

**TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH**  
**KHOA ĐIỆN – ĐIỆN TỬ**  
**BỘ MÔN ĐIỀU KHIỂN TỰ ĐỘNG**

**TRẦN MINH QUANG**  
**NGUYỄN DUY TÂN**

**LUẬN VĂN TỐT NGHIỆP**  
**ĐIỀU KHIỂN MÁY BAY CÁNH BẰNG**  
**BÁM THEO QUỸ ĐẠO**

**KỸ SƯ NGÀNH KỸ THUẬT ĐIỀU KHIỂN & TỰ ĐỘNG HÓA**

**TP. HỒ CHÍ MINH, 2018**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH**

**KHOA ĐIỆN – ĐIỆN TỬ**

**BỘ MÔN ĐIỀU KHIỂN TỰ ĐỘNG**

**TRẦN MINH QUANG – 1413112**

**NGUYỄN DUY TÂN – 1413450**

**LUẬN VĂN TỐT NGHIỆP**

**ĐIỀU KHIỂN MÁY BAY CÁNH BẰNG**

**BÁM THEO QUỸ ĐẠO**

**TRAJECTORY-TRACKING CONTROL OF AN AIRPLANE MODEL**

**KỸ SƯ NGÀNH KỸ THUẬT ĐIỀU KHIỂN & TỰ ĐỘNG HÓA**

**GIẢNG VIÊN HƯỚNG DẪN**

**TS. NGUYỄN VĨNH HẢO**

**TP. HỒ CHÍ MINH, 2018**

TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH

**KHOA ĐIỆN - ĐIỆN TỬ**

**BỘ MÔN: ĐIỀU KHIỂN TỰ ĐỘNG**

---

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc

---

TP. HCM, ngày....tháng.....năm.....

**NHẬN XÉT LUẬN VĂN TỐT NGHIỆP  
CỦA CÁN BỘ HƯỚNG DẪN**

**Tên luận văn:**

**ĐIỀU KHIỂN MÁY BAY CÁNH BẰNG BẮM THEO QUỹ ĐẠO**

**Nhóm Sinh viên thực hiện:**

Trần Minh Quang

1413112

Nguyễn Duy Tân

1413450

**Cán bộ hướng dẫn:**

TS. Nguyễn Vĩnh Hảo

**Đánh giá Luận văn**

1. Về cuốn báo cáo:

Số trang

\_\_\_\_\_

Số chương

\_\_\_\_\_

Số bảng số liệu

\_\_\_\_\_

Số hình vẽ

\_\_\_\_\_

Số tài liệu tham khảo

\_\_\_\_\_

Sản phẩm

\_\_\_\_\_

- Một số nhận xét về hình thức cuốn báo cáo:

2. Về nội dung luận văn:

3. Về tính ứng dụng:

4. Về thái độ làm việc của sinh viên:

**Đánh giá chung:** Luận văn đạt/không đạt yêu cầu của một luận văn tốt nghiệp kỹ sư,  
xếp loại Giỏi/ Khá/ Trung bình

**Điểm từng sinh viên:**

Trần Minh Quang: ...../10

Nguyễn Duy Tân: ...../10

**Người nhận xét**

(Ký tên và ghi rõ họ tên)

TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH

**KHOA ĐIỆN - ĐIỆN TỬ**

**BỘ MÔN: ĐIỀU KHIỂN TỰ ĐỘNG**

---

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc

---

*TP. HCM, ngày....tháng.....năm.....*

**NHẬN XÉT LUẬN VĂN TỐT NGHIỆP  
CỦA CÁN BỘ PHẢN BIỆN**

**Tên luận văn:**

**ĐIỀU KHIỂN MÁY BAY CÁNH BẰNG BẮM THEO QUỹ ĐẠO**

**Nhóm Sinh viên thực hiện:**

**Cán bộ phản biện:**

Trần Minh Quang

1413112

Nguyễn Duy Tân

1413450

**Đánh giá Luận văn**

1. Về cuốn báo cáo:

Số trang \_\_\_\_\_ Số chương \_\_\_\_\_

Số bảng số liệu \_\_\_\_\_ Số hình vẽ \_\_\_\_\_

Số tài liệu tham khảo \_\_\_\_\_ Sản phẩm \_\_\_\_\_

- Một số nhận xét về hình thức cuốn báo cáo:

2. Về nội dung luận văn:

3. Về tính ứng dụng:

4. Về thái độ làm việc của sinh viên:

**Đánh giá chung:** Luận văn đạt/không đạt yêu cầu của một luận văn tốt nghiệp kỹ sư,  
xếp loại Giỏi/ Khá/ Trung bình

**Điểm từng sinh viên:**

Trần Minh Quang: ...../10

Nguyễn Duy Tân: ...../10

**Người nhận xét**

(Ký tên và ghi rõ họ tên)

## **LỜI CẢM ƠN**

Sau gần 6 tháng thực hiện, trải qua một quá trình dài tìm hiểu lí thuyết, vận dụng kiến thức đã học, nhiều kinh nghiệm thực tế cùng những lần thử nghiệm đầy vất vả, hiện tại chúng em đã hoàn thành đề tài luận văn tốt nghiệp của mình. Trong suốt khoảng thời gian đó, chúng em đã được trải nghiệm thực tế với những lí thuyết đã học ở trường, thậm chí những kiến thức từ các nguồn tài liệu khoa học nước ngoài, mà có thể rất khó khăn nếu không có những cách tiếp cận được cung cấp bởi các thầy cô. Đồng thời, chúng em cũng tích lũy được cho riêng mình những kinh nghiệm và kiến thức để có thể trở thành một kĩ sư giỏi trong bước đường sự nghiệp sắp tới.

Để có được kết quả như trên, chúng em thật sự trân trọng sự ủng hộ về mặt tinh thần và vật chất từ phía gia đình; sự đồng hành, chia sẻ và giúp đỡ từ bạn bè, các anh chị các khóa; những kiến thức nền tảng và phương pháp tiếp cận vấn đề một cách khoa học từ các thầy cô trường Đại học Bách Khoa TP. HCM nói chung và các thầy cô khoa Điện – Điện tử nói riêng. Và cuối cùng, chúng em cũng gửi lời cảm ơn đến giáo viên hướng dẫn – thầy Nguyễn Vĩnh Hảo, người đã nhiệt tình giúp đỡ, đưa ra những ý tưởng phát triển, cung cấp những giải pháp để chúng em có được kết quả như ngày hôm nay.

Trân trọng

TP. HCM, ngày 10 tháng 6 năm 2018

Sinh viên thực hiện

Trần Minh Quang – Nguyễn Duy Tân

TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH

**KHOA ĐIỆN - ĐIỆN TỬ**

**BỘ MÔN: ĐIỀU KHIỂN TỰ ĐỘNG**

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc

TP. HCM, ngày....tháng.....năm.....

## ĐỀ CƯƠNG CHI TIẾT

<b>TÊN ĐỀ TÀI: ĐIỀU KHIỂN MÁY BAY CÁNH BẰNG BÁM THEO QUỸ ĐẠO</b>
<b>Cán bộ hướng dẫn: TS. Nguyễn Vĩnh Hảo</b>
<b>Thời gian thực hiện:</b> Từ ngày 15/01/2018 đến ngày 10/06/2018
<b>Sinh viên thực hiện:</b>  <b>Trần Minh Quang - 1413112</b>  <b>Nguyễn Duy Tân - 1413450</b>
<b>Nội dung đề tài:</b>  <i>Mục Tiêu:</i>  Xây dựng một hệ thống điều khiển nền tảng, cơ bản có thể giữ máy bay ổn định độ cao, cân bằng, thực hiện được các trạng thái cơ bản của máy như bay thẳng, bay rẽ trái, rẽ phải và kết hợp những trạng thái này điều khiển máy bay bám theo một số quỹ đạo đơn giản.  <i>Phương pháp thực hiện:</i>  Mô phỏng thuật toán trên Matlab và nhúng xuống board mạch vi điều khiển kết hợp CPLD. Dùng cảm biến chính IMU/GPS, lưu dữ liệu trên thẻ nhớ SD card và truyền dữ liệu về máy tính thông qua RF.  <b>Kết quả mong đợi:</b> <ul style="list-style-type: none"><li>■ Xây dựng mô hình máy bay cánh bằng với hai chế độ điều khiển bằng tay và điều khiển tự động.</li><li>■ Mô phỏng và nhúng thành công thuật toán điều khiển máy bay cánh bằng bám theo quỹ đạo xuống board mạch điều khiển.</li></ul>



**Kế hoạch thực hiện:**

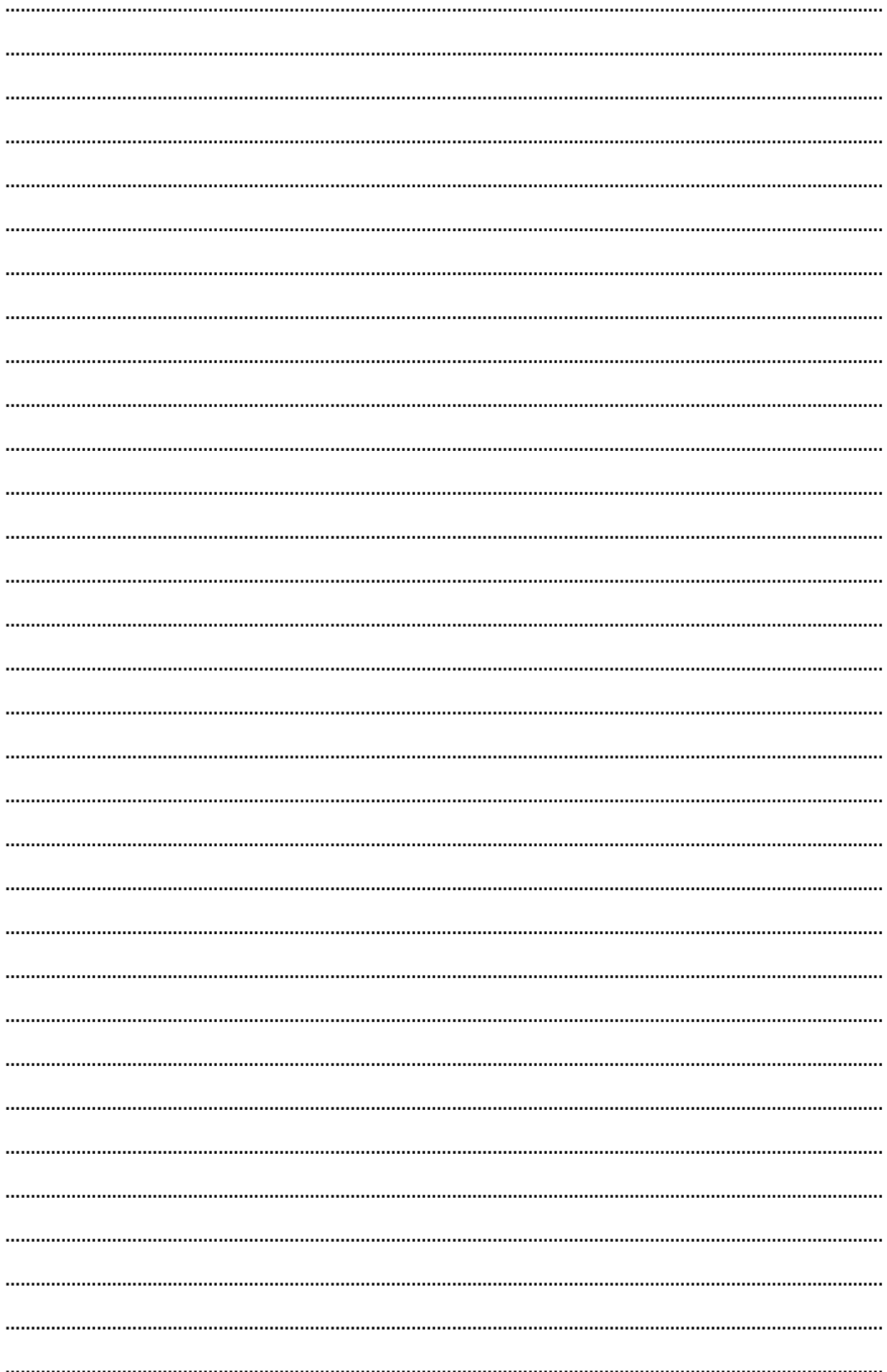
<i>STT</i>	<i>Phân công</i>	<i>Công việc</i>	<i>Thời gian</i>
1	Trần Minh Quang Nguyễn Duy Tân	Xây dựng mô hình máy bay	15/01/2018 – 15/02/2018
2	Trần Minh Quang	Thiết kế phần cứng mạch ra chân và điều khiển	15/01/2018 – 01/02/2018
3	Trần Minh Quang Nguyễn Duy Tân	Mô phỏng thuật toán Stanley trên Matlab	01/02/2018 – 15/02/2018
4	Nguyễn Duy Tân	Lập trình trên CPLD	15/02/2018 – 01/03/2018
5	Trần Minh Quang Nguyễn Duy Tân	Xây dựng giải thuật xử lí, lưu trữ và quản lí dữ liệu từ cảm biến	15/02/2018 – 01/03/2018
6	Trần Minh Quang Nguyễn Duy Tân	Nhúng thuật toán Stanley xuống vi điều khiển STM32F4	01/03/2018 – 15/03/2018
7	Trần Minh Quang Nguyễn Duy Tân	Xây dựng giải thuật bay cân bằng.	15/03/2018 – 15/04/2018
8	Trần Minh Quang Nguyễn Duy Tân	Xây dựng giải thuật bay điểm - điểm.	15/04/2018 – 01/05/2018
9	Trần Minh Quang Nguyễn Duy Tân	Xây dựng giải thuật rẽ trái và rẽ phải.	01/05/2018 – 15/05/2018
10	Trần Minh Quang Nguyễn Duy Tân	Xây dựng giải thuật bám theo quỹ đạo	15/05/2018 – 01/06/2018
11	Trần Minh Quang Nguyễn Duy Tân	Cải thiện tính ổn định và giảm sai số.	01/06/2018 – 10/06/2018

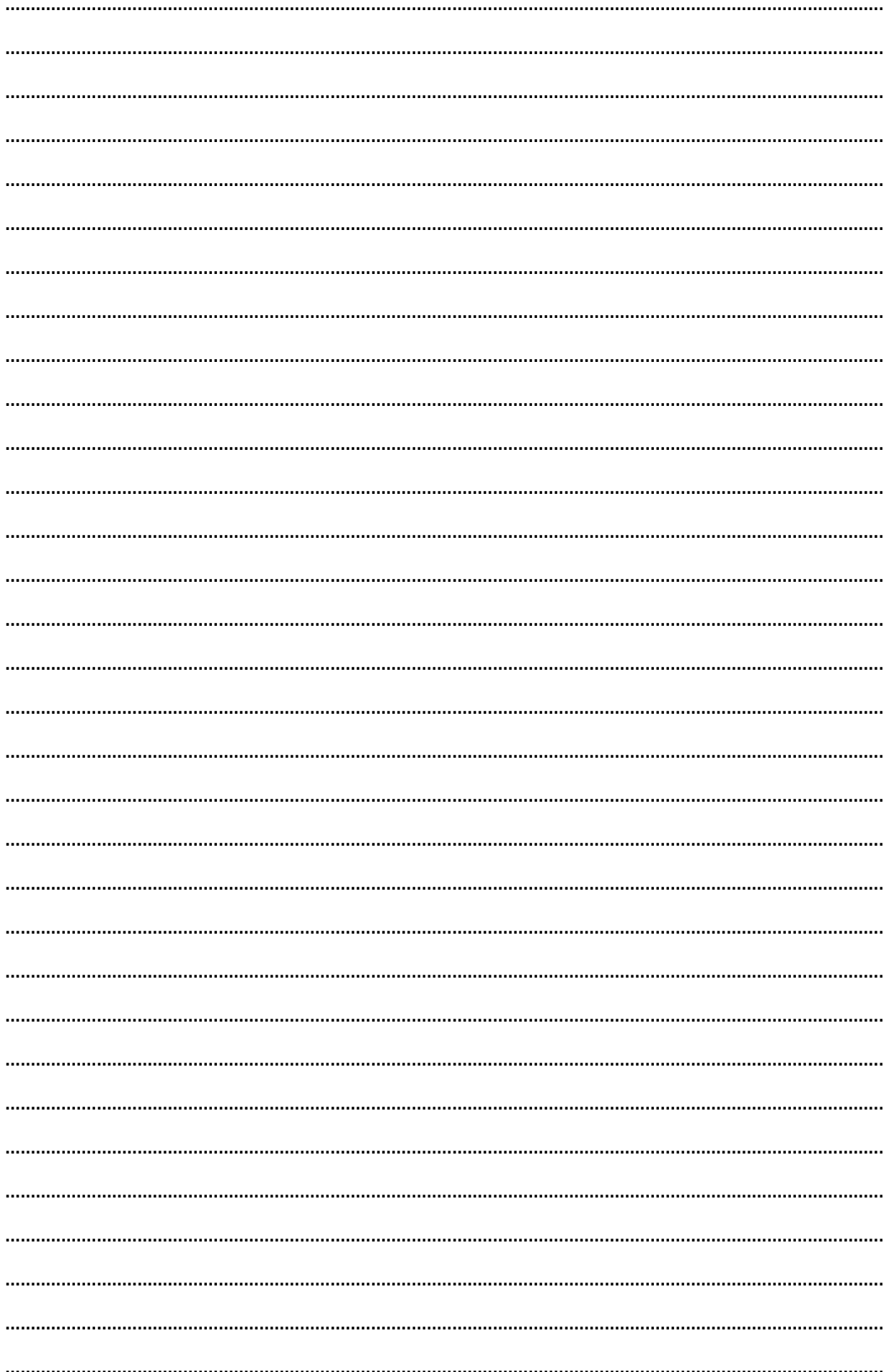
<p><b>Xác nhận của Cán bộ hướng dẫn</b></p> <p>(Ký tên và ghi rõ họ tên)</p>	<p>TP. HCM, ngày....tháng .....năm.....</p> <p><b>Sinh viên</b></p> <p>(Ký tên và ghi rõ họ tên)</p>
--	--

## **DANH SÁCH HỘI ĐỒNG BẢO VỆ LUẬN VĂN**

Hội đồng chấm luận văn tốt nghiệp, thành lập theo Quyết định số .....  
ngày ..... của Hiệu trưởng Trường Đại học Bách khoa TP.HCM.

1. .... - Chủ tịch.
2. .... - Thư ký.
3. .... - Ủy viên.





## MỤC LỤC

TÓM TẮT LUẬN VĂN.....	1
Chương 1. TỔNG QUAN ĐỀ TÀI .....	2
1.1. Tình hình nghiên cứu trong và ngoài nước .....	2
1.2. Lí do chọn đề tài.....	2
1.3. Mục đích nghiên cứu.....	3
Chương 2. CƠ SỞ LÝ THUYẾT.....	4
2.1. Nguyên lí hoạt động của máy bay cánh bằng .....	4
2.2. Giới thiệu sơ lược thuật toán Stanley .....	7
2.3. Giới thiệu sơ lược thuật toán PID .....	8
Chương 3. GIỚI THIỆU MÔ HÌNH, THIẾT BỊ VÀ THIẾT KẾ PHẦN CỨNG.....	10
3.1. Giới thiệu các loại mô hình và lí do lựa chọn.....	10
3.2. Giới thiệu các thiết bị phần cứng .....	12
3.2.1. Remote Control và Receiver .....	12
3.2.2. RF .....	12
3.2.3. CPLD .....	13
3.2.4. Vi điều khiển (MCU) .....	13
3.2.4.1. Giới thiệu sơ lược về ngoại vi UART.....	13
3.2.4.2. Giới thiệu sơ lược về ngoại vi DMA .....	14
3.2.4.3. Giới thiệu sơ lược về ngoại vi PWM .....	14
3.2.4.4. Giới thiệu sơ lược về ngoại vi FSMC.....	14
3.2.4.5. Giới thiệu sơ lược về ngoại vi SDIO .....	15
3.2.5. Các thiết bị chấp hành.....	16
3.2.6. Cảm biến IMU và GPS.....	17

3.2.6.1.	Khái quát cấu trúc cảm biến IMU và GPS .....	17
3.2.6.2.	Cấu trúc xuất dữ liệu của cảm biến IMU và GPS.....	18
3.3.	Thiết kế mạch vi điều khiển, CPLD và mạch ra chân.....	21
3.3.1.	Mạch vi điều khiển và CPLD .....	21
3.3.2.	Mạch ra chân.....	22
Chương 4.	MÔ PHỎNG THUẬT TOÁN STANLEY TRÊN MATLAB .....	24
4.1.	Những thông số cần thiết và những hiệu chỉnh của thuật toán .....	24
4.2.	Thông số cần thiết để mô phỏng .....	28
4.3.	Kết quả mô phỏng và nhận xét tham số $k$ .....	30
Chương 5.	XÂY DỰNG HỆ THỐNG ĐIỀU KHIỂN .....	32
5.1.	Tổng quan hệ thống .....	32
5.2.	Xây dựng hệ thống điều khiển trên CPLD .....	33
5.2.1.	Chuyển chế độ Auto/ Manual trên CPLD .....	33
5.2.2.	Sơ đồ phục vụ việc gửi dữ liệu cho ngoại vi FSMC trên MCU .....	34
5.3.	Xử lý, quản lý và lưu trữ dữ liệu .....	35
5.3.1.	Xử lý dữ liệu đọc về từ cảm biến IMU – GPS .....	35
5.3.2.	Tạo khung dữ liệu để lưu trữ thẻ nhớ và truyền qua RF .....	36
5.4.	Xây dựng hệ thống điều khiển trên vi điều khiển.....	37
5.4.1.	Tổng quan cấu trúc hệ thống trên vi điều khiển.....	37
5.4.2.	Trạng thái Calib .....	38
5.4.3.	Tổng quan hệ thống điều khiển trong chế độ Auto .....	41
5.4.4.	Áp dụng thuật toán PID vào hệ thống.....	42
5.4.5.	Giải thuật quyết định và quản lý trạng thái bay trong khối State Decision .....	44

5.4.5.1. Cơ chế bay dùng để điều khiển máy bay .....	44
5.4.5.2. Tổng quan máy trạng thái trong khối State Decision .....	45
Chương 6. KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN .....	52
6.1. Kết quả mô hình thực tế.....	52
6.2. Kế hoạch quá trình bay thí nghiệm .....	53
6.3. Kết quả và nhận xét.....	55
6.3.1. Kết quả khi máy bay ở trạng thái STABLE.....	55
6.3.2. Kết quả khi máy bay ở trạng thái TRACK.....	56
6.3.3. Kết quả khi máy bay ở trạng thái TURN .....	57
6.3.4. Kết quả khi máy bay thực hiện bám theo quỹ đạo.....	59
6.3.4.1. Đáp ứng máy bay khi bám quỹ đạo.....	59
6.3.4.2. Một số hình ảnh quỹ đạo bay được.....	64
6.4. Hướng phát triển.....	67
PHỤ LỤC.....	69
TÀI LIỆU THAM KHẢO .....	74



## DANH MỤC HÌNH VẼ

Hình 2.1: Mô hình máy bay cánh bằng [13] .....	4
Hình 2.2: Nguyên lí hoạt động của Throttle [14] .....	5
Hình 2.3: Nguyên lí hoạt động của Elevator .....	5
Hình 2.4: Nguyên lí hoạt động của Aileron .....	6
Hình 2.5: Nguyên lí hoạt động của Rudder [15] .....	6
Hình 2.6: Mô hình thuật toán Stanley áp dụng trên robot Stanley.....	7
Hình 2.7: Sơ đồ khối thuật toán PID.....	8
Hình 3.1: Mô hình tàu lượn (Surfer) [4] .....	10
Hình 3.2: Mô hình máy bay phản lực [4] .....	11
Hình 3.3: Mô hình máy bay cánh trên [4] .....	11
Hình 3.4: Remote Control và Receiver .....	12
Hình 3.5: Module truyền nhận RF .....	12
Hình 3.6: Vi điều khiển STM32F407VTG6 và CPLD MAX II .....	13
Hình 3.7: Sơ đồ hoạt động của các chân tín hiệu ngoại vi FSMC [8] .....	15
Hình 3.8: Độ rộng xung điều khiển xuất ra từ receiver .....	16
Hình 3.9: Động cơ brushless và ESC.....	17
Hình 3.10: IMU-9DOF được chế tạo tại PTN Bộ môn Tự Động ĐHBK TPHCM	17
Hình 3.11: Sơ đồ khối cảm biến IMU.....	18
Hình 3.12: Dữ liệu xuất ra từ cảm biến IMU .....	18
Hình 3.13: Cấu trúc một mẫu dữ liệu trả về từ cảm biến IMU.....	18
Hình 3.14: Sơ đồ các header dùng để nạp chương trình từ mạch nạp cho vi điều khiển và CPLD.....	21
Hình 3.15: Hình ảnh mạch điều khiển (được thiết kế ở phòng thí nghiệm Bộ môn Tự động) .....	21
Hình 3.16: Sơ đồ mạch ra chân.....	22
Hình 3.17: Mạch ra chân .....	23
Hình 4.1: Quỹ đạo di chuyển mong muốn.....	24

Hình 4.2: Xác định độ lớn góc $\psi$ trên hệ trục tọa độ .....	25
Hình 4.3: Xác định sai số bám quỹ đạo x .....	25
Hình 4.4: Xác định góc điều hướng trên mặt phẳng tọa độ .....	26
Hình 4.5: Dấu của thành phần $\arctan \frac{k.x(t)}{u(t)}$ ở các vị trí khác nhau so với quỹ đạo .....	27
Hình 4.6: Chất điểm chuyển đoạn thẳng quỹ đạo để bám .....	28
Hình 4.7: Khái quát mô hình mô phỏng thuật toán Stanley trên Matlab .....	30
Hình 4.8: Kết quả mô phỏng với điểm xuất phát (0, 4), $v = 4$ , $k = 40$ , $T = 0.0531$	
Hình 4.9: Kết quả mô phỏng với điểm xuất phát (2, -10), $v = 4$ , $k = 4$ , $T = 0.05$ .....	31
Hình 5.1: Khái quát sơ đồ hệ thống .....	32
Hình 5.2: Sơ đồ khối chức năng chuyển chế độ trên CPLD .....	33
Hình 5.3: Điều kiện chọn xung tín hiệu.....	34
Hình 5.4: Sơ đồ khối đọc giá trị độ rộng xung từ CPLD .....	35
Hình 5.5: Cấu trúc một mẫu dữ liệu từ cảm biến.....	35
Hình 5.6: Quá trình xử lý dữ liệu từ cảm biến IMU - GPS.....	36
Hình 5.7: Một đoạn dữ liệu mẫu ghi vào thẻ nhớ .....	37
Hình 5.8: Cấu trúc khung dữ liệu ghi vào thẻ nhớ .....	37
Hình 5.9: Lưu đồ giải thuật hệ thống .....	37
Hình 5.10: Các vị trí tương đối của cặp tọa độ liên tiếp .....	39
Hình 5.11: Hai đề xuất về cách chuyển đoạn thẳng quỹ đạo .....	40
Hình 5.12: Sơ đồ tổng quan chế độ Auto .....	41
Hình 5.13: Sơ đồ thuật toán PID có hiệu chỉnh dành cho góc Roll và góc Pitch .....	42
Hình 5.14: Cơ chế rẽ trái/ phải của máy bay.....	44
Hình 5.15: Sơ đồ máy trạng thái trong khối State Decision.....	45
Hình 5.16: Sơ đồ trạng thái Turn Right và Turn Left.....	46
Hình 5.17: Sơ đồ khối trạng thái State_Decision.....	47

Hình 5.18: Sơ đồ hoạt động của khối State Distribute .....	48
Hình 5.19: Cách xác định sai số góc Yaw .....	49
Hình 6.1: Các thiết bị điện bên trong khoang máy bay .....	52
Hình 6.2: Mô hình máy bay được lắp đặt hoàn thiện.....	53
Hình 6.3: Quỹ đạo mong muốn nhìn từ vệ tinh.....	54
Hình 6.4: Kết quả góc Roll và Pitch ở trạng thái STABLE .....	55
Hình 6.5: Kết quả góc Roll và Pitch ở trạng thái TRACK .....	56
Hình 6.6: Kết quả góc Roll, Pitch và Yaw ở trạng thái TURN.....	58
Hình 6.7: Kết quả góc Roll và Pitch khi bay bám quỹ đạo .....	59
Hình 6.8: Kết quả góc Yaw và sai số góc Yaw khi bay bám quỹ đạo .....	60
Hình 6.9: Sự thay đổi trạng thái và đoạn thẳng quỹ đạo khi bay bám quỹ đạo .....	61
Hình 6.10: Kết quả bay quỹ đạo .....	67
Hình PL.1: Sơ đồ thiết kế mạch điều khiển – A.....	69
Hình PL.2: Sơ đồ thiết kế mạch điều khiển – B.....	70
Hình PL.3: Sơ đồ thiết kế mạch ra chân – A.....	71
Hình PL.4: Sơ đồ thiết kế mạch ra chân – B.....	72
Hình PL.5: Máy bay lúc cất cánh.....	73
Hình PL.6: Máy bay đang bám quỹ đạo trên không.....	73

## **DANH MỤC BẢNG**

Bảng 3.1: Các chân được sử dụng của ngoại vi FSMC [8].....	15
Bảng 3.2: Cấu trúc dữ liệu chuỗi GPGGA .....	20
Bảng 3.3: Cấu trúc dữ liệu GPS của chuỗi GPVTG.....	21

## DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Từ gốc	Ý nghĩa
ANSI	American National Standards Institute	Viện Tiêu chuẩn Quốc gia Hoa Kỳ
CMD	Command	Chân xuất lệnh của chuẩn SDIO
CPLD	Complex Programmable Logic Device	Thiết bị logic phức hợp lập trình được
CPU	Central Processing Unit	Đơn vị xử lý trung tâm
DMA	Direct Memory Access	Truy cập bộ nhớ trực tiếp
DO	Data Output	Chân xuất dữ liệu của chuẩn SDIO
ESC	Electronic Speed Control	Bộ điều khiển tốc độ động cơ brushless
FPGA	Field-programmable Gate Array	Mạch logic tích hợp cỡ lớn có khả năng lập trình được
FSMC	Flexible Static Memory ContRoller	Bộ điều khiển bộ nhớ tĩnh linh hoạt
GPS	Global Positioning System	Hệ thống Định vị Toàn cầu
IMU	Inertial Measurement Unit	Đơn vị Đo lường Quán tính
MCU	MicrocontRoller Unit	Vi điều khiển
PC	Personal Computer	Máy tính cá nhân
PID	Proportional Integral Derivative	Bộ điều khiển tỉ lệ tích phân vi phân

PWM	Pulse-width Modulation	Điều chế độ rộng xung
RC	Remote Control	Điều khiển từ xa
RF	Radio Frequency	Tần số vô tuyến
RX	Receive	Dây nhận của UART
SCK	Synchronous Clock	Chân cấp xung clock đồng bộ của chuẩn SDIO
SDIO	Secure Digital Input/ Output	Chuẩn giao tiếp thẻ nhớ trên STM32F4
TX	Transmit	Dây truyền của UART
UART	Universal Asynchronous Receiver-Transmitter	Bộ truyền nhận bất đồng bộ đa năng
UAV	Unmanned Aerial Vehicle	Phương tiện bay không người lái

## **TÓM TẮT LUẬN VĂN**

Luận văn trình bày những vấn đề sau:

1. Thiết kế mạch nguồn và đệm tín hiệu cho board mạch vi điều khiển đồng thời xây dựng, lựa chọn mô hình máy bay đáp ứng được một số điều kiện thời tiết, khối lượng các board mạch cũng như tốt nhất cho giải thuật điều khiển.
2. Tìm hiểu nguyên lý hoạt động của máy bay cánh bằng. Từ đó tiến hành bay thử nghiệm thực tế và lựa chọn nguyên lý điều khiển phù hợp.
3. Tìm hiểu thuật toán Stanley, tiến hành mô phỏng thử nghiệm trên Matlab
4. Xây dựng giải thuật điều khiển máy bay theo mô hình máy trạng thái kết hợp với thuật toán Stanley.
5. Nhúng toàn bộ giải thuật điều khiển lên vi điều khiển STM32F407VGT6 đồng thời giao tiếp cảm biến IMU/GPS, xử lý, quản lý và lưu trữ dữ liệu.
6. Tiến hành bay thử nghiệm và đánh giá kết quả.

## **Chương 1. TỔNG QUAN ĐỀ TÀI**

### **1.1. Tình hình nghiên cứu trong và ngoài nước**

UAV (Unmanned aerial vehicle) hay còn được gọi là máy bay không người lái đang ngày một phát triển với tốc độ cực kì cao bởi những ứng dụng cực kì to lớn mà nó mang lại. Với những ưu điểm siêu việt như có thể điều khiển từ xa, tự động thực hiện một số công việc không cần tác động của con người, độ an toàn và chính xác cao mà ngày nay các UAV đang được áp dụng trong nhiều lĩnh vực dân sự, quân sự, kinh tế và giải trí...

Trên thế giới hiện đang có rất nhiều phòng nghiên cứu đẩy mạnh xây dựng các hệ thống điều khiển máy bay nhằm tạo ra các UAV ngày càng có khả năng thực hiện các công việc phức tạp và giảm thiểu tối đa những sai sót của con người. Ở các nước phát triển các UAV được ứng dụng mạnh mẽ trong quân sự, nhằm thực hiện các mục tiêu quan trọng như trinh thám, đi vào những vùng có nồng độ phóng xạ cao... Đây đều là những công việc hết sức nguy hiểm và có thể ảnh hưởng đến tính mạng con người.

Trong nước, Chính phủ cũng đã và đang đẩy mạnh phát triển các mô hình UAV nhằm đáp ứng những nhu cầu của đất nước. Không chỉ thế, các UAV đang ngày càng được ưa chuộng trong cuộc sống hằng ngày như máy bay flycam, các hội yêu thích và chơi UAV vào cuối tuần hay việc điều khiển các mô hình máy bay đang là một đề tài có một sức hút to lớn đối với các bạn sinh viên ở các trường đại học công nghệ kĩ thuật [1].

### **1.2. Lí do chọn đề tài**

Hòa cùng với không khí nghiên cứu phát triển các trạm mặt đất cũng như các hệ thống điều khiển máy bay cánh bằng trong những năm gần đây của thầy và trò trường Đại Học Bách Khoa. Cùng với đó là mong muốn xây dựng được một hệ thống điều khiển máy bay bám theo quỹ đạo hoàn chỉnh mang tính nền tảng có thể phát triển được và có thể kết hợp với các trạm mặt đất của các đề tài khóa



trên để thực hiện được một số ứng dụng cơ bản trong đời sống. Chính vì những lí do đó chúng em quyết định thực hiện đề tài: “Điều khiển máy bay cánh bằng bám theo quỹ đạo”.

### **1.3. Mục đích nghiên cứu**

Đề tài hướng tới xây dựng một mô hình cơ bản gồm hai chế độ lái bằng tay và lái tự động có thể chuyển đổi qua lại dễ dàng. Chế độ bằng tay dùng để điều khiển máy bay trong quá trình cất cánh hạ cánh và lấy độ cao ban đầu. Khi đã đạt tới độ cao mong muốn ta tiến hành bật sang chế độ tự động để máy bay bám theo quỹ đạo đã được định sẵn.

Trong phạm vi của luận văn, chúng em tập trung vào hai phần chính:

*Phần 1: Xây dựng mô hình máy bay phù hợp và các mạch điều khiển:*

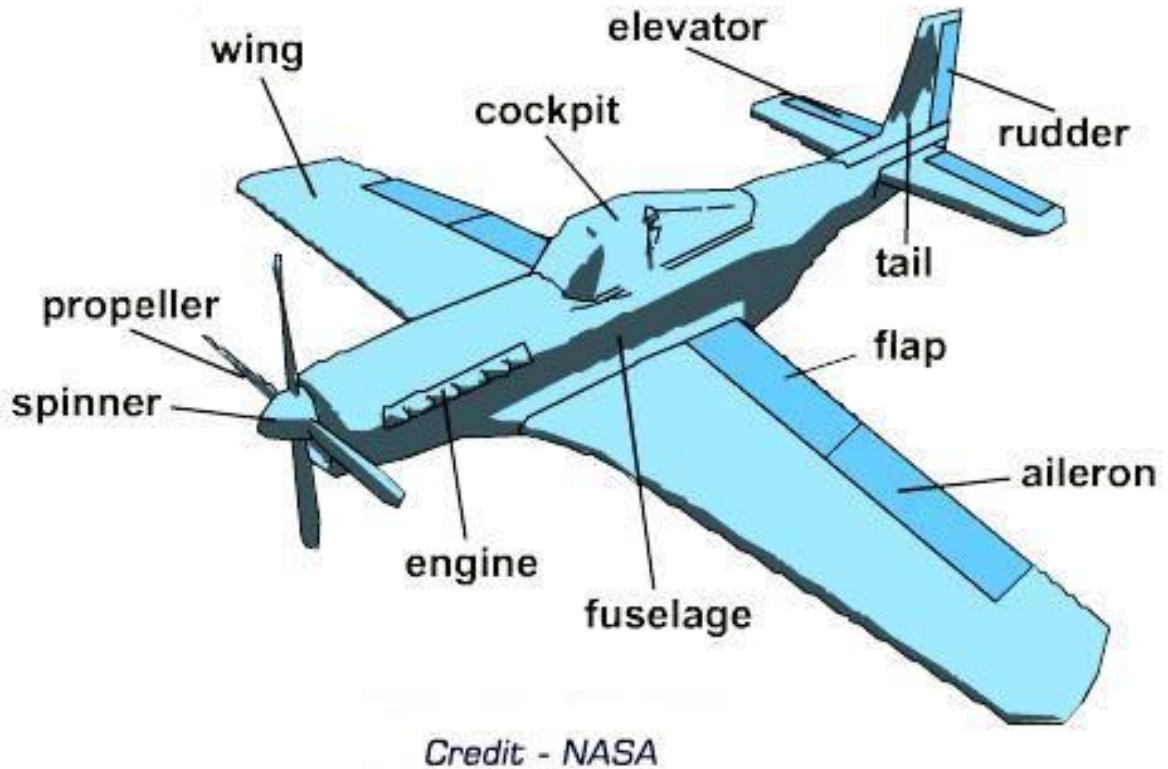
- Lắp ráp và bay thực tế một số mô hình máy bay thực tế và chọn ra mô hình phù hợp nhất.
- Thiết kế driver chuyển đổi mode auto/manual trên nền CPLD
- Thiết kế mạch nguồn và mạch đệm đủ cấp cho mạch điều khiển và các servo

*Phần 2: Xây dựng hệ thống điều khiển bám theo quỹ đạo:*

- Tìm hiểu và mô phỏng thuật toán Stanley trên Matlab
- Xây dựng hệ thống điều khiển máy bay bám quỹ đạo theo mô hình máy trạng thái.
- Giao tiếp cảm biến IMU/GPS, xử lí, lưu trữ và quản lí dữ liệu.
- Thực hiện bay thực tế điều chỉnh thông số cho thích hợp.

## Chương 2. CƠ SỞ LÝ THUYẾT

### 2.1. Nguyên lý hoạt động của máy bay cánh bằng



Hình 2.1: Mô hình máy bay cánh bằng [13]

Máy bay cánh bằng là một loại máy bay có hình dạng như Hình 2.1.

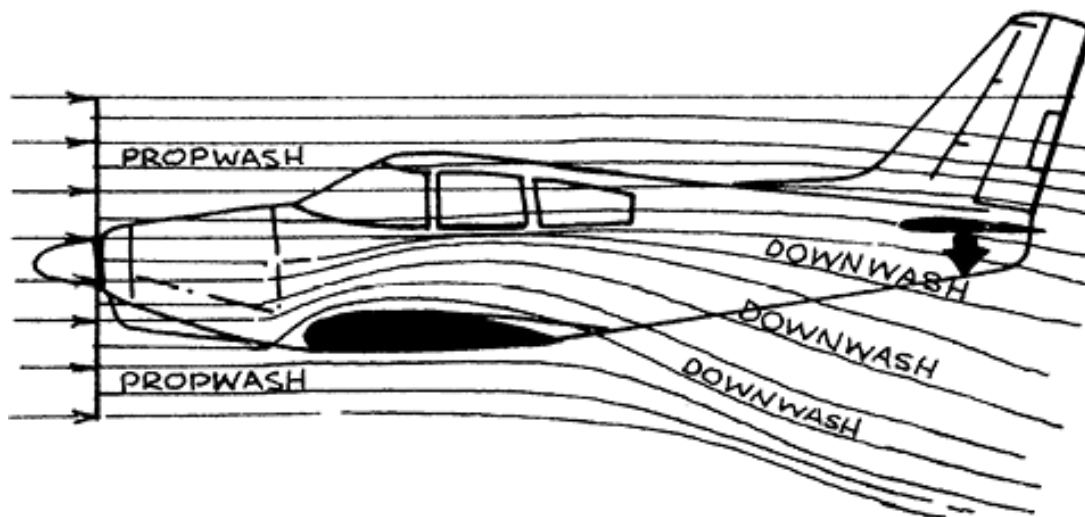
Cấu tạo cơ bản của máy bay cánh bằng gồm có:

- Engine: Động cơ chính, tạo gió cho máy bay
- Wing: Cánh giữa, tạo lực nâng cho máy bay
- Aileron: Cách phụ dùng điều khiển góc Roll
- Elevator: Cách phụ dùng điều khiển góc Pitch
- Rudder: Cánh phụ dùng điều khiển góc Yaw

Về nguyên lý để điều khiển máy bay cánh bằng ta cần những kênh sau:

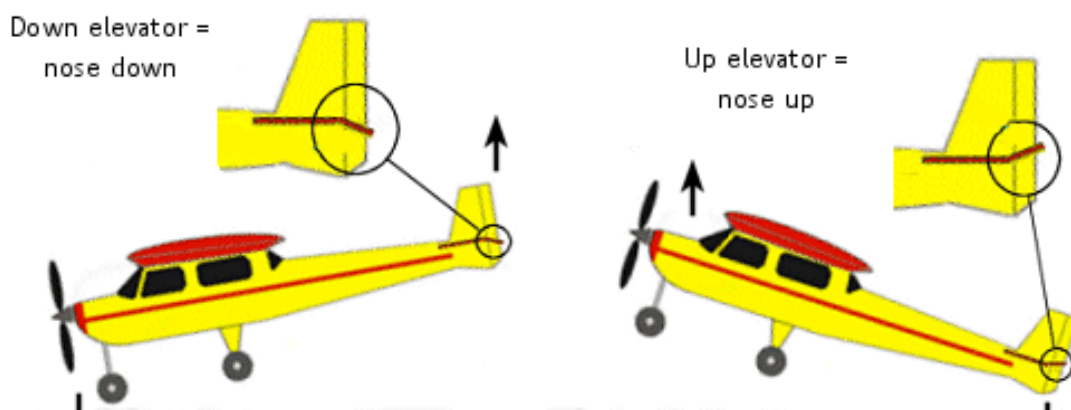
- Throttle (kênh ga): Kênh này điều khiển một động cơ Brushless (động cơ này sẽ được giới thiệu ở mục 3.2.5) có tốc độ quay lớn thổi gió mạnh nhằm

tạo lực kéo giúp máy bay tiến về phía trước. Từ đó gây nên sự chênh lệch áp suất trên cánh giữa, tạo lực nâng cơ bản cho máy bay.



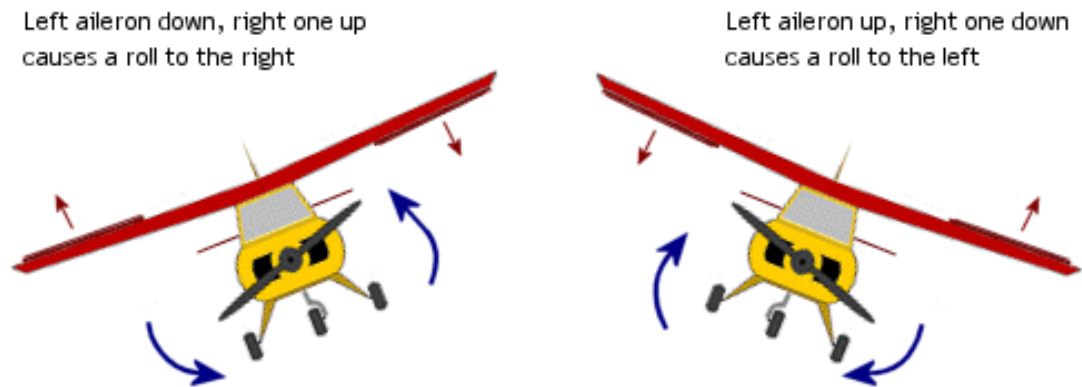
Hình 2.2: Nguyên lí hoạt động của Throttle [14]

- Elevator: Đây là kênh dùng để điều khiển góc Pitch của máy bay. Khi cánh Elevator được nâng lên, thì toàn bộ không khí sẽ tác động lên cánh đuôi theo hướng đi xuống và nâng đầu của máy bay lên. Và ngược lại, khi cánh đuôi hạ xuống lực gió sẽ đẩy phần đuôi lên làm hạ đầu của máy bay xuống. Cánh Elevator đóng vai trò đặc biệt quan trọng trong quá trình cất cánh và hạ cánh của máy bay. Ta cần kết hợp nhuần nhuyễn Elevator và Throttle thì máy bay mới có thể cất cánh và hạ cánh an toàn.



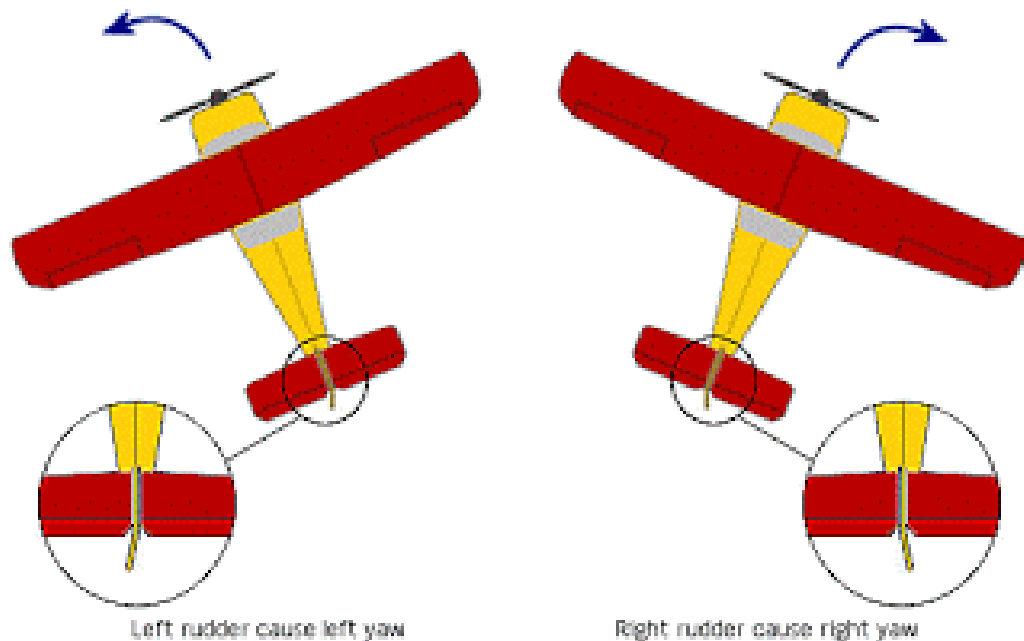
Hình 2.3: Nguyên lí hoạt động của Elevator

- Aileron: Đây là kênh dùng để điều khiển độ nghiêng của máy bay gồm có hai cánh phụ luôn di chuyển ngược chiều nhau, giúp máy bay nghiêng sang trái, sang phải theo một góc mong muốn. Kết hợp với cánh Elevator để giúp máy bay rẽ trái, rẽ phải.



*Hình 2.4: Nguyên lí hoạt động của Aileron*

- Rudder: Đây là kênh dùng để thay đổi hướng bay của máy bay, tuy nhiên ta chỉ nên dùng để thay đổi nhỏ hướng của máy bay hoặc trong một số trường hợp đặc biệt vì khi dùng cánh này rẽ một góc lớn sẽ dẫn tới mất độ cao của máy bay.



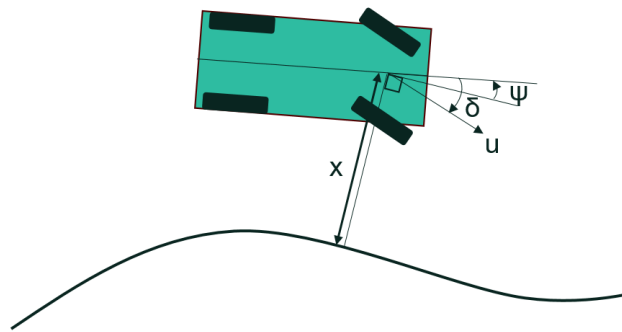
*Hình 2.5: Nguyên lí hoạt động của Rudder [15]*

Như vậy ta đã đi sơ lược qua nguyên lý hoạt động của máy bay cánh bằng. Dựa vào những nguyên lý căn bản ta sẽ xây dựng nên giải thuật điều khiển máy bay bay theo quỹ đạo mong muốn.

## 2.2. Giới thiệu sơ lược thuật toán Stanley

Thuật toán phù hợp cho việc điều khiển các phương tiện (xe đạp, xe hơi, máy bay,...) bám theo quỹ đạo là thuật toán Stanley. Thuật toán Stanley là thuật toán chính được sử dụng để điều khiển góc lái cho robot Stanley, robot giành giải quán quân trong cuộc thi The Grand Challenge được tổ chức bởi Defense Advanced Research Projects Agency (DARPA) vào năm 2003. Thuật toán điều khiển góc lái (steering control) của Stanley là một thuật toán rất hữu ích không những trong điều khiển góc lái của xe hơi mà còn có thể ứng dụng để điều khiển góc lái của máy bay cánh bằng.

Công thức của thuật toán Stanley là một hàm hồi tiếp phi tuyến của sai số quỹ đạo bám (cross track error)  $x$ , được đo từ trung tâm của trục bánh lái trước đến điểm gần nhất trên quỹ đạo, và kết quả trả về sai số quỹ đạo bám sẽ hội tụ về 0 theo quy luật hàm mũ.



Hình 2.6: Mô hình thuật toán Stanley áp dụng trên robot Stanley

Công thức của thuật toán Stanley như sau:  $\delta(t) = \psi(t) + \arctan \frac{k \cdot x(t)}{u(t)}$ .

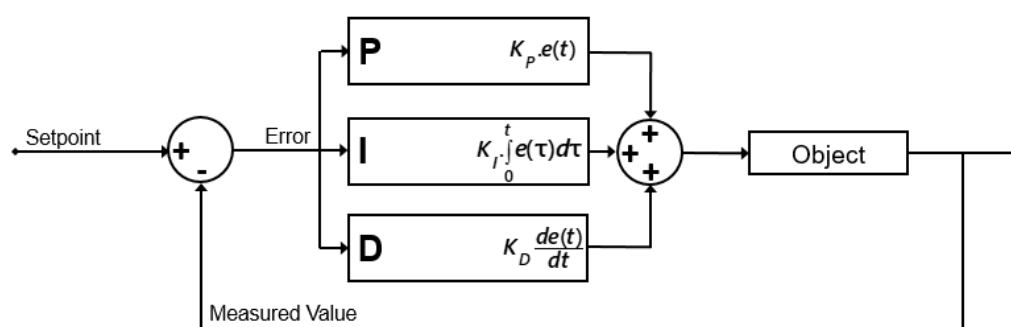
Trong đó,  $\delta(t)$  là góc điều hướng để xe bám quỹ đạo;  $\psi(t)$  là góc thể hiện hướng của một đoạn quỹ đạo gần nhất, được xác định so với hướng di chuyển của thiết

bị;  $u(t)$  là vận tốc tức thời của thiết bị;  $x(t)$  là sai số bám quỹ đạo, được tính từ trung tâm trục bánh trước đến điểm gần nhất trên quỹ đạo;  $k$  là thông số độ lợi, sai số bám quỹ đạo càng lớn thì  $k$  sẽ quyết định việc bám vào quỹ đạo nhanh hay chậm [2].

### 2.3. Giới thiệu sơ lược thuật toán PID

Một bộ điều khiển vi tích phân tỉ lệ (bộ điều khiển PID - Proportional Integral Derivative) là một cơ chế phản hồi vòng điều khiển (bộ điều khiển) tổng quát được sử dụng rộng rãi trong các hệ thống điều khiển công nghiệp – bộ điều khiển PID là bộ điều khiển được sử dụng nhiều nhất trong các bộ điều khiển phản hồi. Bộ điều khiển PID sẽ tính toán giá trị "sai số" là hiệu số giữa giá trị đo thông số biến đổi và giá trị đặt mong muốn. Bộ điều khiển sẽ thực hiện giảm tối đa sai số bằng cách điều chỉnh giá trị điều khiển đầu vào.

Trong trường hợp không có kiến thức cơ bản (mô hình toán học) về hệ thống điều khiển thì bộ điều khiển PID là sẽ bộ điều khiển tốt nhất. Tuy nhiên, để đạt được kết quả tốt nhất, các thông số PID sử dụng trong tính toán phải điều chỉnh theo tính chất của hệ thống – trong khi kiểu điều khiển là giống nhau, các thông số phải phụ thuộc vào đặc thù của hệ thống.



Hình 2.7: Sơ đồ khối thuật toán PID

Công thức liên tục của thuật toán PID được thể hiện ở công thức bên dưới và hình minh họa Hình 2.7.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t)$$

Để có thể áp dụng thuật toán PID lên các hệ thống lập trình, ta thường sử dụng PID số, hay PID rời rạc. Để có được công thức rời rạc, ta biến đổi công thức trên về miền  $z$  như sau:

$$\frac{U[z]}{E[z]} = K_p + K_i \frac{T_s}{2} \frac{z+1}{z-1} + K_d \frac{z-1}{z.T_s}$$

Sau khi biến đổi ta có được biểu thức sau:

$$U[z] = z^{-1}U[z] + \left( K_p + K_i \frac{T_s}{2} + \frac{K_d}{T_s} \right) E[z] + \left( -K_p + K_i \frac{T_s}{2} - \frac{2K_d}{T_s} \right) z^{-1}E[z] + \frac{K_d}{T_s} z^{-2}E[z]$$

Chuyển biểu thức trên về miền thời gian ta có biểu thức PID rời rạc như sau:

$$u(k) = u(k-1) + a.e(k) + b.e(k-1) + c.e(k-2)$$

Trong đó,  $a = K_p + K_i \frac{T_s}{2} + \frac{K_d}{T_s}$ ,  $b = -K_p + K_i \frac{T_s}{2} - \frac{2K_d}{T_s}$  và  $c = \frac{K_d}{T_s}$  [3].

## Chương 3. GIỚI THIỆU MÔ HÌNH, THIẾT BỊ VÀ THIẾT KẾ PHẦN CỨNG

### 3.1. Giới thiệu các loại mô hình và lí do lựa chọn

Trên thị trường hiện nay có rất nhiều dạng mô hình máy bay cánh bằng, trong số đó có thể kể đến một số dạng mô hình máy bay sau:

- Tàu lượn (Surfer):



*Hình 3.1: Mô hình tàu lượn (Surfer) [4]*

Đặc điểm:

- Lực nâng lớn, không cần dùng động cơ quá mạnh và dễ bay
- Nhỏ gọn, thuận lợi cho việc di chuyển
- Khoan chứa nhỏ
- Không mang được tải trọng lớn
- Tốc độ bay nhanh



- Máy bay phản lực:



*Hình 3.2: Mô hình máy bay phản lực [4]*

Đặc điểm:

- Nhỏ, gọn thuận lợi cho việc di chuyển
- Không có khoang máy bay
- Tốc độ bay rất nhanh
- Khi bay máy có độ mất ổn định cao

- Máy bay cánh trên:



*Hình 3.3: Mô hình máy bay cánh trên [4]*

Đặc điểm:

- Mô hình bay đầm và ổn định
- Khoang chứa lớn
- Lớn và nặng, cần có động cơ lớn, dẫn tới thời gian bay không lâu

- Tốc độ bay vừa phải

Từ những đặc điểm trên chúng em chọn mô hình máy bay cánh bằng cánh trên bởi những ưu điểm thuận lợi của nó trong quá trình điều khiển: Khoang lớn dùng để chứa các mạch điện, máy bay đầm và tốc độ bay vừa phải thuận lợi cho quá trình điều khiển và nhận tín hiệu GPS.

### 3.2. Giới thiệu các thiết bị phần cứng

#### 3.2.1. Remote Control và Receiver

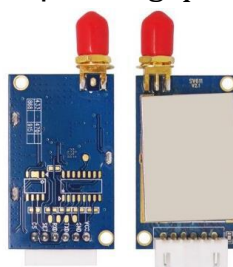
Remote Control (RC) là một tay cầm điều khiển từ xa, phát xung đến một thiết bị gọi là receiver, được lắp trên máy bay. Các tín hiệu nhận được và xuất ra các chân của receiver theo các kênh (channel) tương ứng, các xung có tần số và độ rộng xung phù hợp để điều khiển các servo.



Hình 3.4: Remote Control và Receiver

#### 3.2.2. RF

RF là một thiết bị truyền sóng vô tuyến với khoảng cách truyền xa và tương đối tin cậy. Module RF có thể dễ dàng được giao tiếp thông qua chuẩn UART, rất tiện lợi cho việc truyền nhận dữ liệu thông qua vi điều khiển.

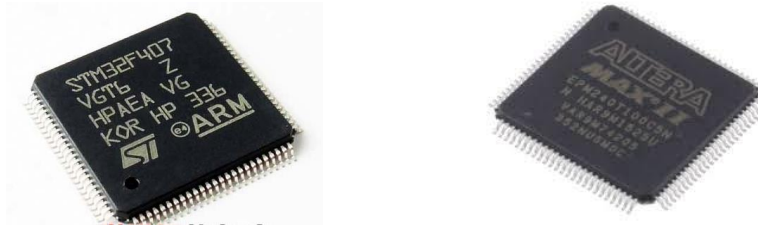


Hình 3.5: Module truyền nhận RF

### 3.2.3. CPLD

CPLD, viết tắt của Complex Programmable Logic Device, là một vi mạch mà người dùng có thể lập trình được. Nó cũng tương tự như FPGA về ứng dụng nhưng không có nhiều cổng logic (logic gates) như FPGA. Về cơ bản, CPLD là một thiết bị lập trình logic có tốc độ xung nhịp lớn hơn nhiều lần so với vi điều khiển thông thường, rất thích hợp cho các ứng dụng đo độ rộng xung của tín hiệu [5].

Thiết bị CPLD được dùng trong hệ thống là MAX II của hãng Altera như Hình 3.6.



Hình 3.6: Vi điều khiển STM32F407VTG6 và CPLD MAX II

### 3.2.4. Vi điều khiển (MCU)

Vi điều khiển được sử dụng trong hệ thống là vi điều khiển STM32F4 Discovery của hãng ST Electronics [6]. Đây là một vi điều khiển được tích hợp nhiều ngoại vi được sử dụng trong hệ thống như: Timer, UART, PWM, FSMC, SDIO,...

#### 3.2.4.1. Giới thiệu sơ lược về ngoại vi UART

UART (universal asynchronous receiver-transmitter) là một thiết bị phần cứng máy tính dùng cho giao tiếp nối tiếp bất đồng bộ, trong đó định dạng dữ liệu và tốc độ truyền dữ liệu đã được thiết lập trước [7]. Giao thức truyền nhận dữ liệu UART rất phổ biến trong việc giao tiếp của vi điều khiển, như giao tiếp giữa các vi điều khiển, hay giao tiếp giữa vi điều khiển và các cảm biến,...

Trong giao tiếp UART cần tối thiểu 3 dây: TX (Transmit), RX (Receive) và GND (Ground); và các thông số được quy định trước như baudrate, số start/ stop bit, parity,...

Trong phạm vi hệ thống, giao tiếp UART sẽ được sử dụng để truyền nhận dữ liệu giữa vi điều khiển và bộ cảm biến IMU – GPS, giữa vi điều khiển và thiết bị RF.

#### **3.2.4.2. Giới thiệu sơ lược về ngoại vi DMA**

DMA (Direct memory access), hay truy cập bộ nhớ trực tiếp, được sử dụng để cung cấp việc truyền dữ liệu tốc độ cao giữa các ngoại vi với bộ nhớ và giữa các bộ nhớ với nhau. Dữ liệu có thể được truyền một cách nhanh chóng bằng DMA mà không cần đến sự can thiệp của CPU. Việc này giúp CPU rảnh rỗi để thực hiện các ứng dụng khác [8].

#### **3.2.4.3. Giới thiệu sơ lược về ngoại vi PWM**

PWM (Pulse-width modulation), hay còn được gọi là điều chế độ rộng xung, là một ngoại vi phổ biến có mặt trên hầu hết các loại vi điều khiển. Chức năng chính của PWM trong hệ thống là tạo ra một xung có chu kỳ và độ rộng như *Hình 3.8* để điều khiển các động cơ servo [8].

#### **3.2.4.4. Giới thiệu sơ lược về ngoại vi FSMC**

FSMC (Flexible static memory contRoller) là một ngoại vi có thể giao tiếp với các bộ nhớ đồng bộ và không đồng bộ và các thẻ nhớ PC 16 bit, được tích hợp sẵn trong vi xử lý STM32 Discovery [8].

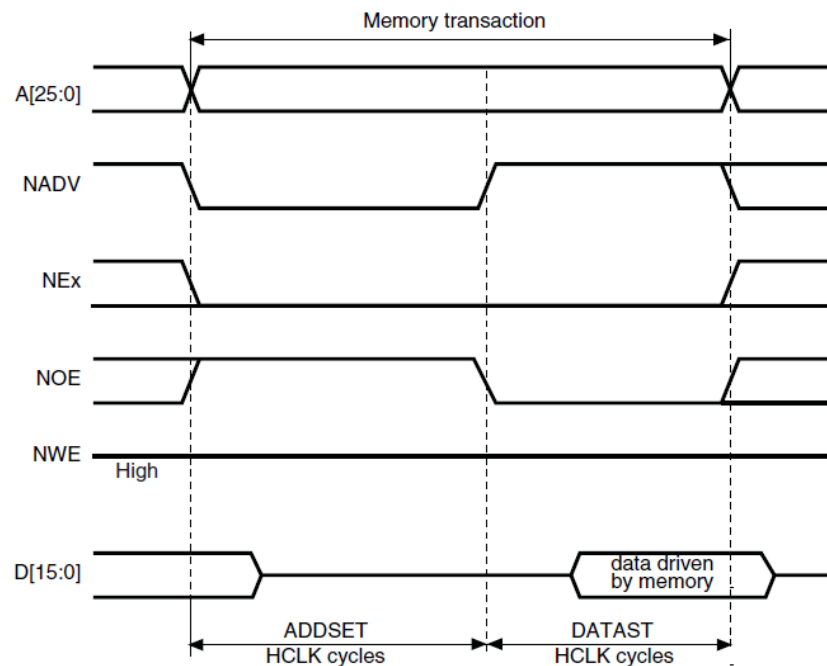
FSMC hỗ trợ cho người dùng nhiều chế độ đọc ghi dữ liệu để chọn lựa, và ở đây chọn chế độ NOR với các chân được mô tả trong *Bảng 3.1*.

Tên chân	I/O	Chức năng
CLK	O	Clock (dùng cho giao tiếp đồng bộ)
A[25:0]	O	Bus địa chỉ
D[15:0]	I/O	Bus dữ liệu (vừa đọc vừa ghi)
NEx	O	Chọn chip, x = 1..4

NOE	0	Cho phép ngõ ra
NWE	0	Cho phép ghi
NL (=NADV)	0	Cho phép chốt
NWAIT	1	Đợi tín hiệu vào từ FSMC

*Bảng 3.1: Các chân được sử dụng của ngoại vi FSMC [8]*

Có nhiều mode hoạt động để ghi đọc dữ liệu thông qua ngoại vi FSMC, Hình 3.7 dưới đây sẽ mô tả các chân tín hiệu trên sẽ hoạt động ở Mode C – mode sử dụng chung bus Address và Data.



*Hình 3.7: Sơ đồ hoạt động của các chân tín hiệu ngoại vi FSMC [8]*

### 3.2.4.5. Giới thiệu sơ lược về ngoại vi SDIO

SDIO (Secure digital input/ output) là một giao thức giao tiếp thẻ nhớ với nhiều ưu điểm về tốc độ truyền dữ liệu so với chuẩn SPI thông thường [8].

Về phần cứng, STM32 Discovery cung cấp 3 chân cho chuẩn giao tiếp SDIO: PC8 chức năng DO, PC12 chức năng SCK và PD2 chức năng DI (CMD).

Về phần mềm lập trình, để giao tiếp thẻ nhớ theo chuẩn giao tiếp SDIO, người dùng cần sử dụng thư viện FatFS.

Thư viện FatFS là một module file hệ thống FAT/exFAT chung cho các hệ thống nhúng nhỏ. Module FatFS được viết tuân thủ theo ANSI C (C89) và hoàn toàn tách biệt với lớp I/O ổ đĩa. Vì vậy, nó độc lập với các nền tảng (platform). Nó có thể được tích hợp vào với các vi điều khiển nhỏ với nguồn giới hạn như 8051, PIC, AVR, ARM, Z80, RX, vvvv [9].

### 3.2.5. Các thiết bị chấp hành

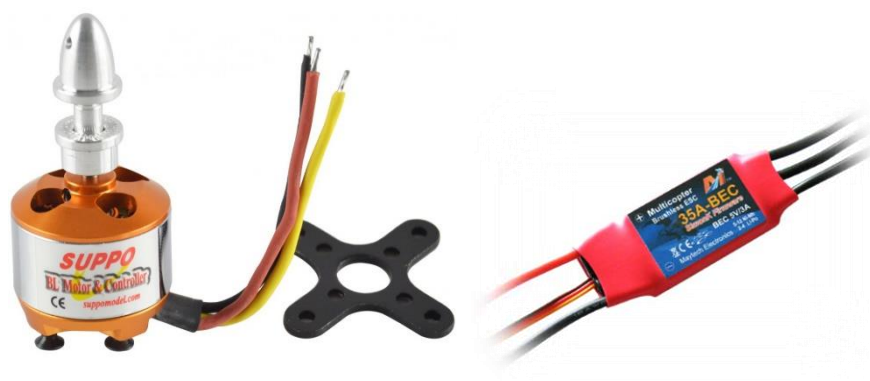
Thiết bị chấp hành trên máy bay gồm có 2 loại động cơ: servo motor và brushless motor. Tùy thuộc vào vị trí của các motor trên máy bay ta có tên gọi của chúng khác nhau, gồm có: Aileron, Elevator, Rudder (servo motor) và Throttle (brushless motor).

Để đồng bộ việc cấp xung cho tất cả ngõ ra, ta cần một thiết bị gọi là ESC (Electronic Speed Control). ESC vừa có tác dụng hạ nguồn 12V thành nguồn 5V cấp cho receiver, vừa chuyển xung điều khiển từ receiver thành tín hiệu điều khiển cho brushless motor.



Hình 3.8: Độ rộng xung điều khiển xuất ra từ receiver

Xung điều khiển cho servo motor từ 1 đến 2 ms tương ứng góc xoay từ  $-90^{\circ}$  đến  $90^{\circ}$ .

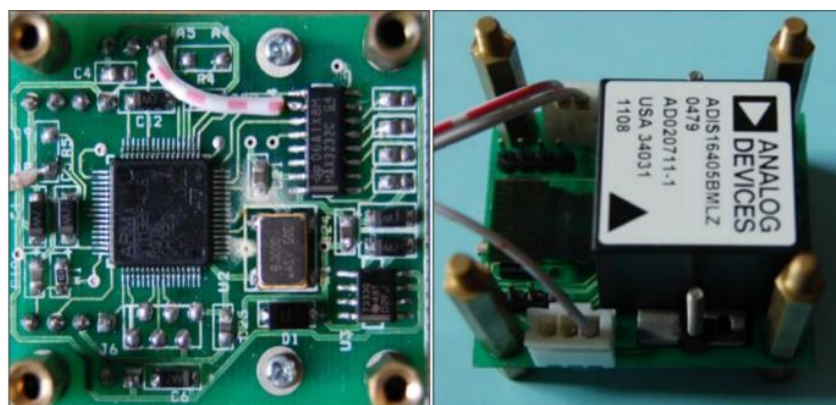


Hình 3.9: Động cơ brushless và ESC

### 3.2.6. Cảm biến IMU và GPS

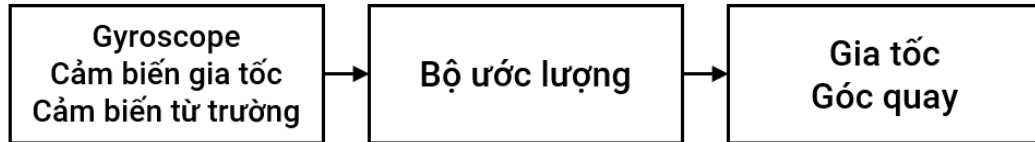
#### 3.2.6.1. Khái quát cấu trúc cảm biến IMU và GPS

IMU (Inertial Measurement Unit) và GPS (Global Positioning System) là bộ cảm biến được chế tạo tại phòng thí nghiệm Điều Khiển Tự Động - Bộ môn Tự Động, Đại học Bách Khoa TP HCM. Hệ thống IMU 9-DOF bao gồm vi điều khiển ARM-Cortex M4 STM32F405 hỗ trợ tính toán dấu chấm động và cảm biến 9-DOF ADIS16405 (gồm ba loại cảm biến Gia tốc ba trục, Vận tốc góc ba trục, Từ trường ba trục).



Hình 3.10: IMU-9DOF được chế tạo tại PTN Bộ môn Tự Động ĐHBK TP HCM

Bên trong IMU có 9 cảm biến bao gồm 3 cảm biến vận tốc góc (gyroscope), 3 cảm biến gia tốc (accelerometer), và 3 cảm biến từ trường (magnetometer) kết hợp lại với nhau trên 1 trục tọa độ 3 chiều trực giao. Hệ thống cảm biến này gọi chung là MARG (Magnetic, Angular rate, and Gravity) [10].



Hình 3.11: Sơ đồ khối cảm biến IMU

### 3.2.6.2. Cấu trúc xuất dữ liệu của cảm biến IMU và GPS

Dữ liệu từ IMU được gửi sau mỗi 10ms, mỗi frame gồm 13 số, 80 byte. Tốc độ baud là 460800.

```

-0010 0084 -0040 0010 -0001 -0002 -0143 -0013 0983 2690 0163 2042 0857
-0010 0084 -0040 0008 0008 0000 -0139 -0016 0979 2692 0151 2041 0856
-0010 0084 -0040 0000 0006 -0005 -0143 -0013 0976 2692 0159 2042 0856
-0010 0084 -0040 0000 -0001 0006 -0146 -0013 0979 2692 0157 2043 0857
-0010 0084 -0040 -0002 0003 -0010 -0146 -0013 0976 2691 0168 2043 0857
-0010 0084 -0040 0012 0008 -0014 -0146 -0013 0976 2691 0169 2043 0857
  
```

Hình 3.12: Dữ liệu xuất ra từ cảm biến IMU

Phi	0x20	Theta	0x20	Psi	0x20	wx	0x20	wy	0x20	wz
0x20	ax	0x20	ay	0x20	az	0x20	Mx	0x20	My	0x20
Mz	0x20	P	0x0d	0x0a						

Hình 3.13: Cấu trúc một mẫu dữ liệu trả về từ cảm biến IMU

Trong đó:

- Byte bắt đầu: 0x0A (Start, 1 byte).
- Phi (5 byte), Theta (5 byte), Psi (5 byte): 3 góc quay theo 3 trục x, y, z tương ứng (đơn vị 0.10)
- $\omega_x$  (5 byte),  $\omega_y$  (5 byte),  $\omega_z$  (5 byte): giá trị vận tốc góc 3 trục x, y, z tương ứng (đơn vị mrad/s).



- ax (5 byte), ay (5 byte), az (5 byte): giá trị gia tốc 3 trục x, y, z tương ứng (đơn vị mm/s<sup>2</sup>).
- mx (5 byte), my (5 byte), mz (5 byte): giá trị từ trường 3 trục x, y, z tương ứng (đơn vị mgauss).
- p (4 byte): giá trị của áp suất của môi trường.
- Byte kết thúc: 0x0D (Stop, 1 byte) [10].

Dữ liệu GPS được truyền sau mỗi 100ms, bao gồm 2 dạng dữ liệu GPGLA, GPVTG.

Ví dụ của chuỗi GLA.

\$GPGLA,170834,4124.8963,N,08151.6838,W,1,08,1.5,280.2,M,46.9,M,,\* 75

Tên trường	Dữ liệu mẫu	Mô tả
Tên xác định loại chuỗi	\$GPGLA	Global Positioning System Fix Data
Thời gian vệ tinh	170834	17:08:34 Z
Vĩ độ	4124.8963, N	41d 24.8963' N hay 41d 24' 54" N
Kinh độ	08151.6838, W	81d 51.6838' W or 81d 51' 41" W
Chất lượng fix: - 0 = Không xác định - 1 = GPS fix - 2 = DGPS fix	1	Dữ liệu từ GPS fix
Số lượng vệ tinh	08	Quan sát được 8 vệ tinh
Horizontal Dilution of Precision (HDOP)	1.5	Độ chính xác tương đối của vị trí ngang

Cao độ	280.2, M	280.2 mét từ mặt nước biển
Height of geoid above WGS84 ellipsoid	46.9, M	46.9 mét
Thời gian từ lần cập nhật DGPS cuối	trống	Không có cập nhật lần cuối
Id trạm tham khảo DGPS	trống	Không có id trạm
Checksum	*75	Sử dụng để kiểm tra lỗi đường truyền

*Bảng 3.2: Cấu trúc dữ liệu chuỗi GPGGA*

Ví dụ chuỗi VTG là:

\$GPVTG,,T,,M,0.00,N,0.00,K\*4E

Số thứ tự trường	Ý nghĩa
0	Tên xác định chuỗi \$GPVTG
1	Track made good (độ thực)
2	T: track made good so với hướng bắc thực
3	Track made good (độ từ trường)
4	M: track made good so với hướng bắc từ trường
5	Tốc độ (hải lí)
6	N: tốc độ tính bằng hải lí
7	Tốc độ so với mặt đất (km/h)
8	K: Tốc độ so với mặt đất tính bằng km/h

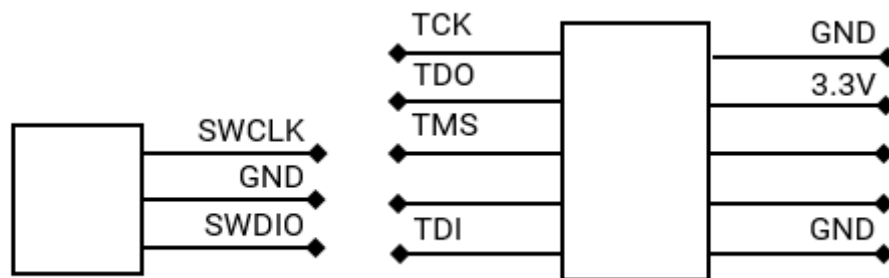
9	Dữ liệu checksum, luôn bắt đầu với *
---	--------------------------------------

*Bảng 3.3: Cấu trúc dữ liệu GPS của chuỗi GPVTG*

### 3.3. Thiết kế mạch vi điều khiển, CPLD và mạch ra chân

#### 3.3.1. Mạch vi điều khiển và CPLD

Mạch điều khiển chứa vi điều khiển STM32 Discovery và CPLD MAXII của hãng Altera (như Hình 3.6) được thiết kế ở phòng thí nghiệm Bộ môn Tự động). Nguồn cấp cho mạch là nguồn 5V, đi qua mạch tạo nguồn 3.3V dùng TPS7333Q. Nguồn 3.3V được dùng để cấp nguồn cho toàn bộ mạch bao gồm cả vi điều khiển và CPLD. Ngoài ra, vi điều khiển và CPLD yêu cầu những chân cần thiết để sử dụng



*Hình 3.14: Sơ đồ các header dùng để nạp chương trình từ mạch nạp cho vi điều khiển và CPLD*



*Hình 3.15: Hình ảnh mạch điều khiển (được thiết kế ở phòng thí nghiệm Bộ môn Tự động)*

mạch nạp và nạp chương trình vào các thiết bị trên. Các chân cần thiết cho mạch nạp được mô tả ở Hình 3.14.

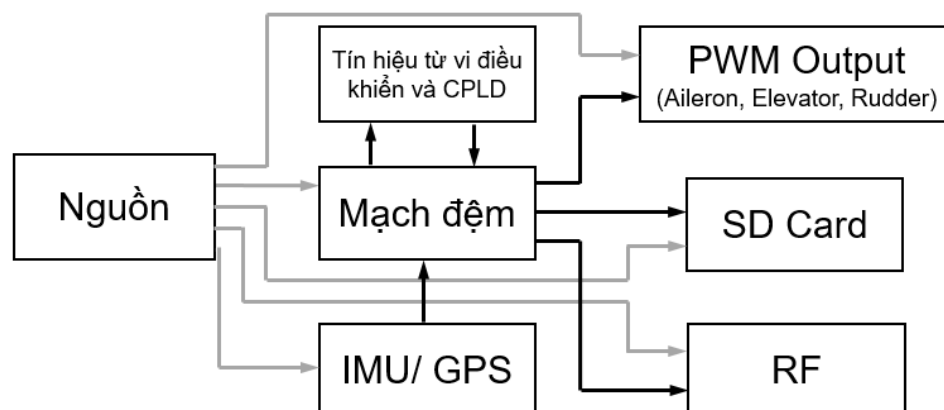
Thạch anh cho vi điều khiển là thạch anh 8 MHz, thạch anh cho CPLD là thạch anh 20 MHz.

Sơ đồ thiết kế (Schematic) được cung cấp ở phần *Phụ lục*. Mạch điều khiển hoàn thiện được thể hiện ở Hình 3.15.

### 3.3.2. Mạch ra chân

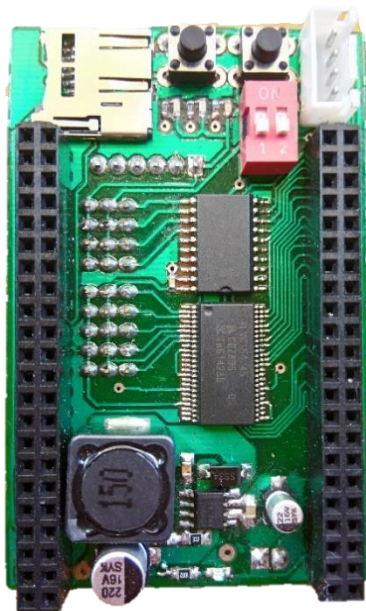
Chức năng chính của mạch ra chân là cấp nguồn và tín hiệu điều khiển cho các cảm biến, các module và các động cơ servo. Nguồn cho nhiều thiết bị chấp hành đòi hỏi dòng cấp là rất lớn, do đó chip nguồn TPS5430 được chọn để làm mạch giảm áp thành 5V và 3.3V cung cấp nguồn toàn bộ mạch. TPS5430 có thể chịu dòng lên đến 3A [11]. Sơ đồ khái quát các thành phần của mạch được thể hiện ở Hình 3.16.

Mạch đệm được sử dụng trong mạch nhằm đảm bảo tín ổn định của các tín hiệu điều khiển PWM và tín hiệu truyền nhận UART (dùng cho RF và cảm biến IMU/ GPS), giảm thiểu nhiễu và suy hao tín hiệu trong mạch và trên đường truyền.



Hình 3.16: Sơ đồ mạch ra chân

Sơ đồ thiết kế (Schematic) sẽ được cung cấp ở phần *Phụ lục*. Mạch hoàn thiện được thể hiện ở *Hình 3.17*.



*Hình 3.17: Mạch ra chân*

## Chương 4. MÔ PHỎNG THUẬT TOÁN STANLEY TRÊN MATLAB

### 4.1. Những thông số cần thiết và những hiệu chỉnh của thuật toán

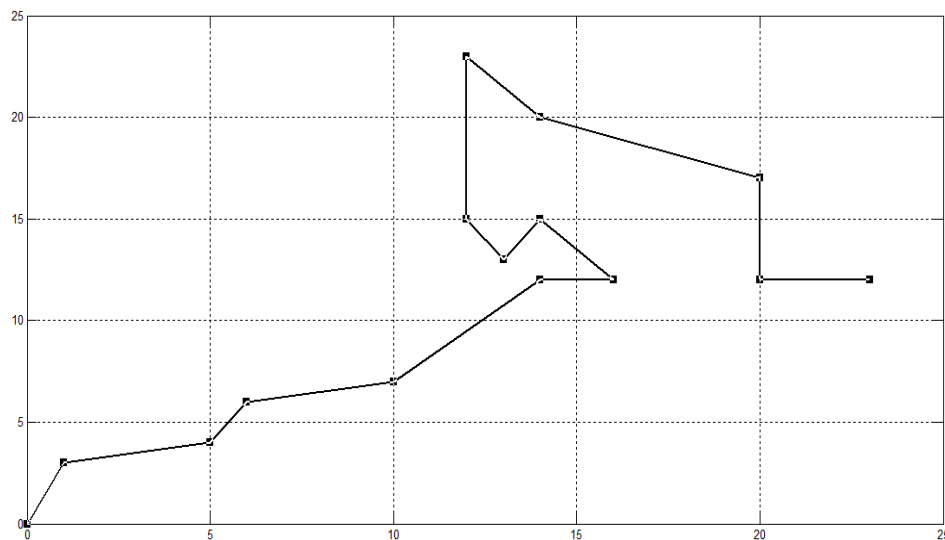
Để có thể mô phỏng thuật toán Stanley trên Matlab, ta giả sử vật di chuyển là một chất điểm có vị trí tọa độ và hướng di chuyển ban đầu được cho trước và vận tốc là hằng số.

Quỹ đạo mong muốn được cho sẵn theo dạng mảng 2 chiều với các tọa độ điểm rời rạc, từ đó ta có thể vẽ được dạng quỹ đạo mong muốn từ mảng nêu trên.

```
tra_x = [0 1 5 6 10 14 16 14 13 12 12 14 20 20 23];
```

```
tra_y = [0 3 4 6 7 12 12 15 13 15 23 20 17 12 12];
```

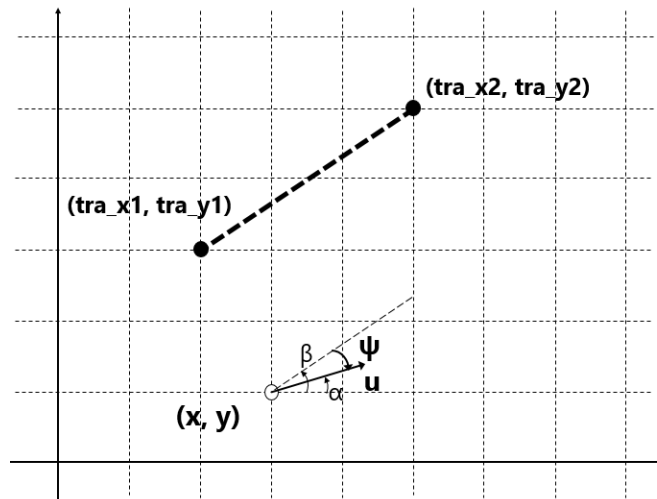
Từ đây vẽ được quỹ đạo như sau:



Hình 4.1: Quỹ đạo di chuyển mong muốn

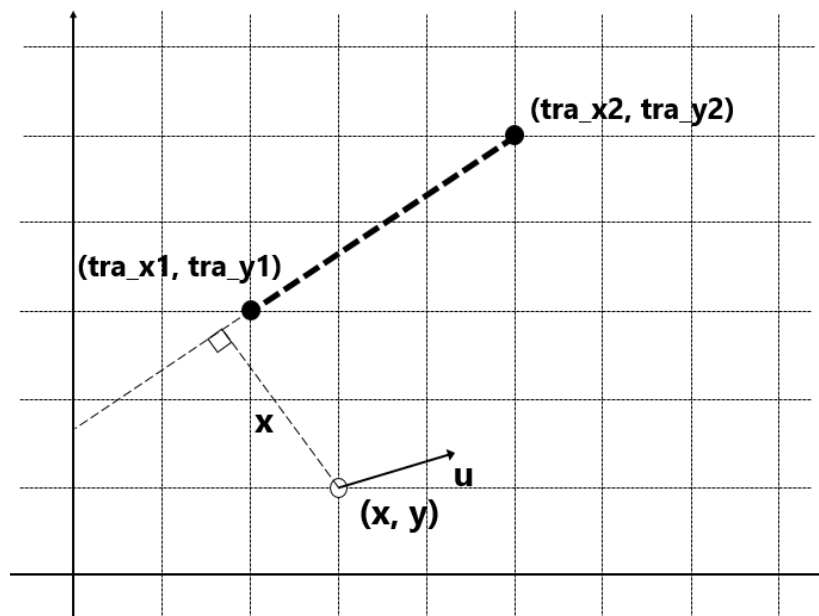
Dựa vào công thức thuật toán ta cần tính toán 2 thông số để cung cấp cho công thức Stanley. Như đã nêu trên, góc  $\psi$  là góc lệch giữa hướng di chuyển của quỹ đạo so với hướng di chuyển của xe.

Để tổng quát và thống nhất về các góc đo khi mô phỏng, tất cả các góc sẽ được tính so với vector  $\vec{i}$  của trục  $Ox$ . Do đó, góc  $\psi$  sẽ bằng độ dốc của quỹ đạo ( $\beta$ ) trừ cho hướng di chuyển của xe ( $\alpha$ ).



Hình 4.2: Xác định độ lớn góc  $\psi$  trên hệ trục tọa độ

Thông số còn lại cần được đưa vào cho công thức Stanley là  $x(t)$  - khoảng cách từ tọa độ hiện tại đến điểm gần nhất trên quỹ đạo. Ở lần chạy đầu tiên, ta sẽ quét tính khoảng cách từ tọa độ hiện tại đến tất cả các đường thẳng trong quỹ đạo để tìm ra khoảng cách  $x$  ngắn nhất, và từ đó bám vào đoạn quỹ đạo đó để tiếp tục di chuyển.

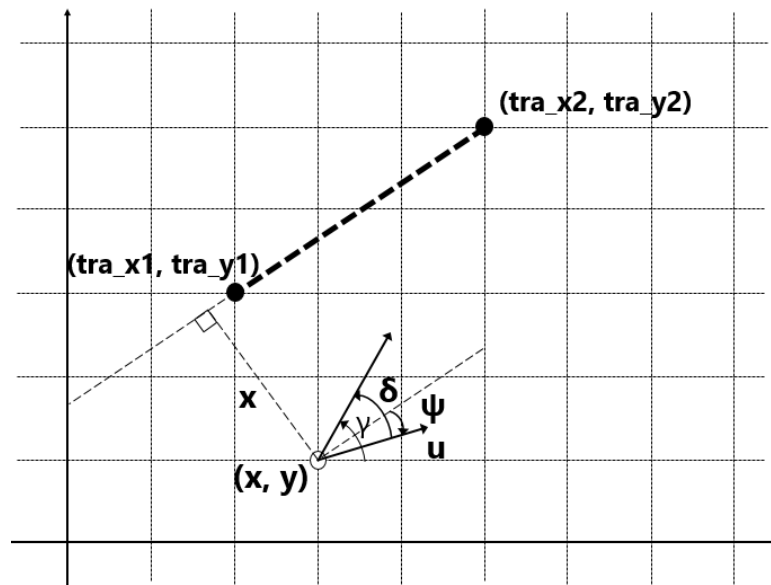


Hình 4.3: Xác định sai số bám quỹ đạo  $x$

Quỹ đạo là tập hợp của nhiều đường thẳng, khoảng cách từ một điểm đến một đường thẳng được tính theo công thức sau:  $x = \frac{|\overrightarrow{AB} \cdot \vec{n}|}{|\vec{n}|}$ . Trong đó,  $\overrightarrow{AB}$  là vector

giữa tọa độ hiện tại và tọa độ điểm trên quỹ đạo;  $\vec{n}$  là vector pháp tuyến của đoạn thẳng quỹ đạo. Để xác định vector pháp tuyến ta chỉ cần xác định vector giữa 2 điểm kề nhau trên quỹ đạo – vector chỉ phương – sau đó đổi vị trí tung độ, hoành độ và thêm một dấu trừ sẽ có được vector pháp tuyến.

Sau khi có được các thông số cần thiết, ta áp dụng vào công thức Stanley để xác định góc lái cho chất điểm. Gọi góc lái là  $\gamma$  ta có công thức sau:  $\gamma = \alpha + \delta$ . Trong đó,  $\alpha$  là góc thể hiện hướng di chuyển của chất điểm;  $\delta$  là góc được tính từ công thức Stanley;  $\gamma$  là góc điều hướng.



Hình 4.4: Xác định góc điều hướng trên mặt phẳng tọa độ

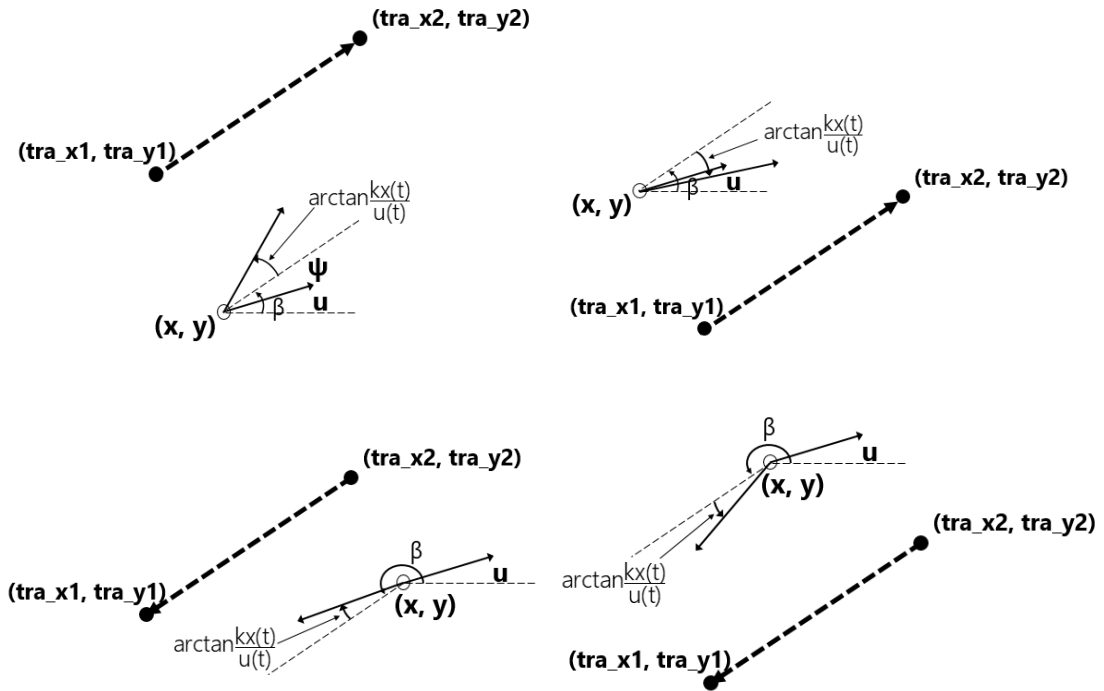
Nhận xét, nếu phân tích kĩ công thức tính góc điều hướng ta sẽ thấy được rằng, công thức Stanley sẽ điều chỉnh góc di chuyển của chất điểm về đúng hướng của quỹ đạo, nhưng song song với quỹ đạo. Sau đó, thành phần  $\arctan \frac{k \cdot x(t)}{u(t)}$  sẽ tiếp tục cộng thêm một góc bù để đẩy góc điều hướng  $\gamma$  vào sát quỹ đạo, góc này lớn



hay nhỏ tùy thuộc vào  $x(t)$ . Ta có thể dễ dàng nhìn thấy qua công thức biểu diễn bên dưới:

$$\gamma = \delta + \alpha = \psi + \arctan \frac{k.x(t)}{u(t)} + \alpha = \beta - \alpha + \arctan \frac{k.x(t)}{u(t)} + \alpha = \beta + \arctan \frac{k.x(t)}{u(t)}.$$

Tuy nhiên, việc cộng bù thành phần  $\arctan \frac{k.x(t)}{u(t)}$  khác nhau đối với các góc phần tư khác nhau trên mặt phẳng tọa độ, và cũng phụ thuộc vào vị trí tương đối của chất điểm so với đoạn thẳng quỹ đạo – trên hay dưới đoạn thẳng.



Hình 4.5: Dấu của thành phần  $\arctan \frac{k.x(t)}{u(t)}$  ở các vị trí khác nhau so với quỹ đạo

Qua quan sát ta nhận thấy cần phải xác định thêm dấu cho biểu thức  $\arctan \frac{k.x(t)}{u(t)}$  để có thể điều hướng chính xác ở mọi góc phần tư trên hệ trục tọa độ.

Giả sử, quy ước chất điểm nằm dưới đoạn thẳng quỹ đạo có dấu trừ, nằm trên đoạn thẳng quỹ đạo sẽ có dấu cộng, khi đó ta dùng công thức sau để xác định dấu  $d = B(Ax_0 + By_0 + C)$  với  $(x_0, y_0)$  là tọa độ điểm cần xác định trên hay dưới đường thẳng, và  $Ax + By + C$  là phương trình đường thẳng.

Độ lớn góc  $\beta$  cũng ảnh hưởng đến dấu của thành phần  $\arctan \frac{k.x(t)}{u(t)}$ , cụ thể nếu

$\frac{\pi}{2} \leq \beta < \frac{3\pi}{2}$  thì dấu của thành phần  $\arctan \frac{k.x(t)}{u(t)}$  bị đảo so với các góc còn lại trên

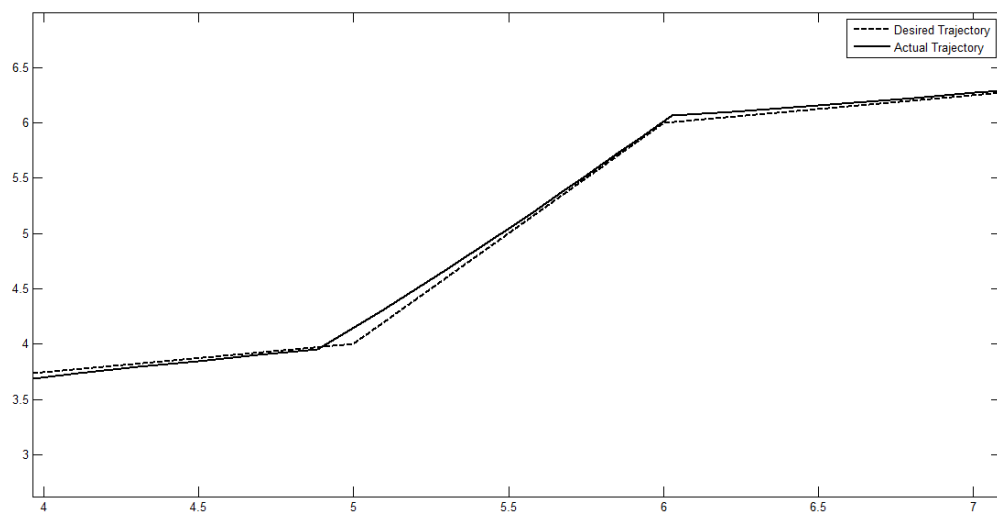
mặt phẳng tọa độ. Do đó, để có được góc điều hướng một cách tổng quát, ta áp dụng như sau trong Matlab:

- Nếu  $\beta < \frac{\pi}{2}$  hoặc  $\beta \geq \frac{3\pi}{2}$  thì  $\delta = \psi + \text{sgn}d \cdot \arctan \frac{k.x(t)}{u(t)}$ .

- Nếu  $\frac{\pi}{2} \leq \beta < \frac{3\pi}{2}$  thì  $\delta = \psi - \text{sgn}d \cdot \arctan \frac{k.x(t)}{u(t)}$ .

#### 4.2. Thông số cần thiết để mô phỏng

Quá trình tính toán góc lái được thực hiện liên tục mỗi khi tọa độ của chất điểm được cập nhật. Trong lúc này, chất điểm cũng phải được yêu cầu tính khoảng



Hình 4.6: Chất điểm chuyển đoạn thẳng quỹ đạo để bám

cách từ nó đến đoạn thẳng quỹ đạo mới, để khi chất điểm di chuyển đến một điểm nhất định, chất điểm sẽ chuyển sang bám vào đoạn thẳng quỹ đạo mới. Tuy nhiên, đây chỉ là cách để mô phỏng chất điểm di chuyển, trên thực tế tọa độ điểm mới sẽ được cập nhật liên tục từ cảm biến GPS mà không cần phải tính toán.

*Hình 4.6* minh họa khi khoảng cách từ chất điểm đến điểm quỹ đạo tiếp theo rơi vào một vùng bán kính cho trước chất điểm sẽ chuyển sang bám vào đoạn thẳng mới.

Khi mô phỏng thuật toán ta cần xác định được vị trí tiếp theo mà chất điểm đi được. Nói cách khác, khi có được góc điều hướng từ thuật toán Stanley, ta cần biết sau một khoảng thời gian bằng chu kì lấy mẫu thì chất điểm sẽ di chuyển đến tọa độ nào. Một cách lí tưởng, ta sử dụng công thức sau để tìm ra tọa độ mới của chất điểm khi mô phỏng:

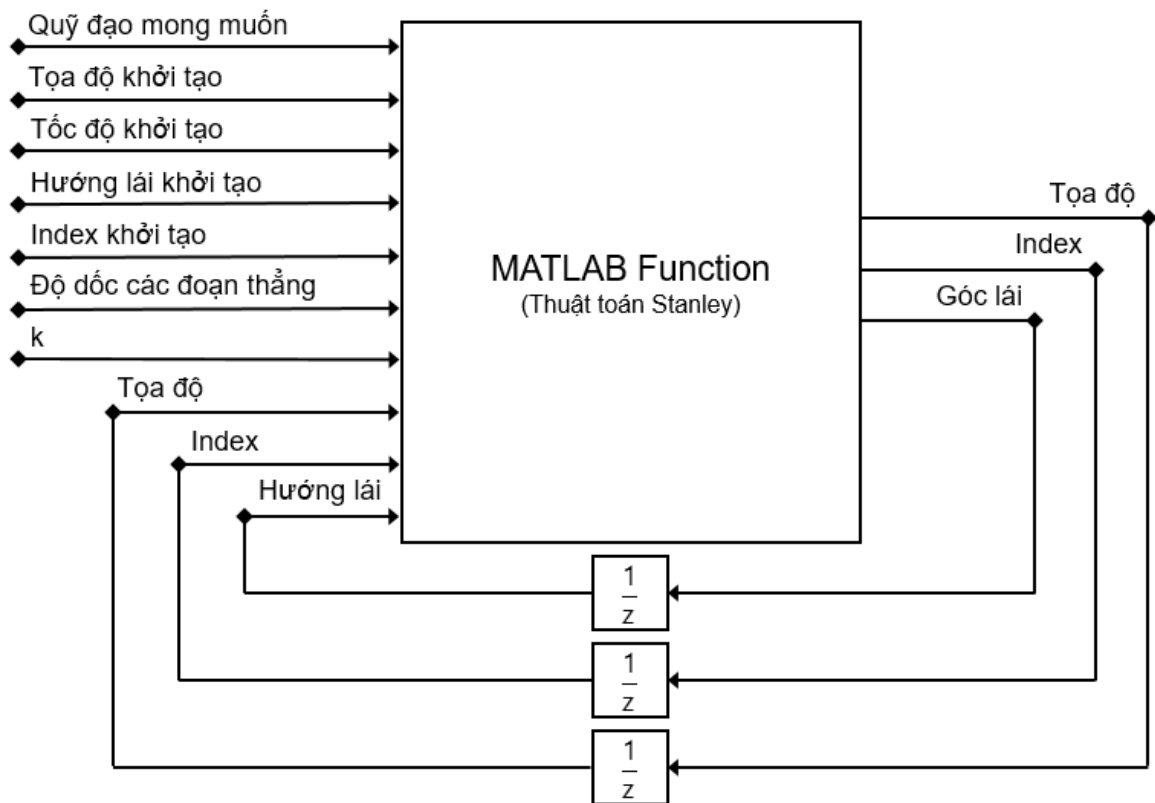
$$x^* = x + u \cdot \cos y \cdot T \text{ và } y^* = y + u \cdot \sin y \cdot T, \text{ trong đó, } T \text{ là chu kì lấy mẫu.}$$

Do vậy, để có được mô hình mô phỏng, ta phải cập nhật liên tục tọa độ lí tưởng này cho vòng lặp tiếp theo.

Từ đây, ta có thể xây dựng được mô hình mô phỏng cho thuật toán Stanley với vật di chuyển là chất điểm như *Hình 4.7*.

Trong *Hình 4.7*, ta thấy ngoài các thông số như đã phân tích, ta thấy có xuất hiện Index khởi tạo và độ dốc các đoạn thẳng. Ở đây, quỹ đạo mong muốn được lưu trữ dưới dạng mảng hai chiều, do đó index ở đây dùng để chỉ thứ tự điểm trong mảng mà chất điểm sẽ bám vào. Để có được index khởi tạo, ta sẽ quét tính khoảng cách từ chất điểm đến tất cả các điểm có trong quỹ đạo, index khởi tạo được chọn sẽ là chỉ số mà khoảng cách từ nó đến chất điểm là ngắn nhất.

Độ dốc đoạn thẳng quỹ đạo chính là góc tạo bởi đoạn thẳng nối 2 điểm liên tiếp nhau trong quỹ đạo với vector  $\vec{i}$ . Các giá trị này được tính trước để giảm khối lượng tính toán trong mỗi vòng lặp. Chi tiết về độ dốc các đoạn thẳng quỹ đạo này sẽ được trình bày cụ thể ở mục 5.4.2.



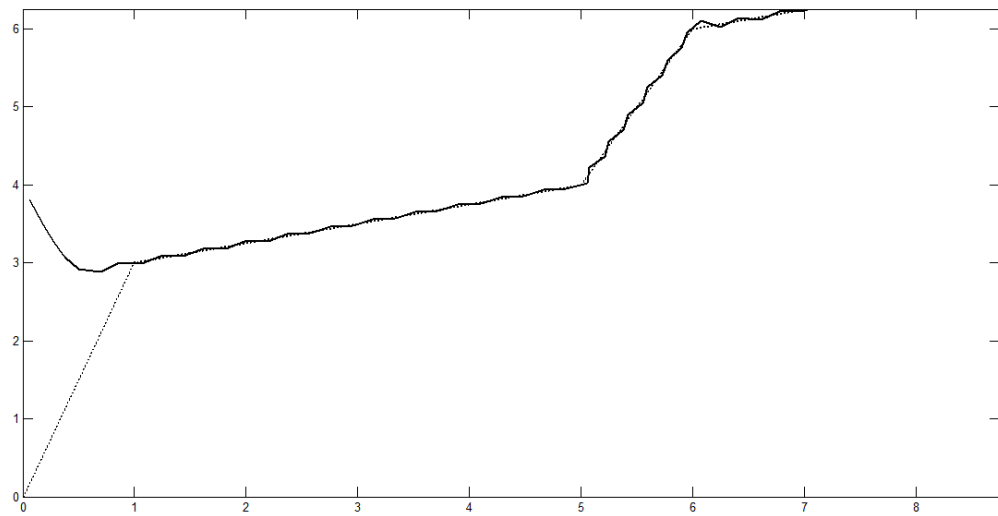
Hình 4.7: Khái quát mô hình mô phỏng thuật toán Stanley trên Matlab

#### 4.3. Kết quả mô phỏng và nhận xét tham số $k$

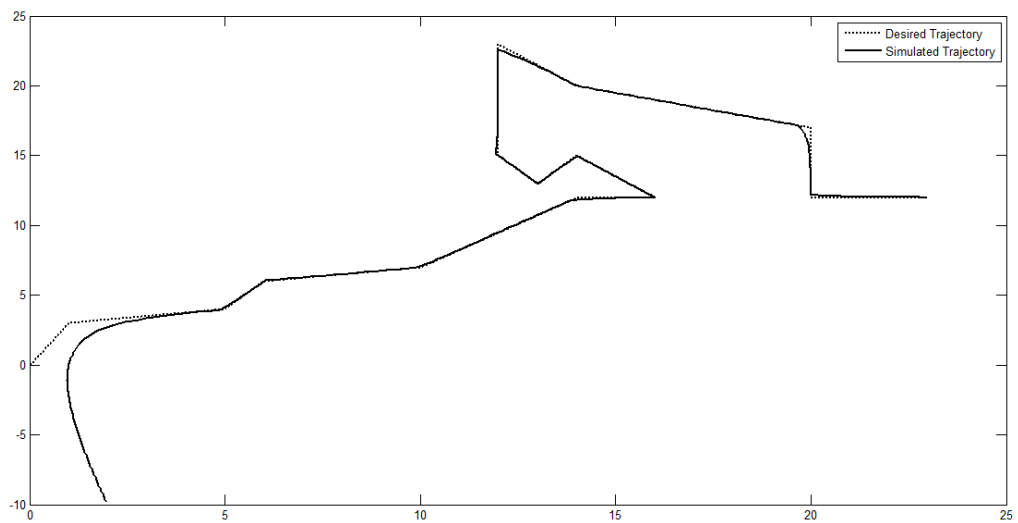
Với các công thức xây dựng đã trình bày ở trên, thuật toán Stanley khi mô phỏng cho kết quả như Hình 4.8, đồng thời khi thay đổi  $k$  ta có được sự thay đổi biểu thị ở Hình 4.9.

Từ Hình 4.8 và Hình 4.9 ta có thể thấy rằng, việc chất điểm bám vào quỹ đạo như thế nào phụ thuộc rất nhiều vào  $k$ . Cụ thể, khi chất điểm ở xa quỹ đạo,  $k$  lớn sẽ giúp chất điểm bám nhanh hơn vào quỹ đạo, nhưng thay vào đó, khi đã vào được quỹ đạo mong muốn,  $k$  chính là yếu tố làm cho chất điểm dao động liên tục xung quanh quỹ đạo.

Sau khi mô phỏng thành công trên Matlab, toàn bộ thuật toán Stanley sẽ được chuyển sang ngôn ngữ C và được sử dụng trong hệ thống được xây dựng trên vi điều khiển STM32F4.



Hình 4.8: Kết quả mô phỏng với điểm xuất phát  $(0, 4)$ ,  $v = 4$ ,  $k = 40$ ,  $T = 0.05$

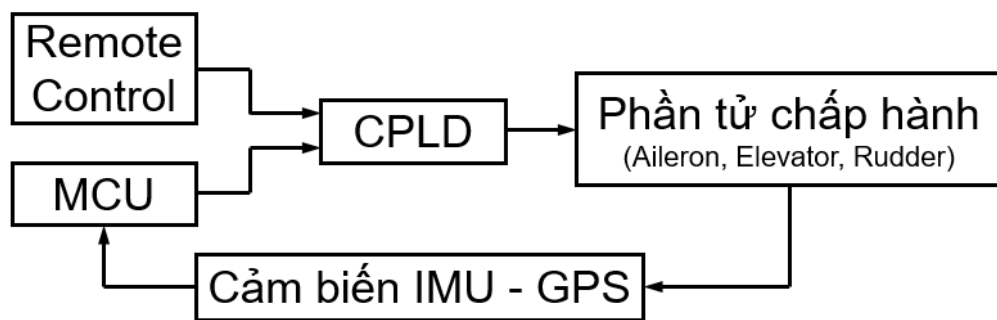


Hình 4.9: Kết quả mô phỏng với điểm xuất phát  $(2, -10)$ ,  $v = 4$ ,  $k = 4$ ,  $T = 0.05$

## Chương 5. XÂY DỰNG HỆ THỐNG ĐIỀU KHIỂN

### 5.1. Tổng quan hệ thống

Trước khi đi sâu vào cấu trúc hệ thống, ta cần nhìn nhận việc điều khiển với những yếu tố khác nhau. Đầu tiên, điều khiển máy bay theo quỹ đạo nhất định chỉ giới hạn ở việc điều khiển khi máy bay đã ổn định trên không trung. Nói cách khác, việc cất cánh và hạ cánh không nằm trong phạm vi dự án, và việc này hoàn toàn là công việc của phi công với bộ điều khiển từ xa (remote control). Từ đây, ta cũng thấy vấn đề đặt ra là cần có một mạch chuyển đổi giữa chế độ người lái và vi điều khiển. Do đó, sơ đồ khái quát của hệ thống được thể hiện dưới biểu đồ như *Hình 5.1*.



*Hình 5.1: Khái quát sơ đồ hệ thống*

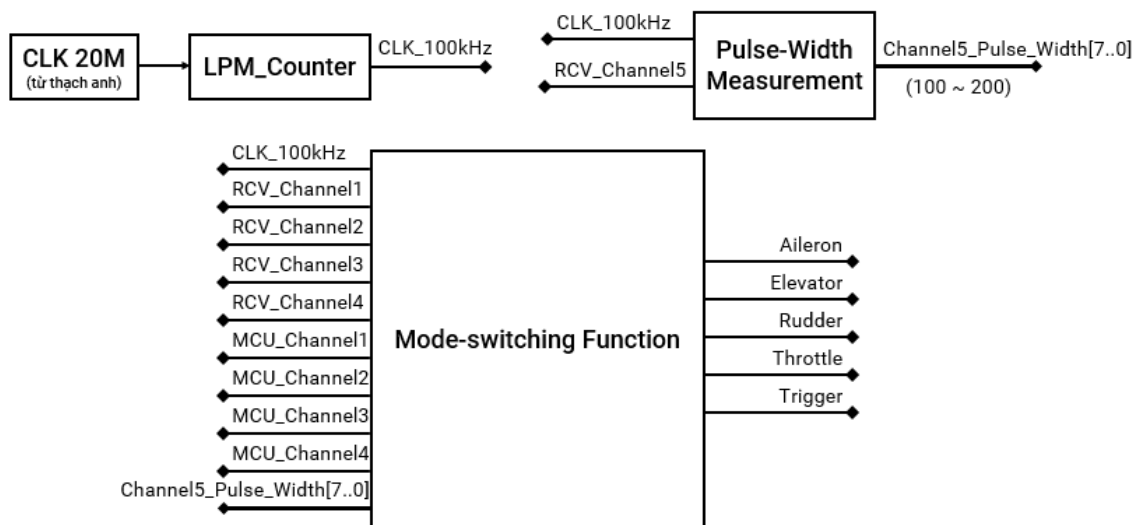
Khi phi công cất cánh và điều khiển máy bay ổn định độ cao, phi công sẽ gạt switch (một trong các switch của remote control) để chuyển sang chế độ tự động của máy bay. Đồng thời, CPLD sẽ liên tục đọc giá trị độ rộng xung của kênh tương ứng với switch này từ receiver, khi giá trị độ rộng xung đạt một mức ngưỡng nhất định, CPLD sẽ xuất tín hiệu yêu cầu vi điều khiển bắt đầu quá trình điều khiển tự động. Như *Hình 5.1*, vi điều khiển sẽ cấp xung cho động cơ thay cho remote control khi chuyển qua chế độ tự động. Cuối cùng, khi muốn hạ cánh, phi công sẽ gạt ngược switch về vị trí ban đầu, CPLD sẽ cho tín hiệu từ remote control đi qua, ngắt tín hiệu từ vi điều khiển, để cấp cho các động cơ.

## 5.2. Xây dựng hệ thống điều khiển trên CPLD

Như đã đề cập ở mục 5.1, CPLD thực hiện chức năng chính trong hệ thống là một công tắc chuyển chế độ bằng tay và tự động trong hệ thống. Chế độ hoạt động được quyết định bằng một mức ngưỡng nhất định của độ rộng xung của kênh tín hiệu tương ứng của switch từ receiver. Khi đã quyết định được chế độ điều khiển hiện tại, chương trình CPLD (viết bằng Verilog và Function Block) cho tín hiệu nào điều khiển các phần tử chấp hành: tín hiệu từ vi điều khiển hay từ remote control.

### 5.2.1. Chuyển chế độ Auto/ Manual trên CPLD

Cụ thể, trong hệ thống sẽ sử dụng Channel 5 của receiver để làm switch chuyển chế độ, các Channel từ 1 đến 4 dùng cho việc điều khiển Aileron, Elevator, Rudder và Throttle. Chương trình cho CPLD được xây dựng theo mô hình như Hình 5.2.

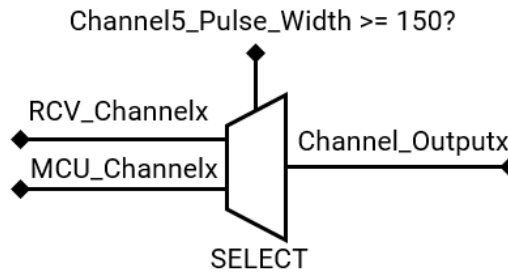


Hình 5.2: Sơ đồ khối chức năng chuyển chế độ trên CPLD

Thạch anh được tích hợp trên mạch để sử dụng cho CPLD là thạch anh 20MHz, do đó để giảm số đường dây (số bit) của ngõ ra bộ đo độ rộng xung ta phải tiến hành giảm tần số để sử dụng. Đây là lí do sử dụng bộ LPM\_Counter với thông số

Modulus là 200. Kết quả, ta có được một xung nhịp có tần số 100kHz cho bộ đo độ rộng xung.

Do xung điều khiển nhận từ receiver có độ rộng từ 1 đến 2 mili giây (như *Hình 3.8*) nên, với xung nhịp 100kHz, giá trị đo được nằm trong tầm 100 đến 200.



*Hình 5.3: Điều kiện chọn xung tín hiệu*

Khi switch trên remote control được bật, độ rộng xung đọc được là 200, tức lớn hơn mức ngưỡng 150, thì hệ thống sẽ hoạt động ở chế độ tự động, ngược lại ở chế độ người lái (như *Hình 5.3*). Nói cách khác, đây là nhiệm vụ chính của Mode-switching Function ở *Hình 5.2*.

### 5.2.2. Sơ đồ phục vụ việc gửi dữ liệu cho ngoại vi FSMC trên MCU

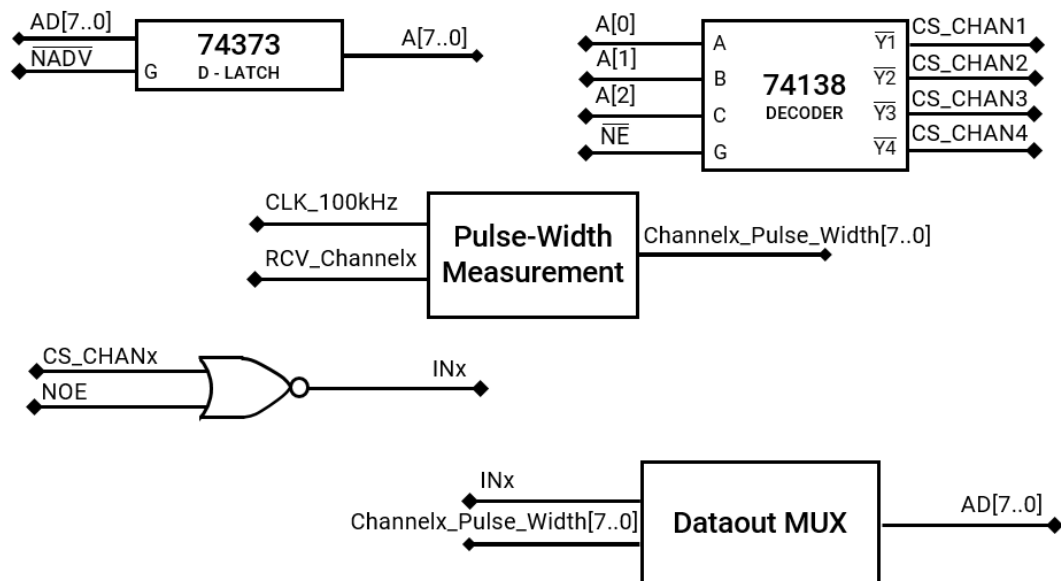
Trong quá trình điều khiển xuất hiện một vấn đề làm ảnh hưởng rất nhiều đến tính ổn định của máy bay trong quá trình chuyển chế độ. Cụ thể là tín hiệu điều khiển từ remote control và receiver không đồng độ nhau khi chuyển chế độ, do đó khi bật switch, máy bay bị giật cánh rất mạnh khi bắt đầu điều khiển. Điều này, đặt ra một vấn đề là làm sao để giảm thiểu sự chênh lệch này.

Sau nhiều lần thử nghiệm, giải pháp được đưa ra là đo độ rộng xung của các kênh điều khiển từ remote control và chuyển đổi thành thành phần  $u(k-1)$  cho thuật toán PID.

Để có thể chuyển giá trị độ rộng xung cho ngoại vi FSMC của vi điều khiển, việc cần làm là phải có một mạch giải mã địa chỉ để đọc giá trị tương ứng với 4 Channel. Sơ đồ khối cho chức năng này được thể hiện ở *Hình 5.4*.



Chế độ FSMC được sử dụng ở chế độ Address và Data dùng chung bus, do đó, ta sử dụng thêm mạch chốt 74373 để chốt địa chỉ. Mạch giải mã địa chỉ 74138 dùng để chọn kênh tương ứng với Aileron, Elevator, Rudder và Throttle. Khi kênh tương ứng được chọn, giá trị độ rộng tương ứng sẽ trả ra bus AD từ hàm Dataout MUX. (Chi tiết về cách thức hoạt động của FSMC đã được giới thiệu ở mục 3.2.4.4).



Hình 5.4: Sơ đồ khối đọc giá trị độ rộng xung từ CPLD

### 5.3. Xử lý, quản lý và lưu trữ dữ liệu

#### 5.3.1. Xử lý dữ liệu đọc về từ cảm biến IMU – GPS

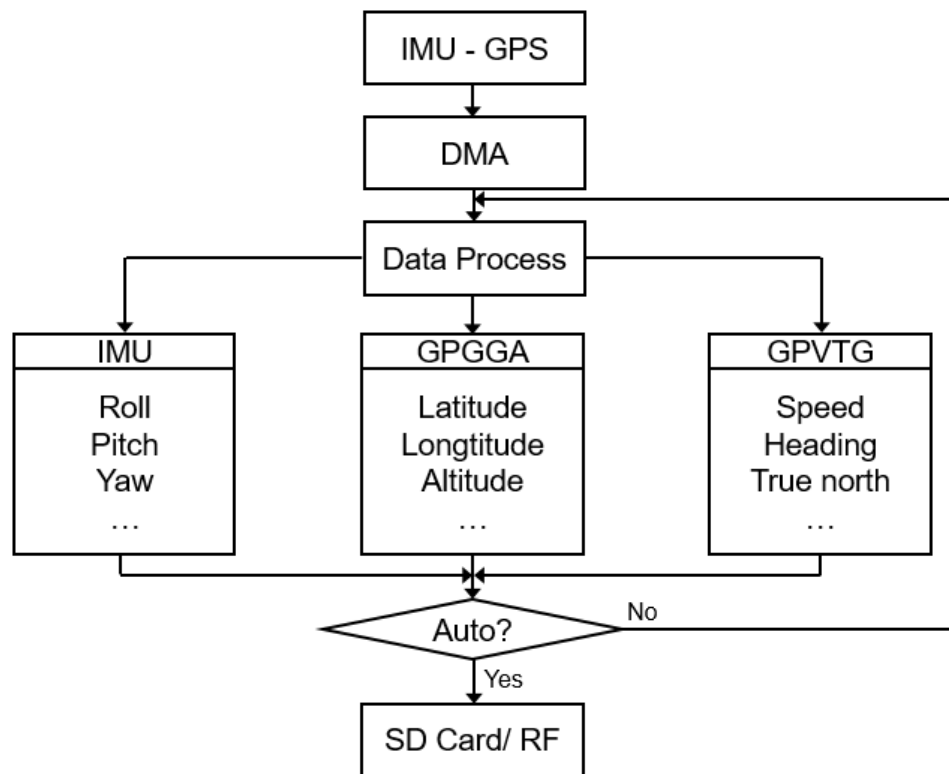
Bộ cảm biến IMU – GPS giao tiếp với vi điều khiển thông qua UART với baudrate 460 800, và cứ 10 chuỗi IMU sẽ có một chuỗi GPVGA và một chuỗi GPVTG. Hình 5.5 là một mẫu dữ liệu lấy về từ cảm biến.

```
-0106 -0136 -0137 -0860 -0150 -0763 0179 -0319 1312 0142 0068 -0040 2472
$GPVTG,156.55,T,,M,21.900,N,40.559,K,A*38
$GPGGA,035130.90,1046.16158,N,10645.69210,E,1,09,0.90,24.6,M,-2.2,M,,*7E
```

Hình 5.5: Cấu trúc một mẫu dữ liệu từ cảm biến

Dựa vào Hình 5.5 ta thấy được rằng mỗi chuỗi có độ dài tương đối lớn, ta không thể sử dụng chương trình ngắt UART để xử lý vì chương trình sẽ tốn rất nhiều thời gian trong chương trình ngắt, ảnh hưởng hưởng rất nhiều đến chu kỳ điều

khiến 20 ms của hệ thống. Do vậy, việc sử dụng ngoại vi DMA là vô cùng phù hợp (sơ lược về DMA được trình bày ở mục 3.2.3.3). Độ dài buffer của DMA cũng cần được cân nhắc để có thể đảm bảo thời gian xử lý dữ liệu phù hợp, vì dữ liệu mới nhận được từ UART sẽ ghi đè lên buffer một cách tuần tự.



Hình 5.6: Quá trình xử lý dữ liệu từ cảm biến IMU - GPS

### 5.3.2. Tạo khung dữ liệu để lưu trữ thẻ nhớ và truyền qua RF

Để thuận tiện cho việc phán đoán và điều chỉnh thông số, đồng thời cũng là cách để lưu trữ dữ liệu để dùng làm số liệu cho báo cáo, việc lưu trữ số liệu vào thẻ nhớ là rất cần thiết. Chỉ một số biến cần thiết sẽ được lưu trữ thành từng dòng riêng biệt cho thẻ nhớ.

Khung dữ liệu cho thẻ nhớ được thể hiện ở Hình 5.7 và một đoạn dữ liệu mẫu được thể hiện ở Hình 5.8.

Setpoint Roll [5 kí tự]	PWM Roll [4 kí tự]	PWM Pitch [4 kí tự]	Roll [6 kí tự]	Pitch [6 kí tự]	Yaw [6 kí tự]	Yaw Heading [6 kí tự]
----------------------------	-----------------------	------------------------	-------------------	--------------------	------------------	--------------------------

Yaw Stanley [6 kí tự]	Vĩ độ [10 kí tự]	Kinh độ [11 kí tự]	Trạng thái [1 kí tự]	Chỉ số quỹ đạo [1 kí tự]
--------------------------	---------------------	-----------------------	-------------------------	-----------------------------

Hình 5.8: Cấu trúc khung dữ liệu ghi vào thẻ nhớ

```

-06.5  592  626 -009.9 -015.7  015.4 187.2 163.3 1046.17775 10645.69773 1 2
-03.7  597  625 -009.2 -015.8  014.0 180.6 163.3 1046.17565 10645.69761 1 2
-01.0  598  624 -006.5 -015.8  012.5 177.2 163.3 1046.17462 10645.69765 1 2
00.0   602  627 -007.1 -015.3  010.1 174.1 163.3 1046.17360 10645.69774 1 2
00.0   598  623 -004.5 -015.8  009.2 170.9 163.3 1046.17258 10645.69789 1 2
-00.5  591  621 -001.6 -016.0  008.3 168.4 089.6 1046.17155 10645.69810 2 2
-03.2  584  621 -000.6 -015.9  006.9 166.5 089.6 1046.17052 10645.69834 2 2

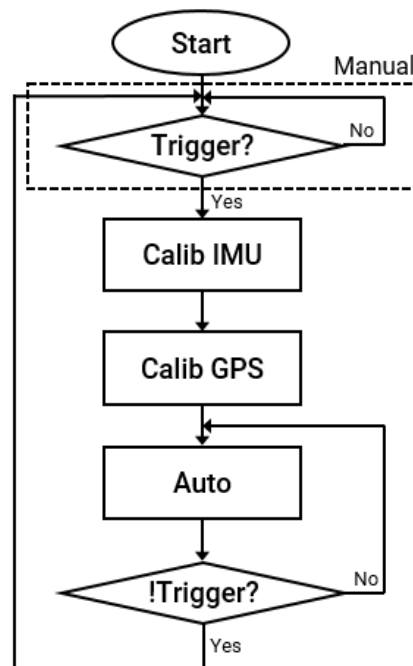
```

Hình 5.7: Một đoạn dữ liệu mẫu ghi vào thẻ nhớ

## 5.4. Xây dựng hệ thống điều khiển trên vi điều khiển

### 5.4.1. Tổng quan cấu trúc hệ thống trên vi điều khiển

Hệ thống điều khiển trên vi xử lý phục vụ hoàn toàn cho chế độ điều khiển tự động với tín hiệu kích quá trình điều khiển tự động từ CPLD (xung Trigger ở Hình



Hình 5.9: Lưu đồ giải thuật hệ thống

5.2). Cấu trúc chính của hệ thống được xây dựng theo một máy trạng thái (state machine) khép kín được thể hiện ở Hình 5.9.

Nhìn chung, hệ thống gồm 3 trạng thái chính: Manual, Calib và Auto. Cụ thể, ở trạng thái đầu tiên, chương trình trên vi xử lý sẽ đi vào chế độ Manual đầu tiên. Trong trạng thái này, nhiệm vụ duy nhất của vi điều khiển là chờ xung kích Trigger từ CPLD. Khi tín hiệu Trigger bật lên 1, hệ thống sẽ chuyển sang trạng thái Calib. Ở trạng thái này, vi điều khiển sẽ thực hiện một số tính toán phục vụ cho việc khởi tạo giá trị đầu cho các thuật toán PID và Stanley.

Trạng thái Auto đảm nhận nhiệm vụ điều khiển chính của hệ thống. Chu kì làm việc của trạng thái Auto là 20 ms, tương ứng với chu kì tối thiểu và một động cơ servo có thể đáp ứng được. Trong chu kì làm việc 20 ms đó, hệ thống sẽ xử lý dữ liệu từ IMU – GPS, tính toán các giá trị điều khiển của Aileron và Elevator dựa vào thuật toán PID và Stanley. Đồng thời, hệ thống cũng kiểm tra tín hiệu Trigger từ CPLD để xem xét việc chuyển sang trạng thái Manual.

Ngoài những ngoại vi phổ biến được sử dụng trong hầu hết các hệ thống khác như Timer, GPIO, UART,... trong hệ thống này còn sử dụng thêm 2 ngoại vi là FSMC và SDIO.

#### **5.4.2. Trạng thái Calib**

Trong trạng thái Calib, nhiệm vụ được chia thành 2 nhiệm vụ chính: Calib IMU và Calib GPS. Nhiệm vụ chính của Calib IMU là tính toán và thiết lập các thông số cần thiết từ dữ liệu IMU như là thiết lập  $K_P$ ,  $K_I$  và  $K_D$  của bộ điều khiển PID; tính toán thành phần  $u(k-1)$  (nguyên nhân đã được giải thích ở mục 5.2.2); cài đặt giá trị đặt ban đầu cho các góc.

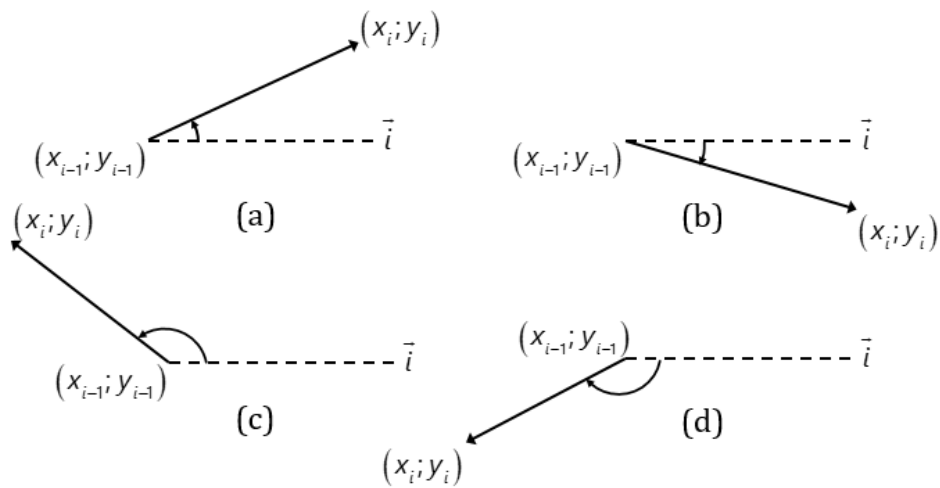
Nhiệm vụ thứ hai, Calib GPS, có chức năng phục vụ cho việc khởi tạo giá trị cho thuật toán Stanley: tính toán độ dốc các đoạn thẳng của quỹ đạo; nạp các giá trị khởi tạo của thuật toán như vận tốc và trí trí đầu.

Để có thể tính toán được độ dốc của các đoạn thẳng quỹ đạo, ta dựa vào 2 điểm liên tiếp để xác định độ dốc của đoạn thẳng (do lấy trục  $Ox$  làm gốc như đã đề cập ở mục 4.1). Do vậy, ta có thể đơn giản tính được góc  $\psi$  của từng đoạn thẳng quỹ đạo theo công thức sau:

$$\psi = \arctan \frac{y_i - y_{i-1}}{x_i - x_{i-1}}$$

Trong đó,  $i$  là thứ tự một tọa độ bất kì trong quỹ đạo mong muốn,  $(x_i; y_i)$  và  $(x_{i-1}; y_{i-1})$  là tọa độ 2 điểm liên tiếp bất kì trong quỹ đạo.

Tuy nhiên, ở đây lại xuất hiện một vấn đề, giá trị của hàm arctan luôn trả về góc nhọn, tức là các giá trị góc như Hình 5.10c và d sẽ không thể được thể hiện chính xác.



Hình 5.10: Các vị trí tương đối của cặp tọa độ liên tiếp

Như vậy, rất cần thiết để bù một góc  $\pi$  cho các góc như Hình 5.6c và d, nói cách khác ta có điều kiện như sau:

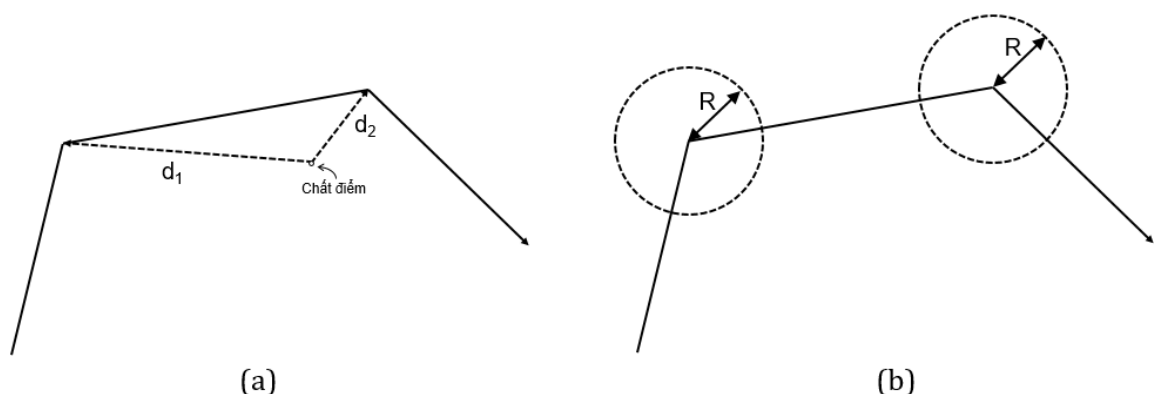
- Nếu  $x_i - x_{i-1} < 0$  thì  $\psi = \pi + \arctan \frac{y_i - y_{i-1}}{x_i - x_{i-1}}$ .

- Nếu  $x_i - x_{i-1} \geq 0$  thì  $\psi = \arctan \frac{y_i - y_{i-1}}{x_i - x_{i-1}}$ .

Ngoài ra, như đã trình bày ở *Chương 3*, ta cũng nhận thấy rằng chất điểm chỉ duy chuyển trên những đoạn thẳng quỹ đạo ngắn, tức các chất điểm buộc phải chuyển từ đoạn thẳng quỹ đạo này sang đoạn thẳng khác để có thể đi qua hết toàn bộ quỹ đạo. Như vậy, câu hỏi đặt ra là làm thế nào để biết một chất điểm đi hết đoạn quỹ đạo hiện tại hay chưa. Ở đây, có 2 cách được đề xuất cho việc quyết định này.

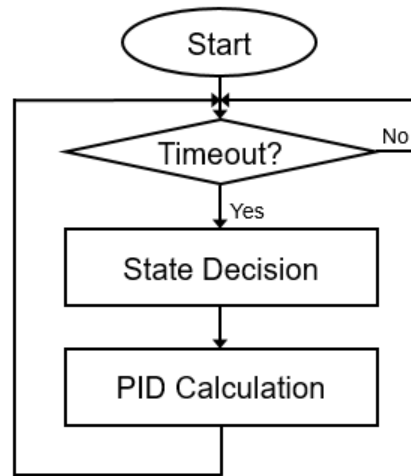
Một là, tính toán khoảng cách từ chất điểm đến 2 đầu mút của đoạn thẳng quỹ đạo đang bám, nếu tỉ lệ giữa 2 khoảng cách này ( $d_1$  và  $d_2$  như *Hình 5.11a*) đạt được một tỉ lệ nhất định thì ta sẽ cập nhật đoạn thẳng quỹ đạo mới. Cách này có ưu điểm là việc chuyển quỹ đạo sẽ không phụ thuộc vào độ dài ngắn của đoạn thẳng quỹ đạo. Tuy nhiên, việc tính toán số phức (tính khoảng cách) 2 lần ở mỗi chu kì tính toán của hệ thống là rất nặng nề đối với vi điều khiển. Do đó, ta sẽ ưu tiên cách thứ 2 hơn.

Hai là, tính khoảng cách từ chất điểm đến đầu mút sau của đoạn thẳng quỹ đạo, khi khoảng cách này nằm trong vùng bán kính  $R$  như *Hình 5.11b* thì ta sẽ cập nhật đoạn thẳng quỹ đạo mới. Như đã nói, cách này có ưu điểm về khối lượng tính toán, nhưng có nhược điểm về việc chọn tọa độ. Do vậy, khi áp dụng thực tế ta phải chọn khoảng cách  $R$  phù hợp để đảm bảo máy bay di chuyển với quỹ đạo hợp lí.



Hình 5.11: Hai đề xuất về cách chuyển đoạn thẳng quỹ đạo

### 5.4.3. Tổng quan hệ thống điều khiển trong chế độ Auto



*Hình 5.12: Sơ đồ tổng quan chế độ Auto*

Như sơ đồ tổng quan của hệ thống điều khiển trong *Hình 5.7* ta thấy hệ thống sẽ di chuyển từ Manual\_Mode đến Calib\_Mode và rồi đến Auto\_mode. Auto\_Mode chính là trung tâm điều khiển chính cho chế độ tự động của máy bay, nó bao gồm toàn bộ quá trình từ xử lý dữ liệu đến xuất các tín hiệu điều khiển động cơ và sao lưu dữ liệu. Ta sẽ lần lượt tìm hiểu cấu trúc của hệ thống từ tổng quan đến chi tiết. Trước hết, sơ đồ tổng quát nhất của chế độ Auto được thể hiện ở *Hình 5.12*.

Từ hình trên ta thấy có ba khối chính trong sơ đồ điều khiển trên:

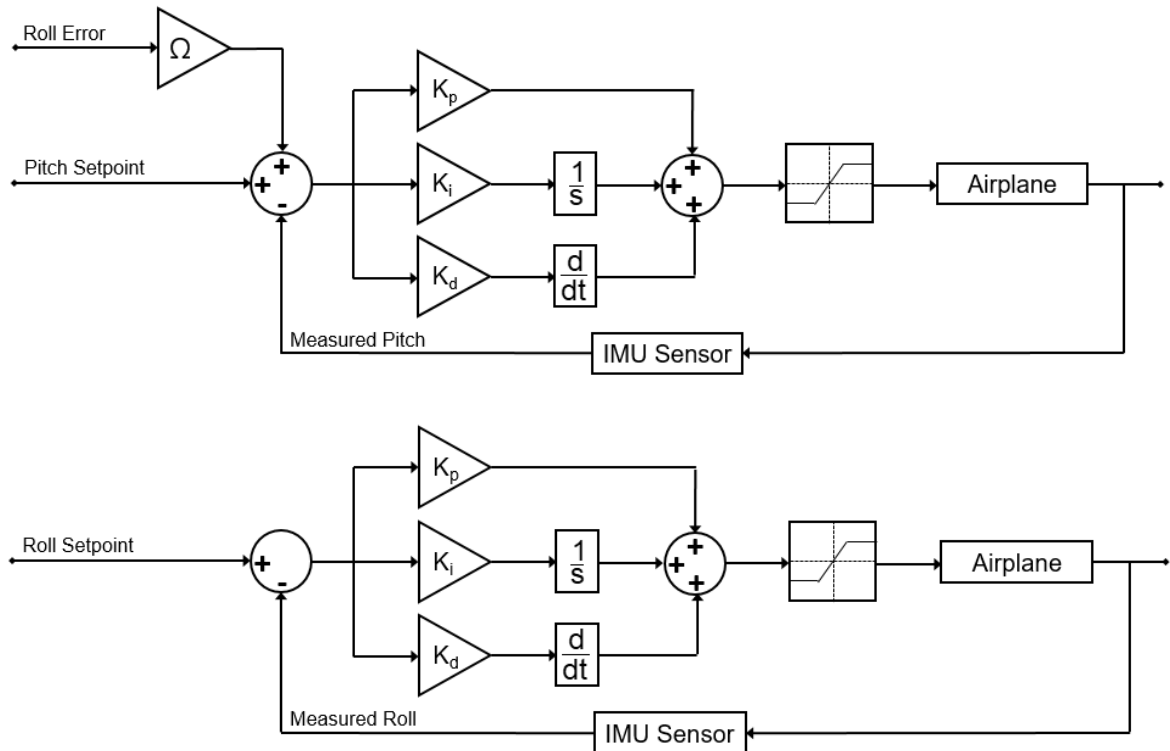
- Khối Time Out: đây là khối quy định chu kì điều khiển của bộ điều khiển PID.
- Khối State Decision: Đây là khối trung tâm đưa ra các quyết định về hướng lái và trạng thái của máy bay. Khối này được xây dựng theo dạng một máy trạng thái và sẽ được trình bày chi tiết ở mục sau của bài báo cáo.
- Khối PID: Đây là khối thuật toán PID, đây là nơi tính toán tín hiệu điều khiển và trực tiếp xuất tín hiệu điều khiển các động cơ.

#### 5.4.4. Áp dụng thuật toán PID vào hệ thống

Thuật toán PID được áp dụng để điều khiển góc Roll và góc Pitch của máy bay. Thuật toán PID được áp dụng cơ bản như lý thuyết, tuy nhiên để điều khiển cụ thể vào máy bay ta cần phải điều chỉnh và bổ sung vào một số thành phần khác. Trước hết ta cũng đến với sơ đồ điều khiển thuật toán PID được áp dụng vào góc Roll và góc Pitch.

Từ sơ đồ *Hình 5.13* ta thấy, so với thuật toán PID kinh điển có ba sự thay đổi chính như sau:

- Các bộ bão hòa tín hiệu điều khiển: các bộ bão hòa này đóng vai trò vô cùng quan trọng, nó quy định phạm vi mà tín hiệu điều khiển có thể xuất ra được, điều này đồng nghĩa với việc quy định biên độ chuyển động của các cánh tương ứng. Mỗi mô hình máy bay cụ thể mang trong mình một đặc tính vật lý khác nhau, do đó các biên độ cánh này cần được chọn vừa đủ để máy bay có thể bay ổn định và không rơi vào những vùng biên độ gây mất khả năng cân bằng của máy bay.



*Hình 5.13: Sơ đồ thuật toán PID có hiệu chỉnh dành cho góc Roll và góc Pitch*



■ Thành phần quan hệ giữa góc Roll và góc Pitch: Trong điều khiển máy bay ta không thể điều khiển tách biệt các góc Roll và Pitch bởi máy bay là một khối và do đó khi một góc thay đổi sẽ dẫn tới các góc khác cũng sẽ thay đổi theo. Chính vì vậy cần có bộ tính toán để bù trừ mối quan hệ giữa các góc. Ở đây khi góc Roll nghiêng khỏi vị trí cân bằng sẽ làm máy bay mất cân bằng và dẫn tới mất độ cao, do đó ta cần ngay lập tức bù phần nghiêng này vào góc Pitch nhằm nâng cánh Elevator lên tỉ lệ với độ nghiêng góc Roll theo một hàm  $f(x)$  để cải thiện độ cao. Cụ thể trong trường hợp này  $f(x) = \Omega * Roll\_Error$  là hàm tuyến tính [12].

■ Thành phần bù tín hiệu điều khiển khi chuyển từ Manual\_Mode sang Auto\_Mode: Khi phi công lái máy bay và giữ cho máy bay cân bằng thì các cánh Elevator và Aileron không luôn nằm ở vị trí cân bằng, thường sẽ bị lệch khỏi vị trí cân bằng nhằm tạo lực để bù với lực gió. Hơn nữa mỗi một lần bay là một lần điều kiện môi trường khác nhau dẫn tới độ lệch cánh này là không cố định và tệ hơn là khi chuyển mode phi công không thể kiểm soát tuyệt đối được mình có thả thẳng máy bay hay không và thường máy bay bị nghiêng và lệch khỏi vị trí cân bằng. Nếu ta chuyển trực tiếp từ Manual\_Mode sang Auto\_Mode sẽ tạo nên sự chênh lệch lớn giữa tín hiệu đang được phi công điều khiển và bộ điều khiển xuất ra làm máy bay giật mạnh và có thể dẫn đến rớt máy bay. Để hiểu cách xử lý tình trạng này, trước tiên ta xem qua những phương trình của thuật toán PID rồi rạc được nhúng xuống vi điều khiển:

$$u(k) = u(k-1) + a.e(k) + b.e(k-1) + c.e(k-2)$$

Trong đó,  $a = K_p + K_i \frac{T_s}{2} + \frac{K_d}{T_s}$ ,  $b = -K_p + K_i \frac{T_s}{2} - \frac{2K_d}{T_s}$  và  $c = \frac{K_d}{T_s}$ .

Ta sẽ giải quyết bằng cách: khi phi công chuyển từ Manual\_Mode sang Auto\_Mode thì ta sẽ tính đo chu kỳ xung điều khiển các động cơ Servo đang được phát đi từ tay cầm điều khiển bằng FPGA, sau đó vi điều khiển sẽ đọc giá trị đó và tính toán ra cần một tín hiệu điều khiển bao nhiêu để đạt được tần số xung tương ứng như vậy, tiếp tục ta sẽ gán giá trị tính được này vào thành phần  $u(k$

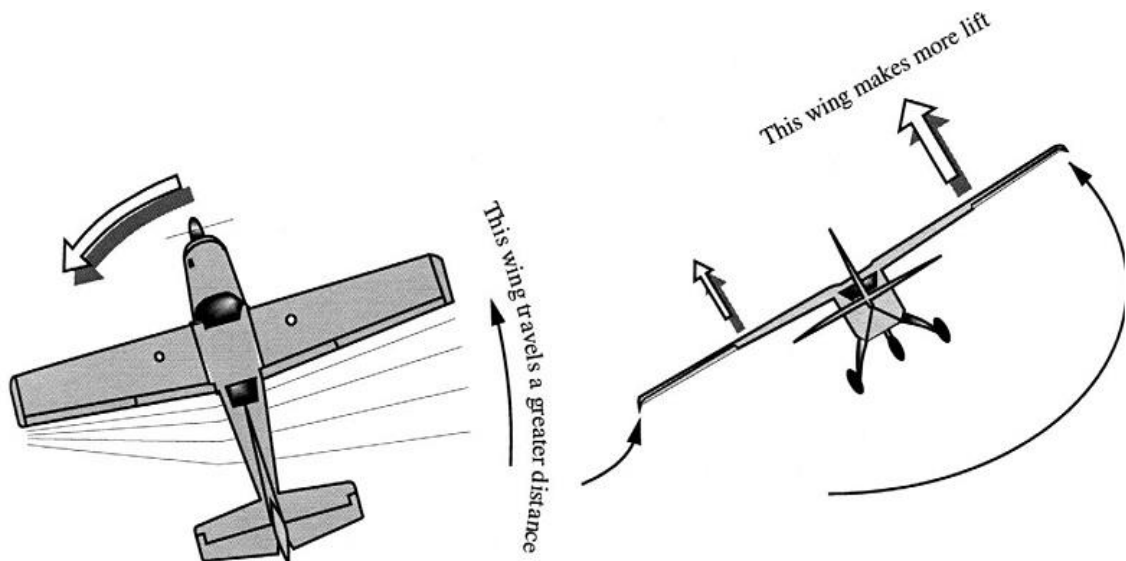
– 1) của công thức trên. Điều này sẽ áp tính hiệu điều khiển hiện tại đang có vào bộ điều khiển và sẽ tiếp tục từ đó tính đi giúp máy bay không bị hiện tượng giật lắc khi phi công chuyển chế độ, bất chấp trạng thái hiện tại của máy bay.

#### 5.4.5. Giải thuật quyết định và quản lí trạng thái bay trong khối State Decision

##### 5.4.5.1. Cơ chế bay dùng để điều khiển máy bay

Để dễ dàng hơn trong việc hiểu giải thuật trong khối State\_Decision ta xem qua cách thức để máy bay có thể rẽ cũng như giữ cân bằng bay thẳng:

- Giữ cân bằng: Để giữ cân bằng máy bay ta sử dụng các cách Elevator và Aileron để giữ máy bay đạt được các Setpoint khi cân bằng.
- Rẽ theo một góc xác định: Để có thể rẽ một góc xác định ta sẽ không sử dụng cánh Ruder như nguyên lí cơ bản của máy bay, vì cánh này rẽ góc quá gấp và làm máy bay mất cân bằng cũng như độ cao rất nhanh. Đối với những máy bay thật trong thực tế người ta cũng không dùng cách này. Thay vào đó việc rẽ sẽ được chia làm ba bước: nghiêng góc Roll một góc xác định, nâng góc Pitch để bù độ cao và khi đạt góc Roll cần thiết thì nâng mạnh để rẽ góc lớn và khi đã rẽ được góc



Hình 5.14: Cơ chế rẽ trái/ phải của máy bay

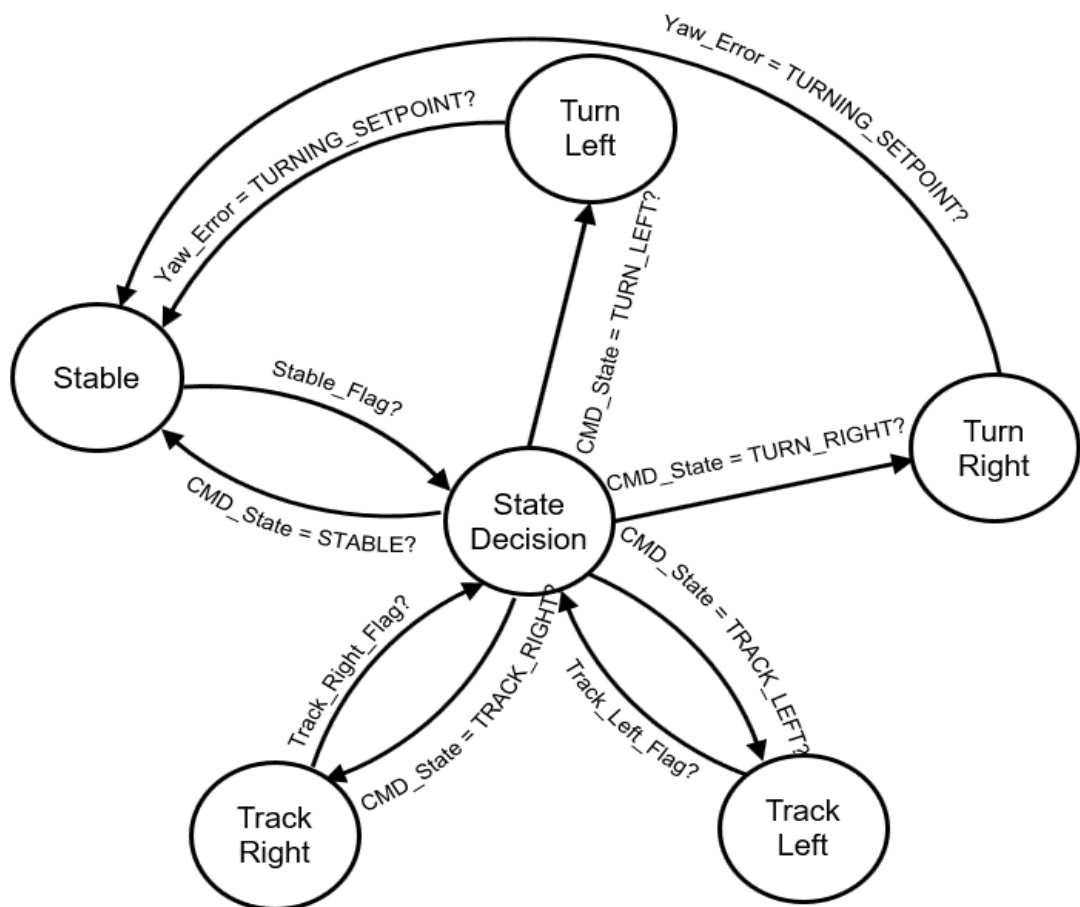
Yaw mong muốn thì trở lại trạng thái cân bằng. Sẽ được nói rõ hơn khi phân tích về các trạng thái của khối State Decision.

#### 5.4.5.2. Tổng quan máy trạng thái trong khối State Decision

Để điều khiển máy bay ta xây dựng một máy trạng thái bao gồm 6 trạng thái chính, đó là: State Decision, Stable, Turn Right, Turn Left, Track Right và Track Left. Trước tiên, ta xem qua mối quan hệ giữa các trạng thái trong *Hình 5.15*.

Ý nghĩa và chức năng của các trạng thái:

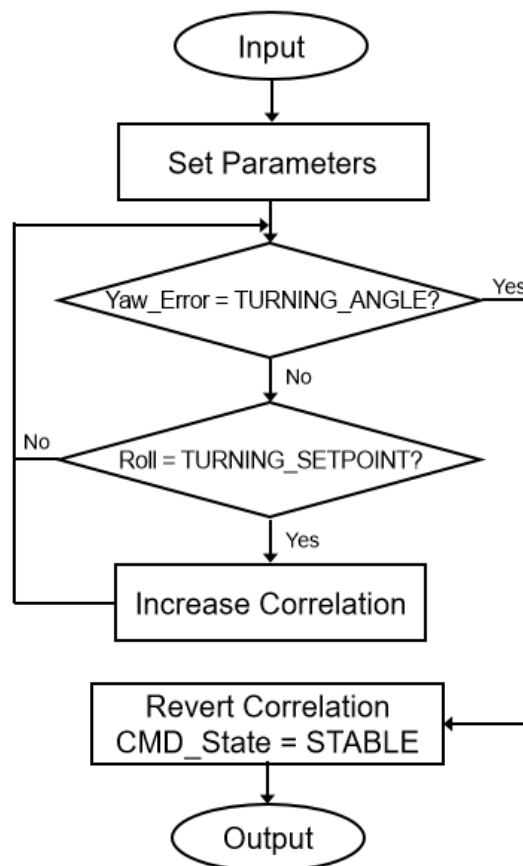
- Trạng thái Stable: Đây là trạng thái mà khi máy bay được yêu cầu ở trong trạng thái này, các Setpoint góc Roll và Pitch sẽ được thiết lập thành các điểm đặt cân bằng. Khi máy bay bám thành công các Setpoint mong muốn thì cờ Stable\_Flag sẽ được bật lên.



*Hình 5.15: Sơ đồ máy trạng thái trong khối State Decision*

- Trạng thái Track Right và Track Left: Đây là hai trạng thái mà khi máy bay được yêu cầu ở trong trạng thái này, Setpoint góc Roll sẽ được thay đổi để tạo một góc nghiêng nhỏ sang trái hoặc sang phải nhằm giúp máy bay bay là vào đường quỹ đạo khi sai số giữa góc Yaw máy bay và góc Yaw mong muốn được trả ra từ thuật toán Stanley chênh lệch nhỏ. Khi đạt được góc Roll mong muốn cờ tương ứng Trackleft\_Flag và Trackright\_Flag sẽ được bật.

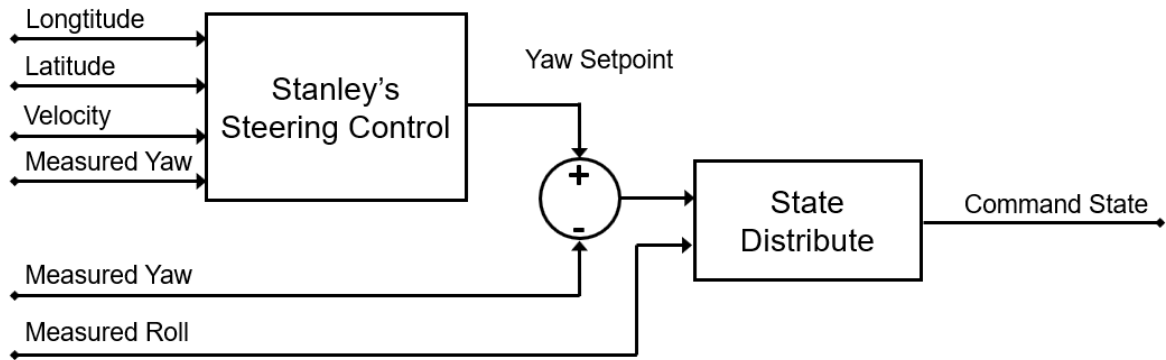
- Trạng thái Turn Right và Turn Left: Đây là hai trạng thái mà khi máy bay được yêu cầu ở trong trạng thái này, Setpoint góc Roll sẽ được thay đổi để tạo một góc nghiêng lớn và khi góc Roll đạt được góc Roll mong muốn thì lập tức tăng mối quan hệ giữa độ nghiêng góc Roll góc Pitch nhằm nâng mạnh cánh Elevator để tạo một góc rẽ lớn. Trong khi đó, góc Yaw sẽ được kiểm tra liên tục xem đã rẽ được góc cần thiết hay chưa. Khi đạt được góc Roll đạt được góc Roll mong muốn



Hình 5.16: Sơ đồ trạng thái Turn Right và Turn Left

cờ Turnleft\_Flag, Turnright\_Flag tương ứng sẽ được bật và khi góc Yaw quay được góc cần thiết thì sẽ lập tức chuyển sang trạng thái Stable. Sơ đồ như Hình 5.16.

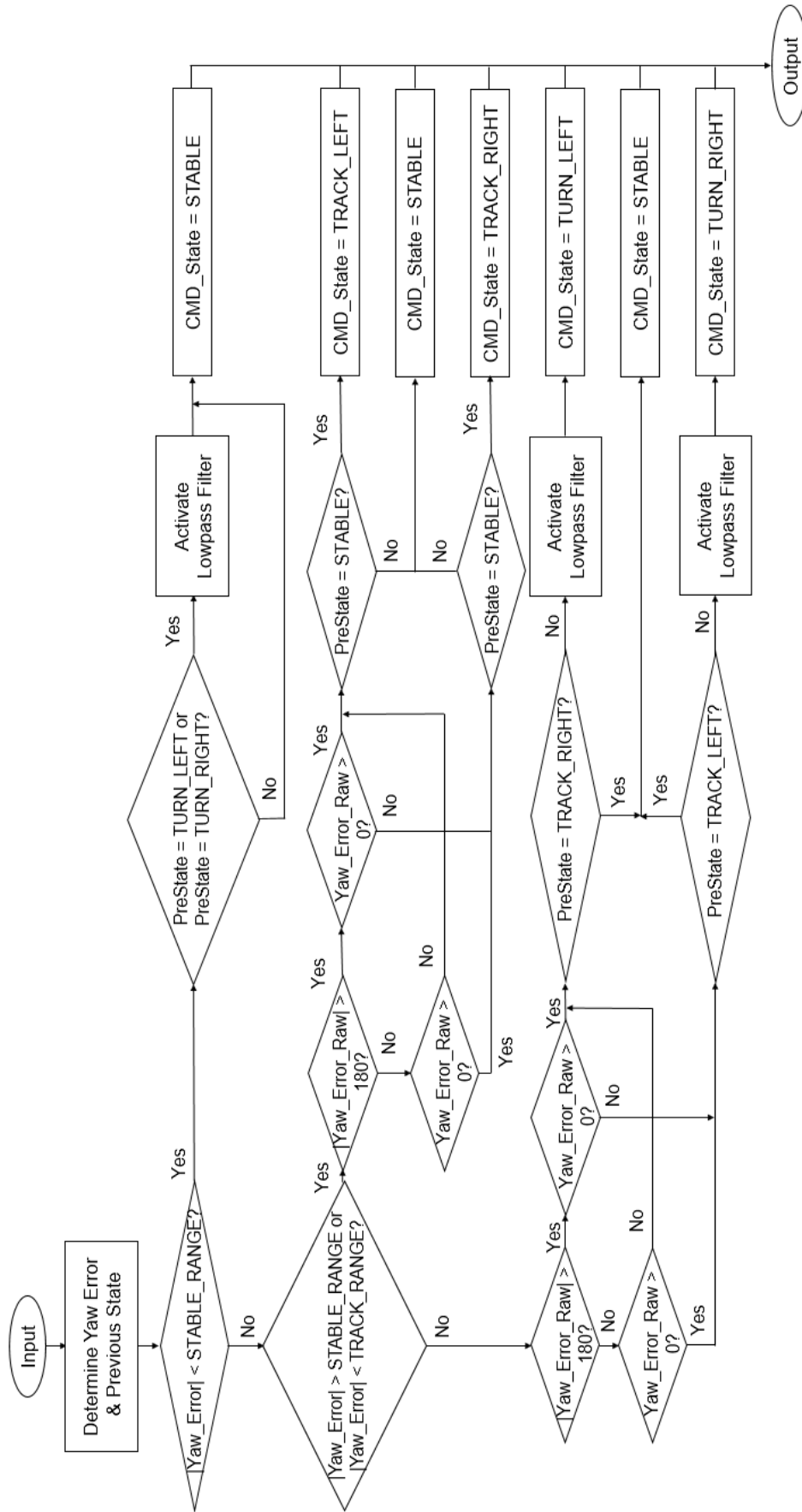
■ Trạng thái State\_Decision: Đây là trạng thái trung tâm và chuyển tiếp giữa các trạng thái, dựa vào các thông tin được đưa ra từ thuật toán Stanley và các cảm biến để quyết định và phân phối trạng thái. Bằng việc kết hợp nhịp nhàng 5 trạng thái ở trên sẽ giúp cho máy bay bám được quỹ đạo. Đây cũng chính là nơi điều khiển để đưa sai số góc Yaw về không. Sơ đồ điều khiển tổng quan của khối State Decision được thể hiện ở Hình 5.17.



Hình 5.17: Sơ đồ khối trạng thái State\_Decision

Khối State\_Distribute (có sơ đồ giải thuật cụ thể như Hình 5.18) là khối cốt lõi điều hành hệ thống, khối này hoạt động cơ bản như sau:

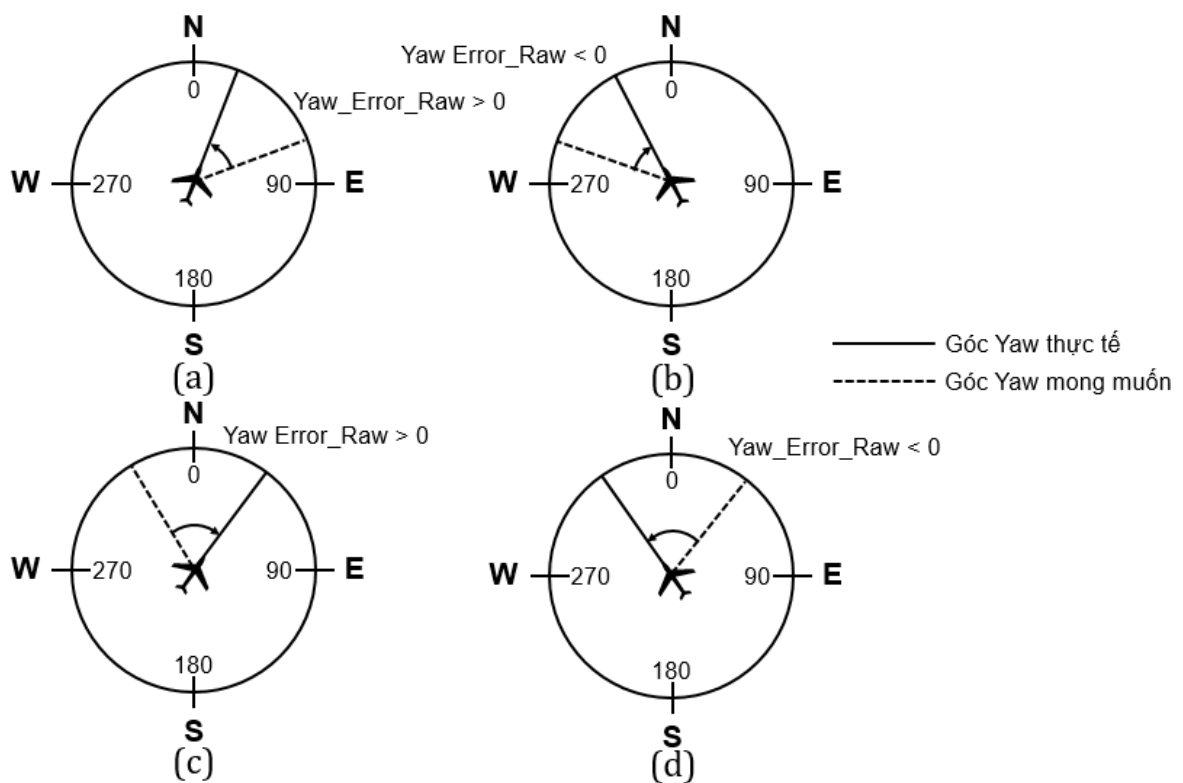
- Input của khối này chính là sai số góc Yaw (Yaw\_Error\_Raw) giữa góc Yaw mong muốn có được từ thuật toán Stanley và góc Yaw hiện thời của máy bay được lấy về từ GPS (góc Heading) và Yaw\_Error là sai số góc Yaw được xử lý từ Yaw\_Error\_Raw sao cho Yaw\_Error chỉ nằm trong khoảng  $[0^\circ; 180^\circ]$ . Ở đây ta không chọn góc Yaw từ trường lấy từ cảm biến IMU mà lấy từ góc Heading của GPS là vì góc lấy từ từ trường thường không chính xác và bị ảnh hưởng nhiều bởi các mạch điện và động cơ phát sinh ra từ trường. Góc Heading của GPS thì tốt hơn tuy nhiên chỉ có khi bắt được tín hiệu GPS và phải đạt được một vận tốc xác định. Đối với máy bay thì hai điều kiện này đều dễ dàng được thỏa mãn.



Hình 5.18: Sơ đồ hoạt động của khối State Distribute

- Vì cảm biến GPS luôn có sai số và máy bay không thể rẽ liên tục trong thời gian dài nên ta sẽ không liên tục bám sát theo giá trị góc Yaw mong muốn từ thuật toán Stanley. Để giữ cho máy bay ổn định hơn ta sẽ chỉ dựa vào mối tương quan giữa góc Yaw GPS và góc Yaw Stanley và từ đó đưa ra quyết định trạng thái kế tiếp của máy bay. Quá trình đưa quyết định đó bao gồm những bước chính như sau:

- Đầu tiên ta dựa vào độ lớn Yaw\_Error (chỉ nằm trong khoảng  $[0^0; 180^0]$ ) giữa góc Heading của máy bay được lấy từ GPS và góc Yaw mong muốn từ thuật toán Stanley để quyết định trạng thái tiếp theo của máy bay sẽ là STABLE, TRACK hay TURN. Ta sẽ chia sai số ra làm ba khoảng chính: STABLE\_RANGE, TRACK\_RANGE và lớn hơn TRACK\_RANGE chính là TURN\_RANGE. Khi sai số của máy rơi vào vùng nào thì máy bay sẽ tiến hành chuyển đến trạng thái đó. Nếu trạng thái tiếp theo là STABLE ta có thể cho máy bay vào ngay trạng thái STABLE. Tuy nhiên



Hình 5.19: Cách xác định sai số góc Yaw

nếu trạng thái tiếp theo là TRACK hay TURN thì ta còn phải xác định thêm là máy bay cần bám hoặc rẽ bên trái hay bên phải.

- Việc xác định máy bay bám hoặc rẽ bên trái ta sẽ dựa vào dấu của sai số để xác định. Quá trình đó được thực thể hiện ở *Hình 5.19*.

Từ *Hình 5.19* ta thấy với sai số Yaw\_Error\_Raw là hiệu giữa góc Stanley và góc Yaw của máy bay thì sẽ có hai trường hợp xảy ra. Nếu giá trị tuyệt đối sai số hai góc này là một góc nhỏ hơn  $180^\circ$  thì khi Yaw\_Error\_Raw  $> 0$  máy bay sẽ cần bay về hướng bên phải và ngược lại khi Yaw\_Error\_Raw  $< 0$  máy bay sẽ cần bay về hướng bên trái. Trong trường hợp giá trị tuyệt đối sai số hai góc này là một góc lớn hơn  $180^\circ$  thì khi Yaw\_Error\_Raw  $> 0$  máy bay sẽ cần bay về hướng bên trái và ngược lại khi Yaw\_Error\_Raw  $< 0$  máy bay sẽ cần bay về hướng bên phải.

- Sau khi đã xác định được hướng rẽ hoặc track cho máy bay, để tăng độ ổn định và nhịp nhàng cho máy bay ta còn cần phải tuân theo một số nguyên tắc sau:

- Sẽ không thể tồn tại trường hợp từ trạng TURN\_LEFT (TURN\_RIGHT) chuyển sang trạng thái TURN\_RIGHT (TURN\_LEFT). Vì cuối mỗi lần máy bay vào các trạng thái này máy bay sẽ ngay lập tức được đưa về trạng thái STABLE mà không cần phải chờ quyết định từ khối State\_Decision như trong *Hình 5.15*. Như vậy, giả sử như máy bay cần đến trạng thái tiếp theo là TURN\_RIGHT thì máy bay chỉ có thể từ ba trạng thái STABLE, TRACK\_LEFT và TRACK\_RIGHT chuyển đến TURN\_RIGHT. Để tránh sự thay đổi Setpoint quá lớn và đột ngột cũng như là giúp máy bay ổn định, nhịp nhàng ta sẽ không cho phép chuyển trực tiếp từ TRACK\_LEFT sang TURN\_RIGHT mà từ TRACK\_LEFT cần phải chuyển qua STABLE trước rồi sẽ tiếp tục xem xét, nếu vẫn còn cần rẽ phải thì mới được chuyển sang TURN\_RIGHT. Tương tự như vậy, từ TRACK\_RIGHT cần phải qua STABLE trước rồi mới tới được TURN\_LEFT. Từ TRACK\_RIGHT (TRACK\_LEFT) cũng phải buộc qua STABLE trước khi muốn tới TRACK\_LEFT (TRACK\_RIGHT).



- Các trạng thái khác nhau sẽ có các Setpoint cho các góc Roll, Pitch khác nhau và khi máy bay được ra lệnh đến một trạng thái nào đó nó buộc phải đạt được trạng thái đó trước khi được quay về khối State\_Decision để quyết định trạng thái kế tiếp. Nhưng vậy, sẽ không có trường hợp trạng thái này chưa hoàn thành xong đã chuyển sang trạng thái khác gây mất ổn định cho máy bay. Để kiểm tra các máy bay đã đạt được trạng thái mong muốn hay chưa ta sẽ tính sai số giữa điểm đặt các góc Roll, Pitch và giá trị đọc về từ cảm biến. Khi sai số này nhỏ hơn một mức ngưỡng định trước thì ta coi như máy bay đạt được trạng thái đó.

- Khi có sự thay đổi điểm đặt lớn để tránh sự rung lắc, dao động của máy bay ta cần thêm vào đó các bộ lọc thông thấp. Các bộ lọc này có chức năng đơn giản là làm giảm độ dốc khi thay đổi điểm đặt lớn giúp máy bay không bị dao động khi chuyển trạng thái.

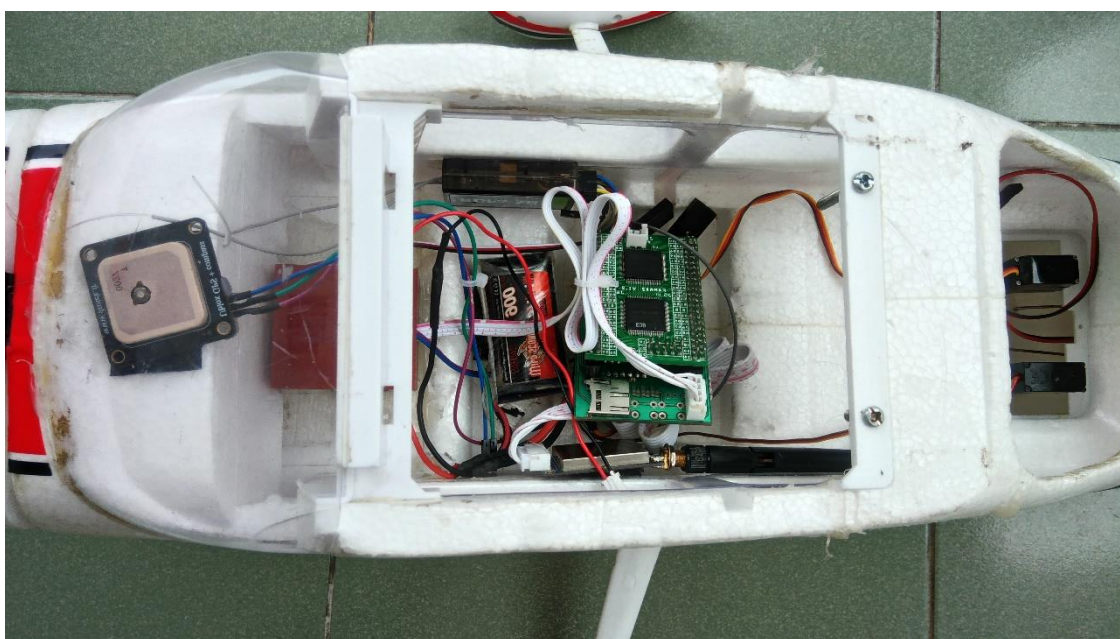
- Sau khi đã có được trạng thái tiếp theo khối State\_Distribute sẽ đặt trạng thái mong muốn vào CMD\_State (Command State). Sau đó các điểm đặt tương ứng sẽ được chuẩn bị và đặt vào khối PID trong chu kì tiếp theo.

## Chương 6. KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN

### 6.1. Kết quả mô hình thực tế

Đã xây dựng thành công mô hình máy bay cánh trên với các thông số kỹ thuật như sau:

- *Độ rộng cánh:* 1410 mm.
- *Chiều dài:* 1100 mm.
- *Trọng tải bay:* 1100 g.
- *Hệ thống lái:* động cơ brushless 900KV.
- *Độ điều khiển tốc độ:* 30A Brushless ESC.
- *Servo:* 6X 9g.
- *Pin:* 11.1V 2200 mAh 25C Li-Po.
- *Tầm điều khiển:* 1000m.



Hình 6.1: Các thiết bị điện bên trong khoang máy bay



*Hình 6.2: Mô hình máy bay được lắp đặt hoàn thiện*

## **6.2. Kế hoạch quá trình bay thí nghiệm**

Tư tưởng chính để điều khiển máy bay chính là chia quá trình bay của máy bay thành các trạng thái cơ bản nhất như STABLE, TRACK, TURN. Từ đó, xây dựng những quy tắc kết hợp những trạng thái cơ bản này để điều khiển máy bay bám theo quỹ đạo. Như vậy, để có thể thực hiện ý tưởng điều khiển này ta cần tập trung xây dựng những trạng thái cơ bản cho thật tốt và ổn định. Từ đó kết hợp những trạng thái cơ bản một cách nhịp nhàng để giúp máy bay bám theo được quỹ đạo. Chính vì lí do đó, chúng em chia quá trình bay thực nghiệm thành các giai đoạn chính như sau:

*Giai đoạn 1:* Bay thực nghiệm, điều chỉnh và lựa chọn những thông số tốt nhất cho từng trạng thái cơ bản của máy bay. Giai đoạn này gồm có ba công đoạn chính:

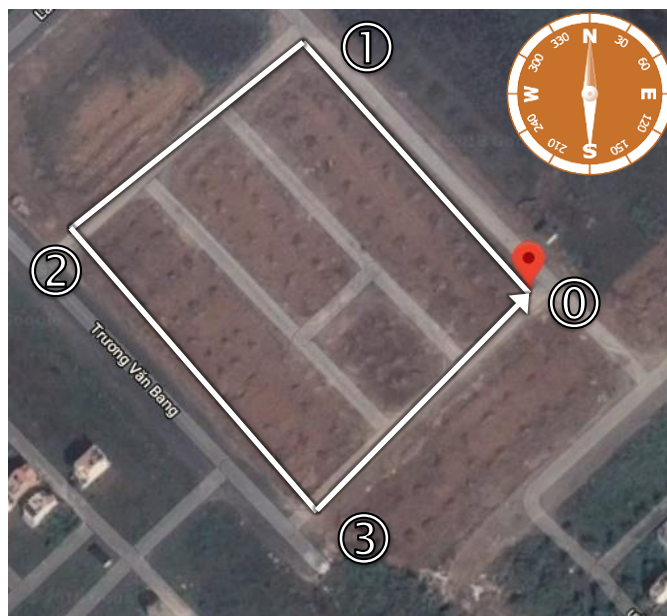
- Điều chỉnh và lựa chọn thông số cho trạng thái STABLE

- Điều chỉnh và lựa chọn thông số cho trạng thái TRACK\_LEFT, TRACK\_RIGHT
- Điều chỉnh và lựa chọn thông số cho trạng thái TURN\_LEFT, TURN\_RIGHT

*Giai đoạn 2:* Xây dựng quỹ đạo bay, điều chỉnh và hoàn thiện thuật toán Stanley. Trong giai đoạn này, ta chủ yếu di chuyển trên mặt đất để kiểm tra các quá trình chuyển điểm đến trong quỹ đạo, tính đúng đắn và hợp lí của góc lái Yaw cần thiết được trả ra từ thuật toán Stanley.

*Giai đoạn 3:* Bay thực nghiệm, hoàn chỉnh giải thuật phân phối kết hợp các trạng thái cơ bản để máy bay bám theo quỹ đạo.

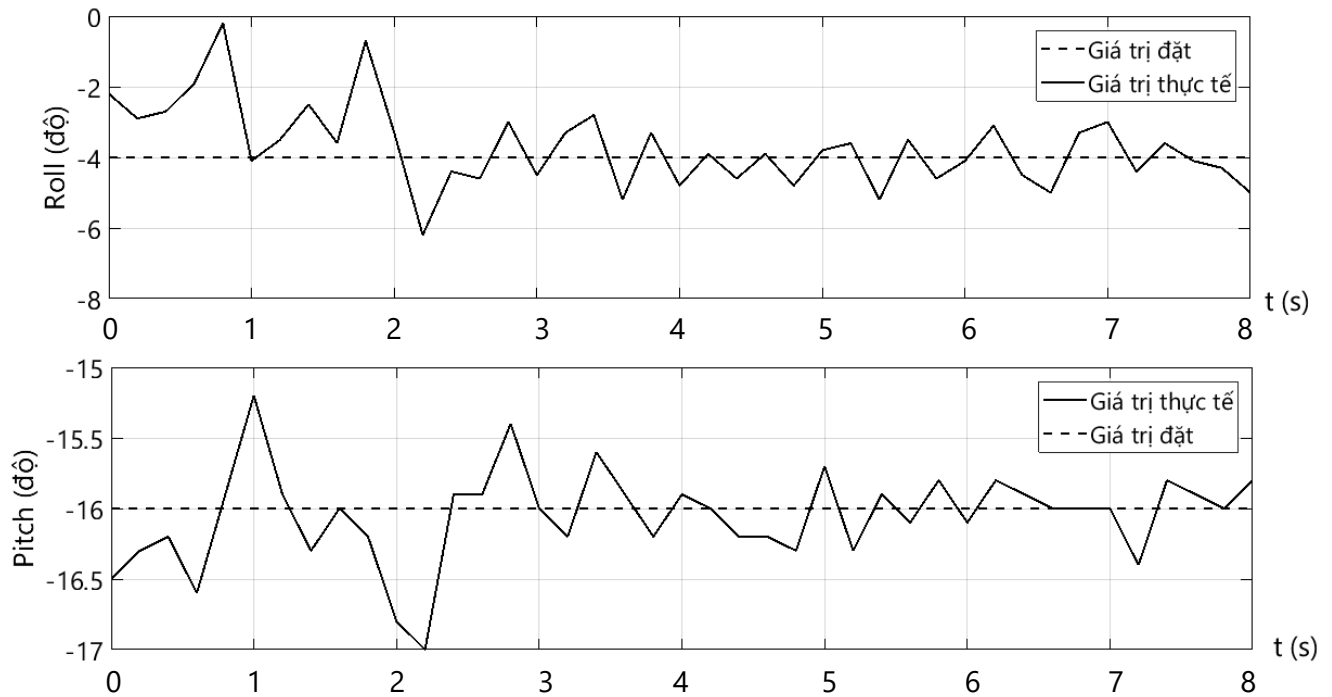
Địa điểm thử nghiệm đường Trương Văn Bang, phường Thạnh Mỹ Lợi, quận 2, thành phố Hồ Chí Minh, Việt Nam. Để dễ dàng cho việc trình bày số liệu về sau, ta quy định bốn điểm chính trong quỹ đạo lần lượt là 0, 1, 2, 3 như *Hình 6.3*.



*Hình 6.3: Quỹ đạo mong muốn nhìn từ vệ tinh*

### 6.3. Kết quả và nhận xét

#### 6.3.1. Kết quả khi máy bay ở trạng thái STABLE



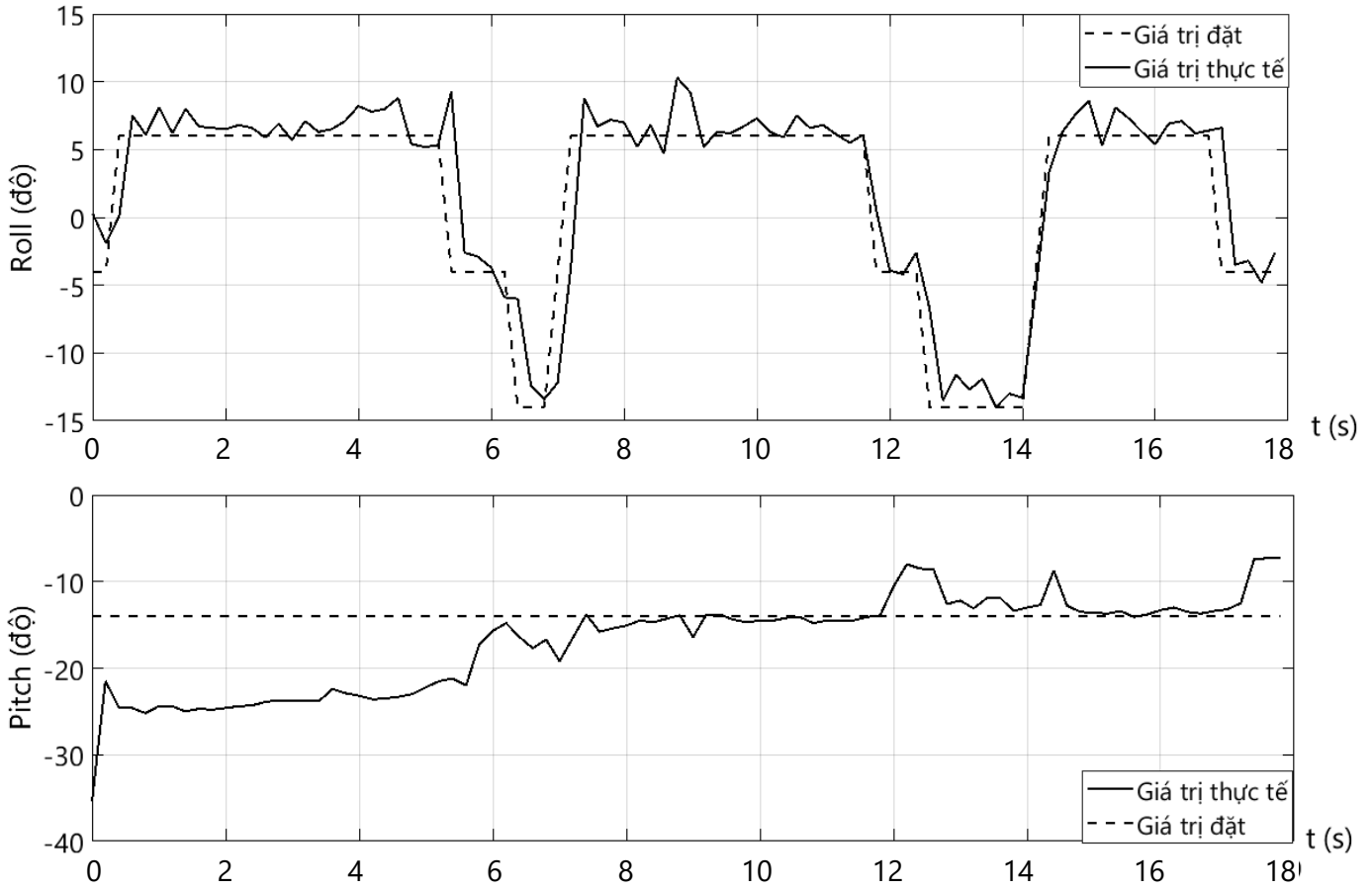
Hình 6.4: Kết quả góc Roll và Pitch ở trạng thái STABLE

Hình 6.4 là kết quả thu được khi máy bay trong trạng thái STABLE với các thông số chính như sau:

- Bộ số PID của góc Roll:  $K_P = 0.035$ ,  $K_I = 0.01$ ,  $K_D = 0.001$ .
- Bộ số PID của góc Pitch:  $K_P = 0.03$ ,  $K_I = 0.01$ ,  $K_D = 0.005$ .
- Hệ số quan hệ giữa Roll và Pitch:  $\Omega = 0.24$ .
- Hệ số điều chỉnh tốc độ bám quỹ đạo trong thuật toán  $K = 0.4$ .
- Điểm đặt góc Roll:  $-4^\circ$ .
- Điểm đặt góc Pitch:  $-16^\circ$ .

*Nhận xét:* Từ Hình 6.4 ta thấy máy bay đã cơ bản giữ cân bằng được trên không với sai số góc Roll và Pitch không vượt quá  $2^\circ$ . Ở đây ta chưa quan tâm tới góc Yaw, vì trong trạng thái STABLE chỉ yêu cầu máy bay cần giữ được trạng thái cân bằng trên không, chưa yêu cầu máy bay phải bám theo góc Yaw xác định nào.

### 6.3.2. Kết quả khi máy bay ở trạng thái TRACK



Hình 6.5: Kết quả góc Roll và Pitch ở trạng thái TRACK

Hình 6.5 là kết quả thu được khi máy bay bay luân phiên TRACK\_LEFT và TRACK\_RIGHT với các thông số chính như sau:

- Bộ số PID của góc Roll:  $K_P = 0.035$ ,  $K_I = 0.01$ ,  $K_D = 0.001$ .
- Bộ số PID của góc Pitch:  $K_P = 0.03$ ,  $K_I = 0.01$ ,  $K_D = 0.005$ .
- Hệ số quan hệ giữa Roll và Pitch:  $\Omega = 0.24$ .
- Hệ số điều chỉnh tốc độ bám quỹ đạo trong thuật toán  $K = 0.4$ .
- Điểm đặt góc Roll:  $-4^\circ$ .
- Điểm đặt góc Pitch:  $-16^\circ$ .
- Điểm đặt TRACK LEFT:  $6^\circ$ .

- Điểm đặt TRACK\_RIGHT:  $-14^0$ .

*Nhận xét:* Từ Hình 6.5 ta thấy máy bay đã cơ bản nghiêng sang trái, sang phải và giữ ổn định góc Pitch không bị hạ thấp đầu và mất độ cao, nhờ có mối quan hệ  $\Omega$  giữa góc Roll và Pitch đã giúp góc Pitch được nâng lên chống lại sự mất cân bằng khi máy bay nghiêng. Hơn nữa ta đã thực hiện được việc không cho phép chuyển trực tiếp từ TRACK\_LEFT (TRACK\_RIGHT) sang TRACK\_RIGHT (TRACK\_LEFT). Máy bay buộc phải chuyển sang và đạt được trạng thái STABLE trước, nhằm tăng độ ổn định của máy bay.

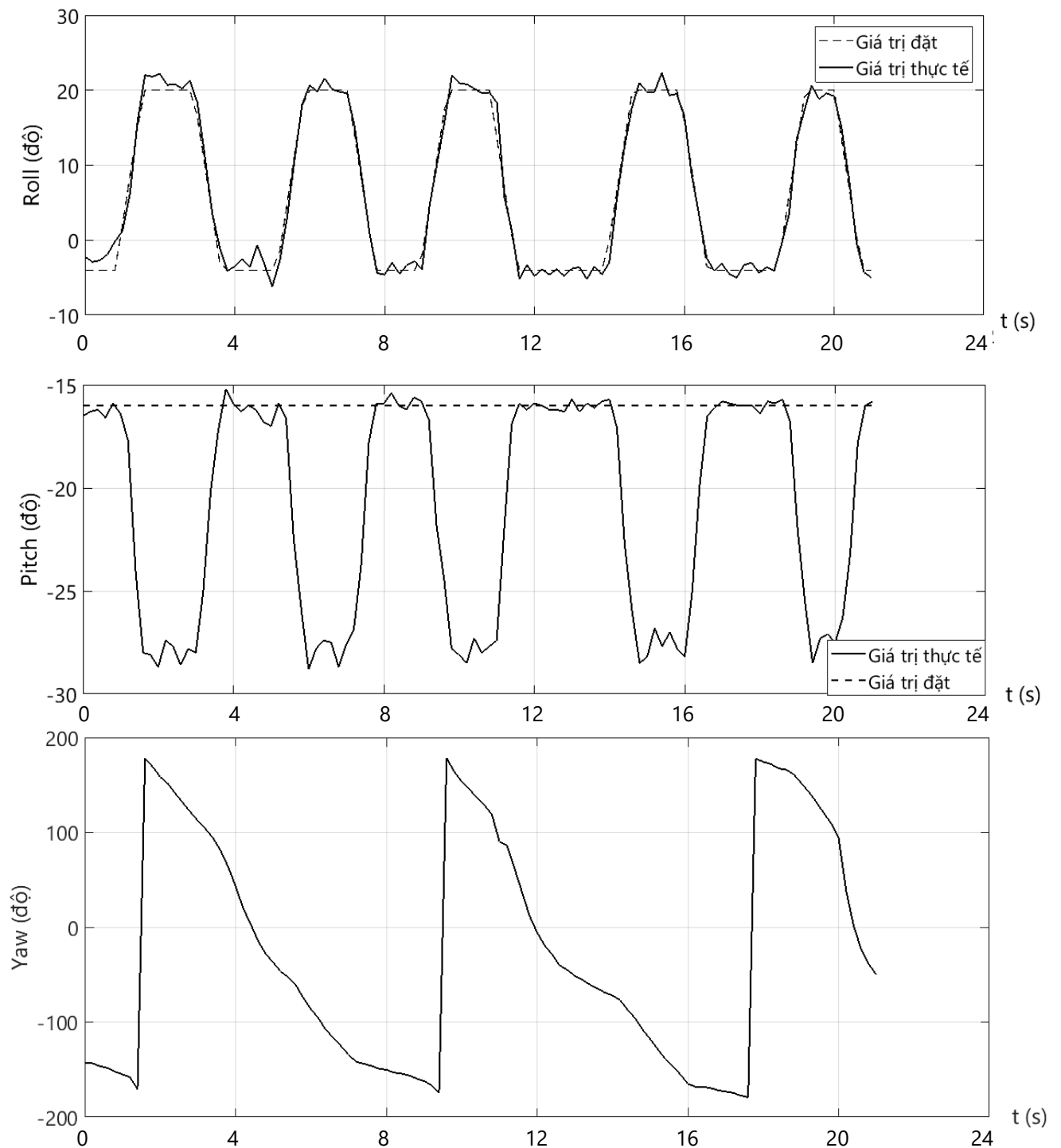
### 6.3.3. Kết quả khi máy bay ở trạng thái TURN

Hình 6.6 là kết quả thu được khi máy bay trong trạng thái TURN\_LEFT với các thông số chính như sau:

- Bộ số PID của góc Roll:  $K_P = 0.035$ ,  $K_I = 0.01$ ,  $K_D = 0.001$ .
- Bộ số PID của góc Pitch:  $K_P = 0.03$ ,  $K_I = 0.01$ ,  $K_D = 0.005$ .
- Hệ số quan hệ giữa Roll và Pitch:  $\Omega = 0.24$ .
- Hệ số điều chỉnh tốc độ bám quỹ đạo trong thuật toán  $K = 0.4$ .
- Điểm đặt góc Roll:  $-4^0$ .
- Điểm đặt góc Pitch:  $-16^0$ .
- Điểm đặt TURN\_LEFT:  $20^0$ .

*Nhận xét:* Từ Hình 6.6 ta thấy máy bay đã có thể nghiêng và giữ ổn định ở một góc nghiêng lớn và khi máy bay đạt tới được góc nghiêng lớn ( $20^0$ ) thì lập tức góc Pitch được nâng lên từ  $-16^0$  đến  $-28^0$  (trong trường hợp này, giá trị âm càng lớn thể hiện đầu máy bay được nhấc lên càng cao) điều này sẽ làm cho máy bay rẽ được một góc lớn, ta có thể nhận thấy điều này từ đồ thị góc Yaw từ trường được lấy từ IMU trong Hình 6.6, những khoảng thời gian góc Roll nghiêng lớn và góc Pitch được đẩy lên thì hệ số góc đồ thị góc Yaw tăng lên, tức là máy bay đang chuyển hướng mạnh. Hơn nữa, từ đồ thị Hình 6.6 ta thấy, khi máy bay rẽ được

một góc định trước thì lập tức trả về trạng thái STABLE để giữ cân bằng cho máy bay và chờ trạng thái tiếp theo.

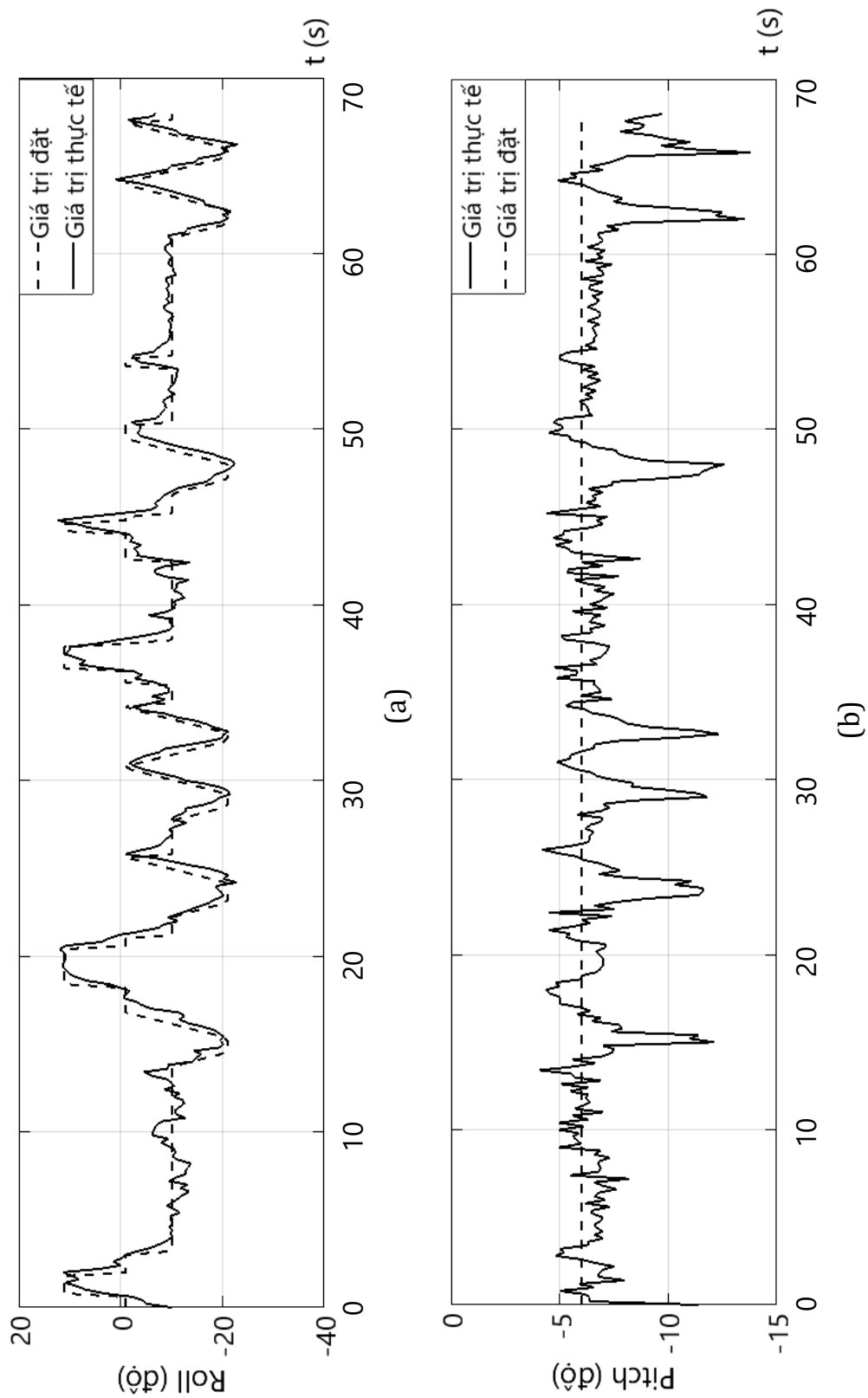


Hình 6.6: Kết quả góc Roll, Pitch và Yaw ở trạng thái TURN

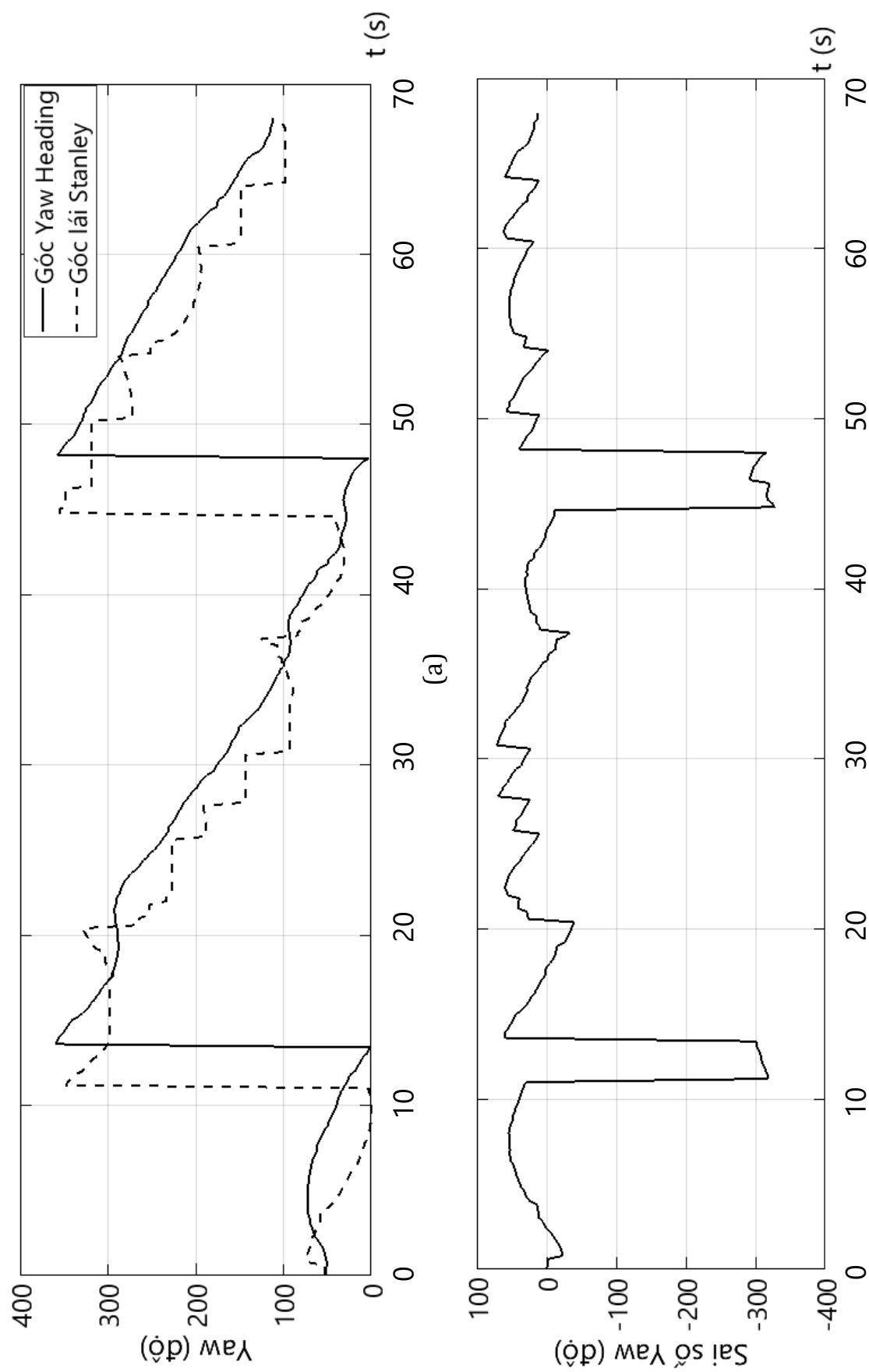


### 6.3.4. Kết quả khi máy bay thực hiện bám theo quỹ đạo

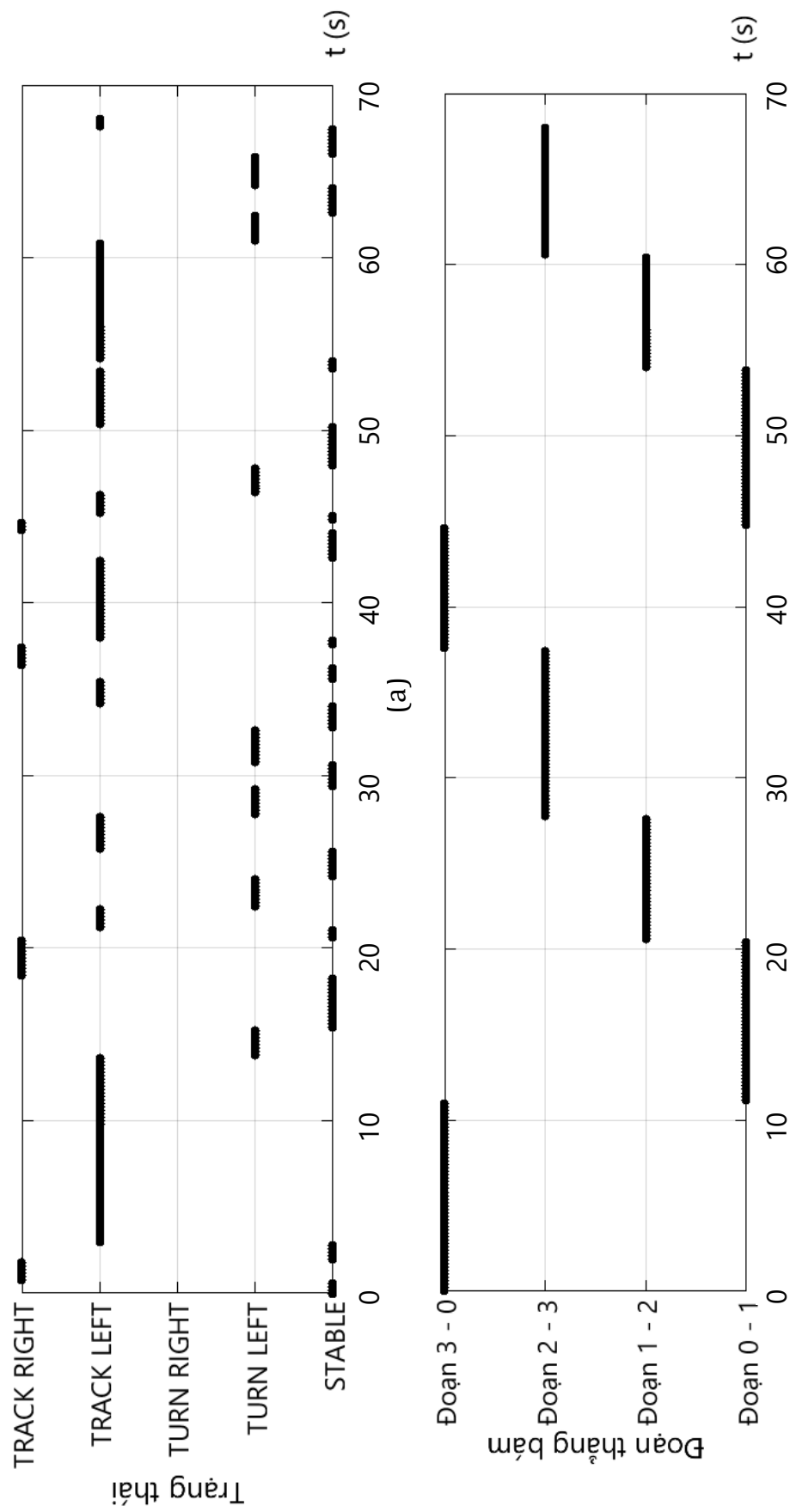
#### 6.3.4.1. Đáp ứng máy bay khi bám quỹ đạo



Hình 6.7: Kết quả góc Roll và Pitch khi bay bám quỹ đạo



Hình 6.8: Kết quả góc Yaw và sai số góc Yaw khi bay bám quỹ đạo



Hình 6.9: Sự thay đổi trạng thái và đoạn thẳng quỹ đạo khi bay bám quỹ đạo

*Hình 6.7, Hình 6.8 và Hình 6.9* là toàn bộ những kết quả thu được về đáp ứng các góc Roll, Pitch, Yaw, góc lái từ thuật toán Stanley, quá trình chuyển điểm cũng như là quá trình chuyển trạng thái hoàn chỉnh khi máy bay bay tự động bám theo quỹ đạo ngày 31/5/2018. Quỹ đạo bay được tương ứng với số liệu này được vẽ trong *Hình 6.8b* với các thông số chính như sau:

- Bộ số PID của góc Roll:  $K_P = 0.005$ ,  $K_I = 0.0015$ ,  $K_D = 0.0001$ .
- Bộ số PID của góc Pitch:  $K_P = 0.015$ ,  $K_I = 0.0001$ ,  $K_D = 0.001$ .
- Hệ số quan hệ giữa Roll và Pitch:  $\Omega = 0.2$ .
- Hệ số điều chỉnh tốc độ bám quỹ đạo trong thuật toán  $K = 0.4$ .
- Điểm đặt góc Roll trong trạng thái STABLE:  $-1^\circ$ .
- Điểm đặt góc Pitch trong trạng thái STABLE:  $4^\circ$ .
- Điểm đặt góc Roll TURN\_LEFT:  $-21^\circ$ .
- Điểm Đặt góc Roll TURN\_RIGHT:  $19^\circ$ .
- Điểm đặt Roll TRACK\_LEFT:  $-10^\circ$ .
- Điểm đặt Roll TRACK\_RIGHT:  $11^\circ$ .
- Bán kính chuyển điểm: 40m
- Khoảng sai số YAW\_ERROR\_STABLE\_RANGE:  $10^\circ$ .
- YAW\_ERROR\_TRACK\_RANGE:  $60^\circ$ .
- Góc Yaw cần rẽ khi vào trạng thái rẽ TURN\_ANGLE:  $25^\circ$ .

*Nhận xét:*

- Từ đáp ứng các góc Roll và Pitch trên *Hình 6.7a và b* ta thấy góc Roll đã có thể bám tốt theo các giá trị đặt được đưa ra từ khối State\_Distribute. Góc Pitch đã có khả năng giữ cân bằng xung quanh điểm cân bằng đặt trước. Đồng thời đã có khả năng kết hợp với góc Roll khi máy bay nghiêng khỏi vị

trí cân bằng, điều này thấy rõ nhất khi máy bay đi vào trạng thái TURN\_LEFT. Ví dụ như ở khoảng giây thứ 24 ta thấy khi máy nghiêng một góc  $21^0$  thì góc Pitch lập tức được đẩy lên cao để vừa cải thiện độ cao vừa giúp máy bay rẽ được một góc lớn. Dù bị đẩy lên cao nhưng ngay sau khi đã rẽ được một góc cần thiết góc Pitch sẽ lập tức được đưa về trạng thái cân bằng. Nhìn chung, đáp ứng các góc Roll và Pitch đã đáp ứng được yêu cầu đặt ra và thể hiện rõ được ý đồ điều khiển.

- Từ đáp ứng góc Yaw và đồ thị biểu thị sai số giữa góc Yaw Heading và góc lái cần thiết được đưa ra từ thuật toán Stanley trên *Hình 6.8a* và *b* ta thấy về cơ bản góc Yaw Heading của máy bay đã có thể bám theo góc lái được trả ra từ thuật toán Stanley. Ngoài ra ta có thể thấy rõ, khi máy bay ở trung các nhóm trạng thái rẽ (TURN) như ở giây thứ 16, 24, 48... (có thể quan sát *Hình 6.9a* để xác định rõ hơn trạng thái của máy bay với thời gian bay tương ứng) thì hệ số góc đồ thị góc Yaw Heading sẽ lớn hơn tức đã có thể rẽ nhanh hơn còn khi máy bay trong các nhóm trạng thái bám (TRACK) và STABLE thì hệ số góc đồ thị góc Yaw Heading sẽ xoắn hơn.

- Từ đồ thị quá trình chuyển trạng thái cũng như quá trình chuyển điểm trên *Hình 6.9a* và *b* ta thấy máy bay đã có thể thay đổi được điểm làm gốc hiện tại và đường bám trên quỹ đạo một cách tuần tự và nhịp nhàng từ đó bám theo quỹ đạo như kết quả *Hình 6.9c*. Thêm vào đó, các trạng thái đã có thể phối hợp với nhau một cách nhịp nhàng dựa trên độ lớn sai số góc Yaw và dấu của sai số này. Nhìn vào quá trình chuyển này ta thấy được một cách rõ ràng những ý đồ và nguyên tắc điều khiển đã được nêu ở *Chương 5* như:

- Kết thúc quá trình rẽ (TURN\_LEFT, TURN\_RIGHT) máy bay sẽ luôn được đưa về trạng thái STABLE để giữ ổn định lại cho máy bay trước khi chuyển sang một trạng thái khác.
- Sẽ không có trường hợp chuyển trực tiếp từ TRACK\_LEFT sang TRACK\_RIGHT hoặc TURN\_RIGHT và ngược lại từ TRACK\_RIGHT sang

TRACK\_LEFT hoặc TURN\_LEFT. Trong những trường hợp này máy bay trước hết phải trở về trạng thái STABLE trước và tiếp tục chờ quyết định từ khối State\_Decision.

#### **6.3.4.2. Một số hình ảnh quỹ đạo bay được**

*Hình 6.10a* là kết quả quỹ đạo bay được ngày 20/4/2018. Các thông số bay trong lần này về cơ bản giống như trong mục 6.3.1 tuy nhiên có một vài thông số khác như sau:

- Bán kính chuyển điểm: 30m.
- YAW\_ERROR\_TRACK\_RANGE:  $90^{\circ}$ .
- TURN\_ANGLE:  $45^{\circ}$ .

Từ đồ thị quỹ đạo ta thấy máy bay về cơ bản máy bay có thể bám theo được quỹ đạo, tuy nhiên các góc rẽ nhỏ máy bay bám chậm và không tìm thấy điểm số 3 trong quỹ đạo. Điều này có thể giải thích là do thông số YAW\_ERROR\_TRACK\_RANGE quá lớn ( $90^{\circ}$ ) mà sai số giữa góc Yaw Heading và góc lái từ thuật toán Stanley không đạt tới được giá trị này, điều này làm máy bay hầu như không đi vào được các trạng thái TURN\_LEFT, TURN\_RIGHT nên không tạo được những góc rẽ lớn. Hơn nữa, bán kính chuyển điểm là 30m vẫn chưa đủ lớn khiến máy bay dễ bị lạc mất không tìm thấy điểm tiếp theo. Những nhược điểm này sẽ được khắc phục trong các lần bay tiếp theo.

*Hình 6.10b* là kết quả quỹ đạo bay được ngày 15/5/2018. Các thông số bay trong lần này về cơ bản giống như trong mục 6.3.4.1 tuy nhiên có một vài thông số khác như sau:

- Bán kính chuyển điểm: 40m.
- YAW\_ERROR\_TRACK\_RANGE:  $60^{\circ}$ .
- TURN\_ANGLE:  $35^{\circ}$ .
- Điểm đặt góc Roll TRACK\_LEFT:  $-12^{\circ}$ .

- Điểm đặt góc Roll TRACK\_RIGHT: 13°.

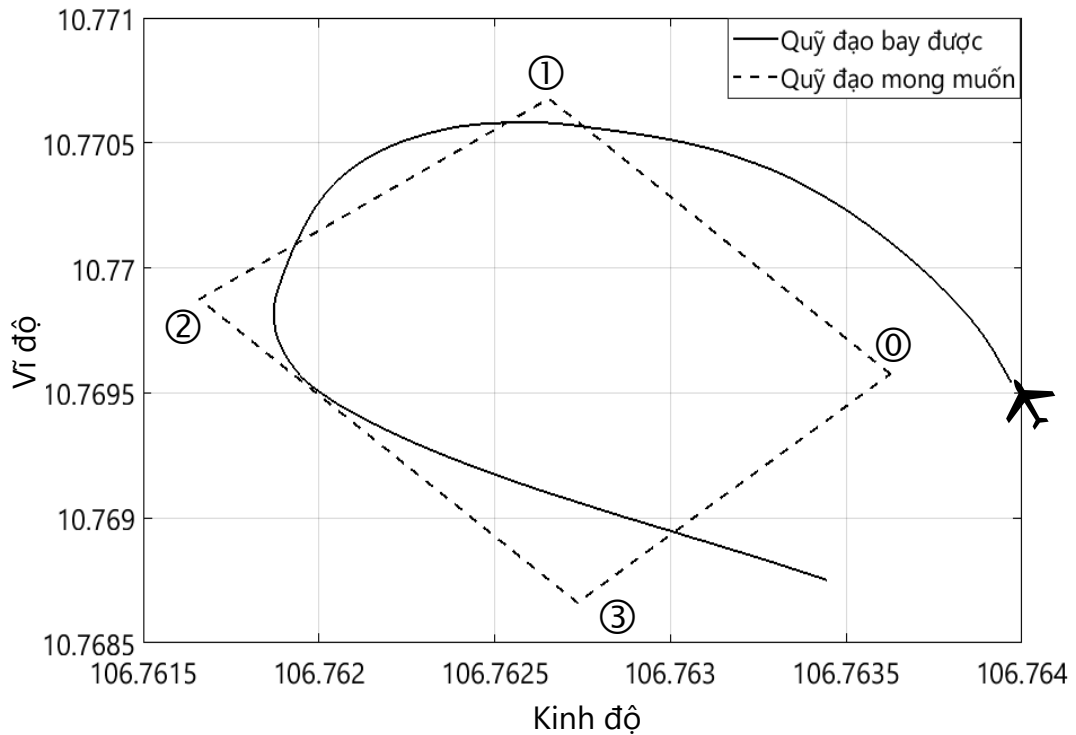
Để cải thiện kết quả của lần bay ngày 20/4/2018 chúng em đã tiến hành giảm thông số YAW\_ERROR\_TRACK\_RANGE xuống còn 60° nhằm giúp máy bay có thể đi vào các trạng thái TURN\_LEFT, TURN\_RIGHT. Bên cạnh đó, tăng bán kính chuyển điểm lên 40m nhằm giúp máy bay có thể tìm được điểm chuyển tốt hơn và cuối cùng là tăng các điểm đặt góc Roll trong các trạng thái TRACK\_LEFT và TRACK\_RIGHT nhằm cải thiện tốc độ bám. Kết quả trên *Hình 6.10b* cho thấy máy bay đã có thể bám nhanh hơn, tạo những góc rẽ lớn, đã có thể đi vào các trạng thái TURN\_RIGHT, TURN\_LEFT và không còn bị mất điểm số 3 trên quỹ đạo. Tuy nhiên, do góc rẽ quá lớn dẫn tới hiện tượng đường bay của máy bay bị dao động lớn xung quanh đường quỹ đạo mong muốn và dẫn tới máy bay bị đẩy quá xa khỏi quỹ đạo khi bám theo đường từ điểm 3 đến điểm 0 và tiếp tục không thể tìm thấy điểm 0. Những nhược điểm này sẽ tiếp tục được cải thiện trong lần bay tiếp theo.

*Hình 6.10c* là kết quả quỹ đạo bay được ngày 31/5/2018. Các thông số bay trong lần này về cơ bản giống như trong mục 6.3.4.1 tuy nhiên có một vài thông số khác như sau:

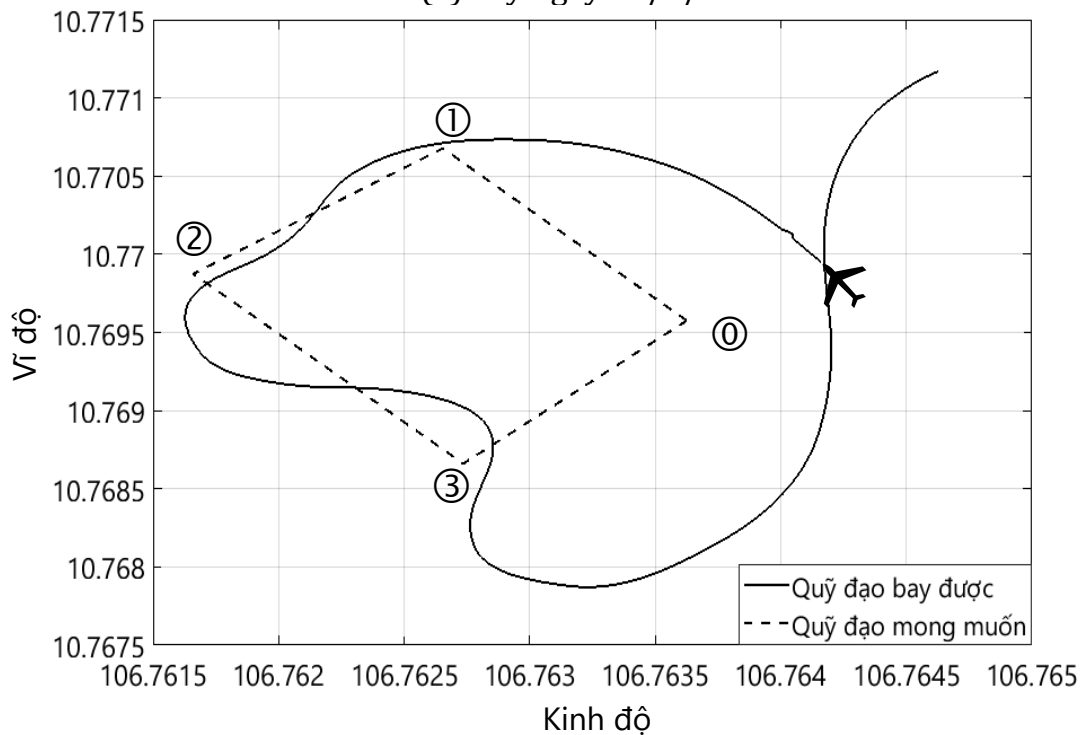
- Bán kính chuyển điểm: 40m.
- YAW\_ERROR\_TRACK\_RANGE: 60°.
- TURN\_ANGLE: 25°.
- Điểm đặt góc Roll TRACK\_LEFT: -10°.
- Điểm đặt góc Roll TRACK\_RIGHT: 11°.

Để cải thiện kết quả của lần bay ngày 15/5/2018 chúng em đã tiến hành giảm thông số TURN\_ANGLE xuống còn 25° và đồng thời giảm các điểm đặt góc Roll trong các trạng thái TRACK\_LEFT, TRACK\_RIGHT nhằm giảm bớt góc rẽ của máy bay. Từ kết quả có được trên *Hình 6.10c* ta thấy máy bay đã bám theo quỹ đạo

tốt hơn rất nhiều không còn hiện tượng không tìm thấy điểm chuyển cũng như dao động xung quanh quỹ đạo mong muốn. Máy bay đã có thể bay bám quỹ đạo hồi tiếp hai lần. Tuy nhiên, sau lần bay thứ hai vì không còn đủ nguồn nuôi nên buộc phải hạ cánh. Đó cũng là kết quả tốt nhất hiện tại mà chúng em có được.

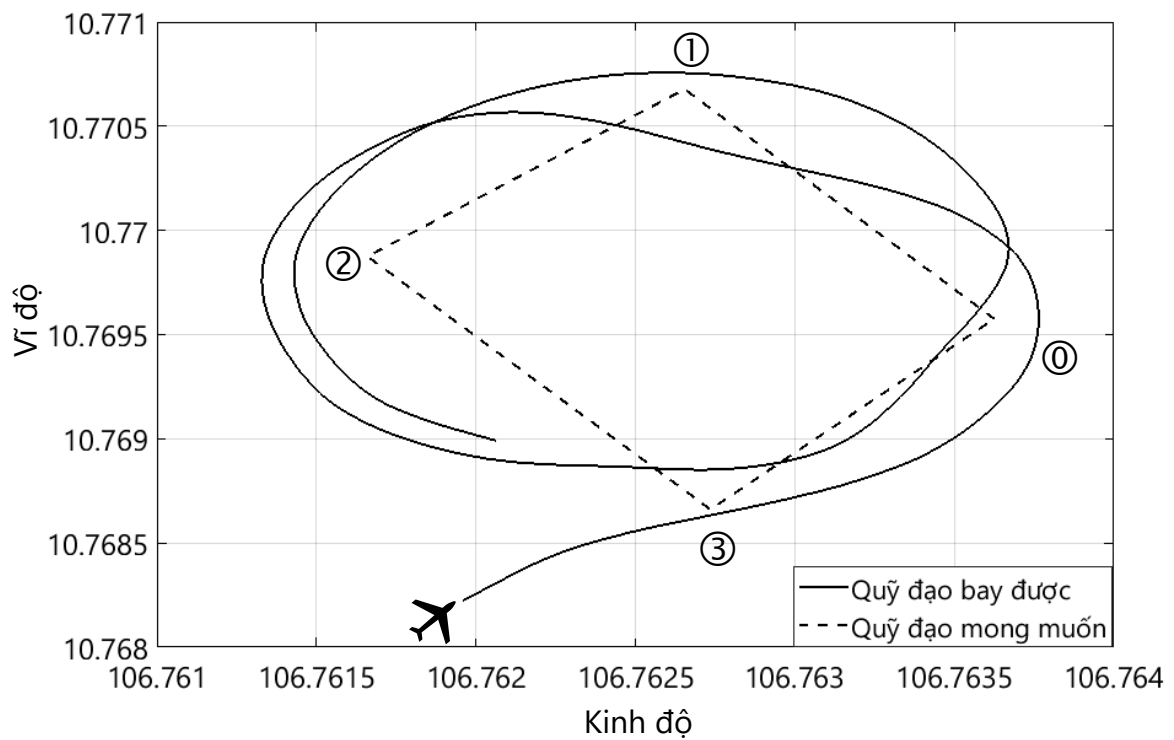


(a) Bay ngày 20/4/2018



(b) Bay ngày 15/5/2018





(c) Bay ngày 31/5/2018

Hình 6.10: Kết quả bay quỹ đạo

#### 6.4. Hướng phát triển

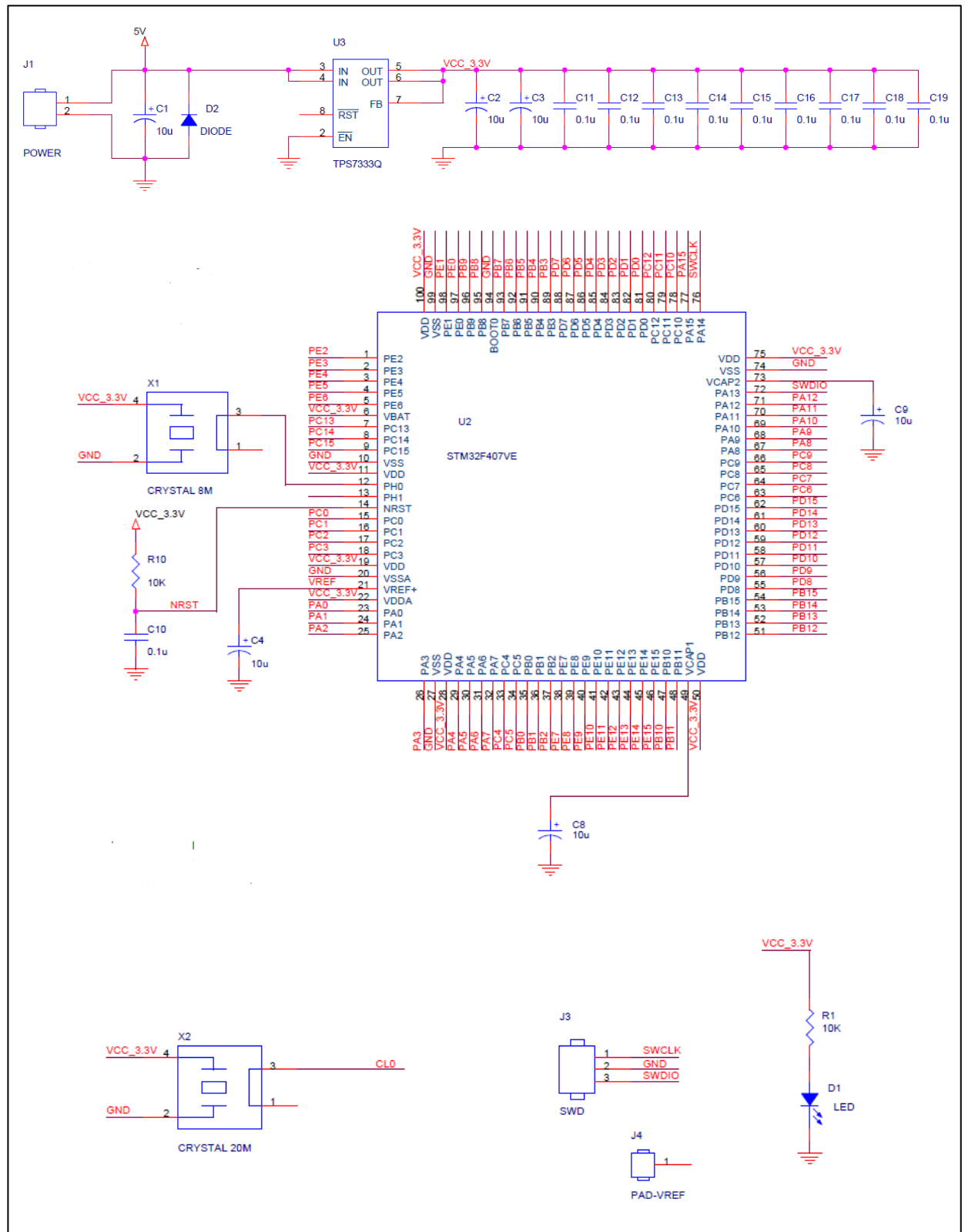
- Kết hợp thêm một vài cảm biến khác như cảm biến hướng gió, áp suất gió để tăng khả năng ổn định của máy bay trong các điều kiện môi trường khác nhau.
- Điều khiển độ cao máy bay. Đề tài hiện tại chỉ chủ yếu điều khiển các góc Roll, Pitch, Yaw và ổn định độ cao máy bay chứ chưa thể điều khiển độ cao máy bay theo một chiều cao mong muốn.
- Áp dụng một số giải thuật khác để máy bay có thể bám sát được quỹ đạo hơn. Chúng em đề xuất giải thuật trong đường link sau:

<https://github.com/ArduPilot/ardupilot/pull/101>

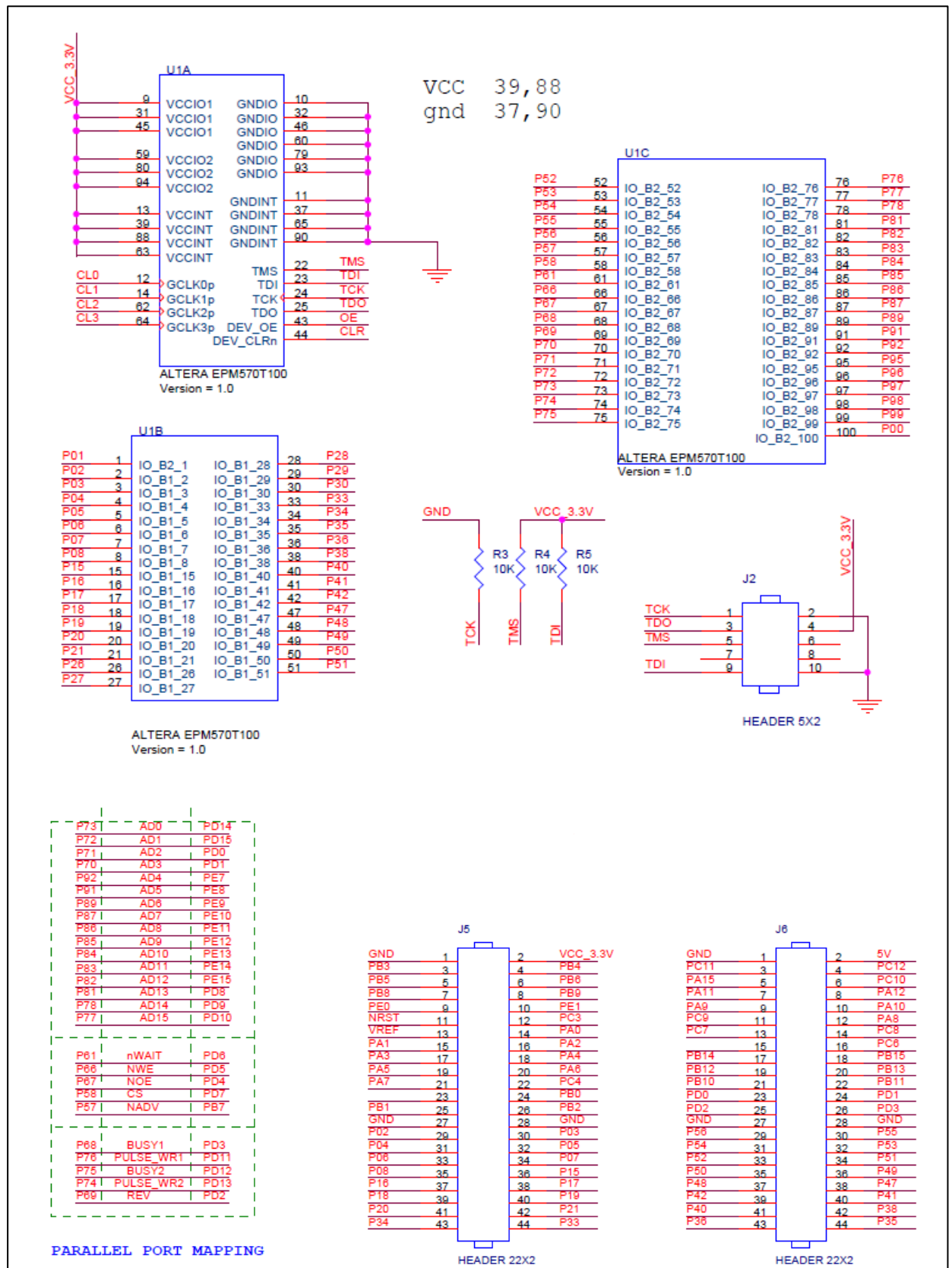
- Phát triển tiếp tục giải thuật để máy bay có thể tự động cất cánh và hạ cánh.

- Kết hợp với các trạm mặt đất đã có ở các năm trước để tạo thành một hệ thống hoàn chỉnh, có khả năng thực hiện một vài nhiệm vụ nhất định.

## PHỤ LỤC

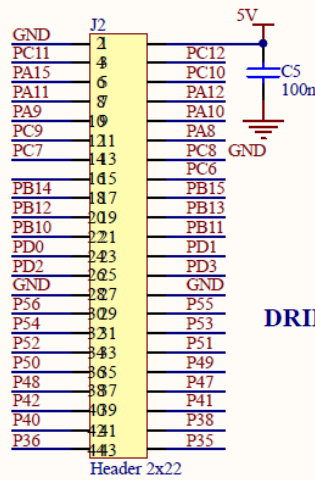
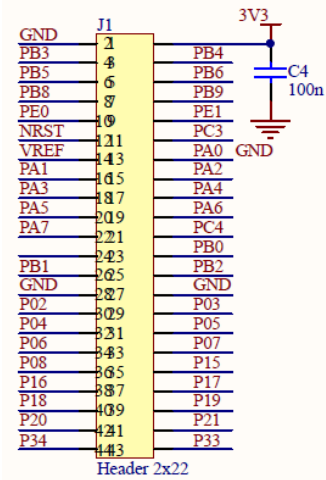


Hình PL.1: Sơ đồ thiết kế mạch điều khiển – A

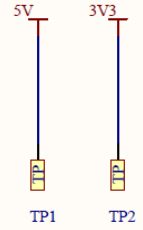


Hình PL.2: Sơ đồ thiết kế mạch điều khiển – B

## STM32\_MAXII



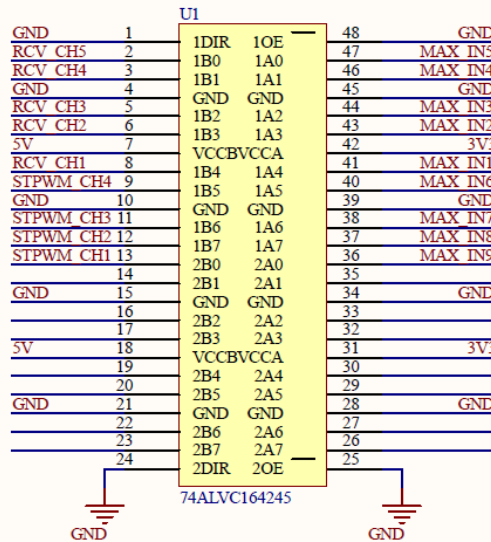
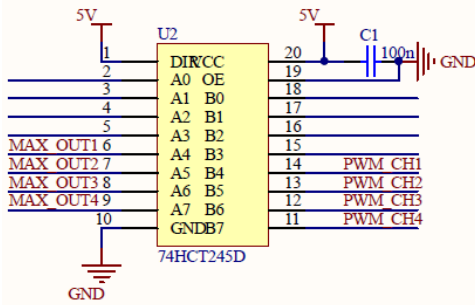
## TESTPOINT



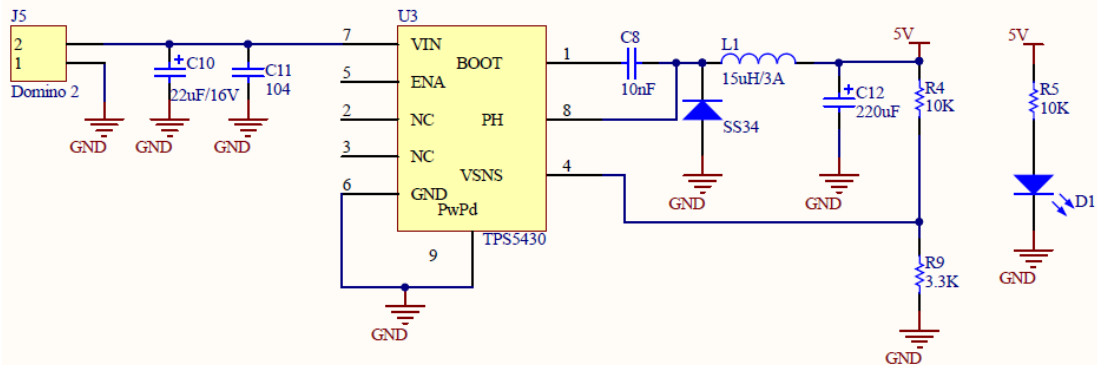
## DRILL

PC6	P35
PB15	P38
PB13	P41

## TRANSCEIVER



## POWER SUPPLY

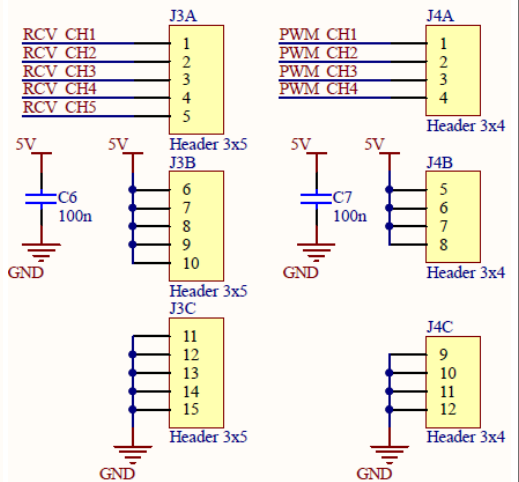


Hình PL.3: Sơ đồ thiết kế mạch ra chân – A

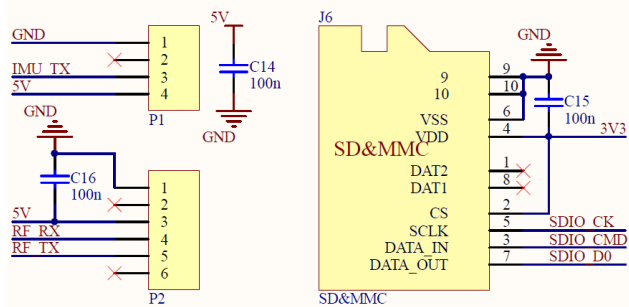
## PORT MAPPING

IMU TX	UART2 RX	PA3	MAX IN1	P15
RF RX	UART4 TX	PA0	MAX IN2	P16
RF TX	UART4 RX	PA1	MAX IN3	P17
STPWM CH1	TIM3 CH1	PA6	MAX IN4	P18
STPWM CH2	TIM3 CH2	PA7	MAX IN5	P19
STPWM CH3	TIM3 CH3	PB0	MAX IN6	P08
STPWM CH4	TIM3 CH4	PB1	MAX IN7	P07
SDIO CK		PD2	MAX IN8	P06
SDIO_CMD		PC12	MAX IN9	P05
SDIO D0		PC8	MAX OUT1	P20
MODE SL1		PC4	MAX OUT2	P21
MODE SL2		PB2	MAX OUT3	P34
LED1		PA2	MAX OUT4	P33
LED2		PA4	LED3	PA5
BT1		PB8	BT2	PE0

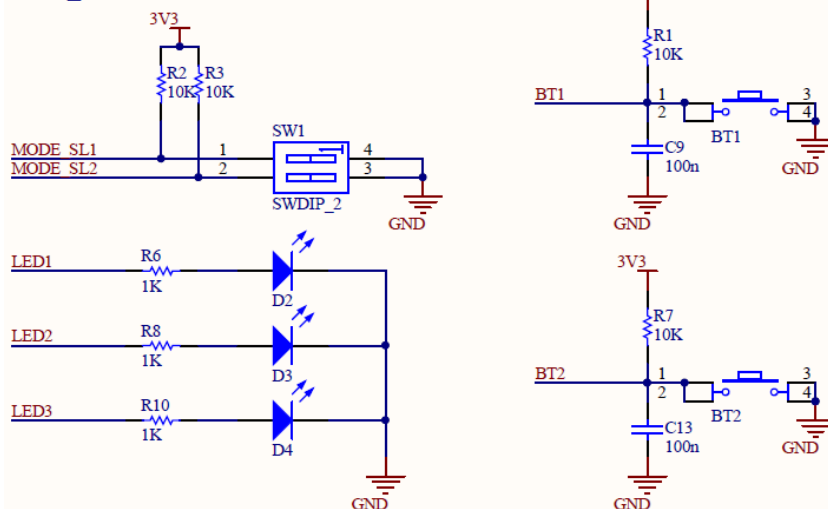
## RECEIVER\_PWMOUT



## SDCARD\_IMU\_RF



## LED\_BUTTON



Hình PL.4: Sơ đồ thiết kế mạch ra chân – B



*Hình PL.5: Máy bay lúc cất cánh*



*Hình PL.6: Máy bay đang bám quỹ đạo trên không*

## TÀI LIỆU THAM KHẢO

- [1] Nguyễn Thanh Xuân và Hồ Nhật Tân, “Thiết kế trạm mặt đất phục vụ điều khiển tự động máy bay cánh bằng,” Thành phố Hồ Chí Minh, 2014.
- [2] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. v. Niekerc, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian và P. Mahoney, “Stanley: The Robot that Won the DARPA Grand Challenge,” *Journal of Field Robotics*, số 23, pp. 662-663, 683-684, 2006.
- [3] Nguyễn Thị Phương Hà và Huỳnh Thái Hoàng, Điều khiển tự động, Thành phố Hồ Chí Minh: Nhà xuất bản Đại học Quốc gia TP. HCM, 2011.
- [4] Quân Phong, "Quân Phong RC," [Online]. Available: <http://quanphongrc.vn>.
- [5] "Intel FPGA," Intel Coroperation, [Online]. Available: <https://www.altera.com/products/cpld/max-series/max-ii/overview.html>.
- [6] "ST - Life augmented," STMicroelectronics, [Online]. Available: <http://www.st.com/en/evaluation-tools/stm32f4discovery.html>.
- [7] "Wikipedia - The Free Encyclopedia," Wikimedia Foundation, Inc., [Online]. Available: <https://en.wikipedia.org>.
- [8] STMicroelectronics, "RM0090 Reference Manual," 2011.
- [9] ChaN, "The Electronic Lives Manufacturing," [Online]. Available: [http://elm-chan.org/fsw/ff/00index\\_e.html](http://elm-chan.org/fsw/ff/00index_e.html).



- [10] Lê Văn Chức, "Xây dựng trạm mặt đất và điều khiển máy bay cánh bằng," TP. Hồ Chí Minh, 2016.
- [11] Texas Instruments, "TPS543x 3-A, Wide Input Range, Step-Down Converter," 2006.
- [12] Ardupilot Dev Team, "ArduPilot," 2016. [Online]. Available: <http://ardupilot.org/plane/docs/Roll-Pitch-controller-tuning.html>.
- [13] M. Brown, "Hooked on RC Airplane," [Trực tuyến]. Available: <http://www.hooked-on-rc-airplanes.com/parts-of-an-airplane.html>.
- [14] ASA, "Learn to fly," Aviation Supplies & Academy, Inc, 27 June 2016. [Trực tuyến]. Available: <http://learntoflyblog.com/2016/06/27/aircraft-performance-Pitch-stability-stability-part-2/>.
- [15] P. Carpenter, "RC Airplane World," [Online]. Available: <https://www.rc-airplane-world.com/how-airplanes-fly.html>.