

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ



THIẾT KẾ, CHẾ TẠO
VÀ KHẢO SÁT ROBOT SCARA

BÁO CÁO GIỮA KỲ

Môn: Các cơ cấu chấp hành trong robot

HÀ NỘI - 2022

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

THIẾT KẾ, CHẾ TẠO
VÀ KHẢO SÁT ROBOT SCARA

BÁO CÁO GIỮA KỲ

Môn: Các cơ cấu chấp hành trong robot

Giảng viên hướng dẫn: TS. Dương Xuân Biên

Thành viên:
Lê Phần Nam
Hoàng Văn Kiên
Đỗ Đức Mạnh

HÀ NỘI - 2022

LỜI NÓI ĐẦU

Theo liên đoàn Robot Quốc tế (IFR), năm 2015 có 60 robot/ 10.000 lao động, con số đó đã tăng lên 77 trong năm 2017 và lớn hơn rất nhiều trong năm 2022. Tại Việt Nam cũng không ngoại lệ, ngành robot đang phát triển theo xu hướng thế giới, do đó nhu cầu giảng dạy về robot tăng nhanh chóng. Tuy nhiên, các trường phổ thông, trường Đại học chưa có nhiều bộ thực hành do chi phí cao hoặc khó tiếp cận với học sinh, sinh viên vì động học đòi hỏi ở mức cao như robot Delta hay cánh tay robot nhiều trục tự do. Do đó chúng tôi đã quyết định nghiên cứu và chế tạo một mẫu robot SCARA quy mô nhỏ, giá thành thấp.

Robot của chúng tôi có khối lượng tổng thể là 1kg, không gian hoạt động có bán kính 25cm và được thiết kế có khả năng di chuyển theo quỹ đạo cho trước như vẽ hình, gấp vật có tải trọng dưới 100g. Với nhu cầu phục vụ cho giáo dục, chúng tôi đã xây dựng vỏ robot bằng công nghệ in 3D, sử dụng bo mạch điều khiển dễ tiếp cận là Arduino kết hợp với động cơ bước, một loại động cơ phổ biến với độ chính xác cao.

Qua nghiên cứu robot SCARA, chúng tôi đã được tiếp cận với công việc thiết kế cơ khí, kiểm bèn một tay máy robot và có thể làm chủ động cơ bước. Độ chính xác về cơ khí cũng như điều khiển được thể hiện qua nhiều lần khảo sát lặp lại. Đặc biệt chúng tôi đã rút ra được nhiều kinh nghiệm để có thể nghiên cứu sâu hơn về robot SCARA, thậm chí phát triển các mô hình robot phức tạp hơn trong thời gian tiếp theo.

LỜI CẢM ƠN

Để có được kết quả này, chúng tôi xin gửi lời cảm ơn chân thành đến thầy TS. Dương Xuân Biên, người đã tận tình chỉ bảo, hướng dẫn tìm hiểu và định hướng trong suốt quá trình nghiên cứu và làm việc.

Do điều kiện chi phí và thời gian nghiên cứu hạn chế, khó tránh khỏi những sai sót. Rất mong nhận được sự thông cảm và đóng góp ý kiến từ người đọc.

Chúng tôi xin chân thành cảm ơn!

MỤC LỤC

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG BIỂU

CHƯƠNG 1. THIẾT KẾ ROBOT	1
1.1. Mục tiêu nhiệm vụ của robot	1
1.2. Mô tả tiến trình hoạt động của robot.....	1
1.3. Cấu tạo của robot.....	1
1.4. Chức năng của các bộ phận trên robot.....	3
1.4.1. Các cơ cấu truyền động	3
1.4.2. Homing.....	4
CHƯƠNG 2. CÁC CƠ CẤU CHẤP HÀNH TRÊN ROBOT	5
2.1. Động cơ bước.....	5
2.1.1. Chức năng	5
2.1.2. Cấu tạo cơ bản.....	5
2.1.3. Nguyên lý hoạt động	6
2.1.4. Phương thức điều khiển	7
2.2. Động cơ servo	9
2.2.1. Chức năng	9
2.2.2. Cấu tạo cơ bản.....	9
2.2.3. Nguyên lý hoạt động	10
2.2.4. Phương thức điều khiển	10
2.3. Công tắc hành trình.....	11
2.3.1. Chức năng	11
2.3.2. Cấu tạo cơ bản.....	11
2.3.3. Nguyên lý hoạt động	12
2.3.4. Phương thức điều khiển	12
2.4. Biến trở	13
2.4.1. Chức năng	13

2.4.2. Cấu tạo cơ bản.....	13
2.4.3. Nguyên lý hoạt động	14
2.4.4. Phương thức điều khiển	14
CHƯƠNG 3. CHẾ TẠO VÀ TÍCH HỢP HỆ THỐNG TRÊN ROBOT	16
3.1. Chế tạo cơ khí.....	16
3.1.1. Chế tạo khung cơ khí	16
3.1.2. Các cơ cấu truyền động.....	16
3.1.3. Khảo sát thiết kế.....	18
3.2. Chế tạo mạch.....	19
3.2.1. Mạch shield V4 CNC mini.....	19
3.2.2. Drive a4988	21
3.3. Codes chương trình điều khiển.....	23
3.4. Thực nghiệm, đo đạc đánh giá sai số	25
3.4.1. Homming.....	25
3.4.2. Chuyển động điểm đến điểm	26
3.4.3. Đánh giá sai số chuyển động điểm đến điểm.....	26
3.4.4. Chuyển động theo một quỹ đạo cho trước	28
3.5. Kết quả và đánh giá	29
3.5.1. Kết quả	29
3.5.2. Định hướng phát triển	30
TÀI LIỆU THAM KHẢO	31

DANH MỤC HÌNH ẢNH

Hình 1.1. Quy trình hoạt động của Robot.....	1
Hình 1.2. Cấu tạo của Robot	2
Hình 1.3. Cơ cấu bánh răng - dây đai	3
Hình 1.4. Cơ cấu bánh răng - thanh răng	4
Hình 1.5. Vị trí công tắc hành trình.....	4
Hình 2.1. Động cơ bước.....	5
Hình 2.2. Cấu tạo động cơ bước.....	6
Hình 2.3. Nguyên lý hoạt động của động cơ bước	7
Hình 2.4. Sơ đồ mạch điều khiển động cơ bước	8
Hình 2.5. Động cơ servo.....	9
Hình 2.6. Sơ đồ mạch điều khiển động cơ servo	10
Hình 2.7. Cấu tạo công tắc hành trình	12
Hình 2.8. Cấu tạo biến trở.....	14
Hình 2.9. Sơ đồ mạch điều khiển động cơ servo bằng biến trở.....	15
Hình 3.1. Các thông số của Robot.....	16
Hình 3.2. Cơ cấu truyền động bánh răng - dây đai của Robot	17
Hình 3.3. Cơ cấu truyền động bánh răng - thanh răng của Robot.....	17
Hình 3.4. Khảo sát độ bền robot.....	18
Hình 3.5. Sơ đồ chân của CNC Shield V4	19
Hình 3.6. Ghép nối Arduino, A4988 và CNC Shield V4.....	20
Hình 3.7. Chân của A4988.....	21
Hình 3.8. Cách nối chân driver a4988	22

<i>Hình 3.9. Giao tiếp Serial port của Matlab và Arduino</i>	<i>23</i>
<i>Hình 3.10. Kết nối giữa Arduino và động cơ.....</i>	<i>24</i>
<i>Hình 3.11. Lưu đồ thuật toán quá trình homing của robot</i>	<i>25</i>
<i>Hình 3.12. Sơ đồ hoạt động chi tiết của robot.....</i>	<i>26</i>
<i>Hình 3.13. Đồng hồ so INSIZE 2308-5A và cách bố trí đo sai số.....</i>	<i>27</i>
<i>Hình 3.14. Sai số khi robot di chuyển từ điểm (123,6.15,0) đến điểm (0,0,0).....</i>	<i>28</i>
<i>Hình 3.15. Robot thực hiện vẽ một đường thẳng theo quỹ đạo cho trước.....</i>	<i>29</i>

DANH MỤC BẢNG BIỂU

<i>Bảng 3.1. Các thông số vật liệu dùng cho thiết kế robot.....</i>	<i>18</i>
<i>Bảng 3.2. Chân kết nối động cơ của CNC Shield V4</i>	<i>20</i>
<i>Bảng 3.3. Các chế độ hoạt động của driver</i>	<i>22</i>
<i>Bảng 3.4. Sai số giữa các lần đo</i>	<i>27</i>

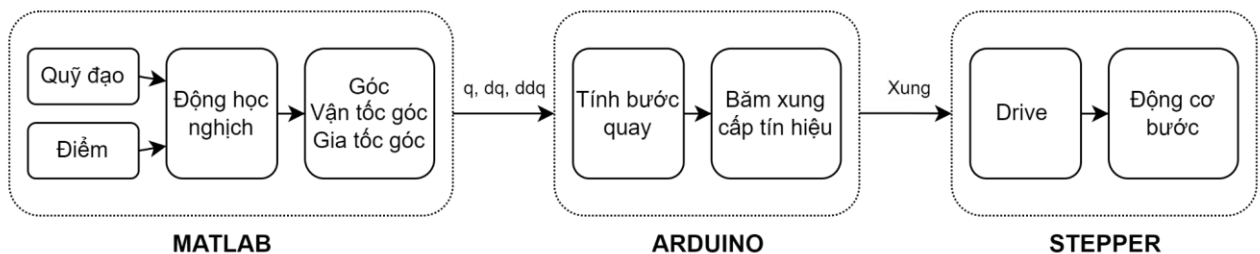
CHƯƠNG 1. THIẾT KẾ ROBOT

1.1. Mục tiêu nhiệm vụ của robot

Với mục tiêu hướng đến giáo dục nhưng vẫn giữ các đặc điểm tương tự các sản phẩm công nghiệp. vì vậy, sự sáng tạo trong thiết kế là sự cần thiết. Báo cáo cũng đưa ra nghiên cứu của chúng tôi về kích thước của robot và sự tối ưu trong phần cứng.

Robot SCARA của chúng tôi sẽ được xây dựng với quy mô nhỏ, giá thành thấp, sử dụng bo mạch điều khiển đơn giản có 2 chức năng chính để tiếp cận với học sinh, sinh viên là gấp đặt vật và vẽ hình theo quỹ đạo cho trước.

1.2. Mô tả tiến trình hoạt động của robot



Hình 1.1. Quy trình hoạt động của Robot

Tiến trình hoạt động của robot gồm 3 giai đoạn chính gồm: Tính toán trên matlab, xử lý trên arduino và điều khiển động cơ bước.

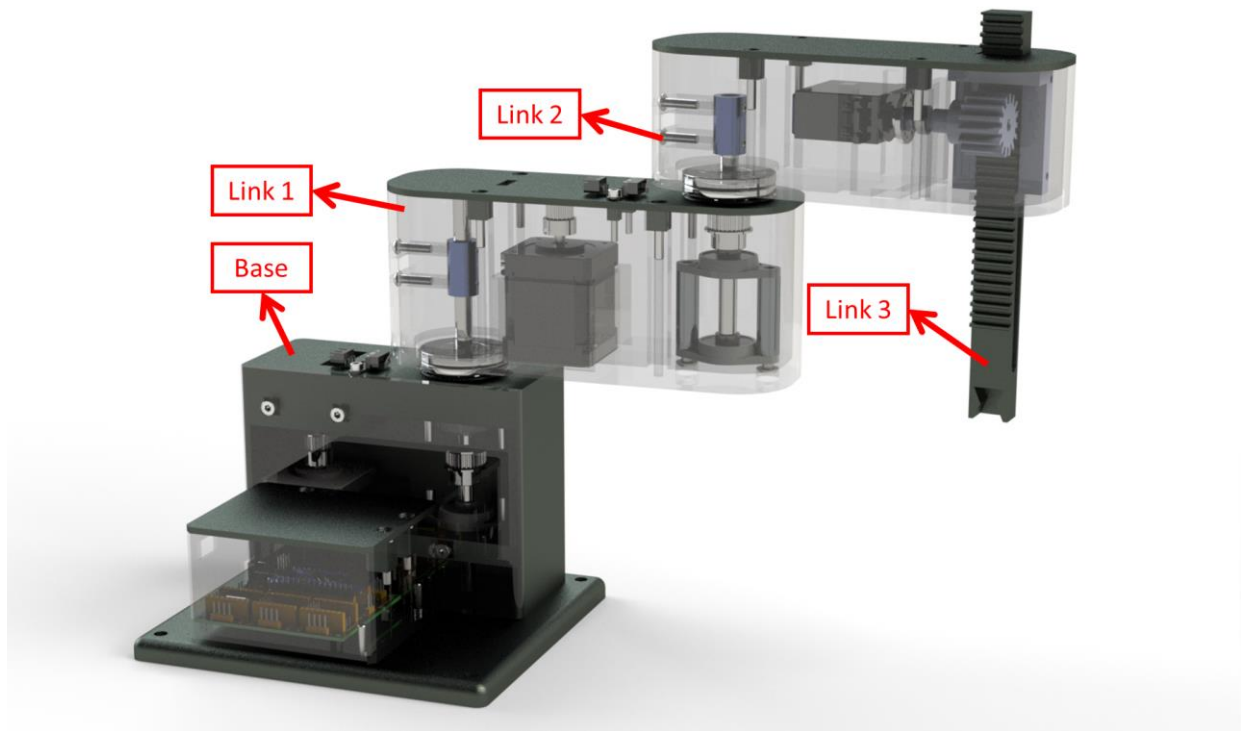
Người dùng thực hiện chọn chế độ trên Matlab User Interface, từ đó sử dụng động học nghịch tính được chuỗi góc cần di chuyển, vận tốc trên từng thời điểm và gia tốc tương ứng. Lúc này, matlab sẽ giao tiếp với arduino, gửi tới arduino các đại lượng góc, vận tốc, gia tốc góc tương ứng để arduino băm xung điều khiển động cơ.

Các cơ cấu chấp hành sẽ giúp truyền động từ động cơ tới các link, kết hợp sự chuyển động của các link tạo nên vị trí của điểm thao tác E di chuyển tới vị trí mong muốn

1.3. Cấu tạo của robot

Khung robot được thiết kế dựa trên nhu cầu thực tế và đòi hỏi sự thẩm mỹ. Chúng tôi thiết kế robot thành một khối duy nhất, bao gồm động cơ, các cơ cấu truyền động, bo mạch.

Robot SCARA về cơ bản có 2 khớp quay bản lề, tuy nhiên khi chế tạo có thể thêm khớp tịnh tiến và khớp quay quanh trục tùy mục đích. Với mục tiêu ứng dụng cho vẽ và gấp đặt vật, chúng tôi sẽ thêm vào robot trục thứ 3 là trục tịnh tiến.



Hình 1.2. Cấu tạo của Robot

Hình trên thể hiện các bộ phận của robot.

Robot gồm 3 khung chính: Base, Link 1, Link 2 và một trục tịnh tiến Link 3.

a. Đế Base

Đế base được thiết kế to, rộng, có hộp đựng arduino và dây bên trong. Đảm bảo đủ khối lượng để giữ thăng bằng cho cả robot.

Bên trong đế có gắn động cơ để điều khiển link 1, nhờ cơ cấu truyền động dây đai - bánh răng. Chúng tôi sẽ nói rõ hơn trong mục 1.4.

b. Link 1

Link 1 có bộ phận liên kết với động cơ ở đế Base và cũng có một động cơ để điều khiển link 3. Việc chia động cơ sẽ giúp trọng tâm của cả robot gần về phía trọng tâm đế Base hơn, giúp robot đứng vững khi thao tác.

c. Link 2

Link 2 có vị trí gắn servo, gắn với bánh răng thực hiện nhiệm vụ tịnh tiến link 3.

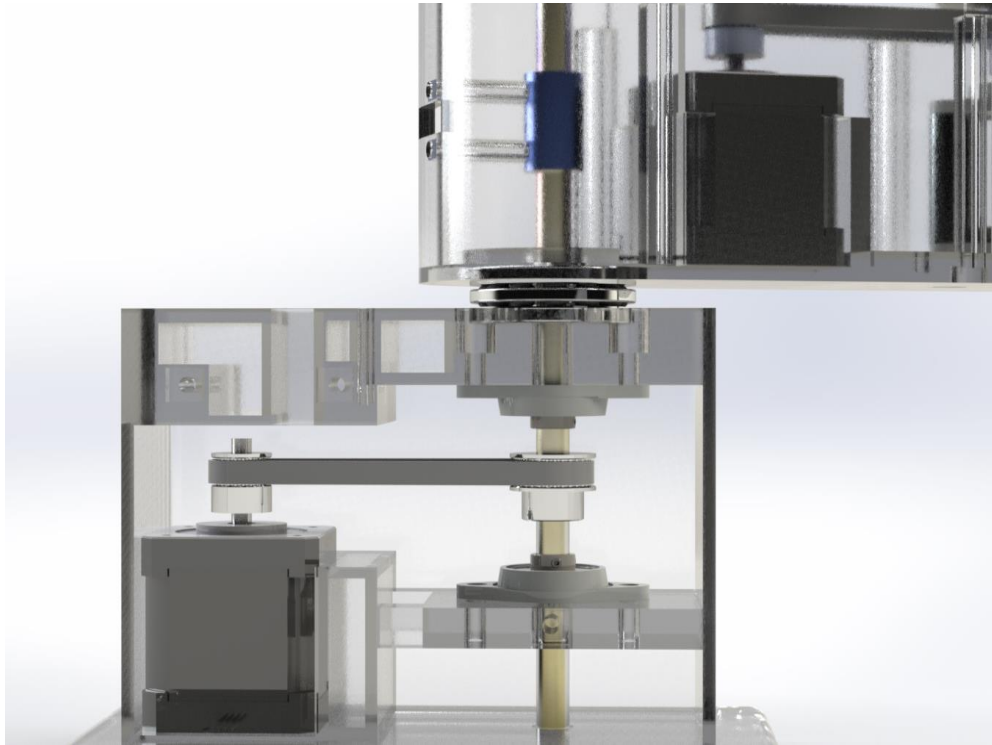
d. Link 3

Là trục tịnh tiến, ở cuối link chúng ta có một phần ghép nối để ghép các cơ cấu thực hiện việc gấp, viết ... theo nhu cầu.

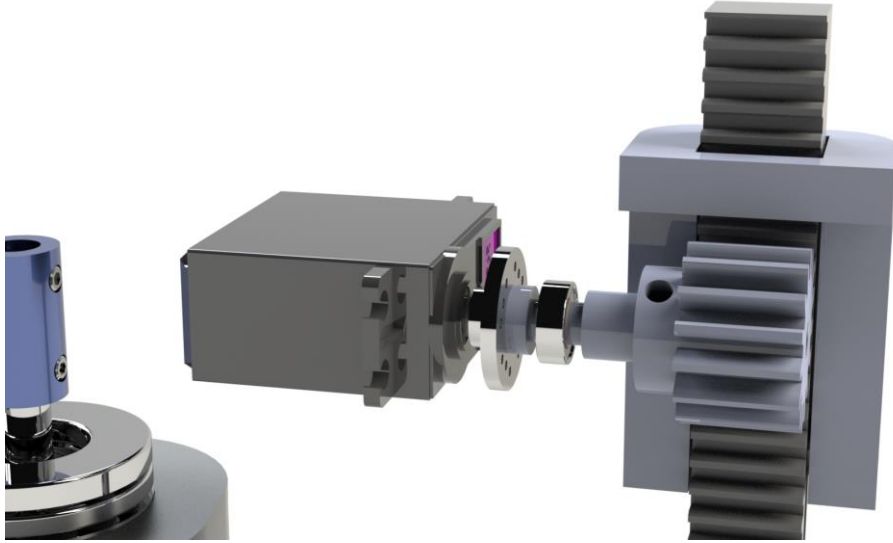
1.4. Chức năng của các bộ phận trên robot

1.4.1. Các cơ cấu truyền động

Robot sử dụng dây đai và bánh răng pulley để thực hiện việc dẫn động, với động cơ, sử dụng trục crom liên kết các link và hệ thống bánh răng - thanh răng di chuyển link 3.



Hình 1.3. Cơ cấu bánh răng - dây đai



Hình 1.4. Cơ cấu bánh răng - thanh răng

1.4.2. Homing

Hiện tại robot đang hoạt động chưa có phản hồi, do bộ phận truyền động sử dụng động cơ bước nên việc sử dụng công tắc hành trình hỗ trợ homing là cần thiết. Do đó, robot SCARA của chúng tôi được thiết kế với 2 công tắc trên thân, đảm bảo được về cả chức năng và thẩm mỹ.



Hình 1.5. Vị trí công tắc hành trình

CHƯƠNG 2. CÁC CƠ CẤU CHẤP HÀNH TRÊN ROBOT

2.1. Động cơ bước

2.1.1. Chức năng

Động cơ bước là loại động cơ có thể quy định được tần số góc quay của nó. Nếu góc bước của nó càng nhỏ thì số bước trên mỗi vòng quay của động cơ càng lớn tương ứng với độ chính xác của vị trí thu được càng lớn.

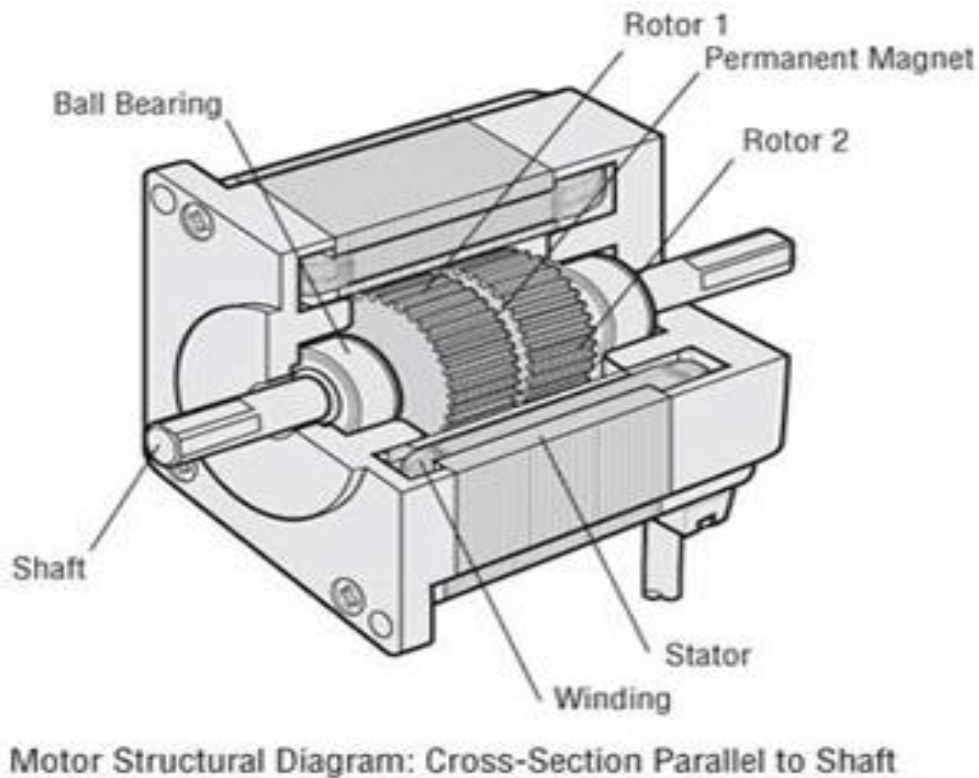
Động cơ bước được ứng dụng nhiều trong ngành tự động hoá, chúng được ứng dụng trong các thiết bị cần điều khiển chính xác: Điều khiển robot, điều khiển bắt, bám mục tiêu trong các khí tài quan sát, điều khiển lập trình trong các thiết bị gia công cắt gọt, điều khiển các cơ cấu lái phương và chiều trong máy bay. Trong điện tử tiêu dùng, động cơ bước cũng có nhiều ứng dụng khác như cho chức năng lấy nét và thu phóng camera kỹ thuật số tự động trong máy ảnh hay máy in 3D.



Hình 2.1. Động cơ bước

2.1.2. Cấu tạo cơ bản

Động cơ bước bao gồm 2 phần chính là Rotor và Stator. Trong đó, Rotor là một dãy các lá nam châm vĩnh cửu được sắp xếp chồng lên nhau. Trên các lá nam châm này lại được chia thành các cặp cực sắp xếp đối xứng nhau. Stator được cấu tạo bằng sắt từ, chúng được chia thành các rãnh nhỏ để đặt cuộn dây.



Hình 2.2. Cấu tạo động cơ bước

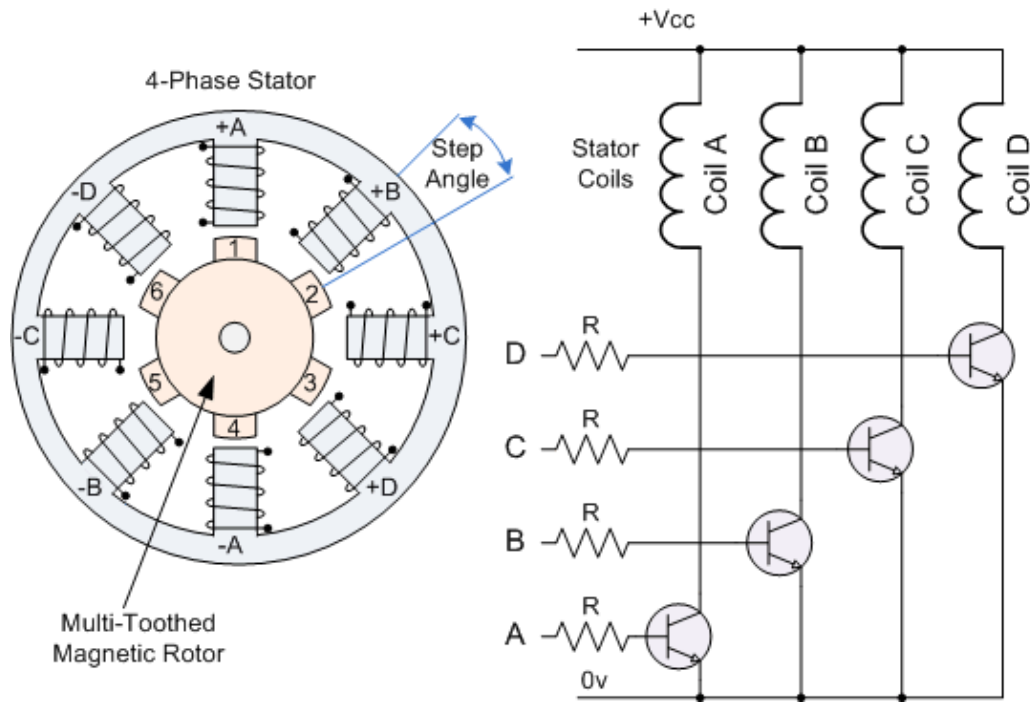
2.1.3. Nguyên lý hoạt động

Nguyên lý hoạt động của động cơ bước:

Khi cấp điện cho pha A, từng cặp cuộn dây A bố trí đối xứng nhau có cùng cực tính là nam (S) và bắc (N). Lúc này các cuộn dây hình thành các vòng từ đối xứng.

Khi cấp điện cho pha B, lúc này từ trở trong động cơ lớn, momen từ tác động lên trục rotor làm cho rotor quay theo chiều giảm từ trở. Rotor quay cho tới khi từ trở nhỏ nhất và khi momen bằng không thì trục động cơ dừng, Rotor đạt đến vị trí cân bằng mới.

Tương tự như vậy khi cấp điện cho pha C, động cơ hoạt động theo nguyên tắc trên và Rotor ở vị trí như hình. Quá trình trên lặp lại và các động cơ quay liên tục theo thứ tự pha A, B, C. Để động cơ quay ngược chiều chỉ cần cấp điện cho các pha theo thứ tự ngược lại.



Hình 2.3. Nguyên lý hoạt động của động cơ bước

2.1.4. Phương thức điều khiển

a. Đầu vào

Các phương thức điều khiển:

Điều khiển dạng sóng: là phương pháp điều khiển cấp xung điều khiển lần lượt theo thứ tự chọn từng cuộn dây pha.

Điều khiển đủ bước: là phương pháp điều khiển cấp xung đồng thời cho hai cuộn dây pha kế tiếp nhau.

Điều khiển nửa bước: là phương pháp điều khiển kết hợp cả hai phương pháp điều khiển dạng sóng và điều khiển đủ bước.

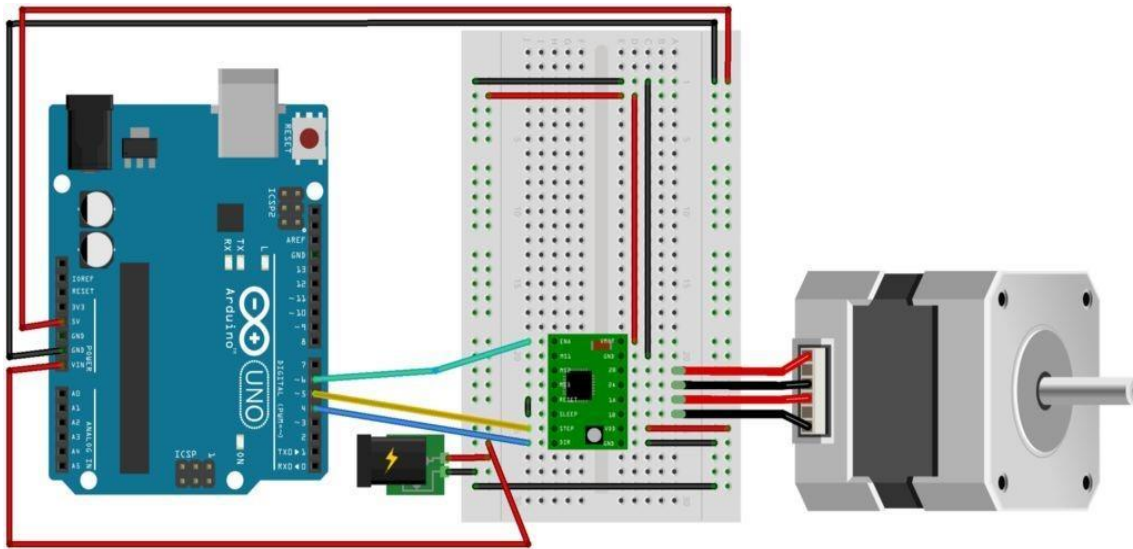
Điều khiển vi bước: là phương pháp mới được áp dụng trong việc điều khiển động cơ bước cho phép động cơ bước dừng và định vị tại vị trí nửa bước giữa hai bước đủ.

b. Đầu ra

Đầu ra của điều khiển là động cơ bước sẽ quay số góc theo tín hiệu đầu vào.

c. Codes chương trình kết nối, điều khiển

Để điều khiển động cơ bước ta cần có driver động cơ, ở đây ta sẽ dùng driver A4988. Sơ đồ mạch điều khiển được thể hiện trong hình.



Hình 2.4. Sơ đồ mạch điều khiển động cơ bước

Code điều khiển động cơ bước trong Arduino và giải thích theo từng dòng:

```
#include <AccelStepper.h>;          //Thu viện điều khiển động cơ bước
AccelStepper stepper(1,5,4);        // pin 5 step, pin 4 dir
void setup()
{
    Serial.begin(9600);
    pinMode(6,OUTPUT);              // Enable
    digitalWrite(6,LOW);            // Đặt Enable xuống low để khởi động động cơ
}
void loop()
{
    if (stepper.distanceToGo() == 0) //kiểm tra thử động cơ bước có còn
    {
        // đang chạy hay không, nếu không còn chạy thì...
        delay(1000);
        stepper.moveTo(rand() % 400); //chuyển đến tọa độ 0 - 399 (Random)
        stepper.setMaxSpeed((rand() % 400) + 200); //chỉnh tốc độ.
        stepper.setAcceleration((rand() % 200) + 100); //chỉnh gia tốc.
    }
    Serial.println(stepper.distanceToGo());
    stepper.run(); // phải có hàm này ở hàm loop
}
// với mỗi biến stepper thì nói mới chạy được
```

Thư viện AccelStepper sẽ giúp điều khiển vi bước cho động cơ bước, giúp tăng động chính xác khi hoạt động.

2.2. Động cơ servo

2.2.1. Chức năng

Chức năng chính của Servo đó là điều khiển vị trí, thay đổi tốc độ chính xác, điều chỉnh momen phù hợp với những ứng dụng công việc. Trong hệ thống, dây chuyền sản xuất tự động hóa ngày nay không thể thiếu Servo. Nó chính là 1 hệ thống truyền động, hồi tiếp vòng kín.

2.2.2. Cấu tạo cơ bản

Động cơ RC Servo gồm 4 bộ phận: Động cơ một chiều, hộp số, biến trở và mạch điều khiển.



Hình 2.5. Động cơ servo

Trong đó:

Động cơ có tốc độ cao và momen thấp nên phải có hộp số để giảm tốc độ và đồng thời tăng momen để điều khiển vị trí tốt hơn. Sau khi qua hộp giảm tốc, động cơ có tốc độ khoảng 60 vòng/phút.

Biến trở được nối với hộp số hoặc trục động cơ, chính vì vậy mà khi động cơ quay biến áp sẽ quay theo. Trong lúc biến trở quay sẽ ứng với động cơ quay theo một góc tuyệt đối so với vị trí ban đầu. Ở một số loại RC Servo còn tích hợp thêm mạch cầu H để chỉnh động cơ quay theo vị trí mong muốn cho đến khi sai số giữa 2 tín hiệu điện áp bằng 0.

2.2.3. Nguyên lý hoạt động

Về nguyên lý làm việc, Servo được hình thành bởi hệ thống hồi tiếp vòng kín. Các tín hiệu ra của động cơ sẽ nối với mạch điều khiển. Khi vận hành vận tốc và vị trí của động cơ sẽ hồi tiếp về mạch điều khiển này.

Đặc biệt, nếu chuyển động của động cơ bị ngăn cản bởi lý do nào đó, thì cơ cấu tiếp nhận sẽ thấy tín hiệu ra chưa đạt được vị trí mong muốn. Đồng thời, mạch sẽ tiếp tục điều chỉnh sai lệch cho đến khi đạt được điểm chính xác nhất.

2.2.4. Phương thức điều khiển

a. Đầu vào

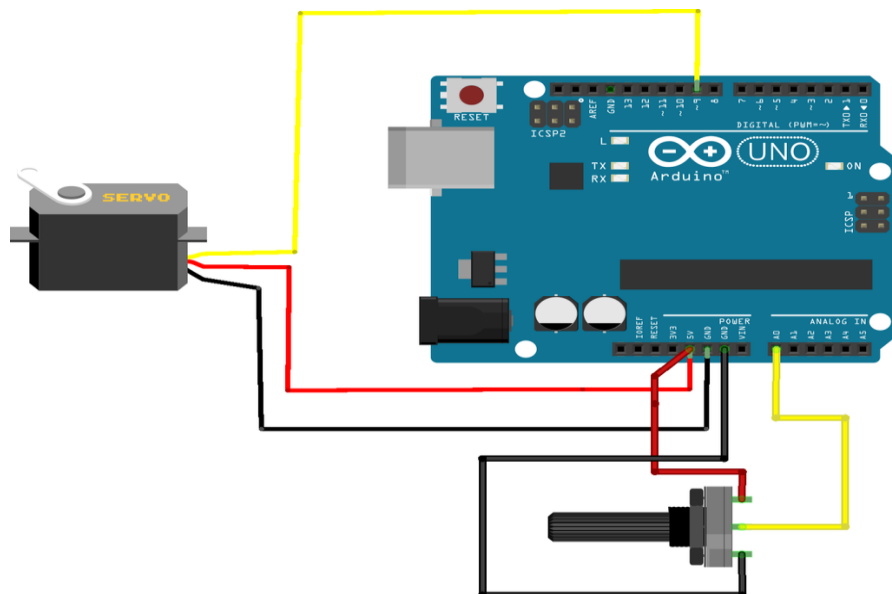
Đầu vào của phương thức điều khiển RC Servo là giá trị góc mong muốn động cơ sẽ thực hiện.

b. Đầu ra

Đầu ra của phương thức điều khiển RC Servo chính là động cơ quay đến góc mong muốn.

c. Codes chương trình kết nối, điều khiển

Kết nối phần cứng với Arduino:



Hình 2.6. Sơ đồ mạch điều khiển động cơ servo

Dây màu đỏ của servo nối với nguồn 5V của Arduino

Dây màu đen của servo nối với GND

Dây màu vàng (cam, xanh) nối với chân có PWM của Arduino

Code điều khiển:

Ví dụ: cho đối với servo sg90 180 độ

```
#include <Servo.h> //để khai báo thư viện Servo có sẵn trong Arduino
Servo myservo;      //khởi tạo đối tượng Servo có tên là myservo
int pos = 0;        //khởi tạo vị trí góc là 0
void setup()
{
    myservo.attach(9); //khai báo chân tín hiệu kết nối
}                    // của servo là chân D9 trong Arduino
void loop()
{
    for(pos = 0; pos < 180; pos += 1)
    {
        myservo.write(pos); //xuất giá trị góc pos cho servo thực hiện,
        delay(15);          // servo sẽ quay chính xác đến giá trị góc pos.
    }
    for(pos = 180; pos>=1; pos--=1)
    {
        myservo.write(pos);
        delay(15);
    }
}
```

2.3. Công tắc hành trình

2.3.1. Chức năng

Công tắc hành trình cũng tương tự như công tắc thường. Tuy nhiên chúng được trang bị thêm 1 cần gạt để giới hạn hành trình đi hoặc dừng để điều khiển một loại thiết bị điện nào khác. Ví dụ như khi tác động vào công tắc hành trình thì thiết bị sẽ dừng ngay tại vị trí đó hoặc cấp điện cho một loại thiết bị khác.

2.3.2. Cấu tạo cơ bản

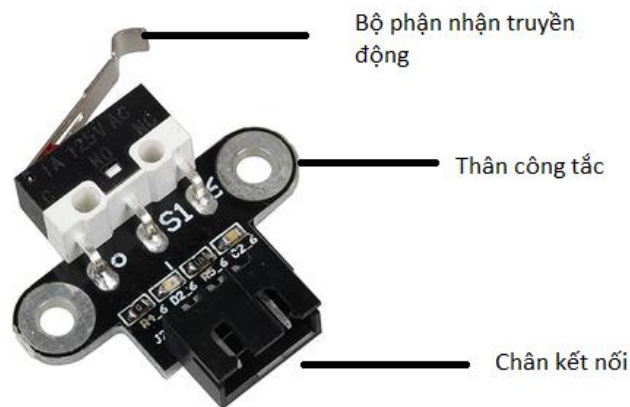
Một công tắc hành trình sẽ được cấu tạo từ các bộ phận như sau:

Bộ phận nhận truyền động: đây là một bộ phận khá quan trọng của một công tắc hành trình, thứ làm nên sự khác biệt giữa chúng và các loại công tắc khác. Chúng được gắn trên

đầu của công tắc có nhiệm vụ nhận tác động từ các bộ phận chuyển động để tác động kích hoạt công tắc.

Thân công tắc: phần thân của công tắc sẽ bao gồm các linh kiện bên trong với lớp vỏ bằng nhựa giúp chúng va đập, bảo vệ các mạch điện bên trong khỏi các tác nhân tác động vật lý.

Chân kết nối: đây được xem là phần tín hiệu ngõ ra cho công tắc vì nó có nhiệm vụ truyền tín hiệu đến các thiết bị khác khi bị tác động bởi bộ phận truyền động.



Hình 2.7. Cấu tạo công tắc hành trình

2.3.3. Nguyên lý hoạt động

Công tắc hành trình thường có 3 chân là: chân COM, chân thường đóng (NC) và chân thường hở (NO). Nguyên lý hoạt động công tắc hành trình là: Ở điều kiện bình thường, tiếp điểm giữa chân COM và chân NC được đấu với nhau. Khi có lực tác động lên cần tác động, tiếp điểm giữa chân COM và chân NC sẽ chuyển sang trạng thái hở, chân COM và chân NO được nối thông với nhau.

Công tắc hành trình là một thiết bị giúp chuyển đổi chuyển động cơ thành tín hiệu điện. Với mục đích là để phục vụ cho quá trình điều khiển và giám sát.

2.3.4. Phương thức điều khiển

a. Đầu vào

Là lực tác động lên bộ phận nhận truyền động.

b. Đầu ra

Là dạng nhị phân (0, 1) cho ta biết công tắc hành trình đang ở trạng thái đóng hay mở.

c. Codes chương trình kết nối, điều khiển

Kết nối công tắc hành trình với Arduino

Chân NC của công tắc hành trình nối với nguồn 5V của Arduino

Chân NO của công tắc hành trình nối với GND

Chân COM nối với chân digital của Arduino

Code nhận giá trị công tắc hành trình

```
const int pin_sw = 3;      //đặt biến nối vào chân D3
int val_sw = 0;           //khởi tạo biến giá trị công tắc hành trình
void setup()
{
    Serial.begin(9600); //Khởi tạo Serial để hiển thị kết quả lên màn hình
    pinMode(pin_sw, INPUT);
}
void loop()
{
    val_sw = digitalRead(pin_sw);      //đọc trạng thái công tắc hành trình
    Serial.print("GIA TRI NHAN DUOC: ");
    Serial.println(val_sw);            //in trạng thái công tắc hành trình
    delay(100);
}
```

2.4. Biến trở

2.4.1. Chức năng

Biến trở (có tên tiếng anh Variable Resistor) là một linh kiện thiết bị điện tử có mức điện trở thuần với khả năng chuyển đổi mức điện áp theo từng ý muốn. Bên cạnh đó, biến trở là một thiết bị dùng để điều chỉnh hoạt động của mạch điện hay dùng trong những ứng dụng giúp biến đổi điện trở để dễ dàng điều khiển các thiết bị cùng hiện tượng khác.

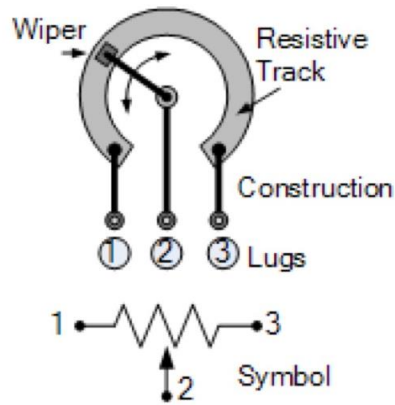
2.4.2. Cấu tạo cơ bản

Nhìn từ bên ngoài, biến trở có cấu tạo gồm 3 bộ phận chính:

Cuộn dây được làm bằng hợp kim có điện trở suất lớn

Con chạy/chân chạy. Cho khả năng chạy dọc cuộn dây để làm thay đổi giá trị trở kháng.

Chân ngõ ra gồm có 3 chân (3 cực). Trong số ba cực này, có hai cực được cố định ở đầu của điện trở. Các cực này được làm bằng kim loại. Cực còn lại là một cực di chuyển và thường được gọi là cần gạt. Vị trí của cần gạt này trên dải điện trở sẽ quyết định giá trị của biến trở.



Hình 2.8. Cấu tạo biến trở

2.4.3. Nguyên lý hoạt động

Đúng như tên gọi của nó là làm thay đổi điện trở, nguyên lý hoạt động chủ yếu của biến trở là các dây dẫn được tách rời dài ngắn khác nhau. Trên các thiết bị sẽ có vi mạch điều khiển hay các nút vặn. Khi thực hiện điều khiển các nút vặn các mạch kín sẽ thay đổi chiều dài dây dẫn khiến điện trở trong mạch thay đổi.

2.4.4. Phương thức điều khiển

a. Đầu vào

Là lực tác động khi ta xoay nút biến trở.

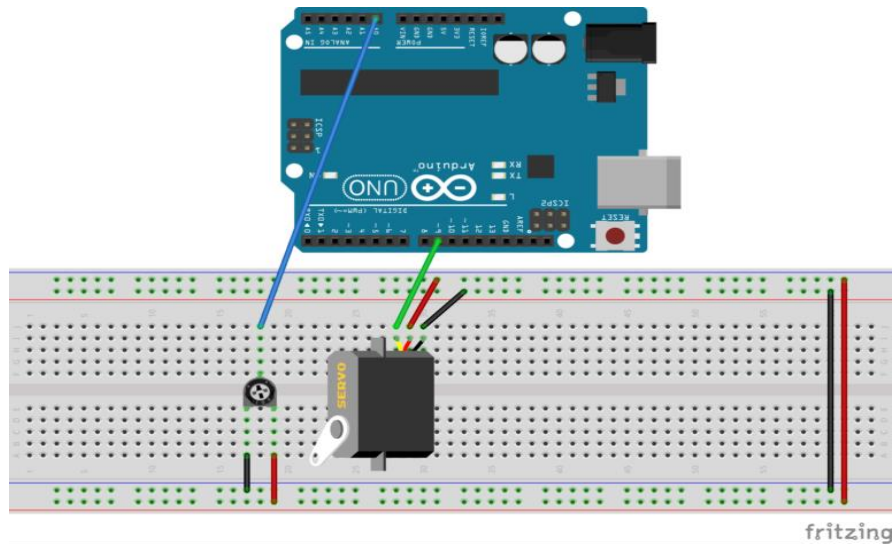
b. Đầu ra

Là giá trị của biến trở theo vị trí của nút xoay.

c. Codes chương trình kết nối, điều khiển

Ví dụ điều khiển động cơ servo bằng biến trở

Kết nối:



Hình 2.9. Sơ đồ mạch điều khiển động cơ servo bằng biến trở

Nối chân tín hiệu của biến trở với chân A0 của Arduino. Chân nguồn 5V và chân GND của biến trở nối tương ứng với Arduino. Động cơ servo được nối như ở phần trên.

Chương trình điều khiển:

```
#include <Servo.h>

Servo myservo;
int potpin = A0;
int val;
void setup()
{
  myservo.attach(9);
}
void loop()
{
  val = analogRead(potpin);    //đọc giá trị biến trở
  val = map(val,0,1023,0,180); //đổi giá trị biến trở thành giá trị góc.
  myservo.write(val);          //khi xoay biến thì servo quay tương ứng.
  delay(15);
}
```


CHƯƠNG 3. CHẾ TẠO VÀ TÍCH HỢP HỆ THỐNG TRÊN ROBOT

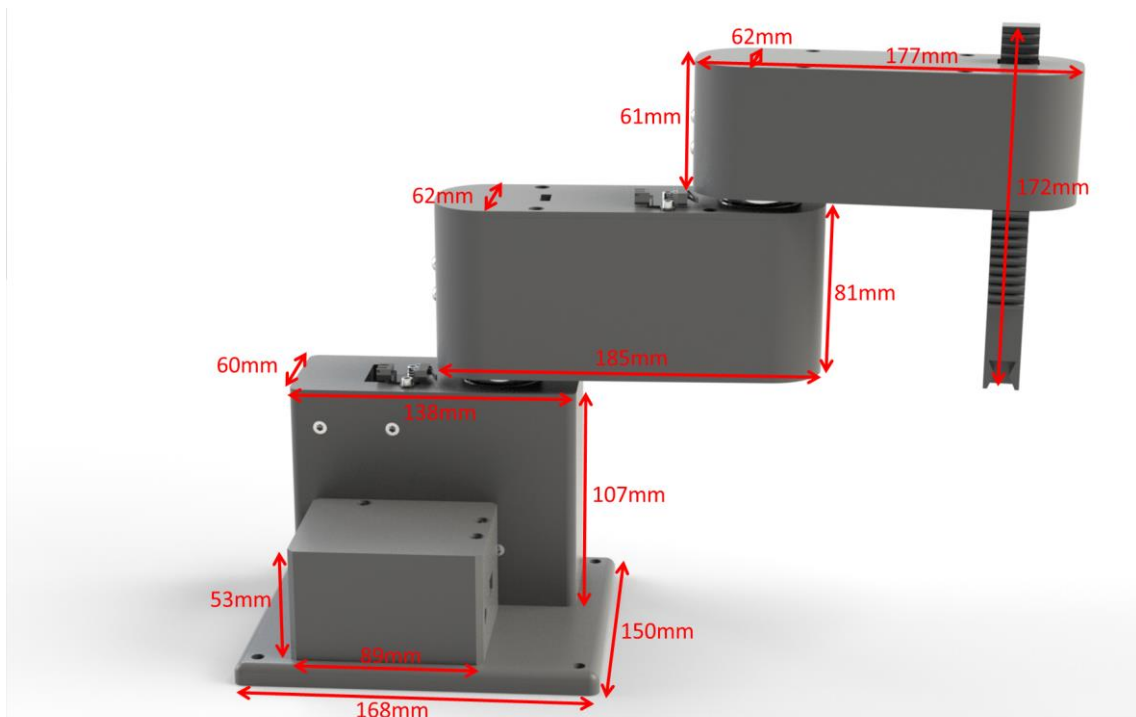
3.1. Chế tạo cơ khí

3.1.1. Chế tạo khung cơ khí

a. Kích thước

Sử dụng phần mềm SolidWorks hỗ trợ, chúng tôi thực hiện vẽ, lắp ghép và mô phỏng mô hình robot trên phần mềm theo kích thước đã tính toán, sau đó thực hiện khảo sát độ bền với phần mềm AnSys.

Dưới đây là kích thước khung cơ khí:



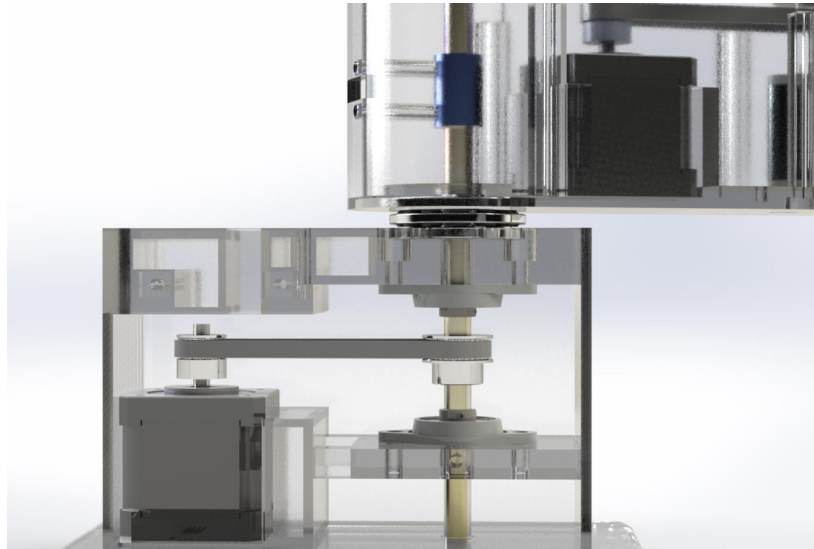
Hình 3.1. Các thông số của Robot

3.1.2. Các cơ cấu truyền động

a. Cơ cấu dây đai - pulley

Đối với các cơ cấu dây đai, trục crom, pulley đã có sẵn và theo một quy chuẩn nhất định thịnh hành trên thị trường vì thế chúng tôi đã mua sẵn và lắp ghép.

Chúng tôi sử dụng đai và pulley GT2 độ rộng 10mm để thực hiện việc truyền động.



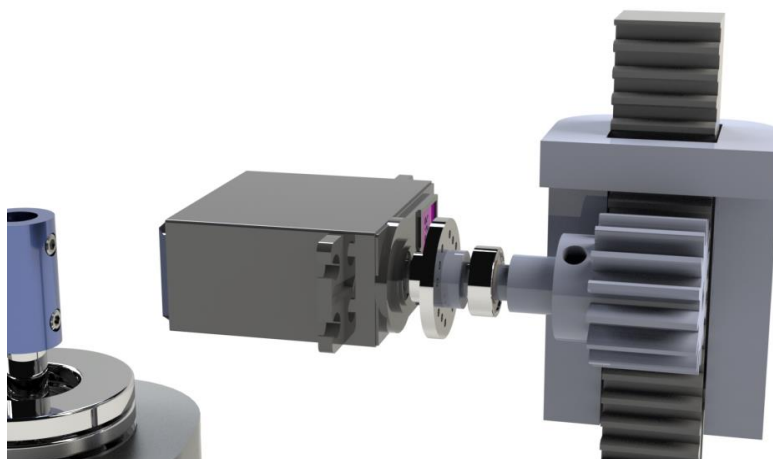
Hình 3.2. Cơ cấu truyền động bánh răng - dây đai của Robot

Trong cơ cấu trên, khi động cơ quay sẽ làm pulley quay, dây đai sẽ truyền động và làm quay pulley thứ 2 đang được gắn với trục crom. Khi này trục crom sẽ quay quanh một trục được cố định bởi hai vòng bi ngang. link phía trên được gắn chặt với trục crom và quay theo vận tốc góc của thanh crom, hoàn thành việc truyền động

Ở đây đảm bảo được yếu tố dây đai căng do chúng tôi đã tính toán khoảng cách hai pulley chính xác, hạn chế việc trượt bước nhiều nhất có thể. Thanh crom được gắn với hai vòng bi cố định, đảm bảo luôn quay quanh một trục cố định vuông góc với mặt phẳng Oxy

b. Cơ cấu thanh răng - bánh răng.

Vì chi phí cao và khó tìm trên thị trường, chúng tôi thực hiện gia công in 3D với cơ cấu này. Thực hiện vẽ mô phỏng trên phần mềm SolidWork theo quy định về chuẩn thanh răng của ISO. Chúng tôi có tính toán sai số do gia công in nhựa 3D, kết quả khá tốt



Hình 3.3. Cơ cấu truyền động bánh răng - thanh răng của Robot

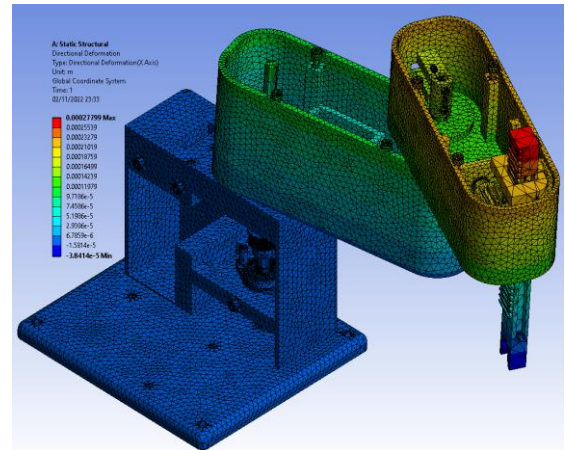
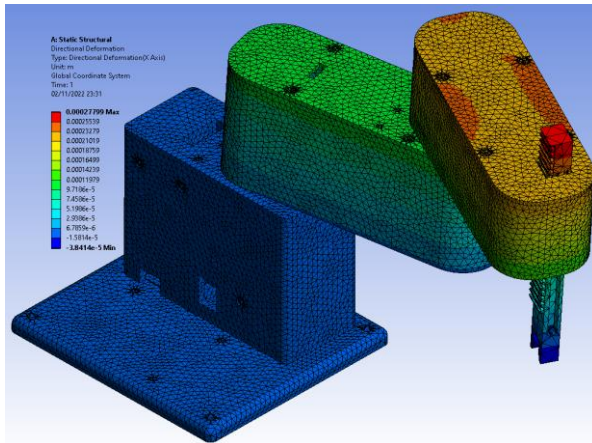
3.1.3. Khảo sát thiết kế

Cơ cấu tay robot được cấu thành bởi ba chi tiết là đế Base, Link 1, Link 2. Thiết kế của các chi tiết này được khảo sát tối ưu bằng cách sử dụng phương pháp phần tử hữu hạn mô phỏng với phần mềm Ansys để đảm bảo có thể chịu được tải nặng ít nhất. Sau đó các chi tiết này được chế tạo bằng vật liệu nhựa PLA-F sử dụng phương pháp gia công thêm lớp (in 3D). Bằng cách này, cơ cấu đảm bảo tính nhẹ, chịu được tải theo yêu cầu thiết kế. Các kết quả tối ưu cho các phần thể hiện dưới đây. Thông số vật liệu sử dụng trong thiết kế được mô tả trong Bảng 3.1.

Bảng 3.1. Các thông số vật liệu dùng cho thiết kế robot

Vật liệu	Mật độ (kg/m^3)	Young's Modulus (Mpa)	Poisson's Ratio	Ứng suất lớn nhất cho phép (Mpa)
Nhựa PLA-F	1400	2000	0.12	50

Dưới đây là chuyển vị của kết cấu khi thực hiện gắp vật nặng 3Kg



Hình 3.4. Khảo sát độ bền robot

Các Link và Base được liên kết với nhau bằng trục crom và có sự đỡ nhau bởi trục bi ở vị trí tiếp giáp các phần.

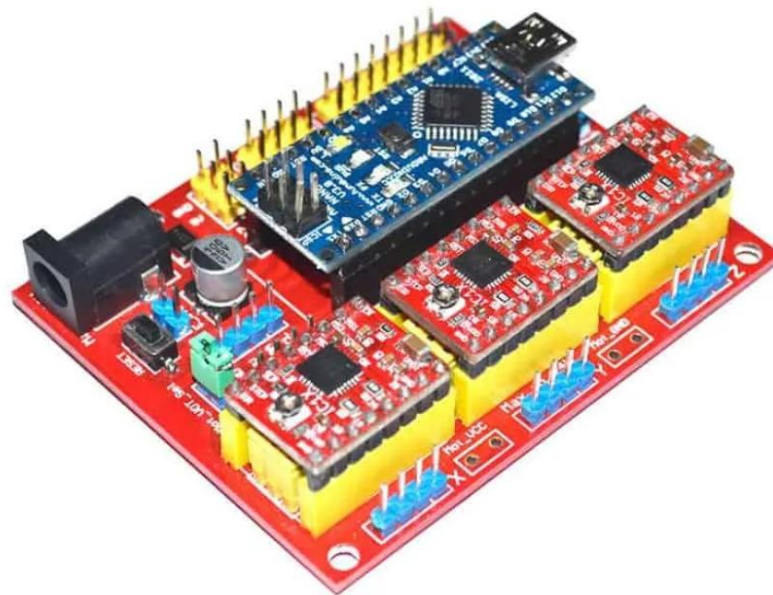
Khi nâng gắp vật với khối lượng 3kg, chúng ta có thể thấy phần đầu trên của tay gắp có sự biến dạng, tuy nhiên hệ số an toàn nhỏ nhất của cả kết cấu vẫn lớn hơn 1, do đó không xảy ra đứt gãy kết cấu.

Bảng 3.2. Chân kết nối động cơ của CNC Shield V4

	X	Y	Z
DIR	D2	D3	D4
STEP	D5	D6	D7
MS1-2-3	jump	jump	jump
EN	D8	D8	D8
VDD	5V	5V	5V
1B 1A 2A 2B	Động cơ	Động cơ	Động cơ
VMOT	12V	12V	12V

c. Cách ghép nối

Arduino và Driver A4988 sẽ được cắm vào CNC Shield V4 như hình. Khi đó các chân của động cơ bước sẽ được nối vào X, Y, Z.



Hình 3.6. Ghép nối Arduino, A4988 và CNC Shield V4

3.2.2. Drive a4988

a. Đặc điểm

Mạch điều khiển động cơ bước A4988 là driver điều khiển động cơ bước cực kì nhỏ gọn, hỗ trợ nhiều chế độ làm việc, điều chỉnh được dòng ra cho động cơ, tự ngắt khi quá nóng.

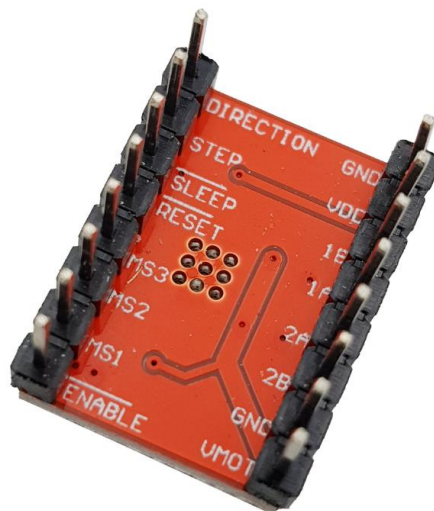
Mạch điều khiển A4988 hỗ trợ nhiều chế độ hoạt động của động cơ bước lưỡng cực

như: Full step, $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$

Công suất ngõ ra lên tới 35V, dòng đỉnh 2A.

b. Sơ đồ chân

Chân của Driver A4988 được thể hiện ở hình.



Hình 3.7. Chân của A4988

Bật tắt động cơ thông qua pin Enable, mức Low là bật module, mức High là tắt module.

Điều khiển chiều quay của động cơ thông qua pin DIR.

Điều khiển bước của động cơ thông qua pin STEP, mỗi xung là tương ứng với 1 bước (hoặc vi bước).

Hai chân Sleep và Reset luôn nối với nhau.

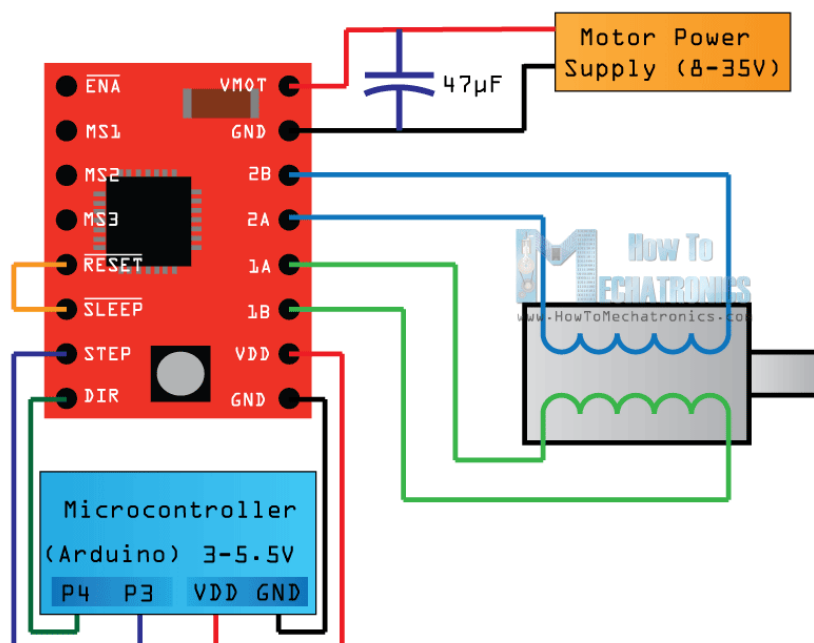
Các chế độ hoạt động của động cơ sẽ được điều khiển qua 3 pin MS1, MS2, MS3. Bảng thể hiện ảnh hưởng 3 pin trên đến các chế độ hoạt động.

Bảng 3.3. Các chế độ hoạt động của driver

MS1	MS2	MS3	Chế độ hoạt động
Low	Low	Low	Full step
High	Low	Low	$\frac{1}{2}$
Low	High	Low	$\frac{1}{4}$
High	High	Low	$\frac{1}{8}$
High	High	High	$\frac{1}{16}$

c. Cách ghép nối

Driver A4988 nhận tín hiệu từ Arduino cho chân STEP và DIR, rồi truyền tín hiệu cho động cơ qua chân 1A,1B,2A,2B. Hình biểu diễn sơ đồ ghép nối driver với Arduino và động cơ bước.

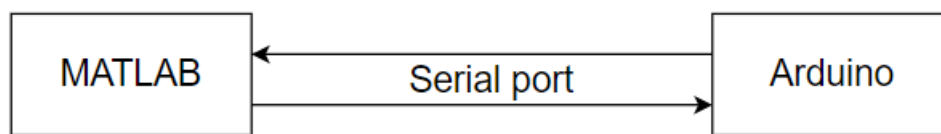


Hình 3.8. Cách nối chân driver a4988

3.3. Codes chương trình điều khiển

Arduino Nano khó khăn trong việc tính động học ngược của cánh tay robot do không đủ bộ nhớ để tính toán. Matlab thực hiện dễ dàng việc tính toán động học thuận hay động học nghịch của cánh tay robot. Do đó chúng tôi thực hiện kết nối Matlab và Arduino qua giao tiếp Serial port từ thư viện trong Matlab.

Matlab sẽ tính toán ra góc khớp và vận tốc của động cơ rồi truyền tới Arduino để Arduino điều khiển động cơ. Hình thể hiện giao tiếp Serial port của matlab và arduino.



Hình 3.9. Giao tiếp Serial port của Matlab và Arduino

Các mã lập trình để giao tiếp Serial port trên Matlab:

```

clear all;
x = serialport('COM3',9600); //Chọn chân kết nối Serial port
//Tên từng dòng của Bảng nhập động học nghịch
promptInverse = {'Enter x (cm) ', 'Enter y (cm) ', 'Enter z (cm) '};
//Tên Bảng nhập động học nghịch
dlgtitleInverse = 'Inverse Kinematics';
//Nhập tọa độ điểm thao tác mong muốn dưới dạng string
inputPoints_str = inputdlg(promptInverse,dlgtitleInverse,[1 60]);
//Chuyển tọa độ trên thành double để tính toán
inputPoints_dou = str2double(inputPoints_str);
//Lưu tọa độ điểm thao tác
E = [inputPoints(1),inputPoints(2),inputPoints(3)];
//Tính toán động học nghịch
dt = 0.1;
t_max = 10;
for t=0:dt:t_max
    [q1,q2,q3] = Inverse_Kinematics_1(E(1),E(2),E(3));
    //Tính số bước từ giá trị góc khớp
    q1_step = q1/pi*3200; // 180 độ tương đương 3200 bước
    q2_step = q2/pi*3200;
    q3_servo = q3/pi*180; //q3 là servo
    //Chuyển giá trị số bước thành string để truyền sang arduino
    q1_str = sprintf('%0f',q1_step); //làm tròn thành số nguyên
    q2_str = sprintf('%0f',q2_step);
  
```

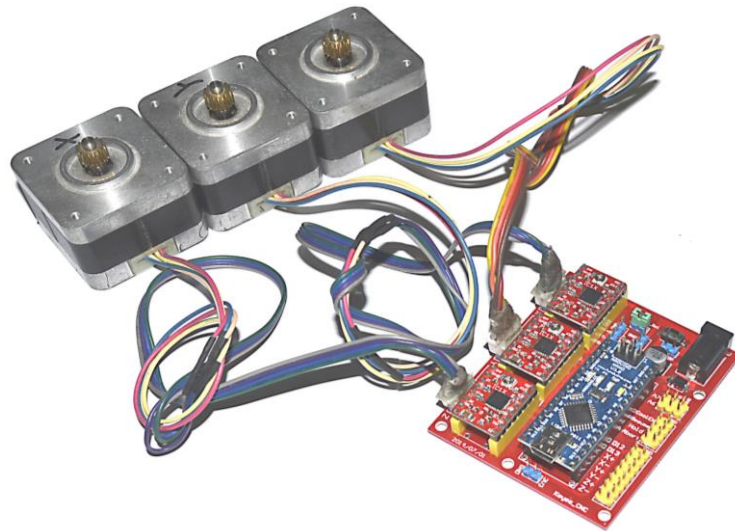


```

q3_str = sprintf('%0f',q3_servo);
q_str = append(q1_str,' ',q2_str,' ',q2_str);
//Chuyển dữ liệu trên vào cổng Serial port
write(x,q_str,"string");
//Kiểm tra lại dữ liệu đã chuyển, tối đa 6 ký tự
read(x,11,"string")
end
clear all;

```

Sau khi nhận được dữ liệu từ Matlab thông qua Serial port, Arduino sẽ xử lý và truyền tín hiệu góc khớp và vận tốc cho động cơ. Hình thể hiện sự kết nối giữa arduino và động cơ.



Hình 3.10. Kết nối giữa Arduino và động cơ

Mã lập trình để giao tiếp Serial port trên Arduino:

```

int msg; // khởi tạo biến lưu dữ liệu truyền
void setup()
{
  Serial.begin(9600); //khai báo sử dụng serial cổng 9600
  stepperInit(); // khai báo step
  servoInit(); // khai báo servo
}
int kt = true // biến kiểm tra
void loop()
{
  if (kt)
  { // chỉ chạy 1 lần duy nhất trong vòng loop
    stepperStart(); // khởi tạo step - homing
    servoSetting(); // cài đặt giá trị servo
    servoHoming(); //test servo
  }
}

```

```

    }
    readAngle(); //đọc giá trị từ serial
    stepperRun(); // chạy các giá trị vừa nhập
}

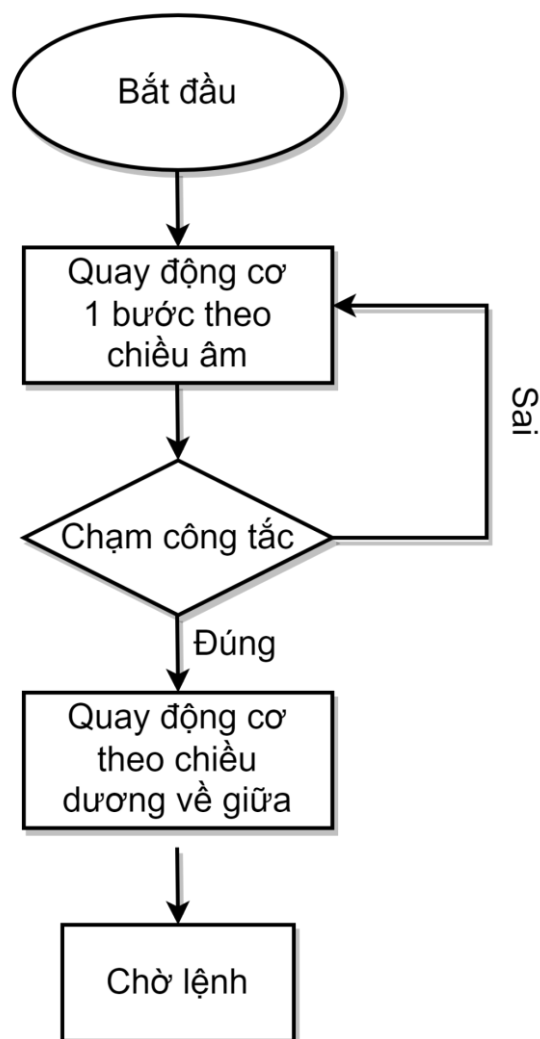
```

Do các mã lập trình quá dài, chúng tôi sẽ trình bày chi tiết tại đường [Link](#)

3.4. Thực nghiệm, đo đạc đánh giá sai số

3.4.1. Homming

a. Quá trình homing



Hình 3.11. Lưu đồ thuật toán quá trình homing của robot

Khi robot khởi động, động cơ bắt đầu quay cùng chiều kim đồng hồ cho tới khi gặp công tắc hành trình. Lúc này arduino sẽ nhận được tín hiệu và ra lệnh cho động cơ dừng lại.

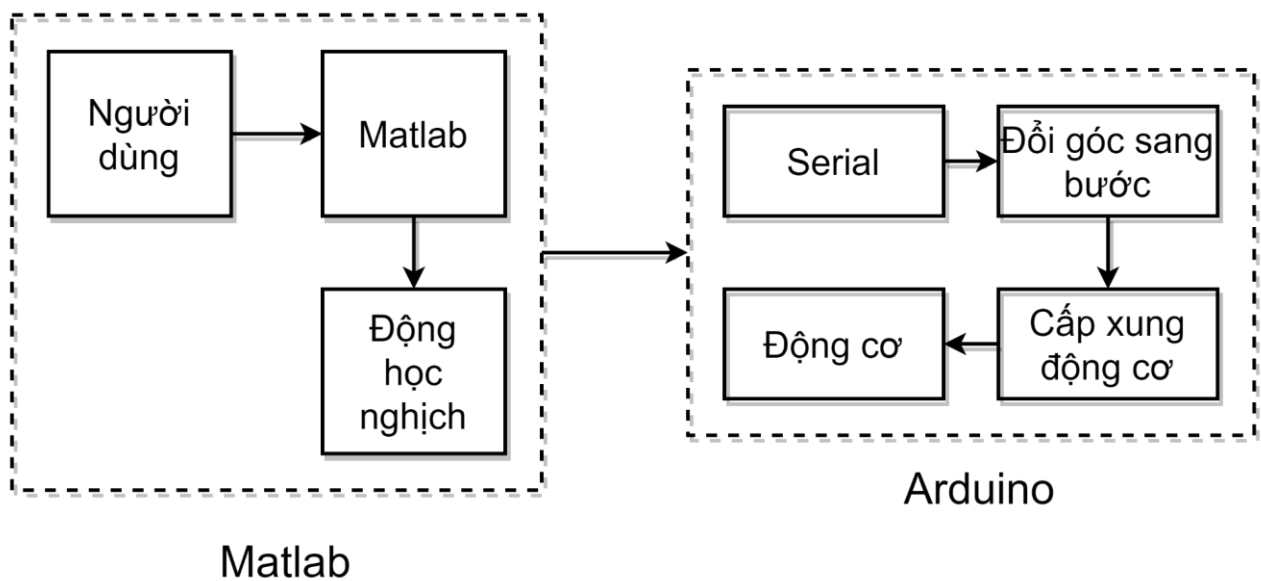
Tiếp đến, robot sẽ di chuyển các link về vị trí chính giữa không gian làm việc và hoàn tất quá trình homing.

b. Video

[Link video](#): Homing và GoTohome (đưa robot về giữa, chờ lệnh điều khiển)

3.4.2. Chuyển động điểm đến điểm

a. Quy trình chuyển động



Hình 3.12. Sơ đồ hoạt động chi tiết của robot

Robot nhận lệnh di chuyển từ điểm tới điểm từ người dùng thông qua giao diện matlab. Matlab sẽ tính toán các chuyển động cần thiết và gửi tới arduino các giá trị góc cần quay của từng động cơ

b. Video

[Link video](#): Homing hoàn chỉnh và đi tới một điểm

3.4.3. Đánh giá sai số chuyển động điểm đến điểm

a. Nguyên nhân

Do chế tạo phần cứng, vì là cơ cấu in 3D nhựa nên không thể tránh khỏi việc bị sai số trong quá trình gia công. Các lớp nhựa bị chảy nhiều hoặc ít hơn dự tính, hoặc các khoảng cách thực tế so với bản vẽ bị lệch đi, dù rất nhỏ nhưng vẫn gây ra sai số.

Mặt khác, do các động cơ, dây đai, pulley ... có hiệu suất thấp hơn 100% hoặc các mạch linh kiện bị nhiễu trong quá trình hoạt động cũng là tác nhân gây ra sai số

b. Khảo sát sai số khi di chuyển

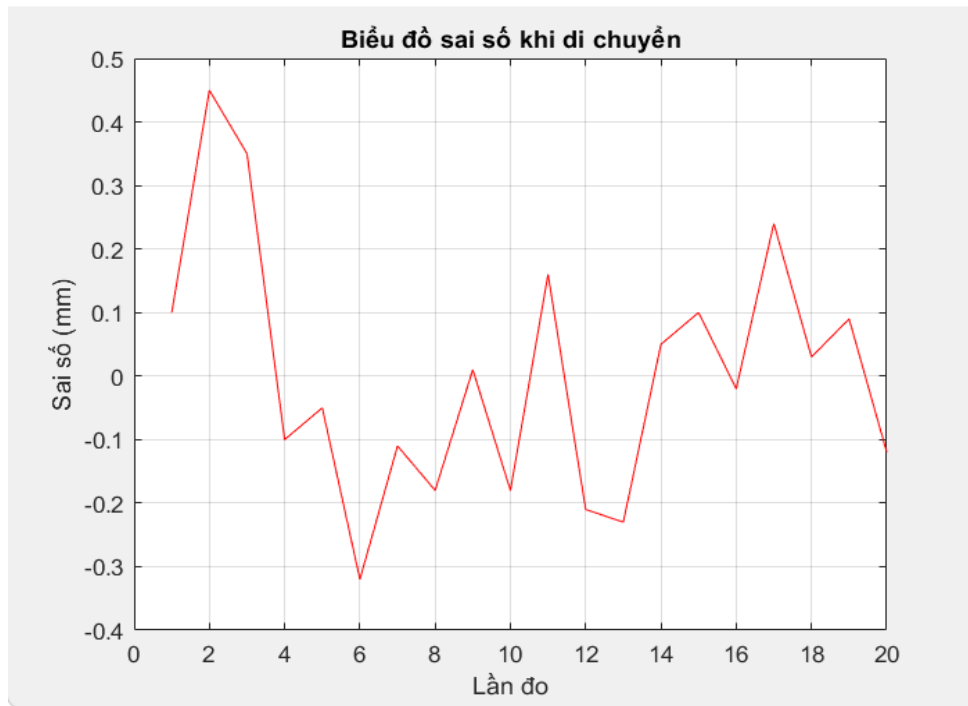
Độ chính xác lặp lại của chuyển động của hệ được khảo sát với quỹ đạo di chuyển từ điểm (123, 6.15, 0) đến điểm (0, 0, 0) sử dụng đồng hồ so INSIZE 2308-5A có độ chia 0.01mm. Kết quả được thể hiện trong bảng dưới đây.

Bảng 3.4. Sai số giữa các lần đo

Lần đo	1	2	3	4	5	6	7
Sai số (mm)	0.1	0.45	0.35	-0.1	-0.05	-0.32	-0.11
Lần đo	8	9	10	11	12	13	14
Sai số (mm)	-0.18	0.01	-0.18	0.16	-0.21	-0.23	0.05
Lần đo	15	16	17	18	19	20	
Sai số (mm)	0.1	-0.02	0.24	0.03	0.09	-0.12	



Hình 3.13. Đồng hồ so INSIZE 2308-5A và cách bố trí đo sai số



Hình 3.14. Sai số khi robot di chuyển từ điểm $(123,6.15,0)$ đến điểm $(0,0,0)$

Độ lệch chuẩn cho ta biết được độ phân tán của giá trị thống kê so với giá trị trung bình, ở từng thời điểm khác nhau. Nếu độ lệch chuẩn thấp thì tính biến động không đáng kể và ngược lại. Ở đây độ lệch chuẩn là 0.19778mm.

c. Video

[Link video](#): Chuyển động vẽ 2 quỹ đạo chữ A chồng lên nhau, khảo sát độ sai số giữa 2 lần chuyển động

3.4.4. Chuyển động theo một quỹ đạo cho trước

a. Quy trình

Khi người dùng ra lệnh quỹ đạo, matlab sẽ tính toán và truyền về cho arduino chuỗi 50 vectơ bao gồm số góc cần quay, vận tốc, gia tốc của từng động cơ

Thực chất chuyển động theo quỹ đạo chính là chuyển động chuỗi điểm, theo đó các điểm rất gần với nhau sẽ tạo ra quỹ đạo mong muốn.



Hình 3.15. Robot thực hiện vẽ một đường thẳng theo quỹ đạo cho trước

b. Video

[Link video](#): chuyển động theo quỹ đạo đường thẳng

[Link video](#): chuyển động theo quỹ đạo chữ A

[Link video](#): chuyển động vẽ dấu tích “✓”

3.5. Kết quả và đánh giá

3.5.1. Kết quả

Hoàn thành mục tiêu khảo sát, gia công, chế tạo và điều khiển robot scara với quy mô giáo dục.

Quy trình thiết kế robot đã nêu ở chương 1, phần 3.4 kiểm thử robot khi hoạt động thực tế, chúng tôi nhận thấy robot đáp ứng được các mục tiêu đặt ra, chuyển động động cơ đạt yêu cầu.

Tuy nhiên vẫn còn xuất hiện sai số, nguyên nhân do sử dụng động cơ, bo mạch rẽ ảnh hưởng tới tín hiệu, và trong quá trình gia công in 3D kích thước bị thay đổi so với mô phỏng.

3.5.2. Định hướng phát triển

Tích hợp thêm cảm biến khoảng cách, giúp phần cân bàn cho điểm E được tự động hoá và chính xác hơn, tạo một chu trình kiểm tra bàn giúp robot có thể điều chỉnh trục E liên tục, viết trên mặt phẳng gồ ghề.

Sử dụng thêm camera, xử lý ảnh phát hiện vật và gấp vật, có thể dùng camera quét ảnh có sẵn và sau đó vẽ lại theo mẫu.

Cải thiện về tốc độ cho robot, nâng cao tính ổn định khi chuyển động với vận tốc cao đồng thời giảm sai số, khi hoạt động của robot.

TÀI LIỆU THAM KHẢO

- [1] PGS. TS. Nguyễn Trường Thịnh, “Giáo trình Kỹ thuật Robot”, 2014.
- [2] “MG996R Servo Motor.” Components101,
- [3] Laz, et al. “How To Control a Stepper Motor with A4988 Driver and Arduino.”
How To Mechatronics, 17 May 2022,
- [4] “Mechanical Endstop.” RepRap
- [5] "SCARA Robots - Fanuc"
- [6] <https://components101.com/motors/mg996r-servo-motor-datasheet>
- [7] https://reprap.org/wiki/Mechanical_Endstop
- [6] <https://docs.arduino.cc/learn/electronics/stepper-motors>
- [7] <http://arduino.vn/bai-viet/181-gioi-thieu-servo-sg90-va-cach-dieu-khien-bang-bien-tro>
- [8] <http://arduino.vn/bai-viet/87-bai-4-doc-dien-ap-dieu-chinh-boi-bien-tro>
- [9] <https://howtomechatronics.com/tutorials/arduino/how-to-control-stepper-motor-with-a4988-driver-and-arduino/>