc định mô phỏng ở thế hệ thứ 2. Các thư viện và công nghệ mô phỏng thế hệ thứ hai tiên tiến hơn nhiều so với thế hệ đầu tiên. Những lợi thế rõ ràng nhất là trong mô phỏng 3D, nơi người dùng có thể xem lại các chuyển động trước của phần hoạt hình mà không cần chạy lại mô phỏng (trong Thế hệ thứ nhất, bạn chỉ có thể thấy hoạt hình trong khi mô phỏng đang chạy).

****

*Hình …: Sơ đồ mô phỏng trong Simulink*

Bên cạnh các khối được tạo ra trong quá trình chuyển đổi từ phần mềm thiết kế sang Matlab thì cần những khối lệnh để điều khiển cánh tay robot. 4 khối “Control system” được tạo ra để điều khiển liên tục 4 khớp xoay của robot sử dụng thuật toán điều khiển PID. Sơ đồ khối cấu thành nên các khối “Control system” được thiết lập từ các khối:

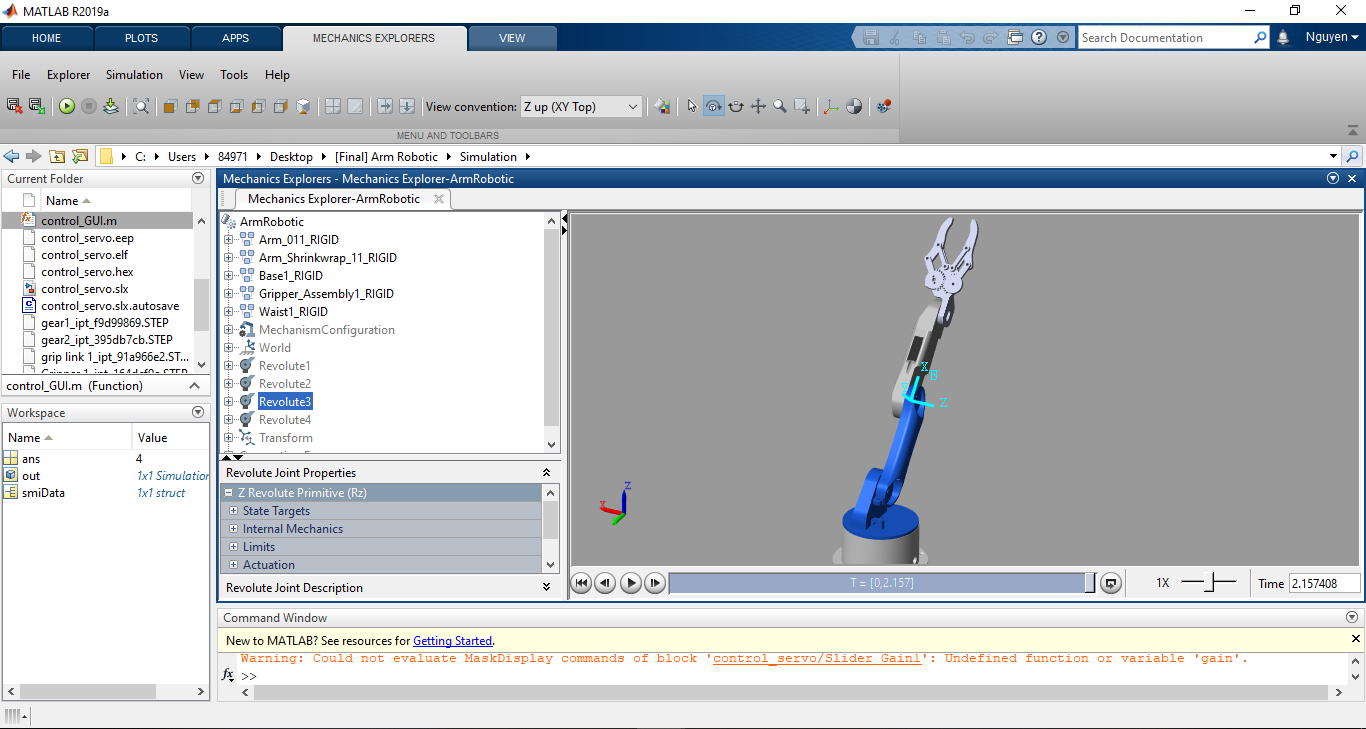
****

*Hình…: Sơ đồ các khối trong khối “Control system”*

|  |  |  |
| --- | --- | --- |
| STT | Khối | Chức năng |
| 1 | **revolute.PNG** | * Khối khớp xoay được tạo ra trong quá trìnhchuyển đổi sang Matlab Simulink * Khối này tích hợp sẵn các cảm biến để trả về các giá trị như tọa độ, vận tốc, gia tốc,… |
| 2 | **constand.PNG** | Khối hằng số đầu vào |
| 3 |  | Khối thay đổi hệ số khuếch đại vô hướng |
| 4 | **Gain.PNG** | Khuếch đại tín hiệu đầu vào |
| 5 | **PID Controller.PNG** | Bộ điều khiển PID |
| 6 | **scope.PNG** | Hiển thị tín hiệu của quá trình mô phỏng |

*Bảng …: Các khối mô phỏng được sử dụng*

1. **Kết quả mô phỏng**

****

*Hình …: Kết quả mô phỏng được hiện thị trong Mechanics Explorers*

* 1. **Xây dựng mô hình thực tế cánh tay robot 4 bậc tự do**
     1. **Tính toán lựa chọn động cơ servo phù hợp cho các khớp quay**

**a. Cấu hình cơ bản của sản phẩm**

- Cân nặng :

- Tải nâng tối đa: 50 g

- Nguồn:

**b. Tính toán động cơ servo**

**A close up of a logo

Description automatically generated**

*Hình …: Mô hình thể hiện các lực*

Để chuẩn bị loại servo thích hợp, ta cần tính toán các thông số mômen dựa từ mô hình lực phía trên. Các thông số đã biết :

0.55 N (Trọng lượng của khớp BC)

0.53 N (Trọng lượng của khớp CD)

0.61 N (Trọng lượng của khớp DE)

0.5 N (Tải)

(Trọng lượng của servo tại C)

(Trọng lượng của servo tại D)

Ta có

Vậy ta có

= 3.66 kg.cm

= 8 kg.cm

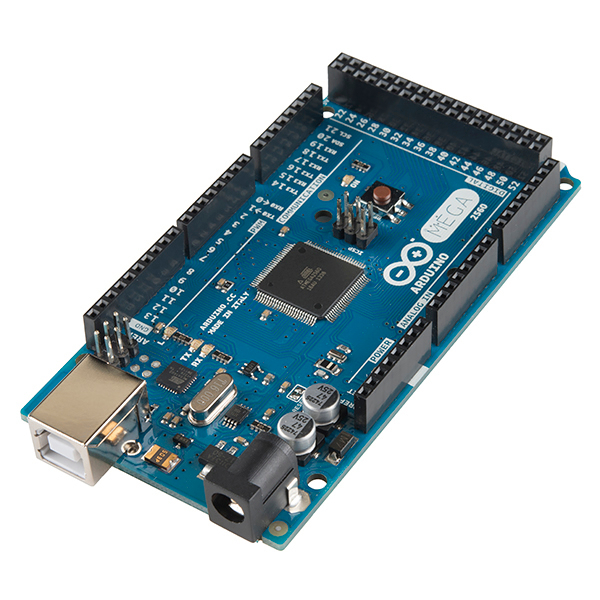
= 1.18 kg.cm

Dựa trên thông số tính toán được, nhóm đã lựa chọn các loại servo sau là servo MG996R và SG90. Với servo MG996R có momen giữ là 9.4 kg.cm (tại 4V) phù hợp với yêu cầu của khớp B và C. Còn servo SG90 có momen giữ là 2.5 kg.cm ( tại 4V) phù hợp với khớp D để quay tay gắp. Cả hai loại servo đều là hai loại phổ biến nên có dễ dàng trong việc sử dụng.

|  |  |
| --- | --- |
|  |  |
| **(a)** | **(b)** |
| *Hình …: Hình ảnh 2 loại độn cơ Servo*   1. *Động cơ servo MG996R; (b) Động cơ servo SG90* | |

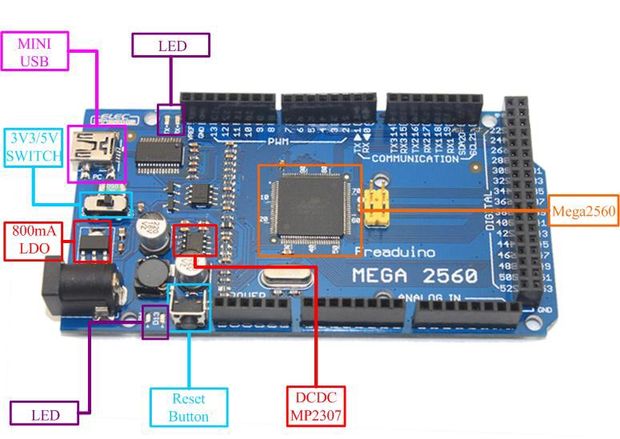
* + 1. **Phần cứng sử dụng**

1. **Arduino Mega 2560**

****

*Hình …: Arduino Mega 2560*

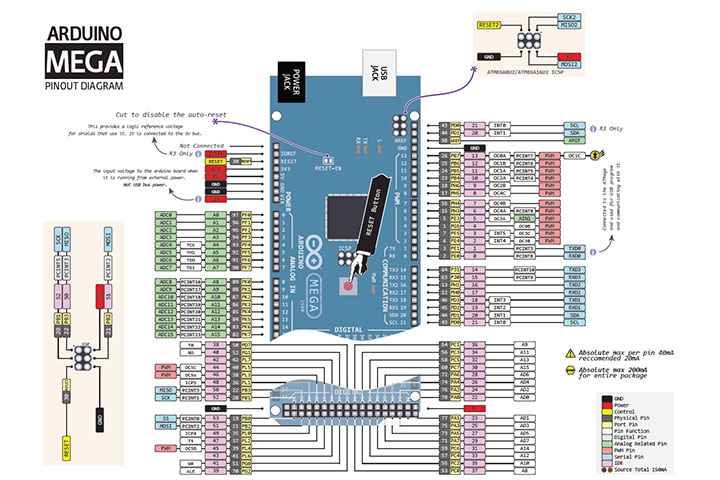
Arduino Mega 2560 là phiên bản hiện đang được sử dụng rộng rãi. Arduino Mega 2560 sử dụng chip ATmega2560 có bộ nhớ flash memory 256 KB, 8KB cho bộ nhớ SRAM, 4 KB cho bộ nhớ EEPROM. Giúp cho người dùng thêm khả năng viết những chương trình phức tạp và điều khiển các thiết bị lớn hơn như robot, …



*Bảng …: Các linh kiện quan trọng cấu tạo trong Arduino*

|  |  |
| --- | --- |
| Vi điều khiển | ATmega2560 |
| Điện áp hoạt động | 5V |
| Điện áp đầu vào (được đề nghị) | 7 - 12V |
| Điện áp đầu vào (giới hạn) | 6 - 20V |
| Số lượng chân  I / O | 54 (14 chân có thể cấp xung PWM) |
| Số lượng chân Input Analog | 16 |
| Dòng điện DC  mỗi I / O | 40 mA |
| Dòng điện DC với chân 3.3V | 50mA |
| Bộ nhớ flash | 256 KB |
| SRAM | 8 KB |
| EEPROM | 4 KB |
| Tốc độ đồng hồ | 16 MHz |
| Chiều dài | 101,52 mm |
| Bề rộng | 53,3 mm |
| Cân nặng | 37 g |

*Bảng …: Thông số kỹ thuật Arduino Mega 2560*

****

*Hình …: Sơ đồ chân của Arduino Mega 2560*

Ngoài ra Arduino Mega 2560 là kit được tích hợp có sẵn thư viện dành cho Matlab/Simulink.

1. **Động cơ servo MG996R**

****

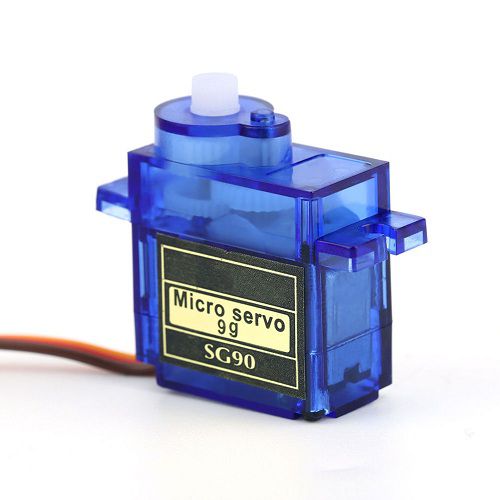
*Hình …: Động cơ MG996R*

Động cơ Servo MG966R là động cơ có mô men xoắn lớn, chạy mượt mà, phù hợp với những mô hình điều khiển có trọng tải lớn như cánh tay robot kim loại. Động cơ Servo MG966R được nâng cấp từ MG995 6V / 11kg, có tốc độ và độ chính xác cao.

|  |  |
| --- | --- |
| Khối lượng | 9 gram |
| Kích thước | 40.7 x 19.7 x 42.9 mm |
| Moomen xoắn | 9.4 kg/cm (4.8 V ), 11 kg/cm (6 V) |
| Tốc độ vận hành | 0.17 s/60º (4.8 V), 0.14 s/60º (6 V) |
| Điện áp làm việc | 4.8 V - 7.2 V |
| Nhiệt độ làm việc | 0 ℃ -55 ℃ |
| Góc quay | 0 - 180 |

*Bảng …: Thông số kĩ thuật của động cơ servo MG996R*

1. **Động cơ servo SG90**

****

*Hình …: Động cơ Servo SG90*

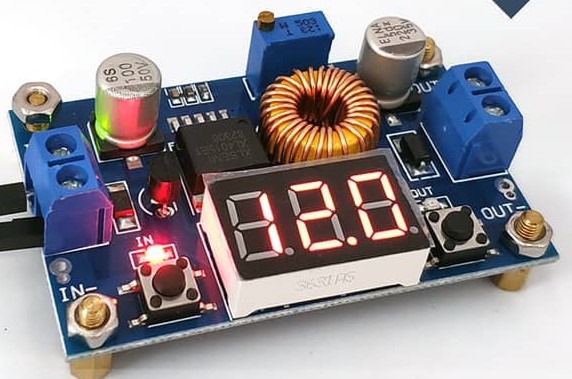
Động cơ RC Servo 9G là động phổ biến dùng trong các mô hình điều khiển nhỏ và đơn giản như cánh tay robot. Động cơ có tốc độ phản ứng nhanh, được tích hợp sẵn Driver điều khiển động cơ, dễ dàng điều khiển góc quay bằng phương pháp điều độ rộng xung PWM.

|  |  |
| --- | --- |
| Khối lượng | 9 gram |
| Kích thước | 23 x 12.2 x 29 mm |
| Moomen xoắn | 2.5kg/cm |
| Tốc độ | 0.3s/60 |
| Điện áp làm việc | 4.2 V - 6 V |
| Nhiệt độ làm việc | 0 ℃ - 55 ℃ |
| Góc quay | 0 - 180 |

*Bảng …: Thông số kĩ thuật của động cơ servo SG90.*

1. **Module giảm áp XL4015**

[**Mạch giảm áp DC XL4015**](https://nshopvn.com/product/mach-giam-ap-dc-xl4015-co-hien-thi-5a/) có hiển thị (5A) sử dụng phương pháp giảm áp xung để giảm áp DC với dòng đầu ra lên đến 5A, mạch có tích hợp đồng hồ Led và phím chức năng chọn hiển thị áp đầu vào và đầu ra để tiện theo dõi.



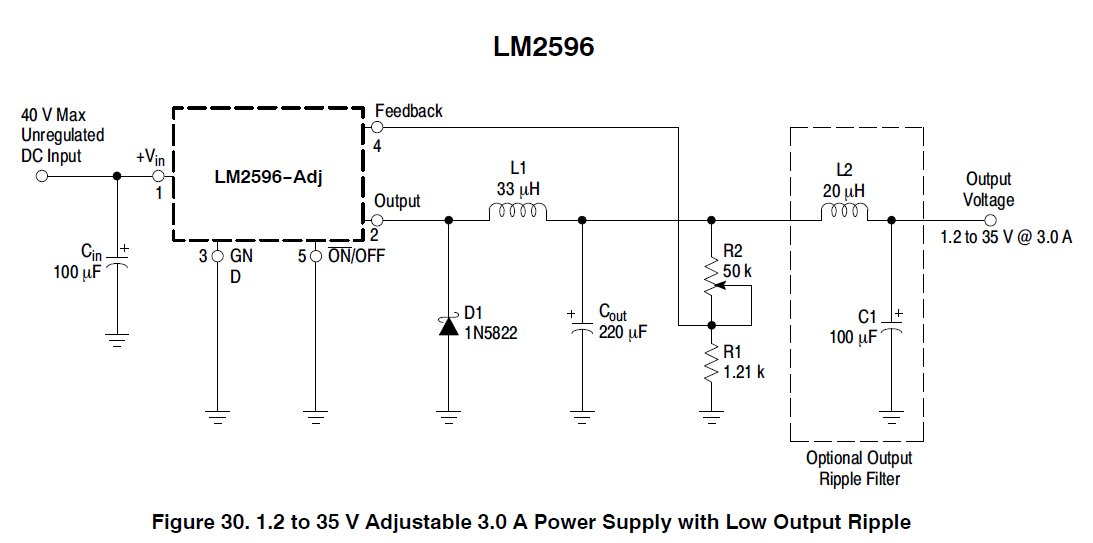
*Hình …: Module giảm áp DC XL4015*

Module giảm áp DC XL4015 sử dụng IC ổn áp LM2596S. LM2596S là một loại IC ổn áp dạng DC – DC. Điện áp đầu vào trong dải từ 4,5V-40V. Điện áp đầu ra điều chỉnh được trong khoảng từ 1,5V-37V, dòng điện áp đầu ra  đạt 3A hiệu suất cao nhờ cơ chế băm xong ở tần số lên tới 150KHz. Trong quá trình hoạt động LM2596 luôn được đặt trong các chế độ bảo vệ quá nhiệt vào quá dòng.

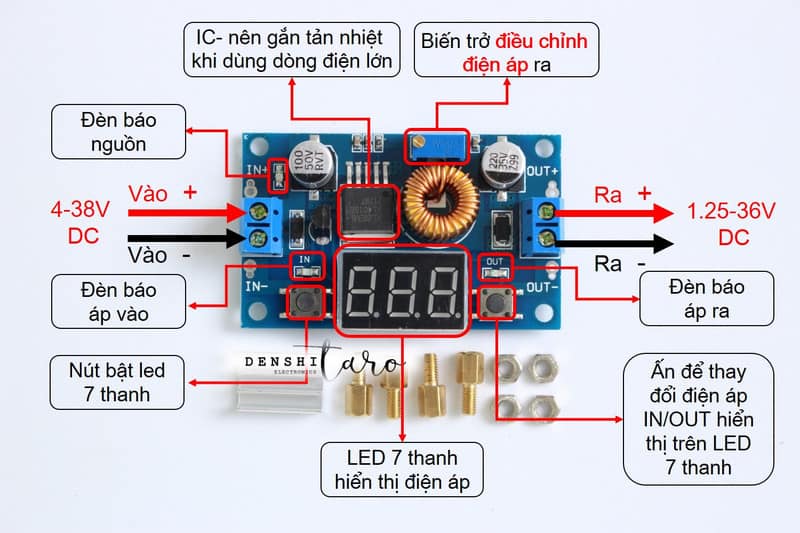
**

*Hình …: IC LM2596S dạng chân dán*

* IC LM 2596 có 5 chân
* Chân 1: V in từ 4,5 ~ 40V
* Chân 2: V out
* Chân 3: GND
* Chân 4:  Feedback ( chân phản hồi điện áp)
* Chân 5: ON/OFF chân tắt bật mức logic



*Hình …: Sơ đồ nguyên lý module giảm áp DC XL4015*



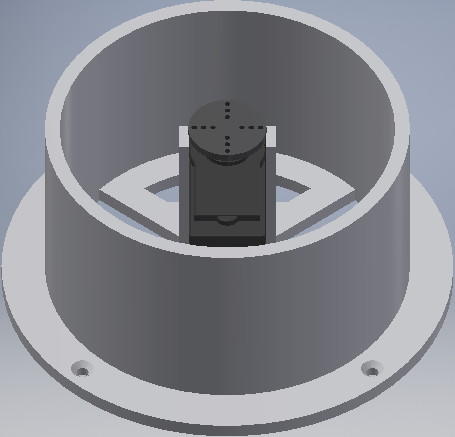
*Hình …: Sơ đồ mạch giảm áp DC XL4015*

|  |  |
| --- | --- |
| IC chính | XL4015 |
| Điện áp đầu vào | 4~38VDC |
| Điện áp đầu ra | 1.25 ~ 36VDC |
| Dòng đầu ra tối đa | 5A |
| Hiệu suất | 96% |
| Tần số xung | 180KHz |
| Nhiệt độ làm việc | -40 ~ 125 độ C |
| Trọng lượng | 25g |
| Kích thước | 60 x 37 x 18mm |

*Hình …: Thông số kỹ thuật mạch giảm áp DC XL4015*

1. **Nguồn cấp**

* Nguồn cấp cho Arduino: Vì điện áp hoạt động của Arduino là 5V, mặt khác Arduino Mega 2560 sẽ được kết nối và điều khiển Servo thông qua phần mềm Matlab trên PC/laptop nên nhóm sử dụng nguồn cấp từ PC/Laptop có điện áp ổn định khoảng 5V.
* Vì nguồn cấp cho động cơ servo MG996R là 4.8 – 7.2V, nguồn cấp cho động cơ servo SG90 là 4.2 – 6V và trên cách tay robot có tổng cộng 5 động cơ servo(*3 động cơ MG996R và 2 động cơ SG90*) phối hợp liên tục và yêu cầu nguồn ổn định nên nhóm sử dụng nguồn ổn định 6V thông qua module giảm áp XL4015.
  + 1. **Lắp ráp mô hình**



*Hình …: Lắp động cơ servo MG996R số 1 với Base(đế robot)*

|  |  |
| --- | --- |
| **2.PNG**  *Hình …: Lắp waist* | *3.PNG*  *Hình …: Lắp động cơ servo MG996R số 2* |

|  |  |
| --- | --- |
| *4.PNG*  *Hình …: Phần 4 - Lắp Link 1*  *6.PNG*  *Hình …: Lắp link 2 và động cơ servo SG90 số 1* | *5.PNG*  *Hình …: Phần 5 - Lắp động cơ servo MG996R số 3*  *7.PNG*  *Hình …: Lắp Gripper* |

****

*Hình …: Mô hình lắp ráp hoàn chỉnh cánh tay robot 4 bậc tự do*

* 1. **Điều khiển** 
     1. **Tổng quan mô hình điều khiển cánh tay robot 4 bậc tự do**

1. **Bài toán điều khiển chuyển động**

Chức năng của hệ thống điều khiển chuyển động là đảm bảo tay robot(End effector) chuyển động bám theo quỹ đạo đặt trước trong môi trường làm việc(không gian làm việc). Chuyển động của tay 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 tón từ lượng đặt vị trí của tay robot trong không gian làm việc thông qua khâu tính toán độc 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(khe hở trong hộp số,…) hoặc thiếu thông tin quan hệ vị trí giữa tay robot và đối tượng.

Hệ thống điều khiển không gian làm việc có chức năng duy trì trự 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 tay trong không gian làm việc bằng không. Khâu tính toán động học ngược sẽ thuộc mạch vòng điều khiển phản hôi. Ưu điểm của phương pháp điều khiển này là tác động trực tiếp các biến không gian làm việc. Nhược điểm là khối tính toán lớn do tồn tại khâu tinh toán động học ngược vòng điều khiển.

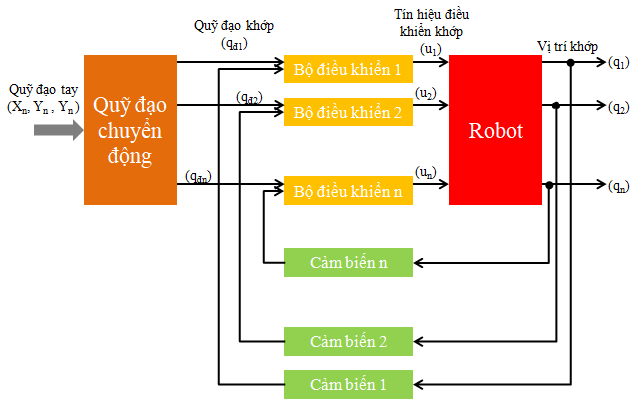
1. **Hệ thống điều khiển độc lập các khớp**

Đồi với các robot có tỉ số truyền của bộ truyền lớn, có thể coi hệ thống robot n bậc tự do sẽ gồm n hệ thống động lập và là hệ thống 1 đầu vào/1 đầu ra(SISO) và sự rằng buộc giữa các khớp được coi là thành phần nhiễu. Trong trường hợp đơn giản, nếu bỏ qua thành phần mô men nhiễu, sơ đồ cấu trúc mô tả một khớp robot có dạng như hình dưới. Sơ đồ dưới mô tả một khớp bất kỳ và không dùng ký hiệu chỉ số I trong các đại lượng của khớp.

<Hình: Sơ đồ cấu trúc của hệ thống chuyển động của một khớp>

1. **Hệ thống điều khiển vị trí**

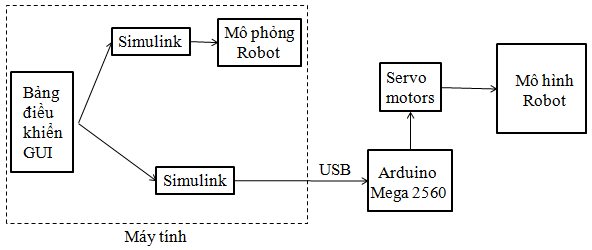
Bộ điều khiển vị trí (Bộ ĐKi)có chức năng điều khiển chuyển động của robot. Khâu tạo quỹ đạo chuyển động sẽ tính toán các quỹ đạo chuyển động mong muốn của tứng khớp(qđi) từ quỹ đạo mong muốn của tay robot (xđ, yđ, zđ). Quỹ đạo chuyển động mong muốn khớp(qđi) là tín hiệu đặt vị trí của các bộ điều khiển vị trí của từng khớp. Tín hiếu hiệu ra của các bộ điều khiển(ui) là tín hiệu điều khiển hệ truyền động của khớp tưng ứng. Mỗi bộ điều khiển vị trí của từng khớp là bộ điều khiển secvo với tín hiệu phản hồi là vị trí của khớp tương ứng được đọ bởi các cảm biến vị trí(CBi). Chi tiết được mô tả trong sơ đồ khổi dưới đây.



*Hình …: Sơ đồ khối hệ thống điều khiển vị trí*

1. **Sơ đồ khối tổng quan hệ thống điều khiển song song mô hình và mô phỏng cánh tay robot 4 bậc tự do.**

Bảng điều khiển được tạo bằng GUI Matlab sau sẽ nhận các giá trị góc xoay của từng khớp(động học thuận) hoặc giá trị vị trí khâu cuối của tay máy(động học nghịch) truyền các thông số tới Simulink mô phỏng để thay đổi giá trị các khối thành phần từ đó tác động vào khớp xoay của mô hình cánh tay robot trên máy tính. Song song với đó, cũng truyền giá trị tới Simulink để truyền giá các thông số đó tới Arduino. Arduino sẽ nhận tín hiệu và điều khiển các động cơ servo từ đó thay đổi các khớp xoay của mô hình thực tế.

**

*Hình …: Sơ đồ khối tổng quan phần điều khiển*

* + 1. **Điều khiển mô phỏng thời gian thực sử dụng thuật toán điều khiển PID**

1. **Giới thiệu về thuật toán PID**

Một bộ điều khiển vi tích phân tỉ lệ (bộ điều khiển PID - Proportional Integral Derivative) là một [cơ chế phản hồi](https://vi.wikipedia.org/w/index.php?title=C%C6%A1_ch%E1%BA%BF_ph%E1%BA%A3n_h%E1%BB%93i&action=edit&redlink=1) [vòng điều khiển](https://vi.wikipedia.org/w/index.php?title=V%C3%B2ng_%C4%91i%E1%BB%81u_khi%E1%BB%83n&action=edit&redlink=1) ([bộ điều khiển](https://vi.wikipedia.org/w/index.php?title=B%E1%BB%99_%C4%91i%E1%BB%81u_khi%E1%BB%83n_(l%C3%BD_thuy%E1%BA%BFt_%C4%91i%E1%BB%81u_khi%E1%BB%83n_t%E1%BB%B1_%C4%91%E1%BB%99ng)&action=edit&redlink=1)) tổng quát được. Bộ điều khiển PID sẽ tính toán giá trị "sai số" là hiệu số giữa giá trị đo [thông số biến đổi](https://vi.wikipedia.org/w/index.php?title=Th%C3%B4ng_s%E1%BB%91_bi%E1%BA%BFn_%C4%91%E1%BB%95i&action=edit&redlink=1) và [giá trị đặt](https://vi.wikipedia.org/w/index.php?title=Gi%C3%A1_tr%E1%BB%8B_%C4%91%E1%BA%B7t_(h%E1%BB%87_th%E1%BB%91ng_%C4%91i%E1%BB%81u_khi%E1%BB%83n)&action=edit&redlink=1) mong muốn. Bộ điều khiển sẽ thực hiện giảm tối đa sai số bằng cách điều chỉnh giá trị điều khiển đầu vào nhắm giữ cho sai số ở mức thấp nhất.

[Giải thuật](https://vi.wikipedia.org/wiki/Thu%E1%BA%ADt_to%C3%A1n) tính toán bộ điều khiển PID bao gồm 3 thông số riêng biệt, do đó đôi khi nó còn được gọi là điều khiển ba khâu: các giá trị [tỉ lệ](https://vi.wikipedia.org/w/index.php?title=T%E1%BB%89_l%E1%BB%87(to%C3%A1n_h%E1%BB%8Dc)&action=edit&redlink=1), [tích phân](https://vi.wikipedia.org/wiki/T%C3%ADch_ph%C3%A2n) và [đạo hàm](https://vi.wikipedia.org/wiki/%C4%90%E1%BA%A1o_h%C3%A0m_v%C3%A0_vi_ph%C3%A2n_c%E1%BB%A7a_h%C3%A0m_s%E1%BB%91), viết tắt là *P,* *I,* và *D.* ham số tỉ lệ ảnh hưởng tới tác động bù trừ cho sai số hiện tại. Tham số tích phân quyết định tác động dựa trên tổng các sai số, và tham số vi phân điều chỉnh tác động dựa vào mức thay đổi của sai số. Kết quả tổng cộng của cả 3 tác động này sẽ được sử dụng để điều chỉnh tiến trình qua các thành phần điều khiển như vị trí của van hay dòng áp của nguồn điện cấp cho hệ thống đốt nóng.

Bằng cách điều chỉnh 3 tham số trong thuật toán PID, bộ điều khiển có thể đưa ra các tác động phù hợp với yêu cầu cụ thể của tiến trình cần điều khiển. Đáp ứng của hệ thống có thể được mô tả bằng khả năng điều chỉnh khi có sai số, mức độ tăng vọt khỏi điểm thiết đặt và mức độ dao động của hệ thống.

Một số ứng dụng chỉ cần sử dụng một hay hai tham số để điều khiển. Lúc đó bộ điều khiển PID thường được gọi là bộ điều khiển PI, PD hay P, I tương ứng với các tham số dùng để điều khiển. Bộ điều khiển PI thường hay được sử dụng trong thực tế, do tác động vi phân thường hay nhạy cảm với nhiễu của các phép đo, và nếu không có tác động tích phân thì hệ thống thường không đạt được trạng thái cần đặt do giới hạn thực tế của các thành phần điều khiển.

Đầu ra của bộ điều khiển PID thường là một biến điều khiển (manipulated variable - MV). Ta có thể viết:

**MV(t) = *Pout  + Iout + Dout***

Trong đó Pout, Iout và Dout lần lượt là tác động của các thành phần tỉ lệ, tích phân và vi phân.

* **Khâu tỉ lệ**

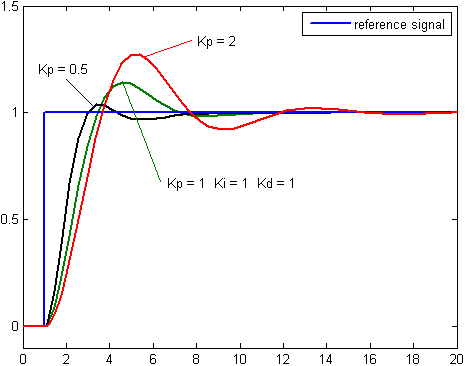
Thành phần tỉ lệ (hay đôi khi còn được gọi là thành phần khuếch đại - gain) làm cho đầu ra thay đổi tỉ lệ với sai số tức thời. Đáp ứng tỉ lệ có thể được thay đổi bằng cách điều chỉnh hệ số tỉ lệ Kp:

***Pout* = *Kp*e(t)**

Trong đó e là sai số, e = SP – PV với SP là điểm đặt trước, PV là biến trạng thái.

Hệ số tỉ lệ càng lớn sẽ làm cho biến điều khiển thay đổi càng lớn khi có thay đổi sai số. Nếu hệ số tỉ lệ quá lớn sẽ làm cho hệ thống mất ổn định hay dao động. Ngược lại, hệ số tỉ lệ nhỏ sẽ làm cho biến điều khiển thay đổi quá ít khi sai số lớn, dẫn đến 1 hệ thống đáp ứng chậm. Nếu hệ số tỉ lệ quá bé sẽ dẫn đến việc biến điều khiển quá nhỏ để có thể phản ứng lại các thăng giáng của hệ thống.

Khi không có thay đổi, một bộ điều khiển hoàn toàn tỉ lệ sẽ không đưa hệ thống về được trạng thái thiết đặt trước, mà sẽ giữ ở một trạng thái cân bằng với sai số phụ thuộc vào hệ số tỉ lệ và độ tăng ích của tiến trình. Mặc dù vậy, cả về lý thuyết điều chỉnh và thực tế trong công nghiệp đều cho thấy thành phần tỉ lệ thường nên đóng vai trò chính trong việc làm thay đổi đầu ra của hệ thống.



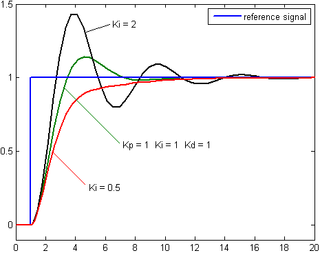
*Hình …: Ảnh hưởng của sai số tỉ lệ tới đầu ra của hệ thống*

* **Khâu tích phân**

Thành phần tích phân tỉ lệ với cả độ lớn của sai số lẫn thời gian kéo dài của sai số. Các sai số trước kia sẽ được tích luỹ và thêm vào đầu ra biến điều khiển sau khi nhân với hệ số tích phân Ki:

***Iout* = *Ki***

Khâu tích phân (khi cộng thêm khâu tỉ lệ) sẽ tăng tốc chuyển động của quá trình tới điểm đặt và khử số dư sai số ổn định với một tỉ lệ chỉ phụ thuộc vào bộ điều khiển. Tuy nhiên, vì khâu tích phân là đáp ứng của sai số tích lũy trong quá khứ, nó có thể khiến giá trị hiện tại vọt lố qua giá trị đặt (ngang qua điểm đặt và tạo ra một độ lệch với các hướng khác).

****

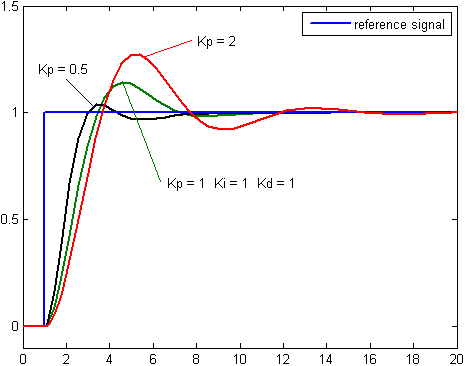
*Hình …: Ảnh hưởng của sai số tich phân tới đầu ra của hệ thống*

* **Khâu vi phân**

Tốc độ thay đổi sai số của tiến trình được tính bằng cách xác định độ dốc của sai số theo thời gian (đạo hàm bậc nhất của sai số theo thời gian), và nhân với hệ số vi phân Kd.

**Dout = *Kd***

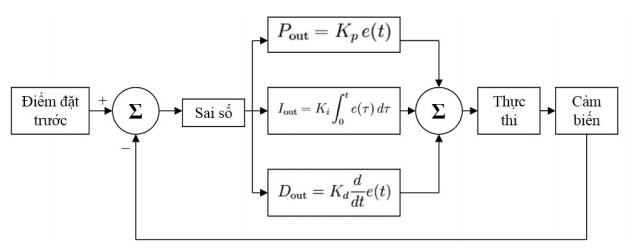
Thành phần vi phân làm giảm tốc độ thay đổi của biến điều khiển khi hệ thống gần đạt được trạng thái thiết đặt. Vì vậy thành phần tỉ lệ được dùng để giảm độ vượt quá gây nên bởi thành phần tích phân và cải thiện độ ổn định của hệ thống. Tuy nhiên thành phần vi phân sẽ khuếch đại nhiễu, do vậy nó rất nhạy cảm với nhiễu của đầu vào bộ điều khiển, và có thể khiến cho hệ thống trở nên mất ổn định khi nhiễu và hệ số vi phân đủ lớn.

****

*Hình …: Ảnh hưởng của sai số vi phân tới đầu ra của hệ thống*

Như vậy, nếu gọi u(t) là đầu ra của bộ điều khiển PID thì thuật toán PID có thể được biểu diễn dưới dạng:

**u(t) = *Kp*e(t) + *Ki* + *Kd***

****

*Hình …: Sơ đồ khối của bộ điều khiển PID*

1. **Điều khiển PID trong mô phỏng cánh tay Robot**

Khi không có điều khiển PID cho từng khớp, mỗi khớp của robot trong mô phỏng sẽ chuyển động 360 xung quay trục quay và hệ tọa độ đã được đặt sẵn. Do vậy để điều khiển mô phỏng xoay tới vị trí như mong muốn thì cần phải sử dụng thuật toán điều khiển PID

Trong mô phỏng động học cánh tay robot sẽ thực hiện điều khiển với 4 góc xoay độc lập, do đó cần dùng 4 bộ điều khiển PID cho từng khớp xoay

* Nếu Giá trị đầu vào là các 4 góc với 4 bậc tự do của tay robot thì đầu ra sẽ là tọa độ điểm cuối của robot(động học thuận tay máy).
* Nếu giá trị đầu vào là tọa độ điểm cuối của tay máy thì giá trị đầu ra sẽ là giá trị của các góc xoay(động học nghịch tay máy).

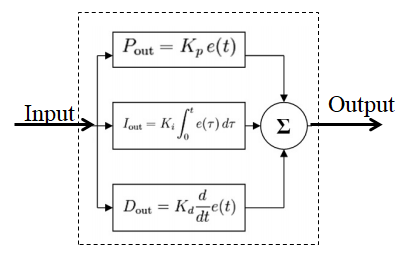


*Bảng …: Sơ đồ tổng quan điều khiển độc lập từng khớp trong mô phỏng dùng thuật toán PID*

Trong Simulink để tinh giảm số lượng các khối, tránh dối mắt mà vẫn đảm bảo được đúng thuật toán điều khiển nhóm đã sử dụng khối “PID Controller” trong Simulink để tính toán song song các giá trị P, I, D.



*Hình …: Khối “PID Controller” trong Simulink*



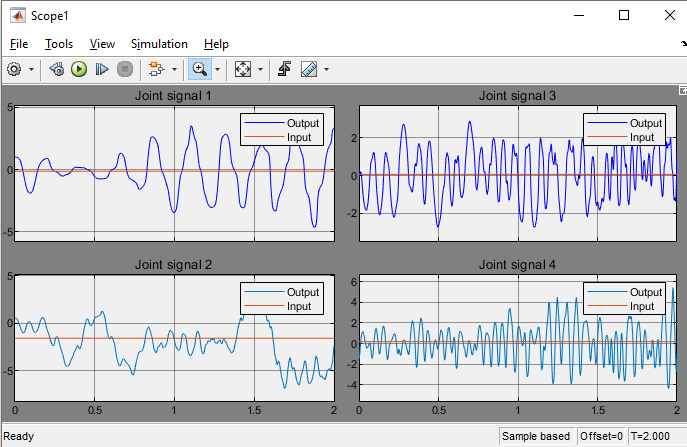
*Bảng …: Cấu trúc cơ bản của khối PID Controller*

Khối PID Controller trong Simulink thực hiện bộ điều khiển PID với tham số miền thời gian liên tục. Đầu ra của khối là tổng, tích phân và đạo hàm của tín hiệu đầu vào. Với các trọng số là các tham số khếch đại tỉ lệ, tích phân và vi phân. Biểu thức của giải thuật PID trong khối PID controller:

***u(t)* = *P* + *I* + *D***

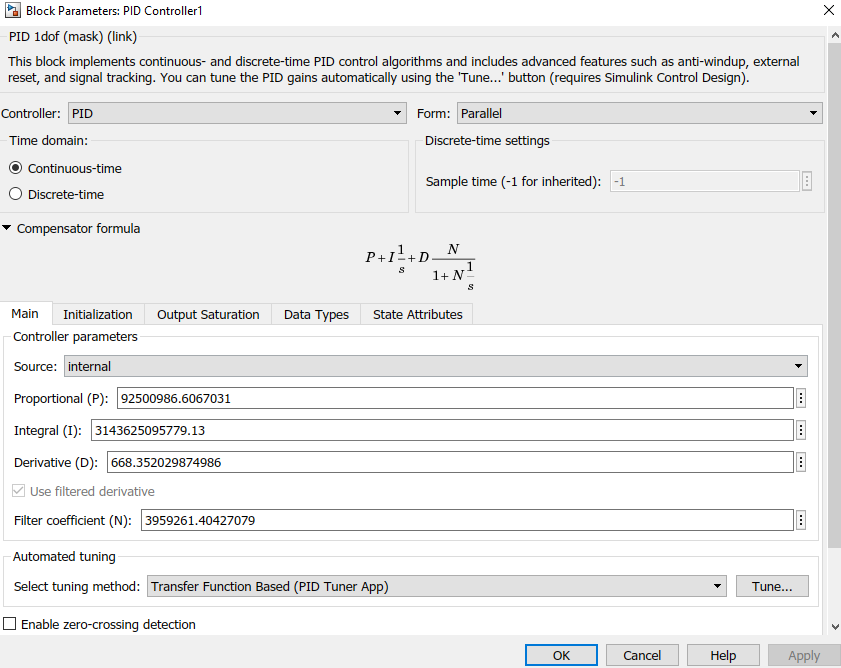
Trong đó:

* P, I, D lẫn lượt là các khâu tỉ lệ, tích phân, vi phân
* s là thời gian.
* N là hệ số lọc phái sinh.

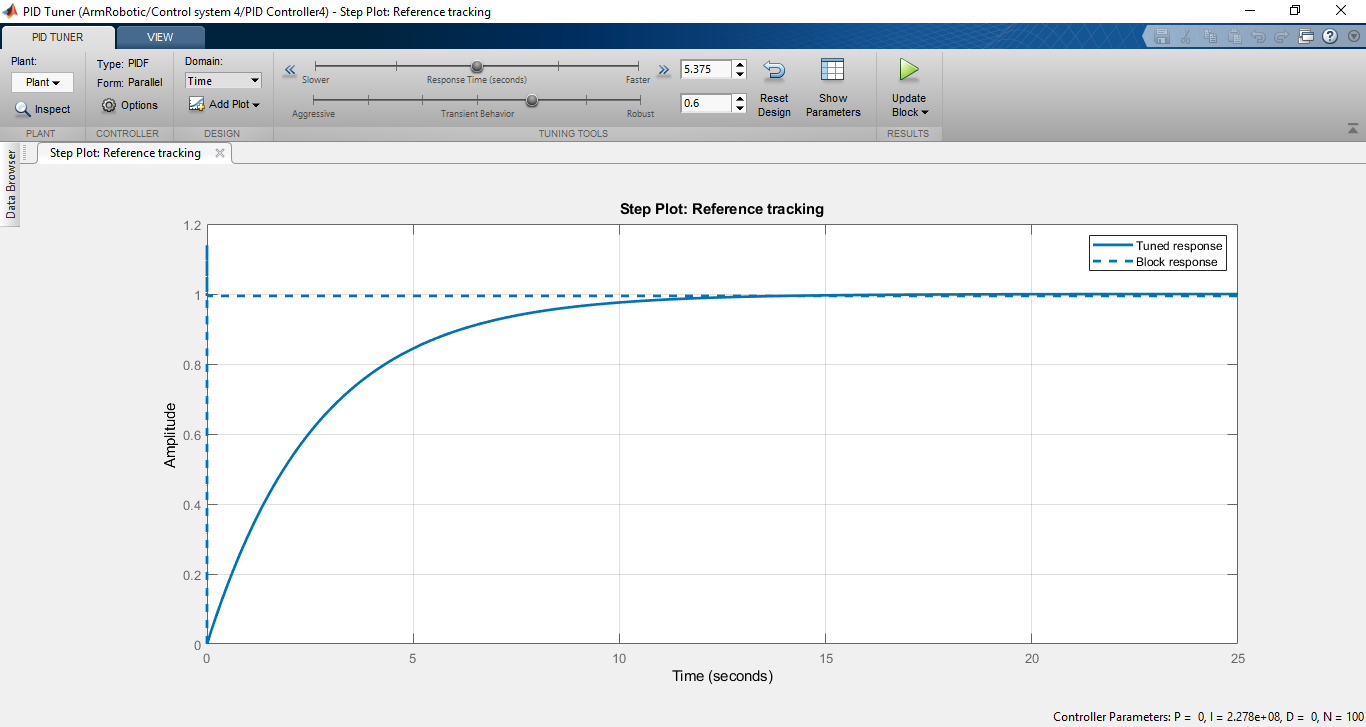


*Hình …: Đồ thị sóng đầu vào và đầu ra của các khớp trong mô phỏng khi chưa sử dụng thuật toán PID để điều khiển*

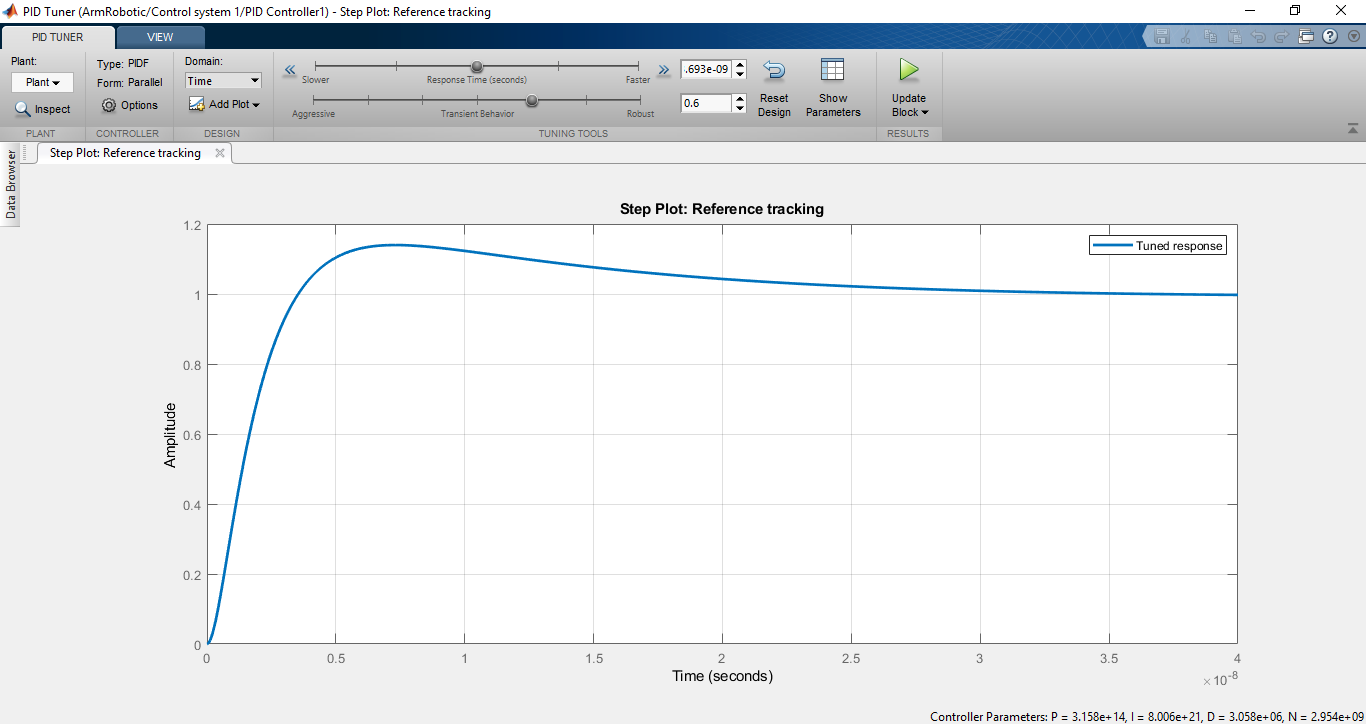
Để tính toán giá trị P, I, D, N ta sử dụng chức năng Tune được tính hợp trong khối “PID Controller” để tự động tính toán



*Hình …: Các giá trị P, I, D, N được khối “PID Controller” tinh toán*



*Hình …: Chức năng Tune trong khối PID Controller ban đầu*

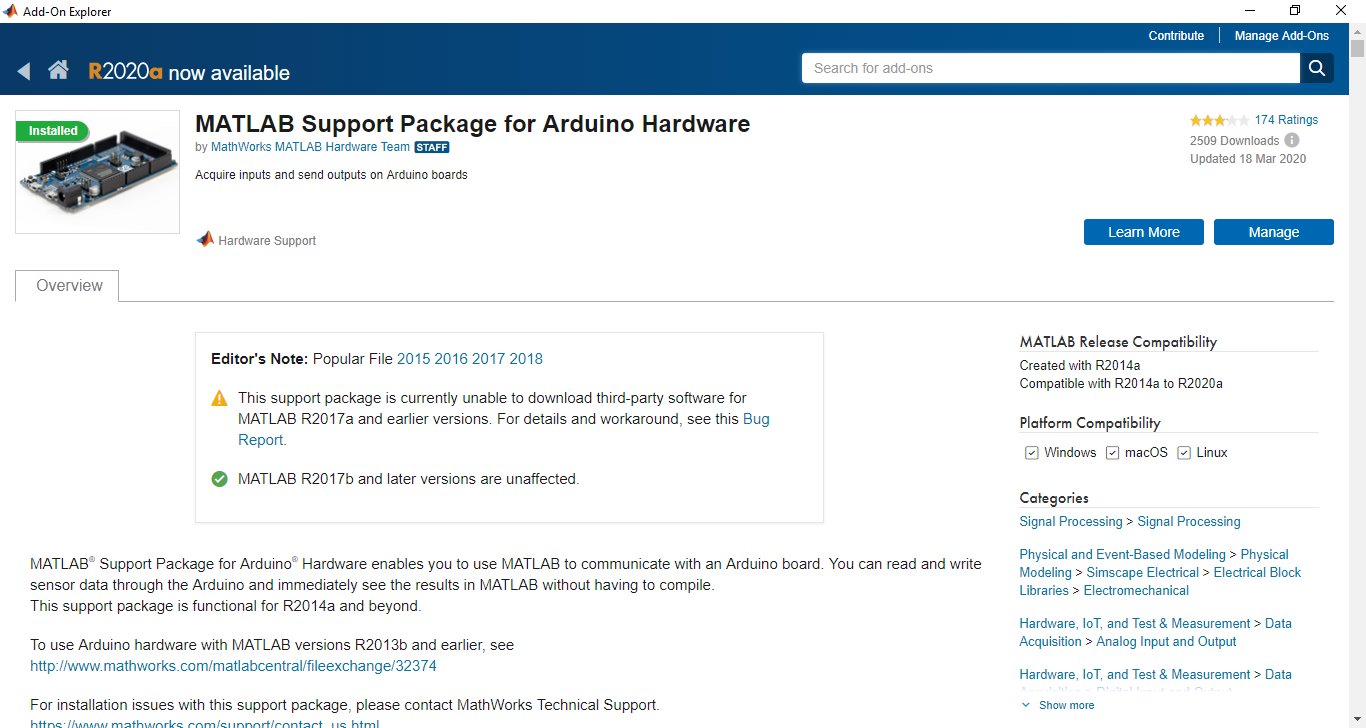


*Hình …: Chức năng Tune trong khối PID Controller khi đã sử dụng thuật toán PID*

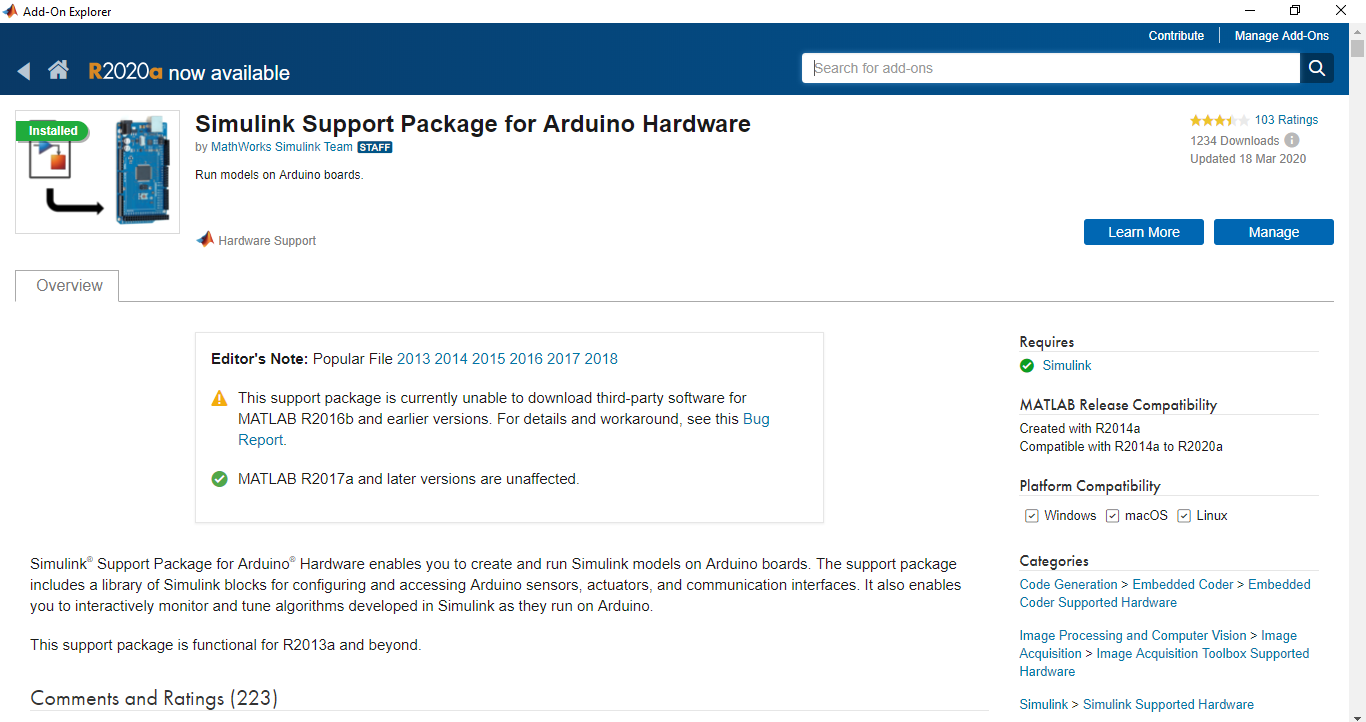
* + 1. **Điều khiển mô hình cánh tay robot.**
  1. **Giới thiệu thư viện giao tiếp giữa Simulink và Arduino Mega 2560**

Để giao tiếp giữa Matlab Simulink và Arduino Mega 2560 thì cần phải cài đặt 2 thư viện mà Matlab hỗ trợ gồm:

* Matlab Support Package for Arduino Hardware: Thư viện này kết nối giữa Matlab và phần cứng bên ngoài(Kit Arduino).
* Simulink Support Package for Arduino Hardware: Thư viện này hỗ trợ kết nỗi giữa Simulink và phần cứn bên ngoài(Kit Arduino).

****

*Hình …: Thư viện Matlab Support Package for Arduino Hardware*

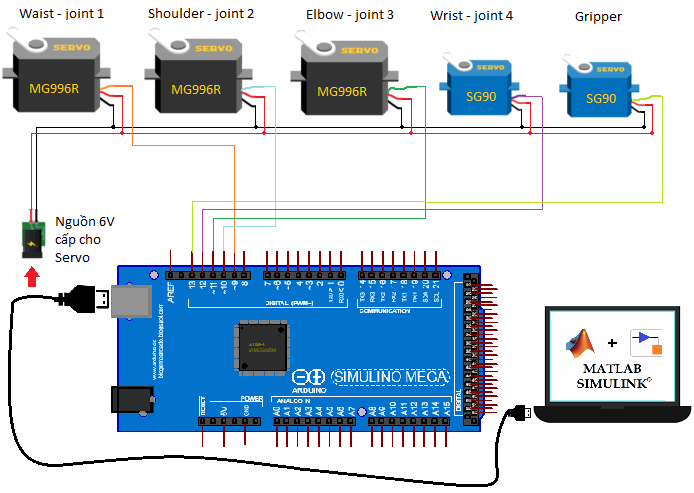
****

*Hình …: Thư viện Simulink Support Package for Arduino Hardware*

2 thư viện trên hỗ trợ với các dùng Arduino như Arduino Uno R3, Arduino Mega 2560,… Sau khí cài đặt các thư viện trên ta có thể giao tiếp chúng giữa Simulink và Arduino thongo qua các khối lệnh.

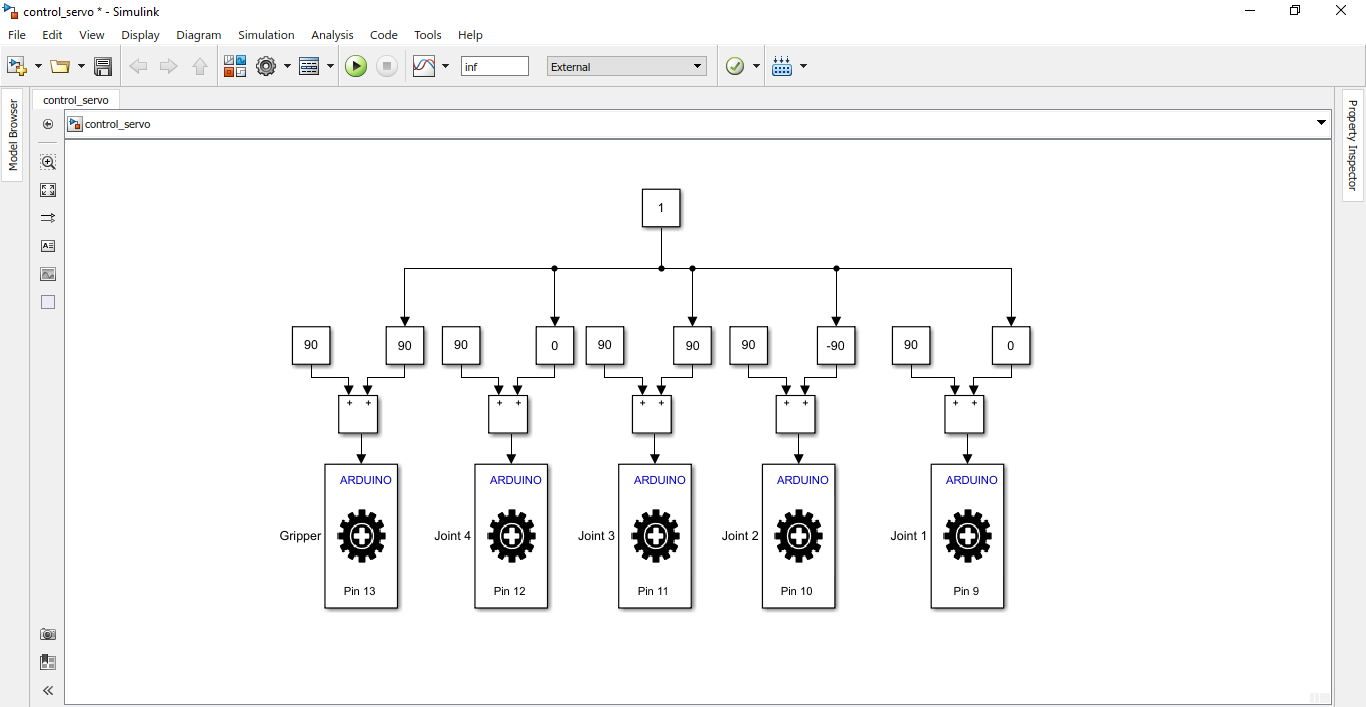
* 1. **Kết nối các động cơ Servo với Arduino**

Arduino Mega 2560 có 14 chân xuất ra xung PWM để điều khiển động cơ từ servo là các chân từ 0 – 14. Vậy nên, để điều khiển 5 động cơ servo cần sử dụng 5 chân PWM từ 9 - 13 trong tổng số 14 chân của Arduino để điều khiển động cơ servo.

****

*Hình …: Sơ đồ kết nối các động cơ Servo với Arduino*

* 1. **Sơ đồ khối Siulink điều khiển động cơ servo**

****

*Hình …: Sơ đồ kết nối các động cơ Servo với Arduino trong Simulink*

Sơ đồ trên sử dụng khối 3 loại khối với chức năng cụ thể của từng khối được mô tả như bảng dưới đây

|  |  |
| --- | --- |
| **constand.PNG** | Khối hằng số đầu vào |
|  | Khối thay đổi hệ số khuếch đại vô hướng |
| **servo wirte.PNG** | Khối đặt góc quay của động cơ servo |

*Bảng …: Các khối mô phỏng được sử dụng.*

Khối “Slider Gain” thay đổi hệ số khuếch đại với giá trị góc mong muốn từ 0 - 180 sau đó truyền dữ liệu vào khối “Standard Servo Write”. Khối “Standard Servo Write” sẽ nhận giá trị đầu vào là các góc quay từ 0 - 180 và tự động truyền xung PWM với độ rộng xung tương ứng với các góc sang các động cơ servo.

* + 1. **Điều khiển đồng thời mô phỏng và mô hình sử dụng GUI Matlab**

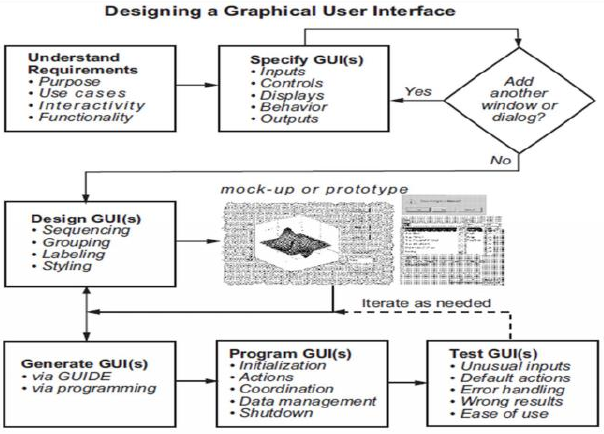
1. **Giới thiệu sơ lược về GUI Matlab.**

* ***Tổng quan về GUI***

GUI được viết tắt của từ “**Graphical User Interface**” dịch theo tiếng việt nôm la là **Giao diện người dùng đồ họa**. Là ta sẽ tiến hành sử dụng chuột để thao tác các thanh công cụ đã được con người lập trình sẵn và tích hợp vào phần mền Matlab. Nó cho phép bạn tương tác giao diện chương trình từ đó bạn nhìn được một cách khách quan nhất và bắt đầu viết code để tương tác giao diện.

Cũng giống như C# trên Window Form GUI Cũng thao tác kéo thả button, textbox, layout,… giúp người dùng nhanh chóng tạo ra giao diện, giảm thời gian ngồi lập trình.

Trong Matlab thì GUI hỗ trợ khá là đầy đủ các chương trình để thực hiện như là tính toán với phép toán LOGIC, lập trình không gian 2D, 3D, đọc dữ liệu từ Excel, sử lý hình ảnh, đọc hiển thị dữ liệu, liên kết đa phương tiện. Giao  
tiếp với người dùng thông qua hình ảnh, các nút nhấn thực thi ...

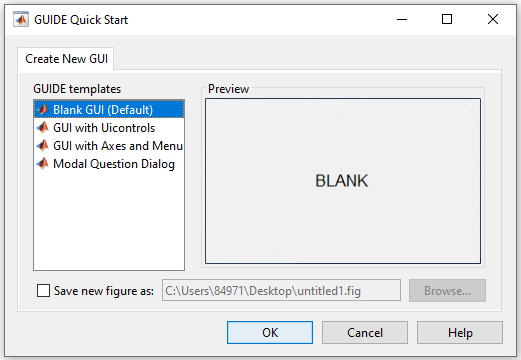


*Hình …: Sơ đồ trình tự thực hiện với GUI*

* ***Khởi tạo GUI***

Thực hiện khởi động Matlab đến GUI theo các bước:

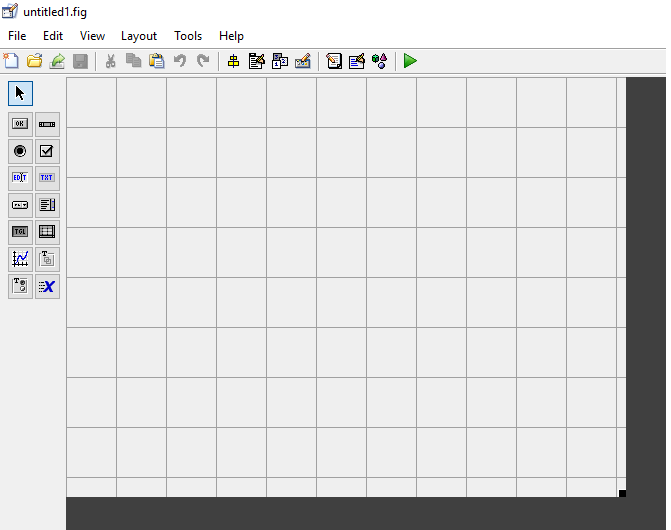
* Khởi động Matlab từ biểu tượng Matlab.
* Trong cửa sổ Comment Windows gõ lệnh “guide” và enter ta được giao diện như hình



*Hình …: Tạo mới GUI*

* Ta có các lựa chọn sau :
* Blank GUI (Default): Hộp thoại GUI trống không có một điều khiển uicontrol nào cả.
* GUI with Uicontrols: Hộp thoại GUI với một vài uicontrol như button, … Chương trình có thể chạy ngay.
* GUI with Axes and Menu: Hộp thoại GUI với một uicontrol axes và button. Các menu để hiển thị đồ thị.
* Modal Question Dialog: Hộp thoại đặt câu hỏi Yes, No.
* Open Exiting GUI: Để mở 1 file có sẵn.

Nhấp Blank GUI (Default) chọn OK để tạo một giao diện bắt đầu với giao

****

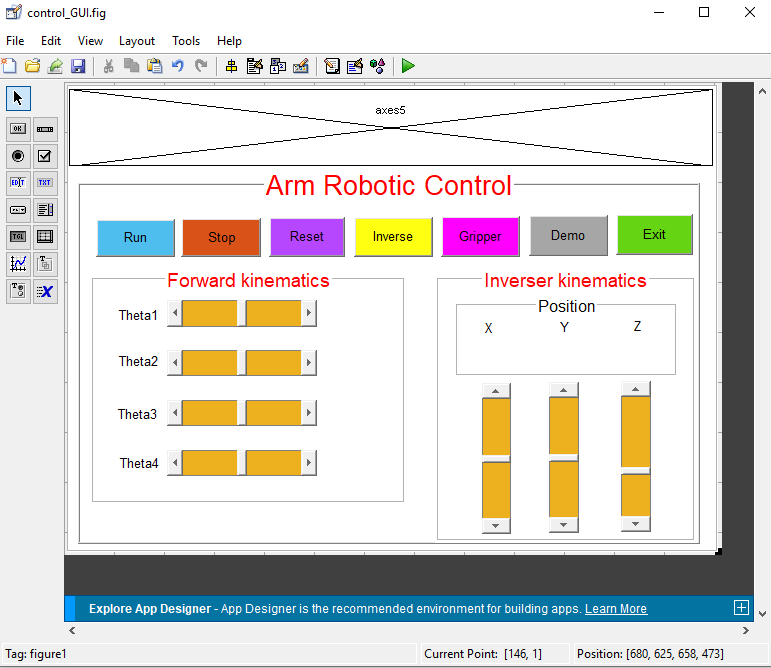
*Hình …: Giao diện GUI sau khi tạo mới*

1. **Thiết kế giao diện GUI điều khiển cánh tay robot**

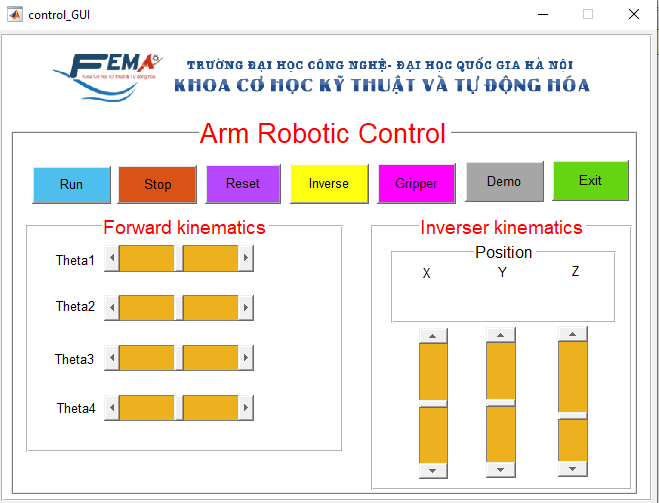
* ***Các tool cơ bản của GUI***
* **Push Button**: Khi nhấn chuột vào thì nút này sẽ xử lý một công việc nào đó mà người dùng mong muốn muốn.
* **Static Tex**t: Dùng để hiện thì thông tin cho một đối tượng nào đó hoặc có thể xuất dữ liệu từ trương trình ra đây.
* **Slider:** Thay đổi giá trị số mà người dùng mong muốn
* **Axes:** Đồ thị
* **Radio Button:** Lựa chọn chức năng khi được nhấn vào
* **Table:** Bảng
* ***Thiết kế giao diện GUI điều khiển cánh tay robot***

Để thiết kế giao diện cho bảng điều khiển cách tay robot nhóm đã sử dụng các tool như: Push Button, Slider, Static Text,…

Để điều khiển cánh tay robot cần các nút bấm chức năng như run, stop, reset, inverse, exit,… Bên cạnh đó là các khối slider để thay đổi các giá trị của góc quay hay tọa độ x, y, z.



*Hình …: Bảng điều khiển sau khi thiết kế*

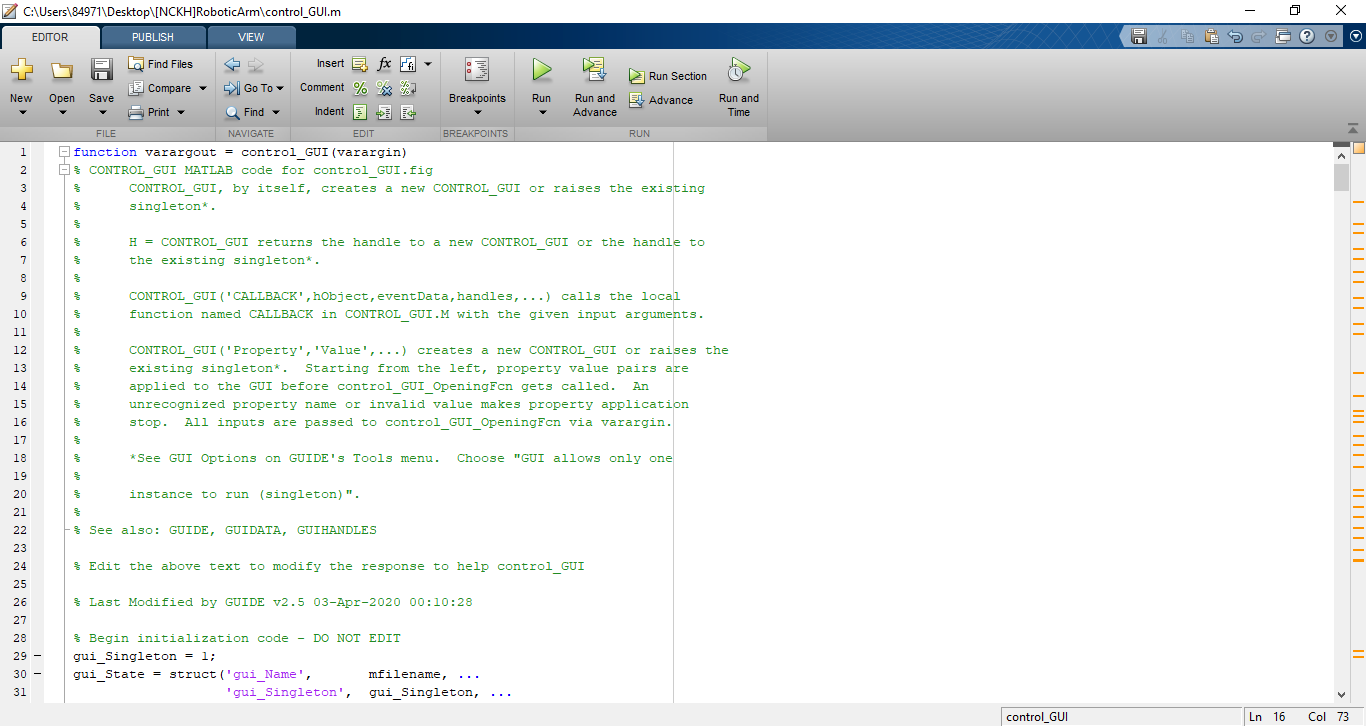
**

*Hình …: Bảng điều khiển hoàn chỉnh*

1. **Lập trình điều khiển song song mô hình và mô phỏng trên giao diện người dùng đồ họa (GUI) được tích hợp trong Matlab**

* ***Giới thiệu về lập trình GUI***
  + - Có 2 phương pháp để lập trình GUI
* *Cách 1*: là cách đơn giản nhất là sử dụng công cụ có sẵn trong GUI Matlab để lập trình. Ưu điểm của cách này là dễ thực hiện và các hàm Function được GUI tạo sẵn.
* *Cách 2*: Các này được lập trình từ siêu tệp Mfile bằng các hàm function do người lập trình tự viết, nó có ưu điểm là tùy biên cao. Tuy nhiên, cách này khó hơn và đòi hỏi người lập trình phải có hiểu biết sâu vè GUI.
* Các bước để lập trình GUI
* *Bước 1*: Xác định mục đích của chương trình.
* *Bước 2*: Xác định các bước để mô phỏng giao diện người dùng cho hợp lý và chính xác.
* *Bước 3*: Viết chương trình và thực thi.

Sau khi thiết kế xong giao diện trong GUI, Matlab sẽ tạo 1 m-file(chương trình soạn thảo văn bản) để người dùng có thể lập trình cho các đối tượng thực thi nhiệm vụ người dùng mong muốn.



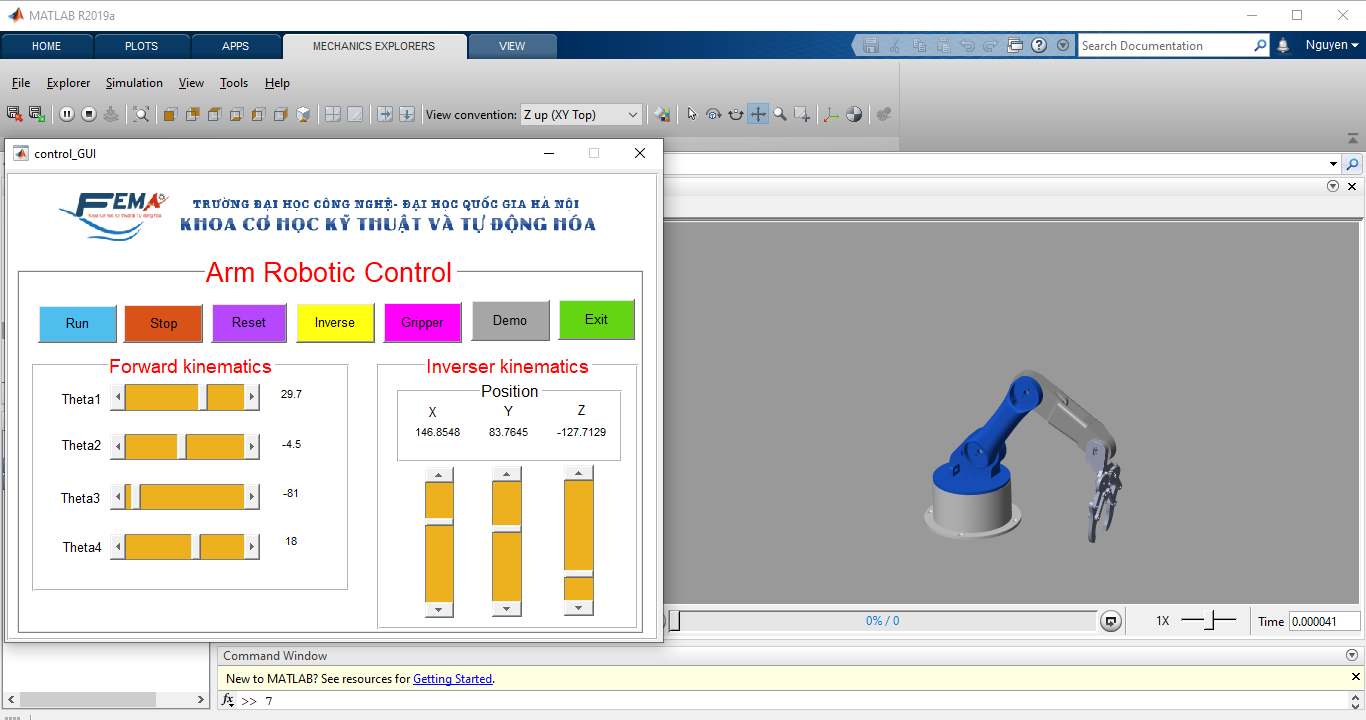
*Hình …: m-file sau khi được tạo*

Hầu hết GUI chỉ thực hiện (trả lời) lệnh người dùng thông qua các tác động của người dùng lên giao diện, người sử dụng không cần biết cấu trúc chương trình vẫn có thể thực hiện được. GUI được thực hiện thông qua các hàm CALLBACK, khi người dùng tác động lên giao diện bằng các cách khác nhau, hàm CALLBACK sẽ được gọi để thực thi.

* ***Lập trình bảng điều khiển cánh tay robot***

Mã lập trình sẽ được trình bày chi tiết trong phần phụ lục.

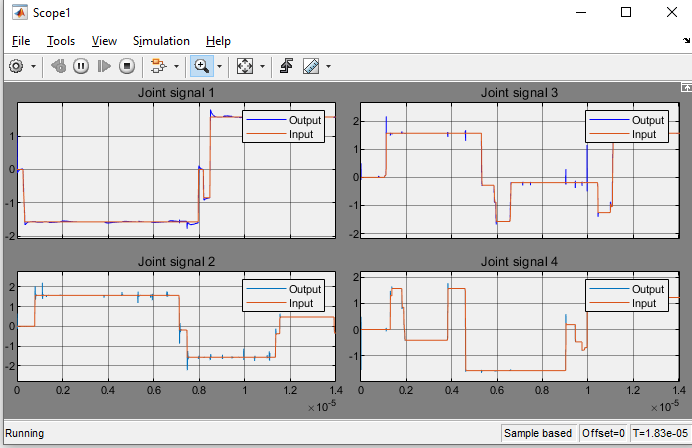
1. **Kết quả**

****

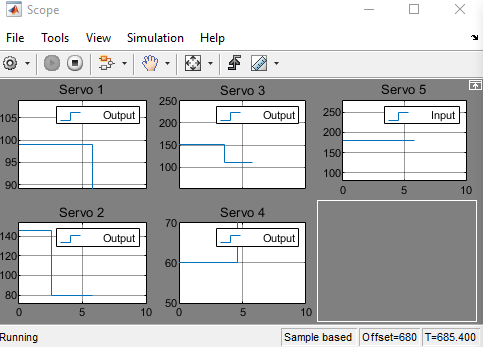
*Hình …: Mô hình robot trên máy tính*

*Với: 1 = 29.7; 2 = -4.5; 3 = -81; 4 = 18*

*Tọa độ điểm cuối tương ứng là x = 146.8548mm; y = 83.7645; z = -127.7129*

****

*Hình …: Sóng tại các khớp xoay trong mô hình*

****

*Hình …: Xung của các động cơ servo trên mô hình*

1. **KẾT LUẬN VÀ HƯỚNG PHÁT TRIỀN.** 
   * + 1. **Kết luận**

Qua quá trình nghiên cứu, thiết kế, xây dựng mô hình, mô phỏng và điều khiển đề tài nghiên cứu của chúng em đã thực hiện được một số công việc sau:

* Nghiên cứu được tổng quan lý thuyết về robot, các khái niệm, các cơ cấu, nguyên lý hoạt động và cụ thể hơn là cánh tay robot 4 bậc tự do. Từ cơ sở trên để thiết kế ra cánh tay robot 4 bậc tự do. Vận dụng các kiến thức về cơ khí, chế tạo máy, in 3D để thiết kế kế cánh tay robot 4 bậc tự do từ phần mềm thiết kế Autodesk Inventor.
* Nghiên cứu phương pháp giải bài toán động học của tay máy. Từ đó làm cơ sở để điều khiển mô hình và mô phỏng các chuyển động của cánh tay robot trên PC/laptop bằng công cụ mô hình hóa, mô phỏng và phân tích hệ thống Matlab/Simulink.
* Xây dựng mô hình hoàn chỉnh cánh tay robot với các chuyển động thông qua chuyển động quay của động cơ servo và được điều khiển bằng Arduino thông qua công cụ Matlab/Simulink.
* Xây dựng giao diện trên PC/laptop từ giao diện người dùng đồ họa (GUI) được tích hợp trên Matlab để điều khiển song song đồng thời cả mô hình thực tế và mô phỏng các chuyển động của cánh tay robot.
  + - 1. **Hướng phát triển**