|  |  |
| --- | --- |
|  | BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HỒ CHÍ MINH** |
| **BÁO CÁO MÔN HỌC**  **ROBOT DI ĐỘNG**  **GIẢI THUẬT ĐIỀU KHIỂN**  **ROBOT DI ĐỘNG UKMARSBOT**  Ngành: Robot và trí tuệ nhân tạo  Lớp: 22DRTA1  **GIẢNG VIÊN : PGS. TS. NGÔ HÀ QUANG THỊNH**   |  |  |  | | --- | --- | --- | | **Sinh viên thực hiện:** | **MSSV:** | **Lớp:** | | Nguyễn Văn Đạt | 2286300010 | 22DRTA1 | | Huỳnh Long | 2286300028 | 22DRTA1 | | Nguyễn Chấn Huy | 2286300020 | 22DRTA1 |   *TP. Hồ Chí Minh, ngày 9 tháng 1 năm 2024* | |

|  |  |
| --- | --- |
|  | BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HỒ CHÍ MINH** |
| **BÁO CÁO MÔN HỌC**  **ROBOT DI ĐỘNG**  **GIẢI THUẬT ĐIỀU KHIỂN**  **ROBOT DI ĐỘNG UKMARSBOT**  Ngành: Robot và trí tuệ nhân tạo  Lớp: 22DRTA1  **GIẢNG VIÊN : PGS. TS. NGÔ HÀ QUANG THỊNH**   |  |  |  | | --- | --- | --- | | **Sinh viên thực hiện:** | **MSSV:** | **Lớp:** | | Nguyễn Văn Đạt | 2286300010 | 22DRTA1 | | Huỳnh Long | 2286300028 | 22DRTA1 | | Nguyễn Chấn Huy | 2286300020 | 22DRTA1 |   *TP. Hồ Chí Minh, ngày 9 tháng 1 năm 2024* | |

*BM01/HDCV01/ĐAMH/VKT*

**Đề số:**…….

**VIỆN KỸ THUẬT HUTECH**

**PHIẾU GIAO ĐỀ TÀI**

**TÊN MÔN HỌC: ROBOT DI ĐỘNG**

**NGÀNH: ROBOT VÀ TRÍ TUỆ NHÂN TẠO**

1. **Họ và tên sinh viên/ nhóm sinh viên được giao đề tài** (sĩ số trong nhóm: 3):

(1) Nguyễn Văn Đạt MSSV: 2286300010 Lớp: 22DRTA1

(2) Huỳnh Long MSSV: 2286300028 Lớp: 22DRTA1

(3) Nguyễn Chấn Huy MSSV: 2286300020 Lớp: 22DRTA1

1. **Tên đề tài**: Giải thuật điều khiển robot di động Ukmarsbot
2. **Các dữ liệu ban đầu**:

* Kiến thức nền tảng: cơ sở về động lực học robot di động, điều khiển tuyến tính và vòng kín, lập trình Arduino, mô phỏng MATLAB/Simulink, xử lý tín hiệu cảm biến.
* Phần cứng tham khảo: Arduino Nano, module Bluetooth HC-05, driver động cơ TB6612FNG, động cơ DC GA12 N20 có encoder cho 2 bánh vi sai, khung xe UKMARSbot, nguồn pin, cảm biến led hồng ngoại IR.
* Phần mềm tham khảo: Arduino IDE, MATLAB/Simulink, Android Studio / MIT App Inventor (để làm app điều khiển), công cụ vẽ sơ đồ mạch/PCB.
* Tài liệu tham khảo: sách/luận văn về động lực học robot di động 2-bánh vi sai, bài báo về điều khiển PID/fuzzy, hướng dẫn HC-05 & Arduino.

1. **Nội dung nhiệm vụ**:

Nghiên cứu và xây dựng mô hình động học, động lực học cho robot di động UKMARSbot.

Mô phỏng hoạt động robot trên MATLAB/Simulink, thể hiện quỹ đạo, vận tốc và góc quay.

Thiết kế và lập trình hệ thống điều khiển robot qua Bluetooth (HC-05) bằng Arduino Nano.

Xây dựng ứng dụng Android để điều khiển robot thực hiện các lệnh cơ bản: Tiến, Lùi, Rẽ Trái, Rẽ Phải, Dừng.

So sánh và đánh giá kết quả mô phỏng với thực nghiệm, phân tích sai số và hiệu chỉnh.

1. **Kết quả tối thiểu phải có**:

Xây dựng và mô phỏng thành công mô hình động học, động lực học của robot UKMARSbot trên MATLAB/Simulink.

Robot thực tế điều khiển được qua Bluetooth bằng ứng dụng Android, thực hiện chính xác các lệnh Tiến, Lùi, Rẽ Trái, Rẽ Phải, Dừng.

Hệ thống hoạt động ổn định, phản hồi đúng lệnh điều khiển và thể hiện được chuyển động cơ bản.

Có báo cáo hoàn chỉnh gồm: mô tả mô hình, kết quả mô phỏng, triển khai phần cứng, kết quả thực nghiệm, so sánh và phân tích sai số.

Video hoặc hình ảnh minh họa robot vận hành và mô phỏng tương ứng.

Ngày giao đề tài: ……./……../……… Ngày nộp báo cáo: ……./……../………

|  |  |
| --- | --- |
| **Sinh viên thực hiện**  *(Ký và ghi rõ họ tên các thành viên)*  **Nguyễn Văn Đạt**  **Nguyễn Chấn Huy**  **Huỳnh Long** | *TP. HCM, ngày … tháng … năm ……….*  **Giảng viên hướng dẫn**  *(Ký và ghi rõ họ tên)* |

**MỤC LỤC**

[I. Giới thiệu đề tài 1](#_Toc212157004)

[1.1. Lý do chọn đề tài 1](#_Toc212157005)

[1.2. Mục tiêu nghiên cứu 1](#_Toc212157006)

[1.3. Phạm vi và giới hạn 1](#_Toc212157007)

[II. Cơ sở lý thuyết và tổng quan 3](#_Toc212157008)

[2.1. Giới thiệu về robot UKMarsbot 3](#_Toc212157009)

[2.2. Khái niệm điều khiển robot di động 3](#_Toc212157010)

[2.3. Kết nối và truyền thông qua Bluetooth 4](#_Toc212157011)

[2.4. Ứng dụng di động trong điều khiển robot 4](#_Toc212157012)

[III. Thiết kế hệ thống 6](#_Toc212157013)

[3.1. Kiến trúc tổng thể hệ thống 6](#_Toc212157014)

[3.2. Phần cứng sử dụng 7](#_Toc212157015)

[3.3. Phần mềm trên robot 8](#_Toc212157016)

[3.4. Ứng dụng điều khiển trên điện thoại 9](#_Toc212157017)

[IV. Mô phỏng phương trình động lực học của robot 10](#_Toc212157018)

[4.1. Xây dựng mô hình động học và động lực học của robot 10](#_Toc212157019)

[4.2. Thiết lập phương trình chuyển động 11](#_Toc212157020)

[4.3. Mô phỏng robot trên MATLAB/Simulink 12](#_Toc212157021)

[4.4. Mô phỏng chuyển động cùng với quỹ đạo, vận tốc, sai số, góc quay 12](#_Toc212157022)

[V. Thuật toán và cơ chế điều khiển 14](#_Toc212157023)

[5.1. Giải thuật điều khiển từ xa qua Bluetooth 14](#_Toc212157024)

[5.2. Giải thuật xử lý lệnh từ ứng dụng 16](#_Toc212157025)

[5.3. Các chế độ điều khiển 16](#_Toc212157026)

[VI. Thực nghiệm và kết quả 18](#_Toc212157027)

[6.1. Kết quả triển khai hệ thống 18](#_Toc212157028)

[6.2. Kết quả mô phỏng động lực học trên MATLAB 19](#_Toc212157029)

[6.3. Đánh giá mô phỏng 20](#_Toc212157030)

[VII. Kết luận và hướng phát triển 21](#_Toc212157031)

[7.1. Kết luận chung 21](#_Toc212157032)

[7.2. Định hướng cải tiến (tích hợp cảm biến, bán tự động, kết nối WiFi, AI…) 21](#_Toc212157033)

[Tài liệu tham khảo 22](#_Toc212157034)

**Mục lục hình ảnh**

[Hình 3.1. Kiến trúc tổng thể hệ thống Ukmarsbot 4](#_Toc212160665)

[Hình 3.2. Nguồn pin cấp 5](#_Toc212160666)

[Hình 3.3. Thiết kế giao diện cho App điều khiển 6](#_Toc212160667)

[Hình 3.4. Block giải thuật cho App điều khiển 7](#_Toc212160668)

[Hình 5.1 Sơ đồ khối mô tả luồng hoạt động của hệ thống 13](#_Toc212160669)

Hình 5.2. Sơ đồ khối miêu tả giải thuật điều khiển.......................................................13

[Hình 6.1. Mô hình hóa hệ thống 16](#_Toc212160670)

# I. Giới thiệu đề tài

## 1.1. Lý do chọn đề tài

Hiện nay, robot di động là một trong những hướng nghiên cứu được nhiều sinh viên quan tâm vì vừa mang tính ứng dụng cao, vừa giúp hiểu rõ hơn về các kiến thức điều khiển, lập trình và cơ điện tử. Trong số các nền tảng robot giáo dục, **UKMarsbot** nổi bật bởi cấu trúc đơn giản, dễ chế tạo và chi phí thấp. Các linh kiện của robot như động cơ DC, cảm biến, mạch điều khiển và module Bluetooth đều có thể dễ dàng tìm mua tại thị trường Việt Nam, giúp sinh viên thuận tiện trong việc lắp ráp, thử nghiệm và mở rộng đề tài.

Từ đặc điểm đó, nhóm quyết định chọn đề tài **“Giải thuật điều khiển robot di động UKMarsbot”** với mong muốn tìm hiểu sâu hơn về cách robot di chuyển và phản hồi khi điều khiển. Đề tài hướng đến việc **xây dựng và mô phỏng phương trình động học của robot bằng MATLAB**, sau đó áp dụng vào **điều khiển thực tế** để quan sát và đánh giá kết quả. Đây là một hướng tiếp cận vừa phù hợp với khả năng của sinh viên, vừa giúp củng cố kiến thức về lập trình, xử lý tín hiệu và điều khiển chuyển động của robot, tạo nền tảng cho các nghiên cứu hoặc ứng dụng phức tạp hơn trong tương lai.

## 1.2. Mục tiêu nghiên cứu

Mục tiêu chính của đề tài là tìm hiểu nguyên lý hoạt động, cấu tạo và đặc tính chuyển động của robot di động UKMarsbot thông qua việc xây dựng và mô phỏng phương trình động học trên phần mềm MATLAB. Thông qua quá trình mô phỏng, có thể quan sát, phân tích quỹ đạo và phản ứng chuyển động của robot, từ đó xác định các tham số điều khiển phù hợp cho mô hình thực tế.

Bên cạnh đó, đề tài hướng đến việc ứng dụng mô hình động học vào điều khiển thực tế nhằm kiểm chứng độ chính xác của kết quả mô phỏng. Giải thuật điều khiển được lập trình để robot UKMarsbot di chuyển theo các lệnh được tính toán từ mô hình MATLAB, qua đó đánh giá mức độ tương đồng giữa mô phỏng và thực nghiệm, cũng như khả năng đáp ứng và độ ổn định của hệ thống điều khiển.

## 1.3. Phạm vi và giới hạn

Đề tài tập trung nghiên cứu và phát triển mô hình động học của robot di động UKMarsbot dựa trên nguyên lý điều khiển vi sai. Phạm vi thực hiện bao gồm việc xây dựng phương trình động học, mô phỏng chuyển động bằng phần mềm MATLAB, và triển khai điều khiển thực tế trên robot thông qua các giải thuật cơ bản. Các yếu tố cơ khí, cảm biến và truyền thông được sử dụng ở mức cơ bản nhằm phục vụ cho quá trình mô phỏng và kiểm chứng mô hình.

Giới hạn của đề tài nằm ở việc chưa xem xét các yếu tố động lực học, nhiễu thực tế từ môi trường, cũng như chưa tích hợp các thuật toán điều khiển nâng cao như PID, fuzzy hay điều khiển thích nghi. Ngoài ra, phạm vi thử nghiệm chủ yếu được thực hiện trong môi trường mô phỏng và trên mô hình robot quy mô nhỏ, chưa áp dụng trong các hệ thống tự hành phức tạp hoặc môi trường thực tế có nhiều biến động.

# II. Cơ sở lý thuyết và tổng quan

## 2.1. Giới thiệu về robot UKMarsbot

Robot **UKMarsbot** là một nền tảng robot di động nhỏ gọn, được thiết kế phục vụ cho mục đích học tập, nghiên cứu và thử nghiệm các thuật toán điều khiển. Robot được phát triển dựa trên cấu trúc **vi sai (differential drive)**, sử dụng hai bánh chủ động và một bánh đỡ tự do, cho phép di chuyển linh hoạt trên mặt phẳng. Với thiết kế cơ khí đơn giản, chi phí thấp và khả năng mở rộng cao, UKMarsbot được sử dụng rộng rãi trong cộng đồng nghiên cứu robot và đặc biệt phổ biến trong các cuộc thi **Micromouse** – nơi robot phải tự động di chuyển và tìm đường trong mê cung.

Về phần cứng, UKMarsbot thường được trang bị **vi điều khiển ATmega328P** hoặc tương đương, hai **động cơ DC có gắn encoder** để đo tốc độ và khoảng cách di chuyển, cùng các **cảm biến hồng ngoại hoặc siêu âm** để phát hiện vật cản. Ngoài ra, robot có thể tích hợp **module Bluetooth hoặc RF** để thực hiện truyền thông không dây, cho phép điều khiển hoặc thu thập dữ liệu từ máy tính hay thiết bị di động. Các linh kiện này đều dễ dàng tìm thấy trên thị trường Việt Nam, giúp việc chế tạo, bảo trì và mở rộng hệ thống trở nên thuận tiện.

## 2.2. Khái niệm điều khiển robot di động

Robot di động (Mobile Robot) là loại robot có khả năng di chuyển trong không gian và thực hiện nhiệm vụ mà không cố định tại một vị trí. Khác với robot công nghiệp gắn cố định, robot di động có thể tự vận hành trong môi trường phức tạp, nhờ đó được ứng dụng rộng rãi trong các lĩnh vực như logistics, giám sát, khám phá, và nghiên cứu khoa học.

Điều khiển robot di động là quá trình **tác động lên hệ thống truyền động** của robot thông qua các tín hiệu điều khiển, nhằm đạt được trạng thái chuyển động mong muốn về vị trí, vận tốc hoặc hướng di chuyển. Quá trình này thường dựa trên mô hình động học của robot, trong đó các thông số như vận tốc bánh xe, góc quay, hoặc vị trí được sử dụng để xác định quỹ đạo và điều chỉnh chuyển động.

Trong thực tế, điều khiển robot di động có thể được phân thành hai nhóm chính: **điều khiển bằng tay (manual control)** và **điều khiển tự động (autonomous control)**. Ở điều khiển bằng tay, robot nhận lệnh trực tiếp từ người điều khiển thông qua các thiết bị như Bluetooth, Wi-Fi hoặc RF. Trong khi đó, điều khiển tự động dựa vào các thuật toán định hướng và cảm biến để robot tự ra quyết định di chuyển. Cả hai phương pháp đều yêu cầu hiểu rõ mô hình động học để đảm bảo tính chính xác, ổn định và an toàn trong quá trình vận hành.

## 2.3. Kết nối và truyền thông qua Bluetooth

Bluetooth là công nghệ truyền thông không dây tầm ngắn, cho phép các thiết bị trao đổi dữ liệu với nhau trong phạm vi khoảng 10 m thông qua sóng vô tuyến ở dải tần 2.4 GHz. Trong các ứng dụng robot di động, Bluetooth được sử dụng phổ biến để **truyền lệnh điều khiển và dữ liệu cảm biến** giữa robot và thiết bị điều khiển như máy tính hoặc điện thoại thông minh.

Module **HC-05** là một trong những module Bluetooth được sử dụng rộng rãi nhờ **giá thành thấp, cấu hình đơn giản và độ ổn định cao**. HC-05 hỗ trợ cả hai chế độ hoạt động: **Master** (chủ động kết nối) và **Slave** (bị điều khiển). Trong hầu hết các mô hình robot di động, HC-05 thường được cấu hình ở chế độ *Slave*, cho phép robot nhận lệnh từ thiết bị điều khiển qua cổng **UART** (Tx, Rx) kết nối với vi điều khiển.

Quá trình truyền thông giữa vi điều khiển và module HC-05 được thực hiện thông qua **giao thức nối tiếp (Serial Communication)** với tốc độ truyền dữ liệu thông thường là 9600 bps. Khi nhận được lệnh từ thiết bị điều khiển, vi điều khiển giải mã dữ liệu và thực hiện các hành động tương ứng như tiến, lùi, rẽ trái, rẽ phải hoặc dừng lại. Với ưu điểm **dễ triển khai, tiết kiệm năng lượng và độ tin cậy cao**, Bluetooth HC-05 là lựa chọn phù hợp cho các hệ thống robot di động điều khiển từ xa trong môi trường thử nghiệm và mô phỏng.

## 2.4. Ứng dụng di động trong điều khiển robot

Ứng dụng di động đóng vai trò trung gian trong việc **giao tiếp và điều khiển robot di động** thông qua kết nối không dây, giúp người dùng dễ dàng thao tác và giám sát hoạt động của robot trong thời gian thực. Nhờ sự phát triển mạnh mẽ của các nền tảng di động như **Android** và **iOS**, việc thiết kế giao diện điều khiển trở nên thuận tiện, linh hoạt và có tính tương tác cao.

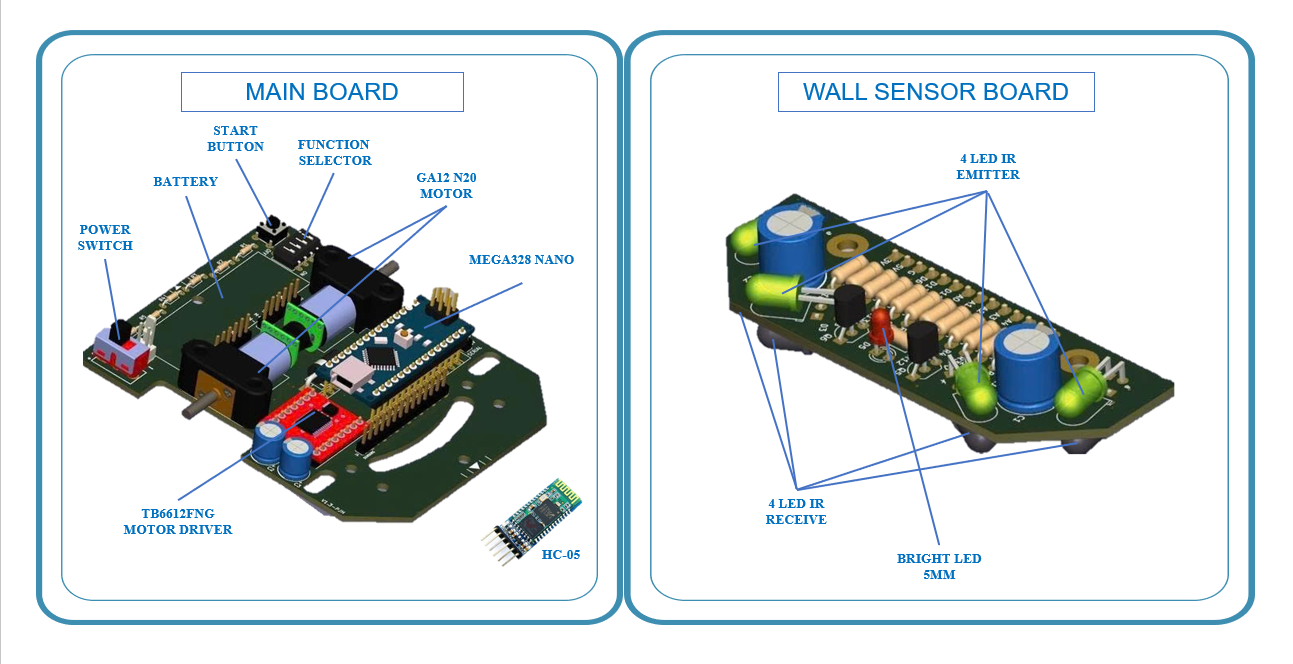
Trong hệ thống này, ứng dụng di động được xây dựng để **gửi các lệnh điều khiển** (như tiến, lùi, rẽ trái, rẽ phải, dừng) tới vi điều khiển của robot thông qua module **Bluetooth HC-05**. Ứng dụng đồng thời có thể **nhận dữ liệu phản hồi** từ robot, chẳng hạn như tốc độ, hướng di chuyển hoặc trạng thái cảm biến, nhằm phục vụ việc giám sát và điều chỉnh.

Việc sử dụng ứng dụng di động trong điều khiển robot mang lại nhiều ưu điểm như **tính cơ động, giao diện trực quan và khả năng tùy biến cao**. Người dùng có thể điều khiển robot dễ dàng mà không cần các thiết bị ngoại vi phức tạp. Ngoài ra, ứng dụng còn có thể mở rộng để tích hợp thêm các tính năng như **lưu trữ dữ liệu, hiển thị bản đồ di chuyển hoặc tự động hóa thao tác**, góp phần nâng cao tính thông minh và hiệu quả của hệ thống robot di động.

# III. Thiết kế hệ thống

## 3.1. Kiến trúc tổng thể hệ thống

Hệ thống điều khiển robot di động UKMarsbot được thiết kế theo **kiến trúc tích hợp giữa phần cứng nhúng và phần mềm điều khiển**, đảm bảo khả năng xử lý dữ liệu cảm biến, điều khiển chuyển động chính xác và giao tiếp thời gian thực với thiết bị điều khiển bên ngoài. Toàn bộ phần cứng được bố trí trên hai bo mạch chính: **bo mạch điều khiển trung tâm (mainboard)** và **bo cảm biến tường (wall sensor board)**.



Hình 3.1. Kiến trúc tổng thể hệ thống Ukmarsbot

**Mainboard** sử dụng **vi điều khiển Arduino Nano** làm bộ xử lý trung tâm, đảm nhiệm việc thu thập tín hiệu, xử lý lệnh điều khiển và điều khiển động cơ. Mạch công suất dùng **driver TB6612FNG** để điều khiển hai động cơ DC (trái và phải), giúp robot di chuyển theo các lệnh được truyền từ mô hình mô phỏng hoặc ứng dụng di động. Mạch được tích hợp các thành phần hỗ trợ như **mạch lọc nguồn (tụ điện 330 µF)**, **mạch đảo tín hiệu**, **mạch xung encoder**, và các đầu nối (headers) cho cảm biến, module Bluetooth HC-05, và công tắc nguồn.

**Bo cảm biến tường (Wall Sensor Board)** gồm 4 cặp **cảm biến hồng ngoại phản xạ** (SFH4545/TEFT4300) được bố trí ở phía trước và hai bên của robot, cho phép phát hiện vật cản hoặc tường trong phạm vi gần. Các tín hiệu phản xạ được khuếch đại và xử lý qua mạch transistor (BC337) trước khi gửi về vi điều khiển. Thiết kế mạch này cho phép robot có thể xác định khoảng cách tới vật thể và điều chỉnh hướng di chuyển khi cần.

Các khối phần cứng được kết nối với nhau thông qua hệ thống chân cắm tiêu chuẩn, giúp dễ dàng thay thế hoặc mở rộng. Dữ liệu điều khiển được truyền giữa robot và **ứng dụng di động** thông qua **Bluetooth HC-05**, tạo thành một **chuỗi điều khiển khép kín** gồm: *ứng dụng → Bluetooth → vi điều khiển → động cơ/cảm biến → phản hồi dữ liệu → ứng dụng*.

Kiến trúc tổng thể này thể hiện rõ sự **phối hợp giữa xử lý tín hiệu, điều khiển động học và truyền thông không dây**, đồng thời được thiết kế tối ưu để sử dụng các linh kiện phổ biến tại thị trường Việt Nam, giúp hệ thống dễ chế tạo, dễ mở rộng và phù hợp cho các nghiên cứu ứng dụng trong lĩnh vực robot di động.

## 3.2. Phần cứng sử dụng

Hệ thống điều khiển robot di động UKMarsbot được xây dựng từ các linh kiện phổ biến, dễ tìm kiếm trên thị trường, đảm bảo tính khả thi trong chế tạo và nghiên cứu. Các thành phần phần cứng chính bao gồm **bộ vi điều khiển**, **cảm biến**, **động cơ dẫn động**, **module Bluetooth**, cùng **pin và mạch cấp nguồn**.

**Bộ vi điều khiển, cảm biến và động cơ:**

Bộ vi điều khiển trung tâm sử dụng **Arduino Nano** (vi xử lý ATmega328P), có kích thước nhỏ gọn, dễ lập trình và tương thích với nhiều module ngoại vi. Hai **động cơ DC** gắn encoder được điều khiển thông qua **mạch cầu H TB6612FNG**, cho phép điều chỉnh tốc độ và hướng quay một cách linh hoạt. Hệ thống cảm biến sử dụng **các cặp LED phát – thu hồng ngoại (SFH4545 và TEFT4300)** bố trí ở phía trước và hai bên robot, đảm nhận nhiệm vụ phát hiện tường hoặc vật cản, giúp robot xác định vị trí tương đối trong không gian làm việc.

**Module Bluetooth:**

Hệ thống sử dụng **module HC-05** để thiết lập kết nối không dây giữa robot và ứng dụng điều khiển trên điện thoại di động. Module này hỗ trợ giao tiếp nối tiếp (UART), tốc độ truyền ổn định và dễ tích hợp với vi điều khiển. Thông qua Bluetooth, các lệnh điều khiển (tiến, lùi, rẽ trái, rẽ phải, dừng) được gửi đến robot, đồng thời robot cũng có thể phản hồi lại các thông tin trạng thái khi cần.

**Pin và mạch cấp nguồn:**



Hình 3.2. Nguồn pin cấp

Nguồn năng lượng cho hệ thống được cung cấp bởi **pin vuông 9V 650mAh** kết hợp với **mạch ổn áp** để cung cấp điện áp 5V cho các linh kiện logic và 3.3V khi cần thiết cho các module cảm biến. Các tụ lọc (330 µF, 100 nF) được bố trí tại các điểm chính trên bo mạch nhằm giảm nhiễu và ổn định điện áp khi động cơ thay đổi tải. Thiết kế mạch nguồn đảm bảo robot có thể hoạt động ổn định trong thời gian dài và giảm thiểu ảnh hưởng của dao động điện áp đến khả năng điều khiển.

## 3.3. Phần mềm trên robot

Phần mềm điều khiển được lập trình trên nền tảng **Arduino IDE**, đóng vai trò trung tâm trong việc xử lý lệnh, điều khiển chuyển động và giao tiếp Bluetooth. Chương trình được tổ chức thành các khối chức năng chính gồm **điều khiển động cơ**, **nhận và xử lý tín hiệu Bluetooth**, cùng **quản lý trạng thái hoạt động của robot**.

**Chương trình điều khiển động cơ:**

Động cơ của robot được điều khiển thông qua **driver TB6612FNG**, sử dụng hai kênh PWM độc lập cho hai bánh xe. Phần mềm thực hiện việc **tạo xung PWM** để điều chỉnh tốc độ quay của động cơ, đồng thời điều khiển các chân logic IN1, IN2, IN3, IN4 nhằm xác định hướng di chuyển (tiến, lùi, rẽ trái, rẽ phải hoặc dừng). Các lệnh này được lập trình theo cấu trúc hàm riêng biệt, giúp dễ dàng mở rộng hoặc hiệu chỉnh trong quá trình thử nghiệm. Ngoài ra, chương trình còn có thể tích hợp phản hồi từ encoder để **tự động cân bằng tốc độ hai bánh**, đảm bảo chuyển động thẳng ổn định.

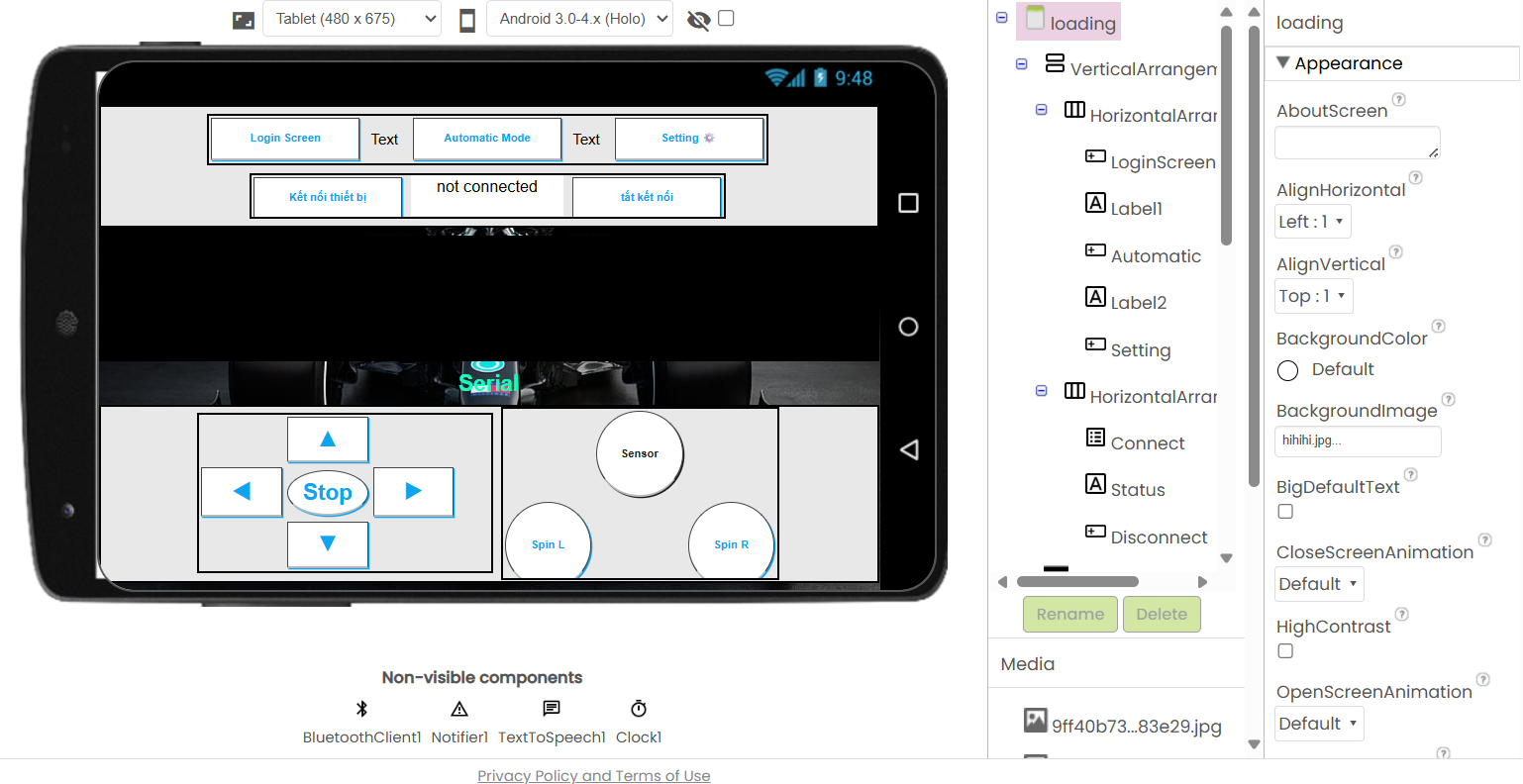
**Thuật toán xử lý tín hiệu Bluetooth:**

Robot sử dụng **module HC-05** để nhận tín hiệu điều khiển được gửi từ ứng dụng di động thông qua chuẩn UART. Chương trình liên tục **lắng nghe dữ liệu từ cổng serial**, phân tích ký tự lệnh (chẳng hạn: 'F' – tiến, 'B' – lùi, 'L' – rẽ trái, 'R' – rẽ phải, 'S' – dừng) và thực thi tương ứng với các hàm điều khiển động cơ đã định nghĩa. Cấu trúc thuật toán đảm bảo việc xử lý lệnh **nhanh, ổn định và không xung đột**, cho phép robot phản ứng tức thời với thao tác từ người dùng.

Phần mềm được thiết kế theo hướng **mở rộng linh hoạt**, cho phép bổ sung các chức năng nâng cao như điều khiển theo quỹ đạo, truyền dữ liệu cảm biến ngược về ứng dụng hoặc tự động điều chỉnh tốc độ theo khoảng cách vật cản. Qua đó, robot UKMarsbot không chỉ dừng ở điều khiển thủ công mà còn có tiềm năng phát triển thành nền tảng nghiên cứu cho các hệ thống robot tự hành thông minh.

## 3.4. Ứng dụng điều khiển trên điện thoại

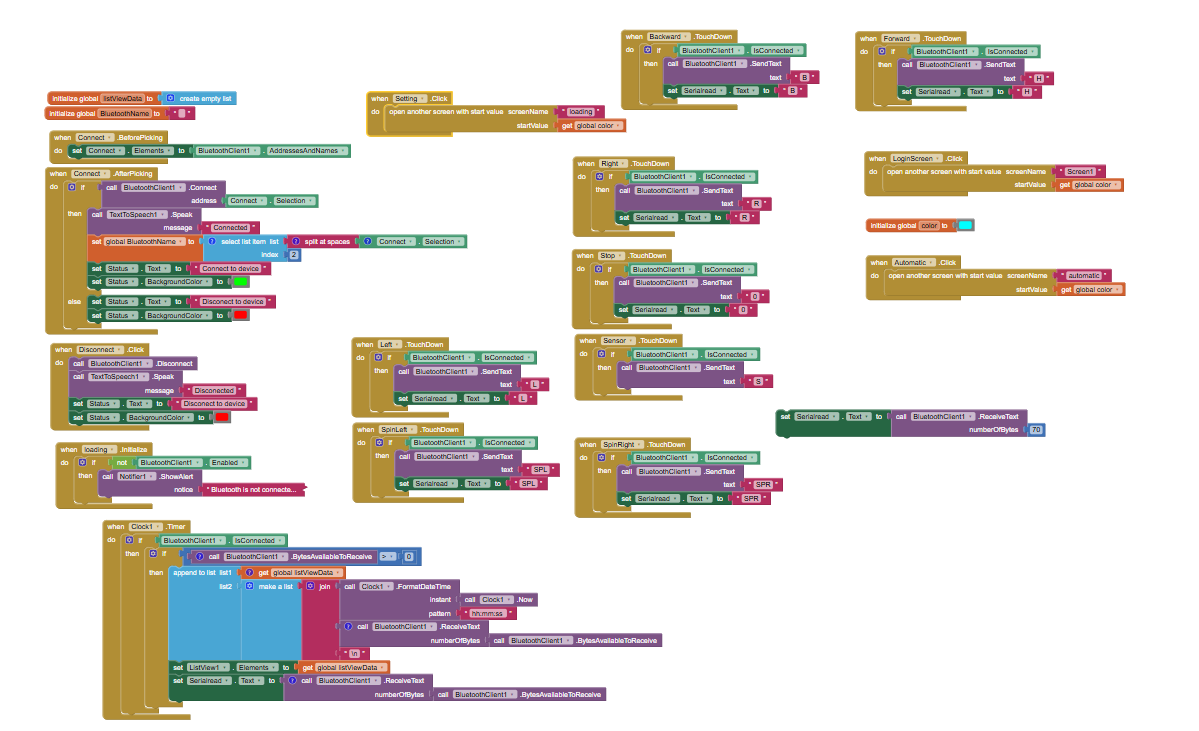
Giao diện người dùng



Hình 3.3. Thiết kế giao diện cho App điều khiển

Ứng dụng điều khiển được thiết kế và lập trình bằng **MIT App Inventor**, một nền tảng kéo-thả trực quan cho phép tạo ứng dụng Android dễ dàng mà không cần lập trình phức tạp. Giao diện của ứng dụng bao gồm các **nút bấm chức năng chính** để điều khiển chuyển động của robot như: **Tiến (Forward), Lùi (Backward), Rẽ trái (Left), Rẽ phải (Right), Dừng (Stop)**, cùng với các nút **Spin Left / Spin Right** để quay vòng tại chỗ. Ngoài ra, ứng dụng còn có **nút Kết nối Bluetooth**, cho phép người dùng tìm kiếm và ghép nối với module **HC-05** gắn trên robot. Giao diện được bố trí thân thiện, dễ thao tác, giúp người dùng điều khiển robot trực quan và nhanh chóng.

Các chức năng chính:



Hình 3.4. Block giải thuật cho App điều khiển

Khi người dùng mở ứng dụng, hệ thống sẽ quét và hiển thị danh sách các thiết bị Bluetooth xung quanh. Sau khi kết nối thành công với **HC-05**, mỗi nút bấm trên giao diện sẽ **gửi một ký tự lệnh tương ứng** (như H, B, L, R, 0, SPL, SPR) qua kết nối Bluetooth đến **Arduino Nano** của robot. Arduino sẽ đọc và xử lý các lệnh này, thực thi hành động tương ứng như di chuyển, quay, hoặc dừng lại. Ngoài ra, ứng dụng còn có thể hiển thị phản hồi từ Arduino (nếu có), giúp người dùng giám sát trạng thái hoạt động của xe.

# IV. Mô phỏng phương trình động lực học của robot

## 4.1. Xây dựng mô hình động học và động lực học của robot

Robot di động UKMarsbot được thiết kế theo cấu trúc **hai bánh chủ động vi sai** (Differential Drive Robot), trong đó hai bánh được điều khiển độc lập bởi hai động cơ DC gắn encoder, cho phép robot có thể di chuyển tiến, lùi hoặc quay tại chỗ. Mô hình động học và động lực học của hệ thống được xây dựng nhằm mô tả chính xác mối quan hệ giữa vận tốc của từng bánh xe và chuyển động tổng thể của thân robot trong không gian phẳng 2D.

**a) Mô hình động học cơ bản**

Gọi:

r: bán kính bánh xe (m)

L: nửa khoảng cách giữa hai bánh (m)

ωL​,ωR​: vận tốc góc của bánh trái và bánh phải (rad/s)

v,ω: vận tốc tuyến tính và vận tốc góc của robot

(x,y,θ): vị trí và góc định hướng của robot trong hệ tọa độ toàn cục

Khi đó, quan hệ giữa vận tốc bánh xe và vận tốc của robot được xác định bởi:

Chuyển động của robot trong không gian phẳng được mô tả bởi hệ phương trình vi phân phi tuyến:

Trong đó, , , lần lượt là đạo hàm theo thời gian của vị trí và góc quay robot.

**b) Mô hình động lực học**

Để phản ánh đặc tính vật lý của hệ thống thực tế, mô hình động lực học được mở rộng bằng cách xem xét quán tính và sai số điều khiển:

với M là khối lượng quy đổi của robot, J là mô men quán tính quanh trục quay là hệ số ma sát nhớt, F và τ là lực và mô men sinh ra bởi hai động cơ.  
Trong mô phỏng MATLAB, để đơn giản hóa, các hiệu ứng động lực học cao cấp này được gộp lại thành sai lệch ngẫu nhiên nhỏ trong vận tốc bánh xe nhằm mô phỏng ảnh hưởng của ma sát, trượt bánh hoặc sai số encoder thực tế.

c) Thông số vật lý của robot sử dụng trong mô phỏng

|  |  |  |  |
| --- | --- | --- | --- |
| **Thông số** | **Ký hiệu** | **Giá trị** | **Đơn vị** |
| Bán kính bánh xe | (r) | 0.017 | m |
| Nửa khoảng cách giữa 2 bánh | (L) | 0.0405 | m |
| Thời gian mô phỏng | (T) | 10 | s |
| Bước thời gian | (dt) | 0.01 | s |

Robot được yêu cầu di chuyển theo **quỹ đạo cung tròn** với bán kính R=0.3 m vận tốc tuyến tính mong muốn ​= 0.2m/s. Từ đó, vận tốc góc tương ứng được tính là:

Vận tốc mong muốn của hai bánh được xác định theo:

**d) Giả lập sai lệch và cập nhật trạng thái**

Để mô phỏng thực tế hơn, vận tốc mỗi bánh được thêm nhiễu Gaussian nhỏ với độ lệch chuẩn 3% nhằm tái hiện sai số encoder và ma sát bánh:

Sau đó, vận tốc và góc quay thực tế của robot được cập nhật theo:

Trạng thái robot được tích phân rời rạc theo bước thời gian dt:

**e) Sai số và tiêu chí đánh giá**

Để định lượng sai khác giữa chuyển động thực tế và quỹ đạo mong muốn, sai số RMS được tính theo công thức:

## 4.2. Thiết lập phương trình chuyển động

**a)Dạng ma trận tổng quát**

Phương trình trên có thể viết gọn dưới dạng ma trận:

Thay v,ω theo vận tốc bánh xe, ta được:

Đây chính là **phương trình chuyển động tổng quát** của robot di động hai bánh vi sai, thể hiện rõ sự phụ thuộc của vận tốc tịnh tiến và góc quay vào các biến điều khiển ωR​ và ωL​.

**b) Dạng rời rạc hóa trong mô phỏng MATLAB**

Trong mô phỏng số, ta rời rạc hóa phương trình theo bước thời gian dt:

Phương trình này được sử dụng trực tiếp trong chương trình mô phỏng MATLAB/Simulink để **cập nhật trạng thái robot theo thời gian**, dựa trên vận tốc tức thời của hai bánh xe.

## 4.3. Mô phỏng robot trên MATLAB/Simulink

Quá trình mô phỏng robot di chuyển hai bánh vi sai được thực hiện trên **MATLAB**, nhằm đánh giá khả năng bám quỹ đạo và độ chính xác của mô hình động học. Trong mô phỏng, các thông số vật lý của robot được định nghĩa bao gồm: **bán kính bánh xe** r=0.017m, **nửa khoảng cách giữa hai bánh** L=0.0405 m, và **bước thời gian lấy mẫu** dt=0.01s.

Robot được điều khiển di chuyển theo **quỹ đạo tham chiếu hình cung tròn** với bán kính R=0.3m và vận tốc tuyến tính mong muốn = 0.2m/s. Từ đó, vận tốc góc tham chiếu được tính theo công thức =. Quỹ đạo tham chiếu của robot được mô tả bằng các phương trình:

Mô phỏng được triển khai theo thời gian, trong đó robot được giả lập với **nhiễu ngẫu nhiên nhỏ (±3%)** để mô phỏng sai lệch do ma sát, trượt bánh hoặc sai số đo encoder. Vận tốc thực tế của từng bánh được tính lại ở mỗi chu kỳ, sau đó dùng để cập nhật vị trí và góc quay của robot theo phương trình động học vi sai.

Kết quả mô phỏng được thể hiện qua **các biểu đồ và quỹ đạo chuyển động** giúp đánh giá độ chính xác của mô hình. Đồ thị hiển thị **đường đi tham chiếu** và **đường đi thực tế**, cho thấy sự bám quỹ đạo của robot tương đối tốt, với sai lệch nhỏ trong phạm vi cho phép. Việc mô phỏng này giúp kiểm chứng khả năng điều khiển ổn định trước khi triển khai thực nghiệm trên phần cứng

## 4.4. Mô phỏng chuyển động cùng với quỹ đạo, vận tốc, sai số, góc quay

Sau khi xác định được mô hình động học, robot được mô phỏng chi tiết để theo dõi **chuyển động thực tế**, **vận tốc**, **góc quay**, và **sai số vị trí** theo thời gian. Mô phỏng cho phép hiển thị trực quan sự khác biệt giữa giá trị lý thuyết và giá trị thực tế của robot khi chịu ảnh hưởng nhiễu và sai số ngẫu nhiên.

Biểu đồ đầu tiên phải thể hiện được **quỹ đạo chuyển động** trên mặt phẳng XY, trong đó đường đứt nét là quỹ đạo tham chiếu, còn đường liền là quỹ đạo thực tế. Ta sẽ có thể quan sát được rằng robot di chuyển ổn định theo cung tròn, chỉ có sai lệch nhỏ do nhiễu vận tốc bánh xe.

# V. Thuật toán và cơ chế điều khiển

## 5.1. Giải thuật điều khiển từ xa qua Bluetooth

Hệ thống điều khiển từ xa của robot UKMarsbot được thiết kế dựa trên giao tiếp Bluetooth không dây, sử dụng module HC-05 để truyền và nhận dữ liệu giữa Arduino Nano và ứng dụng điều khiển trên điện thoại thông minh. Phương pháp này giúp robot có thể được điều khiển linh hoạt, dễ dàng thao tác mà không cần kết nối dây, đồng thời đảm bảo độ ổn định và độ trễ thấp trong truyền nhận tín hiệu.

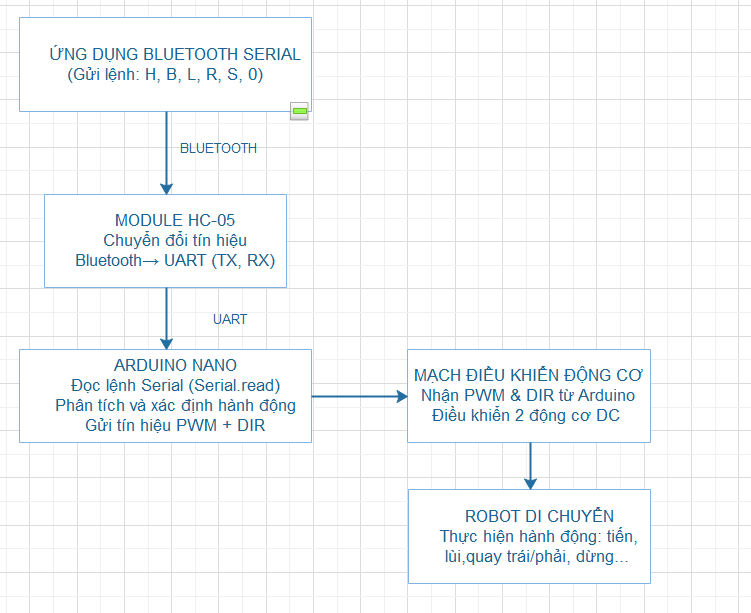
Module Bluetooth HC-05 được nối trực tiếp với Arduino Nano thông qua giao tiếp UART (Serial). Cụ thể, chân TXD của HC-05 được nối vào chân RX (D0) của Arduino, và chân RXD nối vào chân TX (D1) của Arduino. Hai chân VCC và GND lần lượt được cấp nguồn 5V và nối đất chung với mạch điều khiển. Khi kết nối Bluetooth được thiết lập với điện thoại, LED báo trạng thái trên HC-05 chuyển từ nhấp nháy nhanh sang nhấp nháy chậm hoặc sáng liên tục, cho biết đường truyền đã sẵn sàng.

Về nguyên lý hoạt động, ứng dụng điều khiển trên điện thoại (như Bluetooth Terminal, Bluetooth Serial Controller, hoặc app custom) gửi các lệnh dạng chuỗi ký tự thông qua kết nối Bluetooth. Các lệnh này được module HC-05 nhận, chuyển đổi sang tín hiệu UART, và truyền tới Arduino Nano để xử lý. Arduino sau đó phân tích chuỗi lệnh, xác định hành động tương ứng, và phát tín hiệu điều khiển PWM cùng tín hiệu hướng đến hai động cơ DC, giúp robot thực hiện các chuyển động mong muốn như tiến, lùi, rẽ trái, rẽ phải, hoặc dừng lại.

**a) Luồng hoạt động của hệ thống**

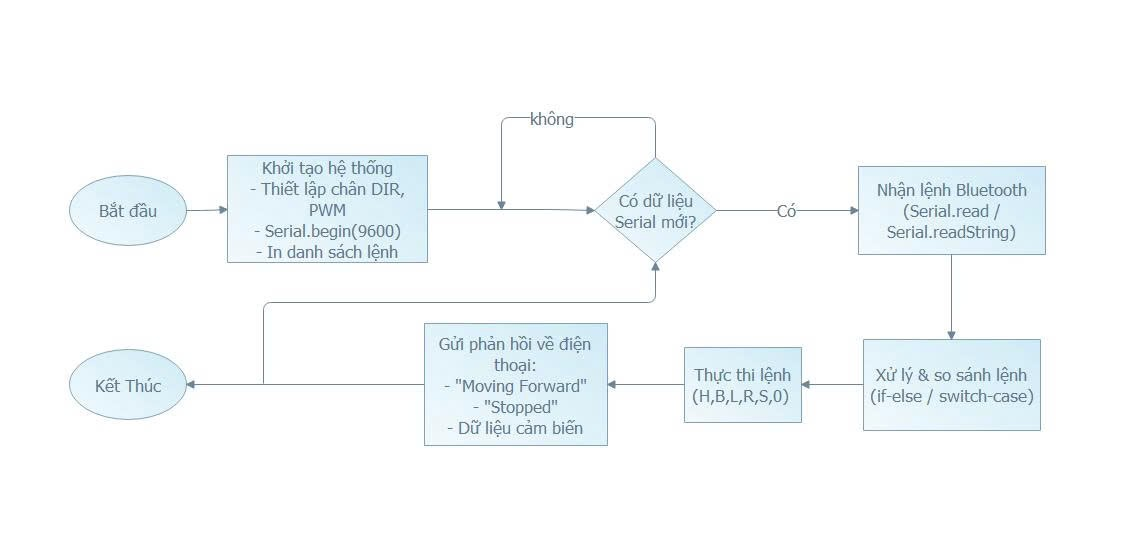
Quá trình điều khiển robot qua Bluetooth được thực hiện dựa trên sự phối hợp giữa ứng dụng điều khiển trên điện thoại, module HC-05 và vi điều khiển Arduino Nano. Đầu tiên, người dùng thao tác trên ứng dụng Bluetooth Serial, gửi các lệnh điều khiển như tiến (H), lùi (B), rẽ trái (L), rẽ phải (R), dừng (0)… Các lệnh này được truyền không dây đến module HC-05, nơi tín hiệu Bluetooth được chuyển đổi thành tín hiệu UART (TX, RX) và gửi trực tiếp đến vi điều khiển Arduino. Việc giao tiếp sử dụng chuẩn truyền thông Serial (UART) với tốc độ 9600 bps, đảm bảo tín hiệu ổn định và độ trễ thấp.

Khi nhận được tín hiệu, Arduino Nano tiến hành đọc dữ liệu từ bộ đệm Serial thông qua hàm Serial.read() hoặc Serial.readStringUntil(). Giải thuật điều khiển sẽ phân tích nội dung lệnh và gọi hàm điều khiển tương ứng, ví dụ goStraight() để đi thẳng, turnLeft90() để rẽ trái 90°, hoặc stopMotors() để dừng khẩn cấp. Sau đó, Arduino phát tín hiệu điều khiển dạng PWM và tín hiệu hướng (DIR) đến mạch driver động cơ DC, giúp điều chỉnh tốc độ và hướng quay của từng bánh xe. Kết quả là robot di chuyển đúng theo yêu cầu của người điều khiển, đồng thời có thể phản hồi trạng thái về điện thoại để người dùng theo dõi quá trình hoạt động.

****

Hình 5.1. Sơ đồ khối mô tả luồng hoạt động của hệ thống

**b) Giải thuật điều khiển trên Arduino**



Hình 5.2. Sơ đồ khối miêu tả giải thuật điều khiển

Giải thuật điều khiển robot qua Bluetooth trên Arduino được xây dựng nhằm đảm bảo việc giao tiếp và phản hồi giữa điện thoại và robot diễn ra liên tục, chính xác và ổn định. Ở giai đoạn đầu, hệ thống tiến hành khởi tạo phần cứng, bao gồm cấu hình các chân điều khiển hướng (DIR) và điều chế độ rộng xung (PWM) cho hai động cơ, đồng thời thiết lập giao tiếp Serial ở tốc độ 9600 bps để kết nối với module Bluetooth HC-05. Sau khi khởi động thành công, Arduino sẽ in ra danh sách các lệnh điều khiển có thể nhận, giúp người vận hành biết được các tùy chọn điều khiển robot từ ứng dụng.

Trong quá trình hoạt động, vi điều khiển liên tục lắng nghe tín hiệu từ cổng Serial – là nơi module HC-05 gửi dữ liệu nhận được từ ứng dụng Bluetooth Serial trên điện thoại. Khi phát hiện có dữ liệu mới, Arduino sẽ đọc ký tự hoặc chuỗi lệnh được gửi đến, sau đó chuyển sang bước xử lý và so sánh lệnh với danh sách các lệnh đã định nghĩa sẵn. Việc so sánh này thường được thực hiện thông qua cấu trúc điều khiển if-else hoặc switch-case, giúp hệ thống nhận biết nhanh chóng loại lệnh cần thực thi.

Dựa trên kết quả so sánh, Arduino sẽ gọi hàm điều khiển tương ứng. Ví dụ, nếu lệnh là H, robot sẽ thực thi hàm goStraight() để di chuyển thẳng; nếu là B, thì thực hiện goBackward() để lùi; còn L hoặc R sẽ kích hoạt các hàm quay trái hoặc phải tương ứng. Ngoài ra, các lệnh đặc biệt như S cho phép đọc cảm biến hồng ngoại, trong khi 0 dùng để dừng robot khẩn cấp.

Sau khi hoàn tất lệnh, Arduino gửi phản hồi trở lại cho ứng dụng trên điện thoại, có thể là thông báo trạng thái (“Moving Forward”, “Stopped”, “Turn Left”), hoặc dữ liệu cảm biến nếu người dùng yêu cầu. Quá trình này diễn ra liên tục trong vòng lặp loop(), tạo thành chu trình điều khiển – phản hồi khép kín, giúp người điều khiển có thể tương tác thời gian thực với robot một cách trực quan và chính xác.

## 5.2. Giải thuật xử lý lệnh từ ứng dụng

Giải thuật xử lý lệnh là thành phần trọng tâm trong hệ thống điều khiển robot, đảm nhận nhiệm vụ tiếp nhận, phân tích và thực thi các lệnh điều khiển được gửi từ ứng dụng Bluetooth Serial trên điện thoại. Thông qua quá trình này, robot có thể phản hồi nhanh chóng và chính xác theo yêu cầu của người điều khiển.

a) Nguyên lý hoạt động

Ứng dụng Bluetooth Serial trên điện thoại đóng vai trò như bộ điều khiển chính, cho phép người dùng gửi các lệnh dạng ký tự đến robot thông qua module HC-05. Các lệnh này được truyền qua giao tiếp UART đến Arduino Nano, nơi chúng được xử lý và chuyển đổi thành các hành động cụ thể.

Khi Arduino nhận được chuỗi lệnh từ cổng Serial, chương trình sử dụng hàm Serial.readStringUntil('\n') để đọc toàn bộ chuỗi cho đến khi gặp ký tự xuống dòng. Sau đó, hàm trim() được gọi để loại bỏ khoảng trắng hoặc ký tự thừa ở đầu và cuối chuỗi. Lệnh sau khi được xử lý sơ bộ sẽ được so sánh với tập hợp các lệnh đã định nghĩa trước.

Nếu lệnh hợp lệ, Arduino gọi hàm điều khiển tương ứng — ví dụ goStraight(), turnLeft90(), spinRight()... — để phát tín hiệu PWM và điều khiển chiều quay của hai động cơ DC.

Trong trường hợp lệnh không hợp lệ, hệ thống sẽ phản hồi bằng một thông báo lỗi gửi lại qua Serial, giúp người dùng nhận biết và gửi lại lệnh đúng.

b) Cấu trúc xử lý lệnh

Giải thuật xử lý lệnh được lập trình trong hàm loop() của Arduino, đảm bảo hoạt động liên tục và phản ứng tức thời với tín hiệu điều khiển. Trình tự hoạt động có thể tóm tắt như sau:

* Khởi tạo vòng lặp: Arduino liên tục kiểm tra xem có dữ liệu đến từ cổng Serial hay không.
* Nhận dữ liệu: Khi phát hiện có tín hiệu, chương trình đọc chuỗi lệnh gửi đến và lưu vào biến cmd.
* Phân tích lệnh: So sánh cmd với các lệnh hợp lệ đã được định nghĩa sẵn.
* Thực thi hành động: Gọi hàm điều khiển tương ứng với từng loại lệnh.
* Phản hồi: In thông báo ra Serial Monitor để xác nhận hành động hoặc báo lỗi nếu lệnh không hợp lệ.

## 5.3. Các chế độ điều khiển

Bảng dưới đây mô tả chi tiết các **lệnh điều khiển Bluetooth** được gửi từ ứng dụng trên điện thoại đến vi điều khiển Arduino Nano thông qua module **HC-05**. Mỗi lệnh là một ký tự hoặc chuỗi ký tự cụ thể, tương ứng với một **hàm xử lý** đã được lập trình sẵn trong chương trình. Khi Arduino nhận được lệnh, nó sẽ gọi hàm tương ứng để thực thi hành động điều khiển robot.

|  |  |  |
| --- | --- | --- |
| **Lệnh** | **Hàm thực thi** | **Mô tả hoạt động** |
| H | goStraight() | Robot di chuyển thẳng về phía trước. |
| B | goBackward() | Robot lùi về phía sau. |
| L | turnLeft90() | Robot quay trái 90 độ. |
| R | turnRight90() | Robot quay phải 90 độ. |
| SPL | spinLeft() | Robot quay tròn tại chỗ sang trái. |
| SPR | spinRight() | Robot quay tròn tại chỗ sang phải. |
| S | readSensors() | Đọc giá trị các cảm biến hồng ngoại (IR). |
| 0 | stopMotors() | Dừng tất cả động cơ ngay lập tức. |
| ? | printHelp() | In lại danh sách lệnh điều khiển. |

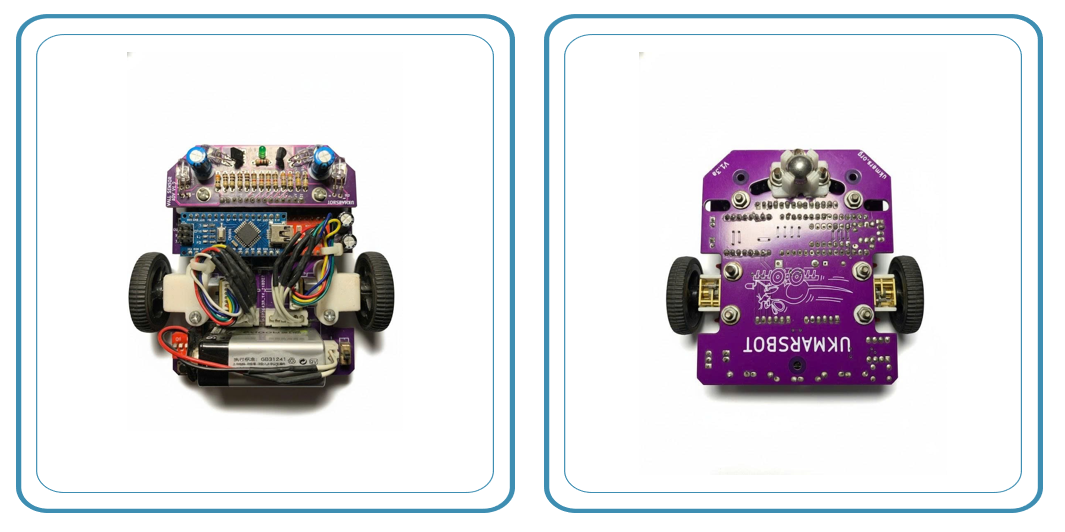
BẢNG 5.1. DANH SÁCH CÁC LỆNH ĐIỀU KHIỂN

Khi người dùng thao tác trên **ứng dụng Bluetooth Serial**, lệnh sẽ được truyền qua giao tiếp UART đến **module HC-05**. Module này chuyển dữ liệu đến **Arduino Nano**, nơi chương trình liên tục đọc các ký tự từ **Serial buffer**. Mỗi ký tự được so sánh với bảng lệnh định nghĩa sẵn; khi trùng khớp, Arduino gọi hàm xử lý tương ứng để điều khiển tốc độ và hướng quay của hai động cơ DC. Cơ chế này giúp robot phản hồi **gần như tức thời**, đảm bảo tính mượt mà và ổn định khi điều khiển từ xa.

# VI. Thực nghiệm và kết quả

## 6.1. Kết quả triển khai hệ thống

Sau khi hoàn thiện mô hình xe, hệ thống điều khiển đã được tích hợp đầy đủ các phần cứng và phần mềm để đảm bảo robot hoạt động ổn định. Xe được lắp ráp hoàn chỉnh gồm: khung cơ khí hai bánh chủ động và một bánh tự do, hai động cơ DC có encoder, module Bluetooth HC-05 để nhận lệnh điều khiển, vi điều khiển **Arduino Nano**, cùng mạch cầu H **L298N** để điều khiển công suất động cơ. Toàn bộ hệ thống được cấp nguồn từ pin Li-ion và có công tắc bật/tắt riêng biệt.



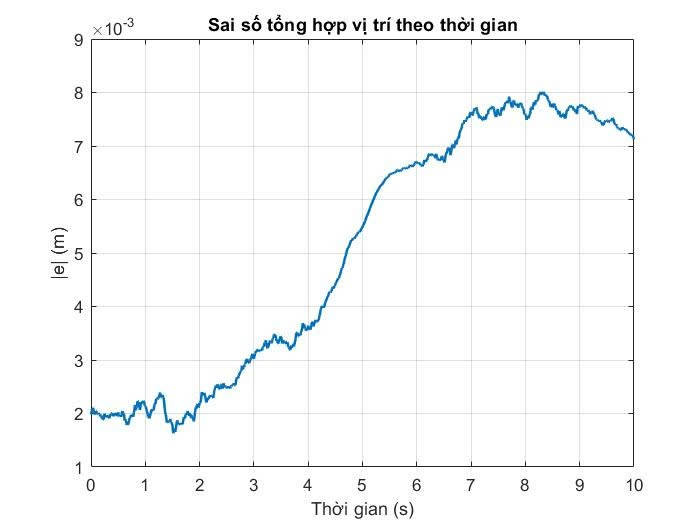
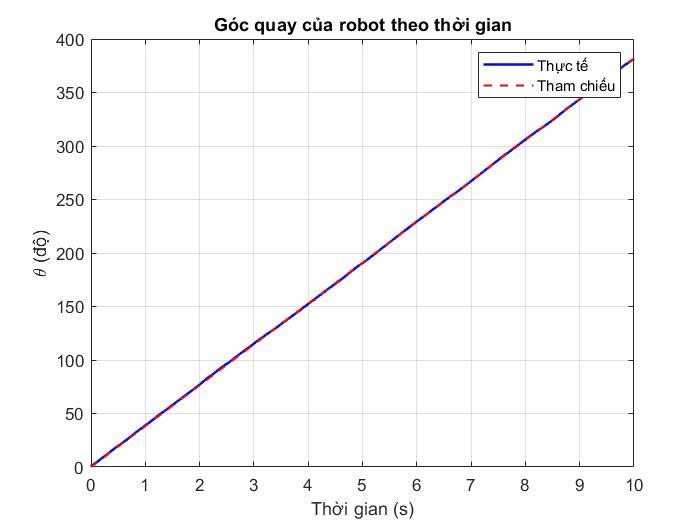
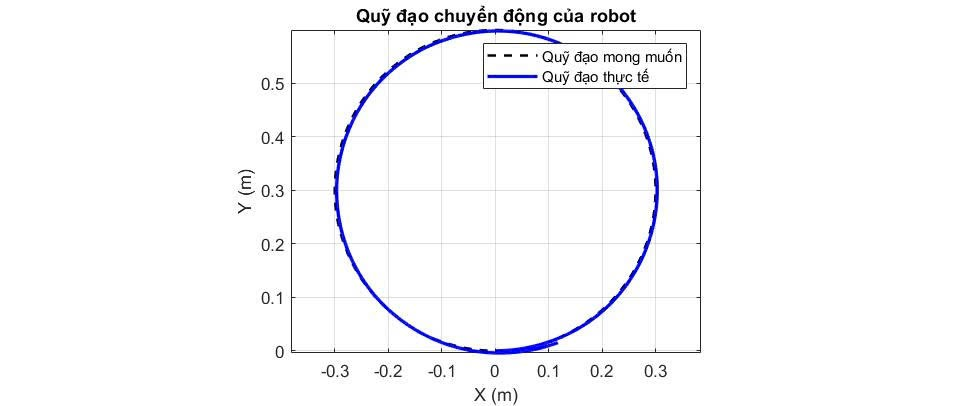
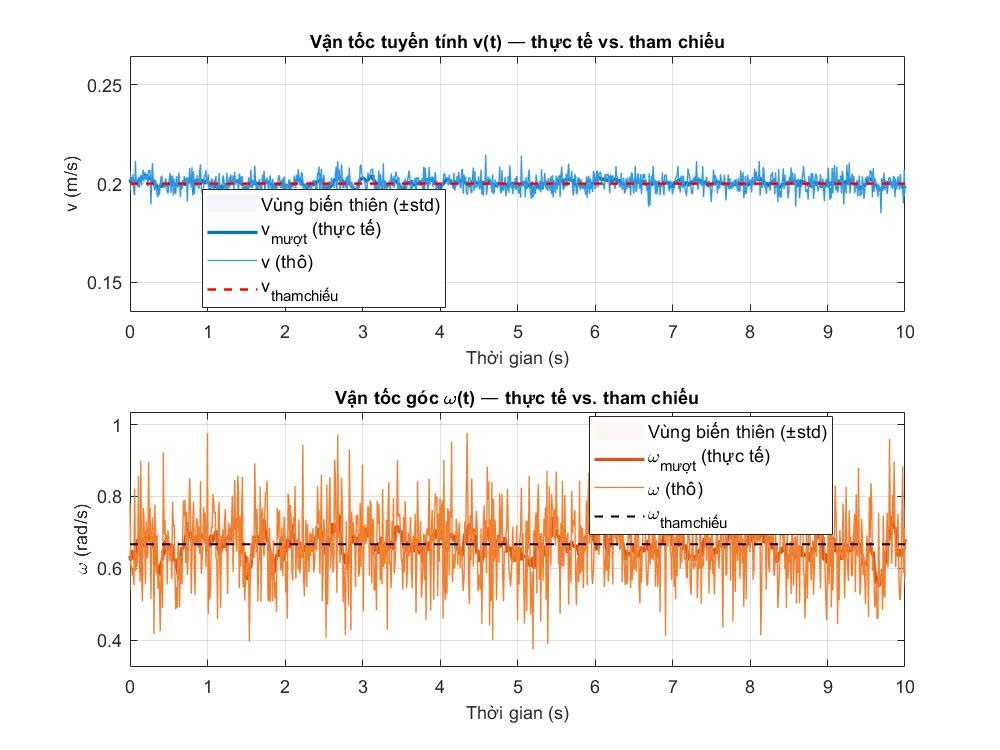
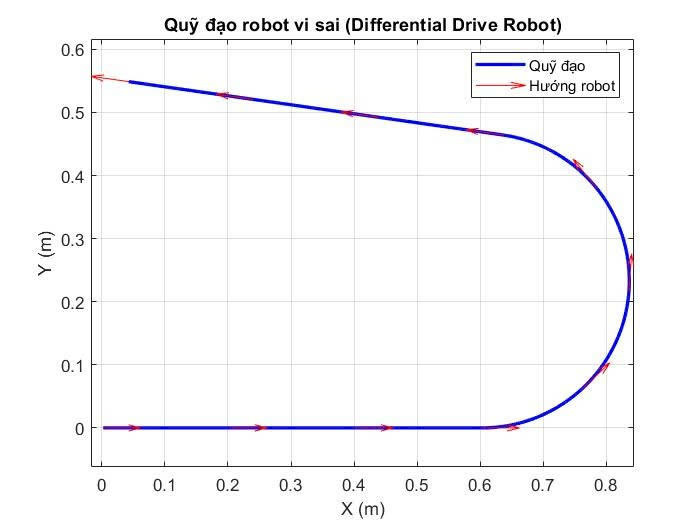
Hình 6.1. Mô hình hóa hệ thống

Về phần mềm, chương trình được lập trình trên Arduino IDE, có khả năng nhận và xử lý lệnh điều khiển được gửi từ ứng dụng điện thoại thông qua Bluetooth. Giao diện điều khiển được thiết kế bằng **MIT App Inventor**, gồm các nút cơ bản như **tiến, lùi, rẽ trái, rẽ phải, dừng**, giúp người dùng điều khiển xe dễ dàng và trực quan. Khi người dùng nhấn nút trên điện thoại, lệnh được truyền đến Arduino, sau đó vi điều khiển xuất tín hiệu PWM và tín hiệu hướng (DIR) tương ứng để điều khiển tốc độ và hướng quay của động cơ.



Hình 6.2. Giao diện app điều khiển xe

## 6.2. Kết quả mô phỏng động lực học trên MATLAB



Hình 6.3. Kết quả mô phỏng MATLAB của hệ thống

Dựa trên kết quả mô phỏng được thể hiện ở Hình trên, mô hình động lực học của robot hai bánh vi sai đã được xây dựng và kiểm chứng trên **MATLAB/Simulink**. Quá trình mô phỏng được thực hiện với các thông số thực tế của hệ thống, bao gồm bán kính bánh xe, khoảng cách giữa hai bánh, vận tốc tham chiếu và thời gian lấy mẫu. Mục tiêu của mô phỏng là đánh giá khả năng bám quỹ đạo, độ ổn định vận tốc, và sai số của robot trong quá trình di chuyển theo quỹ đạo tròn mong muốn.

Kết quả thu được cho thấy robot di chuyển theo quỹ đạo tròn ổn định, đường đi thực tế (đường màu xanh) gần như trùng khớp với quỹ đạo tham chiếu (đường nét đứt màu đen), chứng tỏ mô hình động học và thuật toán điều khiển được xây dựng chính xác. Biểu đồ vận tốc tuyến tính và vận tốc góc cho thấy robot duy trì ổn định quanh giá trị mong muốn, chỉ xuất hiện các dao động nhỏ do nhiễu giả lập (sai số cảm biến và trượt bánh).

Biểu đồ sai số vị trí và sai số góc quay thể hiện mức lệch nhỏ trong quá trình di chuyển, với sai số tổng hợp vị trí luôn nằm trong khoảng vài milimét (dưới 0.01 m). Góc quay của robot tăng tuyến tính theo thời gian và bám sát giá trị tham chiếu, chứng tỏ robot giữ được hướng di chuyển ổn định và phản ứng chính xác với tín hiệu điều khiển.

## 6.3. Đánh giá mô phỏng

Sau khi hoàn thành quá trình mô phỏng robot trên MATLAB/Simulink, việc đánh giá kết quả được thực hiện dựa trên ba tiêu chí chính: **độ chính xác bám quỹ đạo**, **tính ổn định vận tốc và góc quay**, cùng **độ tin cậy của mô hình toán học**.

Về **độ chính xác bám quỹ đạo**, kết quả cho thấy robot di chuyển bám rất sát theo đường tham chiếu đã đặt ra (đường cong hoặc đường tròn). Quỹ đạo thực tế của robot chỉ lệch một khoảng rất nhỏ so với quỹ đạo lý tưởng, sai số vị trí trung bình chỉ khoảng vài milimét, chứng tỏ rằng mô hình động học vi sai và giải thuật điều khiển đã được xây dựng chính xác. Sai số này chủ yếu xuất hiện tại các điểm bắt đầu thay đổi hướng di chuyển, nơi vận tốc góc thay đổi nhanh — điều hoàn toàn chấp nhận được trong thực tế.

Về **tính ổn định của vận tốc và góc quay**, biểu đồ kết quả cho thấy vận tốc tuyến tính và vận tốc góc của robot được duy trì ổn định quanh giá trị tham chiếu, không xuất hiện dao động lớn hoặc hiện tượng quá điều chỉnh. Robot phản ứng nhanh với tín hiệu điều khiển và đạt đến vận tốc mong muốn trong thời gian ngắn, thể hiện khả năng đáp ứng tốt của hệ thống. Góc quay thay đổi tuyến tính theo thời gian, phù hợp với chuyển động quay tròn, khẳng định bộ điều khiển hoạt động ổn định.

Cuối cùng, **độ tin cậy của mô hình toán học** được kiểm chứng thông qua tính tương đồng giữa kết quả mô phỏng và các hành vi mong đợi từ lý thuyết. Mô hình mô phỏng đã phản ánh đầy đủ mối quan hệ giữa vận tốc bánh xe, quỹ đạo di chuyển và hướng của robot. Điều này chứng minh rằng các phương trình động học được thiết lập hoàn chỉnh và có thể sử dụng làm cơ sở cho các bước điều khiển thực nghiệm sau này.

Nhìn chung, kết quả mô phỏng cho thấy mô hình robot hai bánh vi sai có **khả năng điều khiển ổn định, sai số thấp và phản ứng nhanh**, đáp ứng tốt yêu cầu nghiên cứu và hoàn toàn khả thi để triển khai trên mô hình thực tế.

# VII. Kết luận và hướng phát triển

## 7.1. Kết luận chung

Đề tài **“Giải thuật điều khiển robot di động UKMarsBot”** đã được triển khai thành công, đạt được các mục tiêu nghiên cứu ban đầu về mô hình hóa, mô phỏng và điều khiển chuyển động của robot. Thông qua việc xây dựng mô hình động học vi sai (Differential Drive Kinematics) và triển khai mô phỏng trên MATLAB/Simulink, hệ thống đã thể hiện khả năng bám quỹ đạo tốt, điều khiển ổn định và sai số nhỏ trong quá trình di chuyển. Giải thuật điều khiển được thiết kế dựa trên nguyên lý điều khiển vận tốc tuyến tính và góc quay, giúp robot di chuyển chính xác theo lộ trình mong muốn.

Kết quả mô phỏng cho thấy sự tương quan cao giữa quỹ đạo thực tế và quỹ đạo lý tưởng, đồng thời các giá trị vận tốc và góc quay đều đạt độ ổn định tốt trong suốt quá trình thử nghiệm. Điều này khẳng định tính đúng đắn của mô hình và khả năng ứng dụng của giải thuật trong thực tiễn. Ngoài ra, việc kết hợp giữa phần cứng và phần mềm, đặc biệt là ứng dụng điều khiển từ xa qua Bluetooth, giúp hệ thống trở nên linh hoạt, thân thiện và dễ thao tác đối với người sử dụng.

**Ưu điểm của hệ thống** là tính chính xác cao trong điều khiển, khả năng mô phỏng trực quan và dễ dàng mở rộng. Mô hình toán học được xây dựng trên cơ sở lý thuyết chuẩn, các thông số có thể điều chỉnh linh hoạt để phù hợp với nhiều điều kiện vận hành khác nhau. Hệ thống mô phỏng giúp quan sát toàn bộ quá trình chuyển động, từ vận tốc, sai số đến quỹ đạo, hỗ trợ tốt cho quá trình phân tích và tối ưu hóa giải thuật điều khiển.

Tuy nhiên, **hệ thống vẫn tồn tại một số hạn chế**. Mô hình mô phỏng hiện mới chỉ xem xét điều kiện lý tưởng, chưa tính đến các yếu tố thực tế như trượt bánh, ma sát hay nhiễu đo từ cảm biến. Sai số vẫn xuất hiện khi robot thay đổi hướng di chuyển đột ngột, cho thấy cần tinh chỉnh thêm bộ điều khiển hoặc tích hợp cơ chế phản hồi. Ngoài ra, giải thuật hiện tại chưa khai thác các phương pháp điều khiển hiện đại như điều khiển mờ, LQR hay điều khiển dự đoán mô hình (MPC), vốn có khả năng cải thiện độ chính xác và ổn định trong các điều kiện phức tạp hơn.

Khi **so sánh với các phương pháp điều khiển khác**, giải thuật hiện tại cho thấy ưu điểm về tính đơn giản, dễ triển khai và hiệu quả trong các bài toán điều khiển cơ bản. Tuy nhiên, so với các phương pháp điều khiển thông minh như điều khiển mờ (Fuzzy Control) hay điều khiển học tăng cường (Reinforcement Learning), hệ thống còn hạn chế về khả năng tự thích nghi và tối ưu hóa quỹ đạo. Điều này mở ra tiềm năng lớn cho việc kết hợp các phương pháp tiên tiến trong các nghiên cứu tiếp theo.

## 7.2. Định hướng cải tiến

Trong giai đoạn phát triển tiếp theo, hệ thống robot di động **UKMarsBot** sẽ được hướng tới mở rộng khả năng hoạt động từ mô phỏng sang mô hình thực tế, đồng thời tích hợp thêm các thành phần phần cứng và thuật toán thông minh để nâng cao hiệu quả điều khiển. Một số định hướng chính được đề xuất bao gồm:

* **Tích hợp cảm biến** (encoder, IMU, siêu âm hoặc LIDAR) để tăng độ chính xác trong việc xác định vị trí, vận tốc và hướng di chuyển, đồng thời hỗ trợ robot phát hiện và tránh vật cản trong môi trường thực.
* **Phát triển chế độ điều khiển bán tự động hoặc tự hành**, trong đó robot có thể tự điều hướng, tự điều chỉnh quỹ đạo và ra quyết định dựa trên dữ liệu thu được từ cảm biến.
* **Kết nối mở rộng qua WiFi hoặc IoT**, cho phép giám sát và điều khiển robot từ xa thông qua mạng nội bộ hoặc Internet, phục vụ cho các ứng dụng công nghiệp và giáo dục.
* **Ứng dụng trí tuệ nhân tạo (AI)**, đặc biệt là các thuật toán học tăng cường (Reinforcement Learning) hoặc mạng nơ-ron nhân tạo (Neural Network), giúp robot học và tối ưu hóa hành vi di chuyển dựa trên trải nghiệm thực tế.
* **Nghiên cứu kết hợp mô phỏng và thực nghiệm**, nhằm so sánh, đánh giá và hiệu chỉnh mô hình động học, đảm bảo tính nhất quán giữa kết quả mô phỏng và vận hành thực tế.

Những định hướng cải tiến này không chỉ giúp hệ thống **UKMarsBot** trở thành một nền tảng nghiên cứu và giảng dạy hiệu quả trong lĩnh vực robot di động, mà còn mở ra khả năng ứng dụng rộng rãi trong các hệ thống robot thông minh, robot giao hàng, hoặc xe tự hành quy mô nhỏ trong tương lai.

# Tài liệu tham khảo

[1] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, 2nd ed. Cambridge, MA: MIT Press, 2011.

[2] B. Siciliano and O. Khatib (Eds.), *Springer Handbook of Robotics*, 2nd ed. Springer, 2016.

[3] J. Borenstein, H. R. Everett, L. Feng, and D. Wehe, “Mobile robot positioning: Sensors and techniques,” *Journal of Robotic Systems*, vol. 14, no. 4, pp. 231–249, 1997.

[4] M. J. Mataric, “The Robotics Primer,” *MIT Press*, 2007.

[5] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. Hoboken, NJ: Wiley, 2006.

[6] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*, 2nd ed. Springer, 2017.

[7] N. T. Trung and P. T. Khoa, “Thiết kế và điều khiển robot di động hai bánh sử dụng Arduino và cảm biến siêu âm,” *Tạp chí Khoa học và Công nghệ Đại học Bách Khoa Hà Nội*, vol. 62, no. 3, pp. 45–52, 2020.

[8] H. T. Nguyen, T. D. Pham, and V. T. Nguyen, “Modeling and control of differential drive mobile robot using MATLAB/Simulink,” in *Proc. IEEE Int. Conf. on Mechatronics and Automation (ICMA)*, Beijing, China, 2015, pp. 2105–2110.

[9] A. Kumar and D. R. Parhi, “Intelligent navigation of mobile robot using hybrid neuro-fuzzy controller,” *Neural Computing and Applications*, vol. 23, no. 6, pp. 1429–1446, 2013.

[10] T. H. Vo, “Development of Bluetooth-based control system for Arduino robot,” *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 10, no. 4, pp. 100–106, 2019.