

## 1. MỞ ĐẦU

### a. Đặt vấn đề

Sản xuất thông minh là một nền tảng cơ bản và quan trọng của nền công nghiệp hiện đại và kinh tế toàn cầu. Một trong những thách thức chính của sản xuất thông minh là kết nối không gian vật lý và không gian số. Sự phát triển nhanh chóng của mô phỏng, thu thập dữ liệu, truyền thông dữ liệu và các công nghệ tiên tiến khác đã tạo ra sự tương tác lớn hơn giữa không gian vật lý và không gian số. Digital twin đang ngày càng được sử dụng rộng rãi trong nhiều lĩnh vực và ngành công nghiệp khác nhau, và nhận được sự quan tâm từ cả giới học thuật và ngành công nghiệp.

Hiện nay, nhiều doanh nghiệp tại Việt Nam như ASIM, Ahamove, OneFin, Ngân hàng VIB, SDC, Crystal Bay, Joolux, DFAR, Time Universal, Ai20X và HoiAn đã triển khai và áp dụng thành công bản sao số trong quy trình quản lý và sản xuất. Các công ty lớn ở nước ngoài như General Electric, IBM, PTC, Microsoft Corporation, Siemens AG, ANSYS, SAP, Oracle, Robert Bosch, SWIM.AI và ZenithCities cũng đang áp dụng hiệu quả các công nghệ bản sao số.

Tuy nhiên, tại Việt Nam, các nghiên cứu về phát triển hệ thống robot thông minh vẫn đang ở giai đoạn phát triển ban đầu và tập trung vào các phần nhỏ trong cấu trúc của bản sao số. Một cấu trúc tổng thể cho bản sao số chưa được nghiên cứu và phát triển toàn diện. Do đó, việc nghiên cứu và phát triển nền tảng (framework) cho bản sao số là cần thiết.

### b. Nghiên cứu trong và ngoài nước

#### 1. *Ngoài nước:*

Digital twin hay bản sao kỹ thuật số là một đại diện kỹ thuật số của một đối tượng hoặc hệ thống vật lý. Về bản chất, digital twin là một chương trình máy tính lấy dữ liệu trong thế giới thực về một đối tượng hoặc hệ thống vật lý làm đầu vào và tạo ra các dự đoán hoặc mô phỏng đầu ra về cách đối tượng hoặc hệ thống vật lý đó sẽ bị ảnh hưởng bởi những đầu vào đó.

Cơ sở lý thuyết của bản sao số đến từ các ngành khoa học khác nhau như khoa học thông tin, kỹ thuật sản xuất, khoa học dữ liệu và khoa học máy tính. Các lý thuyết phù hợp nhất được chia làm bốn phần [1]: (1) mô hình digital twin, mô phỏng, xác minh; (2) Hợp nhất dữ liệu; (3) Tương tác và cộng tác; và (4) Dịch vụ.

Mô hình hóa và mô phỏng digital twin là cơ sở của việc phát triển các digital twin trong thực tế. Nhiều nhà nghiên cứu đã đề xuất các kiến trúc mô hình khác nhau. Schroeder và cộng sự [2] đã đề xuất một kiến trúc mô hình digital bao gồm năm lớp (tức là lớp thiết bị, lớp giao diện người dùng, lớp dịch vụ web, lớp truy vấn và lớp kho dữ liệu) để quản lý dữ liệu digital twin. Họ cũng phát triển hệ thống thực tế tăng cường để hiển thị thông tin thời gian thực. Họ cũng đề xuất trao đổi dữ liệu giữa các hệ thống không đồng nhất thông qua AutomationML. Phương pháp này gồm 3 giai đoạn: (1) Tạo mô hình; (2) Xác nhận mô hình; Và (3) phát triển hệ thống thông tin. Yun và cộng sự [3] cũng đã đề xuất một mô hình hóa kiến trúc cho các nền tảng digital twin lớn bao gồm một khuôn khổ hợp tác phân tán và một cơ chế truyền thông. Moreno và cộng sự [4] đã sử dụng một máy đọc lỗ để giới thiệu quy trình từng bước để xây dựng mô hình digital twin. Quy trình gồm 5 bước: (1) Xây dựng mô hình 3D; (2) Trích xuất hành vi; (3) Mô hình hóa sự tương tác giữa máy đọc lỗ và các phần tử chuyên động; (4) Mô hình hoạt động; Và (5) Mô phỏng. X. Wang và cộng sự [5] đã sử dụng các nền tảng mã nguồn mở ROS-Gazebo để mô phỏng robot trực tuyến. Garg và cộng sự [6] sử dụng thực tế ảo giúp tăng tính trải nghiệm và tương tác khi mô phỏng và tương tác với robot từ xa. Ngoài ra mô hình digital twin cần được đánh giá đúng mức để đảm bảo độ chính xác của nó trong việc phản ánh thực tế ảo và thực. Do đó, Smarslok và cộng sự [7] đã đề xuất một khuôn khổ để lượng hóa lỗi và đánh giá độ tin cậy, bao gồm một bộ thước đo để đo độ trung thực của các mô hình bản sao số.

Hợp nhất dữ liệu là một công nghệ quan trọng vì digital twin phải xử lý một khối lượng dữ liệu rất lớn từ nhiều nguồn khác nhau như thiết bị, môi trường vật lý, dữ liệu trong quá khứ, .... Ricks và cộng sự [8] đã đề xuất một kỹ thuật giảm thứ tự cho digital twin được áp dụng trong phương pháp tổng quát hóa độ trung thực cao của tế bào để nâng cao hiệu quả xử lý dữ liệu. Cai và cộng sự [9] đã phát triển một phương pháp tích hợp dữ liệu cảm biến và dữ liệu sản xuất làm cơ sở xây dựng digital twin của máy phay đứng, nơi dữ liệu cảm biến được sử dụng để theo dõi các hoạt động gia công và dự đoán độ nhám bề mặt.

Dröder và cộng sự [10] đã ứng dụng thuật machine learning để hiệu chuẩn human-robot. Bielefeldt và cộng sự [11] đã đề xuất một phương pháp đánh giá không phá hủy (NDE) để phát hiện các vết nứt do mỏi. Một nghiên cứu điển hình về cánh máy bay chỉ ra rằng phương pháp này có thể giảm số lượng tính toán một cách hiệu quả. Seshadri và Krishnamurthy [12] đã sử dụng các phản ứng sóng có hướng dẫn để đưa ra các dự đoán trong thời gian thực. Họ tích hợp dữ liệu cảm biến, dữ liệu đầu vào và dữ liệu ảo để mô tả một đối tượng vật lý và chẩn đoán kích thước thiệt hại, vị trí và các thông tin hư hỏng khác.

Pairet et al. [13] giới thiệu công nghệ rô-bốt ngoài khơi sử dụng bộ mô phỏng trung tâm ORCA để đào tạo và thử nghiệm các giải pháp hợp tác giữa người và máy thống nhất ba loại hệ thống sản xuất trên nền tảng kỹ thuật số đôi. Voinov và cộng sự [14] đã nghiên cứu một phương pháp cung cấp khả năng quản lý đáng tin cậy của các hệ thống IoT phức tạp. Martinez et al. [15] đã nghiên cứu tác động của bộ đôi kỹ thuật số đôi với đối mới mô hình kinh doanh dịch vụ y tế bằng cách hiểu đầy đủ cách thiết lập, triển khai và sử dụng sức khỏe số và hiểu tác động của sức khỏe số trong kinh doanh dịch vụ doanh nghiệp. Biesinger và cộng sự [16] đã giới thiệu bộ đôi kỹ thuật số của hệ thống sản xuất body-in-white để đạt được sự tích hợp nhanh chóng của những chiếc xe hơi mới.

## **2. Trong nước:**

Ở Việt Nam, việc nghiên cứu liên quan đến bản sao số cũng đã ngày càng thu hút được nhiều quan tâm của cộng đồng khoa học cũng như doanh nghiệp, đặc biệt là các doanh nghiệp công nghệ. Công ty Robot3T [17], một công ty khởi nghiệp đi đầu trong lĩnh vực robot tại Việt Nam đã phát triển được các sản phẩm robot hợp tác (Colab robot), ứng dụng cả trong công nghiệp và dịch vụ. Và gần đây, việc nghiên cứu phát triển hệ thống bản sao số cho robot cũng đang được công ty này quan tâm và nghiên cứu phát triển. Cho đến nay việc nghiên cứu phát triển các hệ thống robot thông minh đã và đang được nhiều nhóm nghiên cứu đẩy mạnh. Tuy nhiên các nghiên cứu này hiện nay chỉ tập trung phát triển một số phần trong cấu trúc của một hệ thống bản sao số, như chỉ tập trung nghiên cứu phát triển các mô hình điều khiển thông minh [18-20] hoặc tập trung ứng dụng trí tuệ nhân tạo vào nhận dạng, xử lý ảnh [21-23] hoặc mô phỏng [24,25] hoặc phát triển các nền tảng IOT với công nghệ Edge Computing hoặc Cloud Computing [26-28]. Một bản sao số với cấu trúc tổng thể hiện nay vẫn chưa được nghiên cứu phát triển toàn diện tại Việt Nam.

### **c. Mục tiêu đề tài**

Nghiên cứu phát triển mô hình bản sao số cho robot. Trong đó tập trung:

Xây dựng một cấu trúc bản sao số cho robot

Phát triển mô hình số robot bao gồm mô hình hình học 3D và mô hình điều khiển.

Tích hợp mô hình số Robot vào môi trường ảo, đồng thời kết nối vào nền tảng mã nguồn mở ROS.

Kết nối, truyền thông dữ liệu 2 chiều online theo thời gian thực giữa hệ thống robot thực và robot ảo, trong đó tập trung dữ liệu về góc biến khớp, thông số vị trí của điểm cuối tay máy (end-effector) dựa trên nền tảng ROS thông qua bộ giao thức Socket TCP.

### **d. Đối tượng và phạm vi nghiên cứu**

#### **1. Đối tượng nghiên cứu**

Mô hình bản sao số của robot

## **2. Phạm vi nghiên cứu**

Đề tài tập trung nghiên cứu kết nối, thu thập dữ liệu các biến khớp và truyền dữ liệu 2 chiều theo thời gian thực giữa mô hình số và mô hình vật lý.

### **e. Phương pháp nghiên cứu**

#### **1. Nghiên cứu thực nghiệm (Experimental research):**

Thiết lập các thí nghiệm và thực hiện kết nối và tương tác với robot thực để thu thập dữ liệu. Bằng cách tiến hành các thí nghiệm và kiểm soát các yếu tố để kiểm tra, đánh giá và thu thập dữ liệu về hoạt động của robot. Các dữ liệu này sẽ phục vụ cho việc phân tích ở các bước tiếp theo.

#### **2. Nghiên cứu trường hợp (Case study research):**

Chia nghiên cứu thành hai trường hợp: một trường hợp khi robot hoạt động bình thường, và một trường hợp khi robot gặp lỗi. Bên cạnh đó, xem xét một trường hợp khác khi robot hoạt động theo một quỹ đạo nhất định. Việc nghiên cứu các trường hợp này sẽ giúp bạn hiểu rõ hơn về hoạt động và hiệu suất của robot trong các tình huống khác nhau và làm dữ liệu cho việc tối ưu trong những bước phát triển sau.

#### **3. Nghiên cứu phân tích (Analytical research):**

Tập trung vào việc phân tích và đánh giá các dữ liệu thu thập được từ hoạt động của robot. Sử dụng các phương pháp thống kê và phân tích để so sánh và tìm ra độ trễ, sai số giữa mô hình thực và mô hình ảo trong quá trình hoạt động của robot. Bằng cách phân tích dữ liệu, những nhận xét và kết luận về hiệu suất và độ chính xác của robot được đưa ra.

### **f. Mô tả hệ thống**

Hệ thống gồm 3 thành phần chính: (1) Phần ảo; (2) Phần thực; và (3) Phần xử lý trung tâm. Phần ảo là một phần mềm hỗ trợ có khả năng tương tác, điều khiển và mô phỏng. Phần thực là đối tượng vật lý, trong đề tài là robot 2 bậc tự do, có khả năng truyền và nhận dữ liệu cảm biến và tín hiệu điều khiển. Phần xử lý trung tâm dùng để xử lý, kết nối giữa phần thực và phần ảo và lưu trữ dữ liệu. Việc sử dụng bộ xử lý trung tâm giúp tăng khả năng mở rộng và ứng dụng của hệ thống. Việc kết nối và truyền thông dữ liệu 2 chiều online theo thời gian thực giữa bản sao số và bộ xử lý trung tâm thông qua socket TCP. Và việc kết nối và truyền thông dữ liệu giữa bộ điều khiển trên arduino mega 2560 và bộ xử lý trung tâm thông qua giao thức UART.

Hoạt động của hệ thống như sau:

Khi hệ thống đã khởi động thì bản sao số và robot thực sẽ kết nối với bộ xử lý trung tâm. Nếu kết nối thất bại thì hệ thống sẽ thông báo lỗi và chờ kết nối lại. Khi đã kết nối thành công hệ thống sẽ thiết lập các kết nối và dữ liệu bắt đầu được truyền giữa phần bản sao số và robot thực thông qua bộ xử lý trung tâm. Đồng thời dữ liệu sẽ được xử lý và lưu trữ tại bộ nhớ cục bộ và sao lưu định kỳ lên đám mây.

Khi người dùng điều khiển robot từ bản sao số, tín hiệu sẽ được gửi đến bộ điều khiển trung tâm. Ở đây tín hiệu sẽ được xử lý thành giá trị xung để truyền đến bộ điều khiển của robot thực. Bộ điều khiển sẽ tạo xung điều khiển để điều khiển mạch động lực từ đó điều khiển động cơ. Các dữ liệu về encoder được thu thập và truyền về bộ điều khiển để hình thành bộ điều khiển vòng kín. Các giá trị này sẽ được bộ điều khiển gửi đến bộ xử lý trung tâm. Tại đây dữ liệu cảm biến được xử lý và tính toán sau đó được truyền đến bản sao số. Khi nhận được dữ liệu, bộ điều khiển của bản sao số sẽ điều

khởi mô hình đúng với các thông số nhận được. Ngoài ra các thông số cũng được thể hiện trên màn hình để người dùng theo dõi và giám sát.

#### g. Nội dung nghiên cứu

Trong nghiên cứu này, một kiến trúc bản sao số được đề xuất để xây dựng và phát triển mô hình bản sao số cho robot.

Từ kiến trúc hệ thống, nghiên cứu được chia thành những nhiệm vụ chính như sau: Thiết kế và chế tạo robot arm; Xây dựng bản sao số cho mô hình robot; Kết nối và truyền thông dữ liệu; Đánh giá và kết luận.

## 2. NỘI DUNG NGHIÊN CỨU

### 1. Tổng quan hệ thống

Hệ thống được chia làm 3 phần chính: Bản sao số, bộ xử lý trung tâm và mô hình robot thực. Tổng quan hệ thống được mô tả tại hình 1.

#### 1. Bản sao số

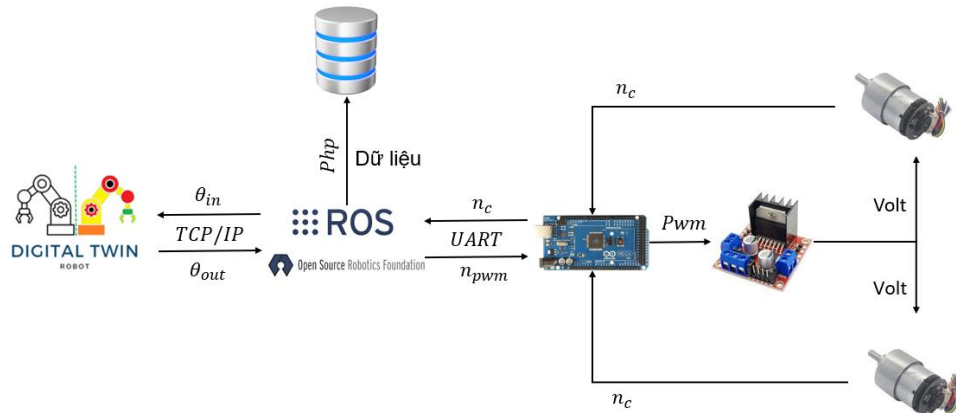
Bản sao số là một phần mềm mô phỏng trạng thái hình học của robot theo dữ liệu thời gian thực. Ngoài ra còn cung cấp cho người dùng các tính năng điều khiển và mô phỏng trước quá trình di chuyển của robot. Người dùng cũng có thể sử dụng chế độ 360 để dùng cho các thiết bị thực tế ảo.

#### 2. Bộ xử lý trung tâm

Đây là phần giúp chuẩn hóa dữ liệu, xử lý và tính toán. Ngoài ra còn hỗ trợ lưu trữ và sao lưu dữ liệu lên nền tảng cloud và server. Việc xử lý trên bộ xử lý trung tâm sẽ giúp mở rộng với nhiều loại robot một cách dễ dàng và tăng tốc độ xử lý của hệ thống.

#### 3. Mô hình robot vật lý

Mô hình robot vật lý bao gồm bộ điều khiển, cảm biến và các cơ cấu chấp hành. Đây là nơi thực hiện các yêu cầu từ người dùng.



**Hình 1.** Tổng quan hệ thống. Trong đó,  $\theta_{in}$  là giá trị góc quay robot ảo cần thực hiện;  $\theta_{out}$  là giá trị góc quay robot thực ảo cần thực hiện; ROS là bộ xử lý trung tâm có cài đặt Ros;  $n_{pwm}$  là số xung bộ điều khiển cần xuất;  $n_c$  là số xung encoder đếm được; Pwm là xung cấp cho mạch động lực. Volt (V) là điện áp cấp cho động cơ.

### 2. Xây dựng đối tượng vật lý

## 2.1. Yêu cầu thiết kế

Mục tiêu đề tài tập trung vào nghiên cứu phát triển bản sao số cho robot trong đó tập trung vào truyền thông dữ liệu 2 chiều theo thời gian thực. Vì vậy mô hình được xây dựng để chứng minh và tăng tính thuyết phục của đề tài. Do đó thiết kế chỉ yêu cầu một robot arm từ 2 tự do trở lên, có khả năng thu thập dữ liệu vị trí góc khớp, kết nối và truyền thông dữ liệu 2 chiều theo thời gian thực.

## 2.2. Lựa chọn động cơ và mạch động lực

Mục tiêu của nghiên cứu là phát triển bản sao số cho robot arm. Vì vậy để tiết kiệm thời gian và chi phí nên robot chỉ yêu cầu ít nhất 2 bậc tự do với khả năng phản hồi vị trí quay của các khớp. Động cơ DC Servo JGB37-520 12VDC với hộp giảm tốc 168:1 được sử dụng. Thông số về động cơ và encoder được trình bày tại bảng 1 và bảng 2.

**Bảng 1. Thông số kỹ thuật động cơ**

STT	Thông số	Giá trị
1	Điện áp định mức	12 VDC
2	Dòng tối đa	1.2 A
3	Tỷ số truyền	168:1
4	Tốc độ sau khi giảm tốc	35 rpm
5	Tốc độ tối đa khi có tải	28 rpm
6	Moment định mức	2.5 Kg.cm
7	Moment tối đa	10 Kg.cm
8	Đường kính trục	6 mm

**Bảng 1. Thông số Encoder**

STT	Thông số	Giá trị
1	Điện áp định mức	5 VDC
2	Số xung	11
3	Số kênh	2 kênh AB

## 2.3. Tính toán thiết kế cơ khí

### 1. Thiết kế robot arm

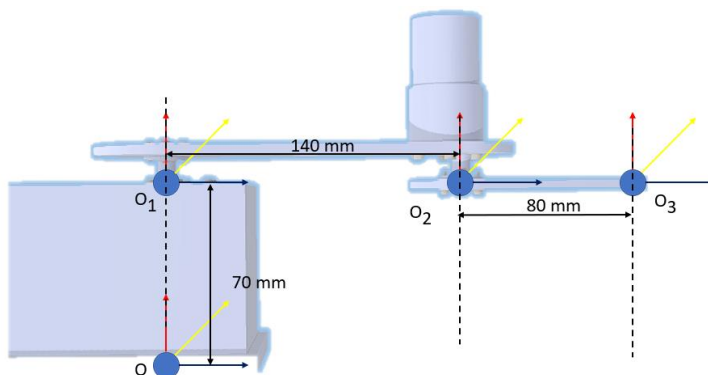
Phần mềm solidword được sử dụng để vẽ bản thiết kế. Vật liệu được sử dụng là mica trong với độ dày 5mm. Bản vẽ gồm 7 chi tiết được gia công bởi máy cắt laser (xem thêm tại phụ lục). Để được sử dụng là hộp điện 20x20 cm để chứa bộ điều khiển và mạch động lực. Ngoài ra còn được đục lỗ để cấp nguồn và cấp tín hiệu điều khiển từ ngoài. Thông số mô hình robot arm được trình bày tại bảng 2.

**Bảng 3. Bảng thông số robot arm**

STT	Thông số	Giá trị
1	Link 1	140 mm
2	Link 2	80 mm
3	Giới hạn khớp 1	[-90,90]
4	Giới hạn khớp 2	[-180,180]

### 2. Tính toán động học thuận

Để tính toán động học thuận (forward kinematics) cho một robot arm 2 bậc tự do, cần biết thông tin về chiều dài các khớp cơ và góc quay của chúng. Điều này cho phép chúng ta tính toán vị trí và hướng của tay robot dựa trên các góc quay của các khớp cơ. Tọa độ và khoảng cách được trình bày tại hình 2.



**Hình 2.** Hệ trục tọa độ và kích thước

**Bảng 2.** Bảng D-H của robot

Khớp	$\theta_i(\text{deg})$	$d_i(\text{mm})$	$a_i(\text{mm})$	$\alpha_i(\text{deg})$
1	$\theta_1$	140	70	0
2	$\theta_2$	80	0	0

Giả sử chúng ta có hai khớp cơ trong robot arm, và chúng ta gọi góc quay của khớp thứ nhất là  $\theta_1$  và góc quay của khớp thứ hai là  $\theta_2$ . Chúng ta cũng cần biết chiều dài các đoạn cơ của robot arm, gọi là  $l_1$  và  $l_2$  (Thông số  $l_1$  và  $l_2$  được lấy từ bản thiết kế). Để tính toán động học thuận, chúng ta sử dụng các công thức sau:

$$\begin{aligned} x &= l_1 * \cos(\theta_1) + l_2 * \cos(\theta_1 + \theta_2) \\ &= 160 * \cos(\theta_1) + 80 * \cos(\theta_1 + \theta_2) \quad (1) \end{aligned}$$

$$\begin{aligned} y &= l_1 * \sin(\theta_1) + l_2 * \sin(\theta_1 + \theta_2) \\ &= 160 * \sin(\theta_1) + 80 * \sin(\theta_1 + \theta_2) \quad (2) \end{aligned}$$

Do robot có kết cấu 2 bậc tự do như hình 3.1 nên tọa độ  $z = 70 \text{ mm}$ .

Trong đó,  $x$  và  $y$  là tọa độ của đầu tay robot. Với các giá trị của  $\theta_1$  và  $\theta_2$ , chúng ta có thể tính được vị trí của đầu tay robot trong không gian. Ngoài ra, chúng ta cũng có thể tính toán hướng của tay robot bằng cách sử dụng công thức sau:

$$\theta = \theta_1 + \theta_2 \quad (3)$$

Trong đó,  $\theta$  là góc quay của tay robot so với trục thẳng đứng. Điều này cho phép chúng ta xác định vị trí và hướng của tay robot dựa trên các góc quay của các khớp cơ và các thông số về chiều dài.

### 3. Tính toán động học nghịch

Để tính toán động học nghịch cho robot arm 2 bậc tự do bằng phương pháp Newton-Raphson, cần sử dụng các bước sau:

Xác định giá trị ban đầu cho góc quay  $\theta_1$  và  $\theta_2$ . Đây có thể là giá trị ban đầu gần với vị trí mục tiêu hoặc giá trị gần với giải pháp hiện tại.

Tính toán giá trị  $f1$  và  $f2$  dựa trên các phương trình sau:

$$f1(\theta1, \theta2) = l1 * \cos(\theta1) + l2 * \cos(\theta1 + \theta2) - x \quad (4)$$

$$f2(\theta1, \theta2) = l1 * \sin(\theta1) + l2 * \sin(\theta1 + \theta2) - y \quad (5)$$

Tính toán ma trận Jacobian  $J$  của hệ phương trình theo công thức sau:

$$J = \begin{bmatrix} \frac{\partial f1}{\partial \theta1} & \frac{\partial f1}{\partial \theta2} \\ \frac{\partial f2}{\partial \theta1} & \frac{\partial f2}{\partial \theta2} \end{bmatrix} \quad (6)$$

Trong đó,  $\partial f1/\partial \theta1$ ,  $\partial f1/\partial \theta2$ ,  $\partial f2/\partial \theta1$ , và  $\partial f2/\partial \theta2$  là các đạo hàm riêng của  $f1$  và  $f2$  theo  $\theta1$  và  $\theta2$ .

Tính toán ma trận delta  $\theta$  (đại lượng thay đổi góc quay) bằng cách giải hệ phương trình tuyến tính  $J * \text{delta } \theta = -[f1, f2]$ .

Cập nhật giá trị góc quay  $\theta1$  và  $\theta2$  bằng cách thực hiện phép cộng:

$$\theta1 = \theta1 + \text{delta } \theta1, \theta2 = \theta2 + \text{delta } \theta2 \quad (7)$$

Lặp lại các bước từ 2 đến 5 cho đến khi giá trị  $f1$  và  $f2$  đạt đến mức chấp nhận được (gần với 0) hoặc đạt đến số lần lặp tối đa được định trước.

Kiểm tra xem giá trị góc quay  $\theta1$  và  $\theta2$  có nằm trong giới hạn cho phép không. Nếu không, áp dụng ràng buộc để đảm bảo nằm trong giới hạn cho phép.

Quá trình này được lặp lại để tìm giá trị góc quay tốt nhất để đạt được vị trí và hướng mong muốn của tay robot.

### 3.2. Xây dựng bộ điều khiển

#### 3.2.1. Mạch điều khiển và mạch động lực

Mạch điều khiển gồm có một vi điều khiển và một mạch động lực để điều khiển động. Vi điều khiển được sử dụng là vi điều khiển arduino mega (bảng 5) và mạch động lực là mạch L298N (bảng 6). Sơ đồ chân và quản mạch điều khiển được trình bày tại hình bảng 7 và hình 3.

**Bảng 3.** Thông số vi điều khiển arduino mega

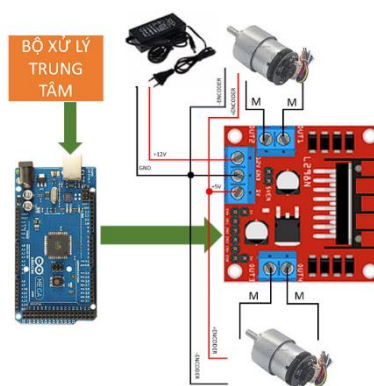
STT	Thông số	Giá trị
1	Vi điều khiển	Atmega2560
2	Điện áp hoạt động	5 VDC
3	Nguồn vào (Khuyến dùng)	7-12 VDC
4	Nguồn vào (Giới hạn)	6-20 VDC
5	Số chân Digital I/O	54 (15 chân pwm)
6	Số chân vào Analog	16
7	Dòng DC mỗi chân I/O	20 mA
8	Dòng DC cho chân 3.3 V	50 mA
9	Bộ nhớ Flash	256 KB
10	SRAM	8 KB
11	EEPROM	4 KB
12	Clock Speed	16 MHz

**Bảng 4.** Thông số mạch động lực L298N

STT	Thông số	Giá trị
1	Driver Chip	Double H-bridge L298N
2	Công suất tối đa	25 W
3	Nguồn động cơ tối đa	46 VDC
4	Nguồn dòng động cơ tối đa	2 A
5	Nguồn cấp	5-35 VDC
6	Dòng	2 A

**Bảng 6.** Bảng kết nối các thiết bị

STT	Arduino Mega	Mạch động lực	Động cơ & Encoder	Nguồn 12 VDC
1	Pin 2	-	Chân A Encoder 1	-
2	Pin 3	-	Chân A Encoder 2	-
3	Pin 4	-	Chân B Encoder 1	-
4	Pin 5	-	Chân B Encoder 2	-
5	Pin 6	IN1	-	-
6	Pin 7	IN2	-	-
7	Pin 8	IN3	-	-
8	Pin 9	IN4	-	-
9	Pin 10	PWM1	-	-
10	Pin 11	PWM2	-	-
11	GND	GND	GND Encoder 1 & 2	GND
12	-	12 VDC	-	VCC
13	-	5 VDC	VCC Encoder 1 & 2	-
14	-	M1	Chân + động cơ 1	-
15	-	M2	Chân - động cơ 1	-
16	-	M3	Chân + động cơ 2	-
17	-	M4	Chân - động cơ 2	-



**Hình 3.** Sơ đồ kết nối phần cứng mạch điều khiển

### 3.2.2. Bộ điều khiển PID

Thuật toán PID được sử dụng để phát triển bộ điều khiển động cơ. Tuy nhiên, bộ điều khiển PID cần có các thông số PID để tối ưu và giúp hệ thống hoạt động ổn định. Để tìm thông số PID, một hàm tìm thông số PID tự động đã được phát triển và tích hợp vào hệ thống. Hàm này được phát triển dựa trên thuật toán



Ziegler-Nichols. Việc tích hợp trực tiếp vào hệ thống giúp cho việc tối ưu và hiệu chỉnh thông số PID một cách nhanh chóng, linh hoạt, dễ dàng và tối ưu.

Do động cơ có encoder là 11 xung và tỷ số truyền là 168:1 nên 1 vòng của khớp quay là  $11 \cdot 168 = 1848$  xung. Vậy 1 độ tương ứng với  $1848/360 \approx 5$  xung. Do giá trị xung từ 0-255 và điện áp đầu vào của động cơ là 12V nên:

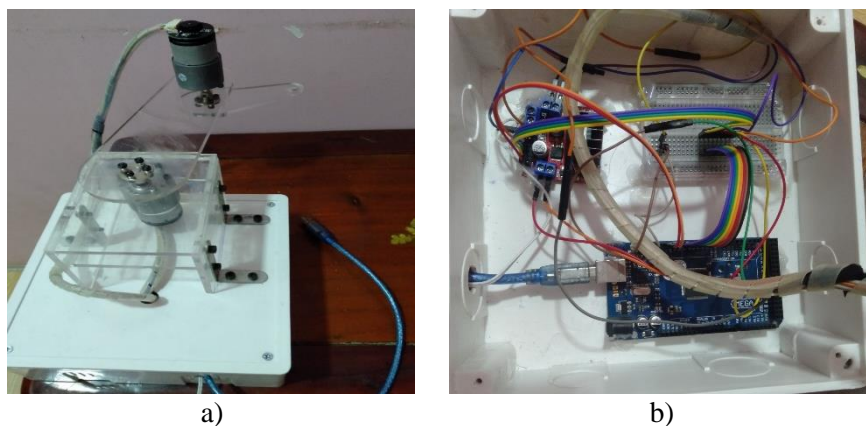
$$pwm(\text{xung điều khiển}) = \frac{u}{12} * 255 \quad (8)$$

Thông tin về điều khiển dựa trên  $u(V)$  được trình bày tại bảng 7.

**Bảng 7.** Chiều quay của động cơ và xung (Pwm) cấp cho mạch động lực theo các trường hợp của điện áp ( $u$ )

$u(V)$	Thông số	Giá trị
$u > 0$	Chiều	1
	Pwm	$u / 12 * 255$
$u < 0$	Chiều	-1
	Pwm	$-u / 12 * 255$
$u = 0$	Chiều	0
	Pwm	0

### 3.3. Kết quả



**Hình 4.** Tổng quan robot: a) Mô hình robot; b) Mạch điều khiển và mạch động lực robot

## 4. Xây dựng bản sao số

### 4.1. Xây dựng mô hình 3D robot và môi trường

Mô hình robot được xây dựng từ các chi tiết của bản thiết kế. Do phần mềm unity và solidword dựa trên nền tảng đồ họa khác nhau, vì vậy cần đưa mô hình về chuẩn đồ họa của unity. Có nhiều giải pháp cho vấn đề này và ở đây, phần mềm blender được sử dụng làm giải pháp trung gian giúp đưa mô hình về chuẩn đồ họa của unity. mô hình 3D được trình bày tại hình 6a.

### 4.2. Thiết kế giao diện

Một giao diện người dùng được xây dựng giúp người dùng giám sát, tương tác và điều khiển hệ thống với nhiều chức năng điều khiển khác nhau. Giao diện điều khiển gồm 2 chế độ xem chính:

#### 4. Chế độ giám sát và điều khiển:

Gồm 2 phần là phần giao diện điều khiển và phần giám sát (hình 6c và 6d). Các thông số về trạng thái robot được thể hiện lên màn hình. Ngoài ra trạng thái của mô hình 3D cũng thay đổi để người dùng dễ hình dung trạng thái của robot. Giao diện điều khiển giúp cho việc tương tác và điều khiển với robot từ xa thuận tiện và dễ dàng.

#### 5. Chế độ 360:

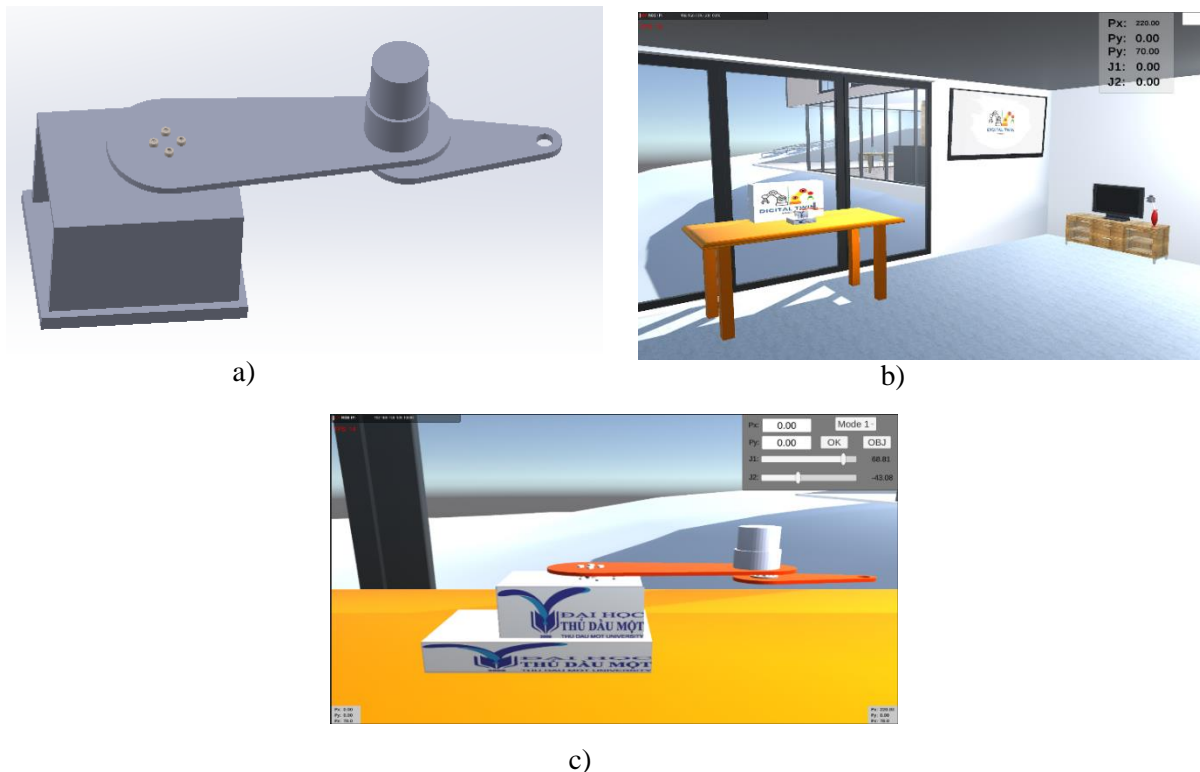
Tương tác và điều khiển góc quay của camera (hình 6b). Điều đó giúp tăng độ trải nghiệm và tính đắm chìm của hệ thống. Ngoài ra chế độ này áp dụng cho thực tế ảo.

Ngoài ra hệ thống còn thêm chế độ mô phỏng giúp người dùng xem trước được quá trình robot sẽ làm việc. Tuy nhiên do dữ liệu và tiết kiệm chi phí nên chức năng này so với thực tế còn cách xa nhau. Cần có thời gian để huấn luyện cho mô phỏng tiệm cận với thực tế.

#### 4.3. Xây dựng bộ điều khiển

Bộ điều khiển phần ảo có chức năng điều khiển mô hình 3D của robot. Trong bộ điều khiển có 2 chế độ gồm: (1) Điều khiển mô hình 3D robot bằng việc sử dụng slider; (2) Điều khiển mô hình 3D robot sử dụng mã lệnh điều khiển với các thông số tùy chỉnh. Tuy nhiên các thông số bị giới hạn trong phạm vi hoạt động của robot. Nếu vượt quá giới hạn thì sẽ robot sẽ không hoạt động và gửi cảnh báo đến người dùng.

#### 4.4. Kết quả



**Hình 5.** Bản sao số: a) Mô hình 3D robot; b) Chế độ 360; và c) Chế độ điều khiển giám sát.

#### 5. Kết nối và truyền thông dữ liệu

## 5.1. Thiết lập kết nối

Hệ thống có 2 kết nối chính: 1. Kết nối giữa bộ xử lý trung tâm và bộ điều khiển; Và 2. Kết nối giữa bộ xử lý trung tâm và bản sao số.

## 5.2. Kết nối giữa bộ xử lý trung tâm và bộ điều khiển:

Trong nghiên cứu này chúng tôi sử dụng thư viện roserial trong ros để kết nối với bộ điều khiển thông qua công Serial (UART).

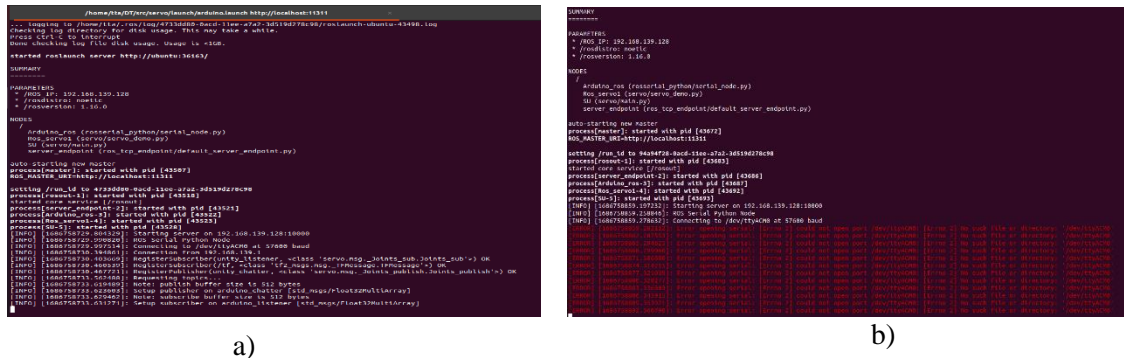
## 5.3. Kết nối giữa bộ xử lý trung tâm và bản sao số:

Để kết nối giữa bộ xử lý trung tâm và bản sao số, giao thức giao tiếp ROS được sử dụng để truyền và nhận dữ liệu giữa hai môi trường này. Phương pháp này cho phép tích hợp giữa bộ xử lý trung tâm và bản sao số trong cùng một mạng cục bộ, giúp truyền dữ liệu và nhận dữ liệu giữa bộ xử lý trung tâm và bản sao số.

## 5.4. Kiểm tra và thông báo lỗi kết nối

Việc kiểm tra và thông báo về các lỗi này là quan trọng để quản lý và xử lý các vấn đề giao tiếp của hệ thống. Bằng cách sử dụng các cơ chế kiểm tra, xử lý lỗi và thông báo, có thể cải thiện tính ổn định và khả năng phản ứng của ứng dụng kết nối này. Hình 7 trình bày thông báo khi kết nối thành công và hiển thị thông báo lỗi khi không kết nối.

## 5.5. Kết quả



Hình 6. Thông báo kết nối: a) Kết nối thành công; b) Kết nối thất bại

## 6. Xử lý và lưu trữ dữ liệu

### 6.1. Xây dựng bộ xử lý trung tâm

Bộ xử lý trung tâm được xây dựng và phát triển trên một máy tính, vì xử lý có cài đặt ROS. Phần mềm xử lý được đóng gói thành giúp cho người dùng dễ dàng sử dụng với các câu lệnh đơn giản.

#### 1. Bộ xử lý trung tâm xử lý những tác vụ sau:

Kết nối và truyền thông dữ liệu 2 chiều theo thời gian thực giữa mô hình vật lý và mô hình số.

Xử lý lệnh điều khiển từ người dùng.

Tính toán động học robot arm.

Tính toán và gửi tín hiệu điều khiển bộ điều khiển.

Xử lý lỗi phát sinh của hệ thống.

Lưu trữ dữ liệu cục bộ và sao lưu dữ liệu trên cloud.

**Bảng 8. Mã lệnh của robot**

STT	Mã lệnh	Mô tả
1	1	Điều khiển góc khớp từng động cơ
2	2	Điều khiển quỹ đạo robot
3	10	Mô phỏng quỹ đạo robot

## 2. Ước tính vị trí của góc quay động cơ

Việc ước tính vị trí động cơ giúp giảm sai số giữa mô hình thực và ảo ảnh hưởng bởi độ trễ của hệ thống. Dữ liệu đầu vào là vị trí góc tại thời điểm trước đó và vị trí góc tại thời điểm hiện tại, một điểm trong tương lai được tính toán. Thuật toán được trình bày như sau:

Đầu vào	:	$P_{cur}, P_{pre}, v_{pre}, t_{cur}, t_{pre}, t_{fur}$
Đầu ra	:	$P_{fur}, P_{pre}, v_{pre}$
Xử lý	:	$v = \frac{P_{cur} - P_{pre}}{t_{cur} - t_{pre}}$ $a = \frac{v_{cur} - v_{pre}}{t_{cur} - t_{pre}}$ $P_{fur} = P_{cur} + v * (t_{fur} - t_{cur}) + \frac{a * (t_{fur} - t_{cur})^2}{2}$ $P_{pre} = P_{cur}$ $v_{pre} = v$

Với:  $P_{cur}$  là vị trí góc quay hiện tại của khớp.  $P_{pre}$  là vị trí góc quay trước đó của khớp.  $v_{pre}$  là vận tốc tại thời điểm trước đó,  $t_{cur}$  là thời gian tại lúc lấy  $P_{cur}$ .  $t_{pre}$  là thời gian trước đó,  $t_{fur}$  là thời gian mà góc quay của khớp ảo được cập nhật.  $P_{fur}$  là giá trị góc quay tại thời điểm  $t_{fur}$ .  $v$  là giá trị vận tốc tức thời tại thời điểm  $t_{cur}$ .  $a$  là gia tốc tức thời tại thời điểm  $t_{cur}$ .

## 3. Xử lý lỗi hệ thống

Khi chưa có kết nối hệ thống sẽ cố gắng kết nối với bộ điều khiển và bản sao số. Khi nào kết nối thành công thì sẽ kiểm tra các kết nối bên trong hệ thống đã đảm bảo chưa và hiển thị ra thông báo. Nếu chưa thì sẽ hiển thị thông báo lỗi kết nối.

Ngoài ra hệ thống cũng kiểm tra kết nối với bản sao số và bộ điều khiển. Nếu trong khoảng 10s không có dữ liệu nào được truyền về thì hệ thống sẽ thông báo lỗi mất kết nối và dừng hoạt động.

Trong quá trình hoạt động, nếu động cơ được lệnh di chuyển mà trong 20s không có sự thay đổi thì hệ thống sẽ thông báo lỗi và ngắt hệ thống. Điều này đảm bảo an toàn cho hệ thống khi có sự cố về động cơ xảy ra.

## 6.2. Lưu trữ dữ liệu

Để lưu trữ dữ liệu cục bộ có thể sử dụng các phương pháp như lưu trữ trong tệp văn bản, lưu trữ trong cơ sở dữ liệu, hoặc sử dụng các cấu trúc dữ liệu trong ngôn ngữ lập trình đang sử dụng. Trong nghiên cứu này sử dụng phương pháp kết hợp phương pháp lưu trữ cục bộ và sao lưu trên nền tảng đám mây. Dữ liệu được lưu vào các tệp csv, sau một khoảng thời gian nhất định thì dữ liệu được sao lưu trên nền tảng cloud. Nền tảng cloud được xây dựng dựa trên MongoDB.

## 6.3. Kiểm tra và thông báo lỗi hệ thống

Các lỗi khi vận hành cũng được chuẩn hóa thành các Error Code (EC) tại bảng 10. Điều đó giúp việc xác định và giải quyết lỗi một cách nhanh chóng và dễ dàng.

**Bảng 10.** Mã lỗi của hệ thống

STT	EC	Loại lỗi
1	000	Không có kết nối với mạch điều khiển
2	001	Không có kết nối với bản sao số
3	002	Không tìm thấy giá trị quỹ đạo
4	003	Lỗi động cơ 1 không hoạt động
5	004	Lỗi động cơ 2 không hoạt động
6	005	Lỗi 2 động cơ không hoạt động
7	006	Lỗi truyền dữ liệu hệ thống

## 6.4. Kết quả

```

[INFO] [1685762390.377469]: data: 0.0, 3.5064935064935066
[INFO] [1685762390.378156]: Processing data: (1.0, 0.0, 7.631099224090576, 0.0, 0.0), [0.0, 3.5064935064935066, 5432.0]
[INFO] [1685762390.378819]: pwm1: 0.0, pwm2: 8.764787280499279
[INFO] [1685762390.379812]: PWM data: 0.0, 8.764787280499279
[INFO] [1685762390.380465]: PWM data: 0.0, 8.764787280499279
[INFO] [1685762390.381258]: data: 0.0, 3.5064935064935066
[INFO] [1685762390.381947]: Processing data: (1.0, 0.0, 7.631099224090576, 0.0, 0.0), [0.0, 3.5064935064935066, 5440.0]
[INFO] [1685762390.382601]: pwm1: 0.0, pwm2: 8.764787280499279
[INFO] [1685762390.383525]: PWM data: 0.0, 8.764787280499279
[INFO] [1685762390.384379]: PWM data: 0.0, 8.764787280499279
[INFO] [1685762390.389946]: data: 0.0, 3.5064935064935066
[INFO] [1685762390.389956]: Processing data: (1.0, 0.0, 7.631099224090576, 0.0, 0.0), [0.0, 3.5064935064935066, 5212.0]
[INFO] [1685762390.391024]: pwm1: 0.0, pwm2: 8.764787280499279
[INFO] [1685762390.392138]: PWM data: 0.0, 8.764787280499279
[INFO] [1685762390.394422]: PWM data: 0.0, 8.764787280499279
[INFO] [1685762390.396352]: data: 0.0, 3.5064935064935066
[INFO] [1685762390.397325]: Processing data: (1.0, 0.0, 7.631099224090576, 0.0, 0.0), [0.0, 3.5064935064935066, 5416.0]
[INFO] [1685762390.398349]: pwm1: 0.0, pwm2: 8.764787280499279
[INFO] [1685762390.401533]: PWM data: 0.0, 8.764787280499279
[INFO] [1685762390.402486]: PWM data: 0.0, 8.764787280499279
[INFO] [1685762390.403352]: data: 0.0, 3.5064935064935066
[INFO] [1685762390.404318]: Processing data: (1.0, 0.0, 7.631099224090576, 0.0, 0.0), [0.0, 3.5064935064935066, 5076.0]
[INFO] [1685762390.405046]: pwm1: 0.0, pwm2: 8.764787280499279
[INFO] [1685762390.405962]: PWM data: 0.0, 8.764787280499279
[INFO] [1685762390.407017]: PWM data: 0.0, 8.764787280499279
[INFO] [1685762390.408296]: data: 0.0, 3.5064935064935066
[INFO] [1685762390.409683]: Processing data: (1.0, 0.0, 7.631099224090576, 0.0, 0.0), [0.0, 3.5064935064935066, 5428.0]
[INFO] [1685762390.410083]: pwm1: 0.0, pwm2: 8.764787280499279
[INFO] [1685762390.410855]: PWM data: 0.0, 8.764787280499279
[INFO] [1685762390.411849]: PWM data: 0.0, 8.764787280499279
[INFO] [1685762390.412827]: data: 0.0, 3.5064935064935066

```

**Hình 7.** Thông báo hệ thống trong quá trình hoạt động của bộ xử lý trung tâm.

## 7. Đánh giá và thảo luận

### 7.1. Phương pháp đánh giá

#### 7.1.1. Bộ xử lý tính toán động học robot

Một công cụ giúp hỗ trợ kiểm tra và đánh giá việc tính toán động học robot được sử dụng giúp hỗ trợ kiểm tra và đánh giá việc tính toán của động học của hệ thống. Công cụ cho phép người dùng nhập đầu vào là vị trí góc của từng khớp và kết quả trả về là vị trí điểm cuối tay máy tương ứng sau đó vị trí điểm cuối này lại được đưa vào tính toán giúp kiểm tra xem góc tương ứng là bao nhiêu và so sánh với dữ liệu đầu phải xem có trùng khớp không.

### 7.1.2 Độ trễ hệ thống

Dữ liệu được thu thập và phân tích về mặt thời gian. Dựa trên dữ liệu thời gian thu thập được sẽ vẽ thành các biểu đồ hỗ trợ việc phân tích. Thuật toán tính toán độ trễ hệ thống được trình bày như sau:

Đầu vào	:	$t_{pre}, t_{cur}, t=0$
Đầu ra	:	$\Delta_{avd}, \Delta_t$
Vòng lặp	:	$\Delta_t = t_{cur} - t_{pre}$ $i = i + 1$ $t = t + \Delta_t$ $\Delta_{avd} = \frac{t}{i}$

Trong đó:  $t_{pre}$  là thời gian tại thời điểm trước đó;  $t_{cur}$  là thời gian tại thời điểm xét;  $\Delta_t$  là độ trễ;  $\Delta_{avd}$  là độ trễ trung bình;  $i$  là số lần dữ liệu được truyền và nhận;  $t$  là tổng độ trễ tương ứng với số lần  $i$ .

### 7.1.3. Sai số hệ thống

Dữ liệu được thu thập và phân tích về mặt thời gian và giá trị tại thời điểm dữ liệu được nhận. Dựa trên dữ liệu đó sẽ được xử lý và chuẩn hóa sau đó vẽ thành biểu đồ để đánh giá sai số.

## 7.2. Đánh giá và thảo luận

### 7.2.1. Đánh giá động học robot

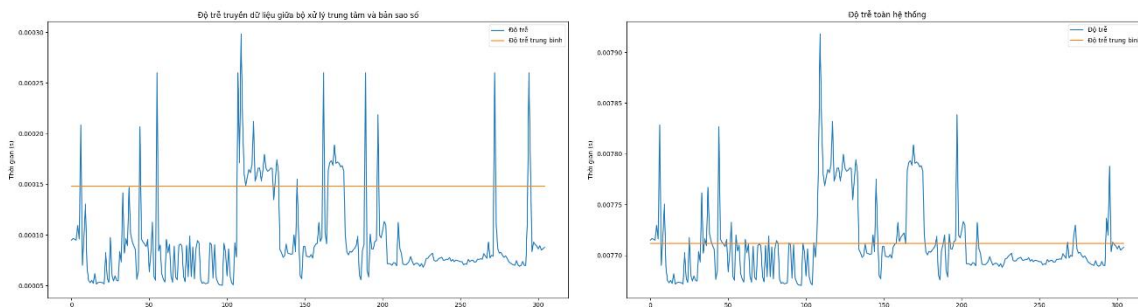
Bảng 11 cho thấy sự so sánh giữa việc tính toán giữa bộ xử lý trung tâm và công cụ hỗ trợ là gần giống nhau. Điều đó chứng tỏ việc tính toán động học robot của bộ xử lý trung tâm được chấp nhận.

**Bảng 11.** So sánh động học giữa bộ xử lý trung tâm và công cụ

STT	Bộ xử lý trung tâm		Công cụ		Vị trí điểm cuối tay máy
	Bộ giá trị góc 1	Bộ giá trị góc 2	Bộ giá trị góc 1	Bộ giá trị góc 2	
1	[50, 50]	[85.5, -50]	[50, 50]	[85.5, -50]	[76.1, 186.03, 70]
2	[49.5, 50]	[85, -50]	[49.49, 50]	[85.01, -50]	[77.73, 185.35, 70]
3	[9.99, 10]	[17.27, -10]	[10, 10]	[17.26, -9.99]	[213.05, 51.67, 70]
4	[-76, 89]	[-17, -89]	[-75.99, -89]	[-16.99, -89]	[111.82, -117.85, 70]
5	[-46.49, 88]	[12, -88]	[-47, 87.99]	[11.99, -88]	[156.29, -48.52, 70]
6	[58.28, 12]	[67, -12]	[58.28, 12.01]	[67, -12.01]	[100.59, 194.4, 70]

### 7.2.2. Đánh giá độ trễ hệ thống

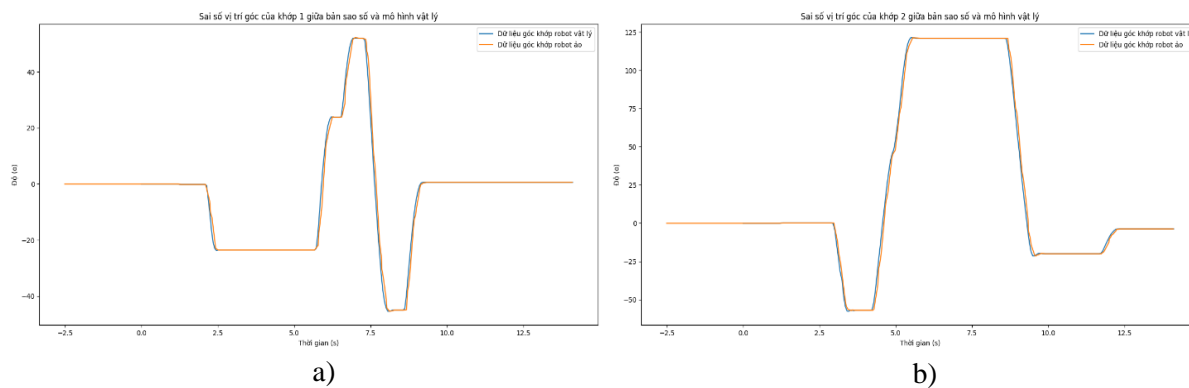
Do độ trễ xử lý nên đã được điều chỉnh khoảng 50 Hz với độ trễ khoảng 20ms để tránh tràn dữ liệu. Độ trễ để xử lý khi nhận được tín hiệu là khoảng 5 ms. Độ trễ khi bộ xử lý gửi dữ liệu đến khi bản sao số nhận được dữ liệu là khoảng 0.15 ms. Thời gian bản sao số xử lý khi nhận được tín hiệu từ bộ xử lý trung tâm là 2.57 ms. Thời gian độ trễ để một dữ liệu từ thu thập đến lúc hoàn thành xử lý trên toàn bộ hệ thống khoảng 7.72 ms. Độ trễ hệ thống được thể hiện tại hình 9.



**Hình 8.** Độ trễ hệ thống. a) Độ trễ xử lý dữ liệu của bộ xử lý trung tâm trong khoảng 0.01-1.9 ms với độ trễ trung bình khoảng 0.15 ms; b) Độ trễ trên toàn hệ thống với độ trễ trung bình 7.72 ms.

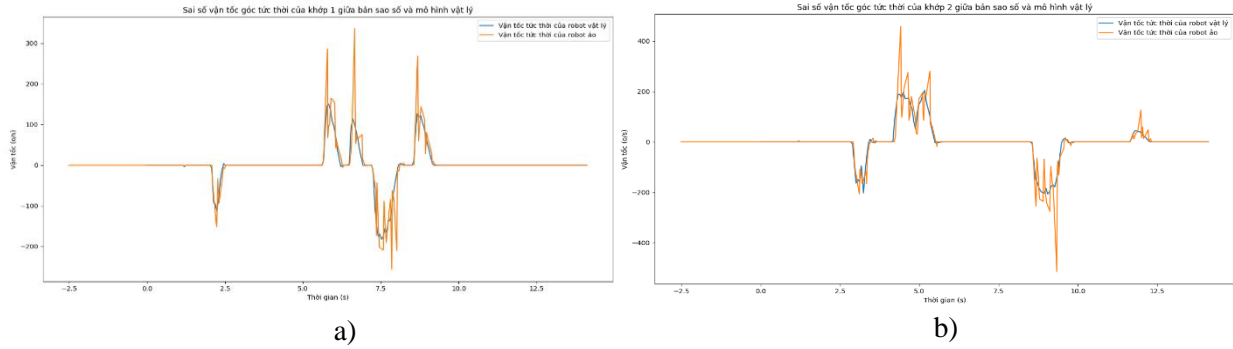
### 7.2.3. Đánh giá sai số giữa mô hình vật lý và mô hình ảo

Sai số giữa mô hình thực vào ảo được thể hiện tại hình 10, 11 khi lấy robot thực làm chuẩn. Có thể thấy rằng sai số giữa mô hình thực và ảo là khá nhỏ và gần như là trùng nhau khi sử dụng robot thực điều khiển robot ảo. Ngược lại thì lấy robot ảo làm chuẩn thì sai số giữa mô hình ảo và mô hình thực tăng lên thể hiện tại hình 12. Vấn đề này là do việc tính toán và điều khiển con ảo khác so với chạy thực tế ở con thực. Điều này có thể khắc phục bằng việc huấn luyện robot ảo từ dữ liệu robot thực.

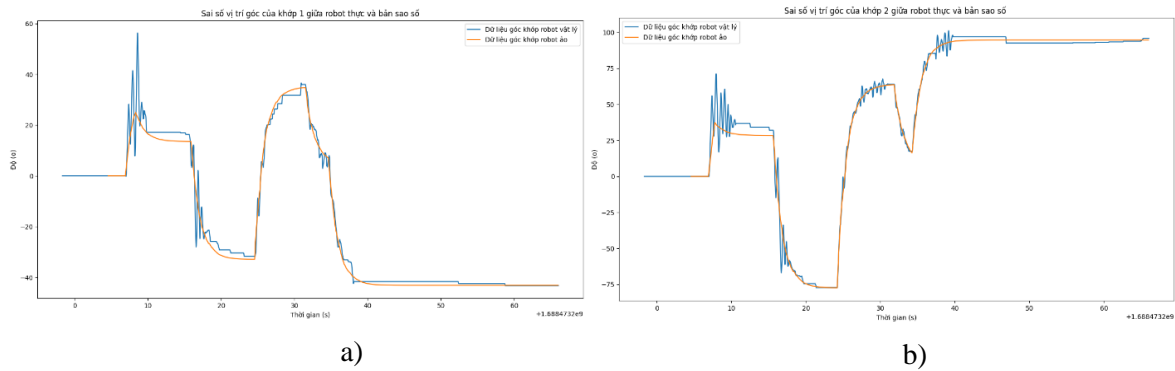


**Hình 92.** Sai số vị trí góc của khớp giữa mô hình vật lý và mô hình bản sao số. a) Sai số vị trí góc tại khớp 1. b) Sai số vị trí góc tại khớp 2.





**Hình 30.** Sai số vận tốc góc tức thời giữa tính toán mô phỏng và thực tế. a) Sai số vận tốc góc tức thời tại khớp 1. b) Sai số vận tốc góc tức thời tại khớp 2.



**Hình 11.** Sai số vận tốc góc tức thời giữa tính toán mô phỏng và thực tế. a) Sai số vận tốc góc tức thời tại khớp 1. b) Sai số vận tốc góc tức thời tại khớp 2.

## 8. Kết luận và hướng phát triển

### 8.1. Kết luận

Nghiên cứu xây dựng được mô hình bản sao số cho robot với các mục tiêu đề ra với độ trễ khoảng 7.72 ms và sai số khá thấp. Tuy nhiên hệ thống chỉ mới thu thập dữ liệu từ một loại cảm biến, chưa đề cập đến các vấn đề dự báo, xử lý va chạm hay huấn luyện robot ảo. Ngoài ra việc sử dụng tính toán để tìm vị trí điểm cuối tay máy hay sự phụ thuộc vào encoder cũng là chưa đủ để đánh giá khách quan sai số giữa robot thực và ảo. Hệ thống cần được phát triển và cải tiến nhiều hơn nữa. Cần lưu ý khi chọn tần số lấy mẫu cảm biến cho bộ điều khiển trung tâm phải lớn hơn độ trễ của hệ thống để tránh tràn dữ liệu.

### 8.2. Hướng phát triển

Cải thiện các mô hình vật lý cũng như mô hình bản sao số để làm cho bản sao số giống mô hình thực hơn và cung cấp thông tin dự đoán bằng cách tích hợp các tính toán đa vật lý, phương pháp điều khiển hiện đại và áp dụng các kỹ thuật Máy học với nhiều dữ liệu thời gian thực hơn.

Ứng dụng VR/AR cho tương tác của hệ thống.