

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN

ĐỒ ÁN TỐT NGHIỆP
CAPSTONE PROJECT

NGÀNH: KỸ THUẬT ĐIỀU KHIỂN VÀ TỰ ĐỘNG HÓA

ĐỀ TÀI:
ĐIỀU KHIỂN PHỐI HỢP HAI DRONE

Người hướng dẫn: TS. NGUYỄN HOÀNG MAI

Cán bộ hướng dẫn: NGUYỄN VĂN VINH

Sinh viên thực hiện:

NGUYỄN HỮU ANH BẰNG

105200291

VÕ MINH HOÀNG

105200301

Lớp: 20TDH1

Đà Nẵng, 6/2025

TÓM TẮT

Tên đề tài: “Thiết kế điều khiển hệ Drone kép nâng tải”.

Sinh viên thực hiện và mã số sinh viên:

Nguyễn Hữu Anh Bằng 105200291

Võ Minh Hoàng 105200301

Lớp: 20TDH1

Trong bối cảnh công nghệ không người lái phát triển mạnh mẽ, việc ứng dụng drone (thiết bị bay không người lái) trong vận chuyển hàng hóa ngày càng phổ biến. Tuy nhiên, hạn chế lớn nhất của các drone đơn lẻ là khả năng nâng tải còn thấp. Đề tài "Thiết kế điều khiển hệ Drone kép nâng tải" nhằm nghiên cứu và phát triển hệ thống điều khiển phối hợp giữa hai drone hoạt động đồng bộ để nâng và vận chuyển tải trọng lớn hơn khả năng của từng drone riêng lẻ.

Nội dung nghiên cứu bao gồm: xây dựng mô hình động lực học của hệ drone kép có liên kết tải, thiết kế bộ điều khiển phân tán hoặc tập trung để đảm bảo sự ổn định và phối hợp giữa hai drone, và thực hiện mô phỏng để đánh giá hiệu quả hệ thống. Ngoài ra, đề tài cũng xem xét đến các yếu tố như nhiễu động môi trường, độ trễ truyền thông và an toàn bay khi vận hành thực tế.

Kết quả nghiên cứu góp phần mở rộng ứng dụng của drone trong các lĩnh vực vận chuyển hàng hóa, cứu hộ cứu nạn, và logistics tại những khu vực khó tiếp cận.

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

TT	Họ tên sinh viên	Số thẻ SV	Lớp	Chuyên ngành
1	Nguyễn Hữu Anh Bằng	105200291	20TDH1	Kỹ thuật điều khiển & Tự động hóa
2	Võ Minh Hoàng	105200301	20TDH1	Kỹ thuật điều khiển & Tự động hóa

1. Tên đề tài đồ án:

Thiết kế điều khiển hệ Drone kép nâng tải

2. Đề tài thuộc diện: ☐ Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện

3. Các số liệu và dữ liệu ban đầu:

Tham khảo nguồn tài liệu trên Internet về Drone, bộ điều khiển PID.

4. Nội dung các phần thuyết minh và tính toán:

a. Phần chung:

TT	Họ tên sinh viên	Nội dung
1	Nguyễn Hữu Anh Bằng	- Tìm hiểu và nghiên cứu các thông tin tài liệu, các bài báo khoa học và tạp chí liên quan về các kỹ thuật, phương pháp của Drone. - Nghiên cứu nguyên lý Drone đơn và hệ Drone kép.
2	Võ Minh Hoàng	- Tiến hành mô phỏng và viết chương trình. - Hiệu chỉnh phần cứng. - Viết báo cáo thuyết minh.

b. Phần riêng:

TT	Họ tên sinh viên	Nội dung
1	Nguyễn Hữu Anh Bằng	- Nghiên cứu tài liệu và xây dựng phương pháp điều khiển. - Nghiên cứu nguyên lý hoạt động của Drone đơn. - Tìm hiểu, nghiên cứu nguyên lý và trạng thái làm việc của hệ Drone kép. - Xây dựng phương trình động lực học. - Xây dựng phương pháp kết nối hệ Drone kép. - Lắp ráp phần cứng, tính toán động học, mô phỏng hoạt động trên Matlab.

		- Đánh giá kết quả mô hình.
2	Võ Minh Hoàng	- Xây dựng thuật toán, lập trình cho module điều khiển thiết bị cơ cấu chấp hành. - Xây dựng thuật toán, lập trình cho module điều khiển vị trí, đo khoảng cách. - Hiệu chỉnh, kiểm tra phần lập trình. - Truyền thông giữa module điều khiển thiết bị cơ cấu chấp hành và module điều khiển vị trí, đo khoảng cách.

5. Các bản vẽ, đồ thị (ghi rõ các loại và kích thước bản vẽ):

a. Phần chung:

TT	Họ tên sinh viên	Nội dung
1	Nguyễn Hữu Anh Bằng	- Nghiên cứu, thảo luận, góp ý về hệ thống Drone kép nâng tải. - Lên ý tưởng phân cứng về hệ Drone kép nâng tải, và mục tiêu của đề tài hướng tới.
2	Võ Minh Hoàng	- Đề xuất phương pháp phân kết nối, truyền thông giữa 2 Drone.

b. Phần riêng:

TT	Họ tên sinh viên	Nội dung
1	Nguyễn Hữu Anh Bằng	- Bản vẽ nguyên lý và trạng thái làm việc của Drone đơn và hệ Drone kép. - Bản vẽ Visio cho mô hình phân cứng. - Bản vẽ mô hình mô phỏng. - Bản vẽ phương pháp kết nối và truyền thông.
2	Võ Minh Hoàng	- Bản vẽ mô hình điều khiển. - Bản vẽ sơ đồ khối điều khiển PID cho các cảm biến. - Bản vẽ lưu đồ thuật toán của 2 Drone.

6. Họ tên người hướng dẫn: **TS. Nguyễn Hoàng Mai**

7. Ngày giao nhiệm vụ đồ án: 17/2/2025

8. Ngày hoàn thành đồ án: 17/6/2025

PHIẾU KIỂM SOÁT TIẾN ĐỘ LÀM ĐỒ ÁN TỐT NGHIỆP
(Phiếu dành cho người hướng dẫn/sinh viên)

TT	Họ tên sinh viên	Số thẻ SV	Lớp	Chuyên ngành
1	Nguyễn Hữu Anh Bằng	105200291	20TDH1	Kỹ thuật điều khiển & Tự động hóa
2	Võ Minh Hoàng	105200301	20TDH1	Kỹ thuật điều khiển & Tự động hóa

Tên đề tài ĐATN: **Thiết kế điều khiển hệ Drone kép nâng tải**

Họ tên người HD: TS. Nguyễn Hoàng Mai Đơn vị: Đại Học Bách Khoa Đà Nẵng

Tuần	Ngày	Khối lượng		GVHD ký tên
		Đã thực hiện (%)	Tiếp tục thực hiện (%)	
1	17/2 – 24/2/2025	- Giao đề tài - Tìm hiểu và nghiên cứu các tài liệu liên quan đến hệ thống điều khiển Drone. - Nghiên cứu các phương pháp điều khiển Drone đơn, hệ Drone kép nâng tải.	- Xây dựng cấu trúc đề tài. - Đề xuất mô hình thiết hệ Drone kép.	
2	25/2 – 4/3/2025	- Nghiên cứu nguyên lý hoạt động của Drone đơn. - Lên ý tưởng phân cứng và thiết kế mô hình. - Nghiên cứu các phương pháp điều khiển hệ Drone chưa có tải.	- Tiếp tục nghiên cứu nguyên lý hoạt động của hệ Drone kép. - Nghiên cứu giải pháp kết nối giữa 2 Drone.	
3	5/3 – 12/3/2025	- Nghiên cứu nguyên lý hoạt động của hệ Drone kép. - Lên ý tưởng, thảo luận về phương pháp kết nối 2 Drone. - Khảo sát tính khả thi của phương pháp.	- Tiến hành tạo mô hình mô phỏng trên Matlab/Simulink.	
4	Duyệt lần 1: Đánh giá khối lượng hoàn thành ____ %: Được tiếp tục làm ĐATN ☐ Không tiếp tục thực hiện ĐATN ☐			

	13/3 – 20/3/2025	- Tiến hành tạo mô hình mô phỏng trên Matlab/Simulink. - Xây dựng sơ đồ khối cho hệ thống điều khiển. - Tích hợp mô hình điều khiển vào Matlab/Simulink.	- Điều chỉnh mô hình mô phỏng. - Xây dựng phương trình động lực học.	
5	21/3 – 28/3/2025	- Xây dựng phương trình động lực học. - Hiệu chỉnh phương trình động lực trên mô hình mô phỏng.	- Nghiên cứu và xây dựng nguyên lý điều khiển. - Lên ý tưởng truyền thông giữa 2 Drone.	
6	29/3 – 5/4/2025	- Xây dựng nguyên lý điều khiển. - Thảo luận, chốt phương pháp truyền thông giữa 2 Drone. - Tính toán lựa chọn động cơ, thiết bị linh kiện.	- Tiến hành lắp ráp phần cứng. - Nghiên cứu tài liệu về lập trình Arduino.	
7	6/4 – 13/4/2025	- Lắp ráp phần cứng, nghiên cứu thêm các module điều khiển. - Thử nghiệm các module.	- Thử nghiệm và tiến hành thực nghiệm để hiệu chỉnh. - Xây dựng bộ điều khiển PID.	
8	Duyệt lần 2: Đánh giá khối lượng hoàn thành ____ %: Được tiếp tục làm ĐATN ☐ Không tiếp tục thực hiện ĐATN ☐			
	14/4 – 21/4/2025	- Thực nghiệm các module và hiệu chỉnh phù hợp. - Xây dựng bộ điều khiển PID và mô phỏng.	- Lập trình cho các module điều khiển. - Tiến hành thực nghiệm và hiệu chỉnh PID cho hệ thống.	
9	22/4 – 29/4/2025	- Hiệu chỉnh phần lập trình các module. - Hiệu chỉnh PID bám sát giá trị mong muốn. - Chỉnh sửa phần cứng, thực nghiệm bay từng Drone không tải.	- Khắc phục, hiệu chỉnh mô hình phần cứng.	

10	30/4 – 7/5/2025	- Hiệu chỉnh mô hình phần cứng. - Thực nghiệm bay Drone kết hợp hiệu chỉnh PID. - Chỉnh sửa phần lập trình các module. - Thực nghiệm bay Drone đơn có tải và lựa chọn mức tải phù hợp.	- Tiếp tục bay thực nghiệm Drone và hiệu chỉnh phần lập trình. - Xây dựng lưu đồ thuật toán.	
11	8/5 – 15/5/2025	- Chỉnh sửa mô hình, hiệu chỉnh phần lập trình. - Xây dựng lưu đồ thuật toán cho từng Drone.	- Tổng kết mô hình phần cứng. - Hiệu chỉnh phần lập trình của hệ thống. - Chỉnh sửa báo cáo thuyết minh.	
12	Duyệt lần 3: Đánh giá khối lượng hoàn thành _____ %: Được tiếp tục làm ĐATN ☐ Không tiếp tục thực hiện ĐATN ☐			
	16/5 – 23/5/2025	- Khắc phục sự cố cho mô hình phần cứng. - Hiệu chỉnh lại phần lập trình và hiệu chỉnh thông số PID. - Bay thực nghiệm đồng thời 2 Drone không tải.	- Tiến hành bay thực nghiệm Drone. - Kết hợp lập trình, hiệu chỉnh bộ PID bám sát đầu vào.	
13	24/5 – 31/5/2025	- Hiệu chỉnh phần lập trình. - Bay thực nghiệm đồng thời hệ Drone có tải. - Chỉnh sửa phần cứng và hoàn thành phần lập trình.	- Tiến hành bay thực nghiệm Drone. - Kết hợp lập trình, hiệu chỉnh bộ PID bám sát đầu vào.	
14	1/6 – 8/6/2025	- Bay thực nghiệm hệ Drone, đánh giá quá trình, kết quả.	- Hiệu chỉnh và đánh giá sự ổn định.	
15	9/6 – 16/6/2025	- Hiệu chỉnh đảm bảo sự ổn định, chỉnh sửa báo cáo cuối cùng.		

LỜI CẢM ƠN

Chúng em xin chân thành cảm ơn đến thầy giáo TS. Nguyễn Hoàng Mai giảng viên bộ môn Kỹ Thuật Điều Khiển và Tự Động Hóa và anh Nguyễn Văn Vinh thuộc Công ty Truyền tải Điện 2 (PTC2) đã hỗ trợ, tạo điều kiện và cung cấp những thông tin bổ ích để chúng em có thể hoàn thành bài báo cáo Đồ án Tốt nghiệp này.

Em xin chân thành cảm ơn Trường đại học Bách Khoa – Đại học Đà Nẵng và đặc biệt là khoa Điện đã tạo điều kiện cho sinh viên như em nghiên cứu thực hiện Đồ án Tốt nghiệp bằng hình thức liên kết doanh nghiệp (Capstone Project) giúp em tích lũy thêm kiến thức và kinh nghiệm thực tế quý báu.

Trong thời gian làm Đồ án Tốt nghiệp, em đã nhận được nhiều sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của thầy cô và bạn bè. Qua đây em xin bày tỏ lòng biết ơn đến các thầy cô giáo giảng viên khoa Điện, ngành Kỹ thuật Điều khiển và Tự động hóa trường Đại học Bách khoa, Đại học Đà Nẵng là những người truyền thụ tri thức, kỹ năng, kinh nghiệm cho em trong suốt những năm học vừa qua. Đó là nền tảng cho việc thực hiện Đồ án Tốt nghiệp này.

Trong bài báo cáo là thành quả của sự nghiên cứu và tìm hiểu mà em đạt được trong thời gian thực hiện Đồ án Tốt nghiệp. Do thời gian thực hiện Đồ án Tốt nghiệp có hạn nên không tránh khỏi sai sót, rất mong nhận được sự góp ý của quý Công ty và Thầy cô giáo bộ môn để tạo điều kiện cho em hoàn thành Đồ án Tốt nghiệp một cách thành công nhất.

Xin chân thành cảm ơn!!!

LỜI CAM ĐOAN LIÊM CHÍNH HỌC THUẬT

Chúng em xin cam kết rằng nội dung đề tài: “**Thiết kế điều khiển hệ Drone kép nâng tải**” được trình bày trong báo cáo Đồ án Tốt nghiệp là kết quả nghiên cứu thực hiện bởi chính chúng em, dưới sự hướng dẫn của thầy TS. Nguyễn Hoàng Mai.

Toàn bộ số liệu và kết quả nghiên cứu trong đồ án là trung thực, không sao chép hay sử dụng kết quả từ bất kỳ đề tài nghiên cứu nào khác.

Nội dung đồ án có tham khảo và sử dụng một số tài liệu, thông tin đã được công bố trên các tạp chí, website, và đã được trích dẫn đầy đủ trong phần danh mục tài liệu tham khảo ở cuối đồ án.

Sinh viên thực hiện

Sinh viên thực hiện

Nguyễn Hữu Anh Bằng

Võ Minh Hoàng

MỤC LỤC

TÓM TẮT.....	2
NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP	3
PHIẾU KIỂM SOÁT TIẾN ĐỘ LÀM ĐỒ ÁN TỐT NGHIỆP	5
LỜI CẢM ƠN.....	8
LỜI CAM ĐOAN LIÊM CHÍNH HỌC THUẬT	9
MỤC LỤC	10
DANH SÁCH CÁC BẢNG, HÌNH VẼ	14
DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT	Lỗi! Thẻ đánh dấu không được xác định.
MỞ ĐẦU	1
CHƯƠNG 1. TỔNG QUAN VỀ ĐIỀU KHIỂN HỆ DRONE KÉP NÂNG TẢI	6
1.1. Lý do chọn đề tài.....	6
1.1.1. Tính cần thiết từ đề tài	6
1.1.1.1. Cơ sở thực tiễn.....	6
1.1.1.2. Cơ sở khoa học	7
1.2. Ý nghĩa về khoa học và thực tiễn	7
1.2.1. Về mặt khoa học	7
1.2.2. Về mặt thực tiễn	7
1.3. Hiện trạng thực tế.....	8
1.3.1. Lịch sử phát triển Drone	8
1.3.2. Các công trình nghiên cứu và cơ sở khoa học ở Việt Nam.....	9
1.3.3. Thực tế	10
1.3.4. Câu hỏi đặt ra	11
1.4. Dự kiến kết quả đạt được.....	11
1.4.1. Quá trình thử nghiệm	11

1.4.1.1. Phương án kết nối hệ Drone kép	11
1.4.1.2. Lí do chọn nguyên vật liệu kết nối	12
1.4.2. Kết quả dự kiến:	13
1.5. Kết luận chương.....	13
CHƯƠNG 2. MÔ HÌNH TOÁN HỌC CỦA DRONE VÀ HỆ DRONE KÉP	14
2.1. Cấu trúc	14
2.1.1. Sơ đồ khối	14
2.2. Nguyên lý hoạt động của drone và hệ drone kép.....	14
2.2.1. Nguyên lý hoạt động của Drone	14
2.2.1.1. Nguyên lý cân bằng của Drone	15
2.2.1.2. Mô tả các trạng thái làm việc của Drone đơn.....	16
2.2.1.2.1. Trạng thái lơ lửng.....	16
2.2.1.2.2. Trạng thái bay lên/xuống	17
2.2.1.2.3. Trạng thái bay nghiêng sang trái/phải " Φ "(Roll).....	17
2.2.1.2.4. Trạng thái tịnh tiến tới/lui " θ "(Pitch).....	18
2.2.1.2.5. Trạng thái xoay qua trái/phải " Ψ "(Yaw)	19
2.2.2. Nguyên lý hoạt động của hệ Drone kép.....	20
2.2.2.1. Nguyên lý cân bằng của hệ Drone kép.....	21
2.2.2.2. Mô tả các trạng thái làm việc của hệ Drone kép	21
2.2.2.2.1. Trạng thái bay lên/xuống	22
2.2.2.2.2. Trạng thái tịnh tiến tới/lui	22
2.2.2.2.3. Trạng thái bay nghiêng sang trái/phải.....	22
2.2.2.2.4. Trạng thái thay đổi quỹ đạo của toàn bộ hệ thống.....	23
2.3. Phương trình động lực	24
2.3.1. Tính chất động lực học của Drone	24
2.3.1.1. Các hệ tọa độ	24
2.3.1.2. Các thông số được sử dụng	25

2.3.1.3. Động lực học của cánh quạt	26
2.3.1.4. Phân tích động lực học của Drone.....	27
2.3.1.4.1. Ma trận chuyển đổi	27
2.3.1.4.2. Chuyển đổi vận tốc góc	28
2.3.1.4.3. Vận tốc dài	29
2.3.1.4.4. Vận tốc góc	30
2.3.1.4.5. Tính chất phi tuyến của động học Drone.....	32
2.3.1.5. Động lực học của hệ Drone kép	34
2.3.1.5.1. Vận tốc dài	34
2.3.1.5.2. Cấu hình hệ Drone kép	35
2.3.1.5.3. Phương trình động lực học	36
2.3.1.5.4. Liên hệ giữa hai Drone	37
2.4. Đề xuất cho hệ thống	38
2.5. Kết luận chương.....	38
CHƯƠNG 3. THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN HỆ DRONE KÉP	40
3.1. Tính toán, lựa chọn động cơ, cánh quạt trong hệ thống và thiết kế linh kiện cho phần động lực.....	40
3.1.1. Tính toán, lựa chọn động cơ, cánh quạt trong hệ thống.....	40
3.2. Nguyên lý điều khiển	42
3.2.1. Phương pháp kết nối hệ Drone kép.....	42
3.2.1.1. Thiết kế hệ truyền tín hiệu (TX).....	42
3.2.1.2. Thiết kế hệ nhận tín hiệu (RX).....	42
3.2.2. Phương pháp điều khiển hệ Drone kép	45
3.2.2.1. Mô hình truyền thông Master – Slave	45
3.2.2.2. Giao tiếp giữa các Arduino.....	46
3.3. Linh kiện phần động lực	46
3.4. Linh kiện phần điều khiển	48
3.4.1. Chi phí đầu tư.....	48

3.5. MÔ HÌNH ĐIỀU KHIỂN CHO HỆ DRONE KÉP.....	49
3.5.1. Bộ điều khiển PID cơ bản	49
3.5.2. Thiết kế bộ điều khiển PID cho hệ Drone kép.....	50
3.5.2.1. Sơ đồ tổng quan của hệ thống điều khiển	50
3.5.2.2. Sơ đồ chi tiết của hệ thống điều khiển	51
3.5.2.3. Các bộ PID của các cảm biến.....	52
3.5.3. Lưu đồ thuật toán	53
3.6. Kết luận chương.....	54
CHƯƠNG 4. MÔ PHỎNG, THỰC NGHIỆM VÀ ĐÁNH GIÁ	55
4.1. Kết quả mô hình	55
4.1.1. Mô hình mô phỏng trên Matlab/Simulink (môi trường không có nhiễu)	55
4.1.2. Kết quả mô phỏng trên Matlab/Simulink (môi trường không có nhiễu)	55
4.1.3. Kết quả của bộ sai lệch	58
4.2. Quá trình thực nghiệm.....	59
4.2.1. Sơ đồ khối PID hệ thống.....	59
4.2.2. Thông số PID hiệu chỉnh qua quá trình thực nghiệm	59
4.2.3. Thực nghiệm bay mô hình	60
KẾT LUẬN	62
TÀI LIỆU THAM KHẢO	63
PHỤ LỤC 1	64
PHỤ LỤC 2	67

DANH SÁCH CÁC BẢNG, HÌNH VẼ

Bảng 1.1: Một số dạng máy bay không người lái trong thực tế.....	10
Bảng 3.1: So sánh tín hiệu PWM và PPM	44
Bảng 3.2: Bảng chi phí linh kiện.....	49
Hình 1.1: Máy bay đầu tiên năm 1900 của anh em nhà Wright.....	8
Hình 1.2: Mô hình đề tài nghiên cứu của nhóm K63AE Học viện Hàng Không.....	9
Hình 1.3: Mô hình đã hoàn thiện của bài báo nghiên cứu [3]	10
Hình 1.4: Drone vận chuyển hàng hóa trong thực tế.....	11
Hình 1.5: Mô hình dự kiến của hệ Drone kép nâng tải	13
Hình 2.1: Sơ đồ khối điều khiển Drone kép bằng vật liệu cứng	14
Hình 2.2: Chiều quay của các động cơ trên 1 Drone.....	15
Hình 2.3: Chiều quay và các góc quay để Drone cân bằng.....	16
Hình 2.4: Chiều quay của Drone ở trạng thái lơ lửng	16
Hình 2.5: Trạng thái Drone di chuyển lên hoặc xuống	17
Hình 2.6: Trạng thái Drone di chuyển nghiêng sang trái hoặc sang phải	18
Hình 2.7: Trạng thái Drone di chuyển tịnh tiến tới hoặc lui.....	19
Hình 2.8: Trạng thái Drone di chuyển xoay sang trái hoặc sang phải.....	19
Hình 2.9: Mô phỏng một mô hình hệ Drone kép	21
Hình 2.10: Chiều di chuyển của 2 Drone khi xoay một góc bất kỳ	24
Hình 2.11: Mô phỏng các đại lượng trên một mô hình Drone	25
Hình 2.12a	27
Hình 2.12b	27
Hình 2.12c	27
Hình 2.13: Tay cầm điều khiển thiết kế tích hợp Arduino	38
Hình 2.14a: Mô hình thực tế Drone 1.....	38
Hình 2.14b: Mô hình thực tế Drone 2	38

Hình 3.1: Phương pháp kết nối – truyền tín hiệu cho hệ Drone kép	42
Hình 3.2: Thiết bị thu – phát sóng vô tuyến RF	43
Hình 3.3: Mô hình mô phỏng tổng quát của việc điều khiển hệ thống Drone	49
Hình 3.4: Sơ đồ tổng quát của một bộ điều khiển PID cơ bản	50
Hình 3.5: Sơ đồ tổng quan của hệ thống	50
Hình 3.6: Sơ đồ chi tiết của hệ thống	51
Hình 3.7: Sơ đồ khối điều khiển PID cho cảm biến HC-SR04	52
Hình 3.8: Sơ đồ khối điều khiển PID cho cảm biến Optical Flow	53
Hình 3.9: Lưu đồ thuật toán của bộ điều khiển PID giữ độ cao cho Drone 1	53
Hình 3.10: Lưu đồ thuật toán của bộ điều khiển PID giữ độ cao cho Drone 2	54
Hình 4.1: Mô hình mô phỏng Matlab/Simulink của hệ thống	55
Hình 4.2: Mô hình mô phỏng từng Drone	55
Hình 4.3: Lực nâng Z đầu vào và lực nâng Z thực tế của 2 Drone	56
Hình 4.4: Giá trị góc Phi Φ	56
Hình 4.5: Giá trị góc Theta θ	57
Hình 4.6: Giá trị góc Psi ψ	57
Hình 4.7: Tọa độ của Drone 1	58
Hình 4.8: Tọa độ của Drone 2	58
Hình 4.9: Kết quả mô phỏng bù trừ lẫn nhau của 2 thiết bị Drone khi $T=30$	58
Hình 4.10: Sơ đồ khối PID của hệ thống	59
Hình 4.11: Bay thực tế cùng độ cao cho 2 Drone	60
Hình 4.12: Giá trị độ cao của 2 Drone phản hồi về LCD	60
Hình 4.13: Test bay thực tế có gắn thanh liên kết theo mô hình	61

MỞ ĐẦU

1. Mục đích thực hiện đề tài

Hiện nay, thời đại 4.0 đòi hỏi sự thuận tiện, nhanh chóng, chính xác, an toàn và ổn định trong mọi công việc, đặc biệt là vận chuyển và các công việc cần điều khiển từ xa, do đó, không gian trên không là một không gian rất lớn để chúng ta tận dụng.

Từ đó, các thiết bị bay không người lái được tạo ra, nhằm phục vụ cho các mục đích khác nhau, có tính linh hoạt, ổn định, có thể điều khiển từ xa tùy thiết bị và đặc biệt quá trình vận chuyển đảm bảo được sự an toàn, tối ưu nhất. Được gọi là Drone.

Song, khối lượng tải cần vận chuyển hoặc mức độ công việc mà Drone phải làm đặt ra vấn đề lớn. Và đây là mục đích thực hiện đề tài.

2. Mục tiêu đề tài

2.1. Mục tiêu tổng quát

Mang lại những lợi ích cho xã hội, khoa học và công nghệ.

Về mặt xã hội, trước nhu cầu di chuyển, vận tải ngày càng tăng và tình trạng ùn tắc giao thông phức tạp, nghiên cứu này nhằm đề xuất khai thác không gian trên không – hiện chủ yếu do máy bay sử dụng – để vận tải. Việc lưu thông song song cả trên mặt đất và trên không sẽ giúp giảm tải giao thông, nâng cao chất lượng đời sống và phát triển xã hội.

Vì đó, chúng ta mong muốn có một công cụ thiết bị để vận chuyển, di chuyển các hàng hóa (tải) mà có thể điều khiển từ xa, không có sự ùn tắc, hiệu quả và an toàn. Song, đối với các tải có trọng lượng lớn, thì một drone không thể nâng được. Và cũng không thể thiết kế một Drone quá lớn, sẽ dẫn đến sự bất cập cũng như khối lượng của Drone sẽ tăng dần. Nên phương án “Thiết kế hệ Drone nâng tải” nhằm mục đích:

- Tiết kiệm được chi phí thiết kế Drone.
- Không cần thiết kế Drone có kích thước quá lớn.
- Chuẩn bị cho những trường hợp có tải trọng lớn.

Về mặt khoa học và công nghệ, đề án nghiên cứu sẽ được coi như là một nguồn để tham khảo cũng như là việc phát triển sau này. Điều này sẽ góp phần giảm chi phí và thời gian trong việc tránh lặp lại các tồn đọng của nghiên cứu trước đó. Từ đó, chỉ tập trung vào việc phát triển những điều tốt cũng như là dành tài nguyên để xây dựng cái mới nhằm đáp ứng nhu cầu đang ngày càng tăng.

2.2. Mục tiêu cụ thể

2.2.1. Nghiên cứu cách thức hoạt động của 1 Drone

Đồ án sẽ dựa vào kết quả nghiên cứu hoạt động của một hệ thống Drone riêng biệt trên bề mặt không gian của các bài báo, đề án khoa học trước đó nhằm xây dựng các hình thái hoạt động tối ưu của một hệ thống.

2.2.2. Nghiên cứu cách thức hoạt động của hệ Drone

Dựa vào việc xây dựng lại hệ thống Drone riêng biệt, đồ án sẽ phát triển hình thức hoạt động song song của nhiều Drone. Cụ thể, đồ án sẽ nghiên cứu hoạt động của hai thiết bị Drone khi được liên kết với nhau. Các hình thức hoạt động trên không vẫn tuân theo các di chuyển cơ bản lên/xuống, tiến tới/tiến lui, và xoay phải/trái.

2.2.3. Mô hình toán học và phương trình động lực học của Drone

Đồ án sẽ sử dụng và nhắc lại các mô hình toán học và phương trình động lực học từ các nghiên cứu trước đây. Nhằm làm rõ xem các dữ liệu trước đây có phù hợp với một hệ nhiều thiết bị Drone hay không.

2.2.4. Thiết kế bộ điều khiển dựa trên bộ điều khiển PID

Việc sử dụng bộ điều khiển PID trong đồ án sẽ đáp ứng các tiêu chí ban đầu đề ra tiếp cận được nhiều đối tượng nghiên cứu, khi mà bộ điều khiển này đang được sử dụng rộng rãi cả trong nghiên cứu và thực tế. Ngoài ra, đồ án sẽ nghiên cứu các nhiều thông số P, I, D khác nhau trong việc sử dụng nhiều PID để điều khiển các thiết bị thành phần của hệ thống nhiều Drone để xét tính ổn định và tính tích hợp của bộ điều khiển. Từ đó, tìm ra thông số phù hợp ổn định nhất.

Đồ án sử dụng phần mềm Matlab/Simulink để mô phỏng đưa ra kết quả đánh giá, như là một công cụ có độ uy tín cao hiện tại để đánh giá kết quả mô hình. Từ đó, giảm thiểu được các sai lệch, tồn đọng có thể xảy ra khi điều khiển một hệ Drone trước khi áp dụng vào việc phát triển một mô hình thực tế.

3. Đối tượng và phạm vi nghiên cứu

3.1. Đối tượng nghiên cứu

- Thiết bị bay: Drone

3.2. Phạm vi nghiên cứu

Phạm vi nghiên cứu của đề tài tập trung vào lĩnh vực tự động hoá trong việc nghiên cứu mô hình toán học, bài toán động lực học chưa thể hiện rõ trong bài nghiên cứu này,

nghiên cứu bộ điều khiển PID để xây dựng bộ điều khiển cho thiết bị. Mô phỏng trên phần mềm Matlab Simulink và áp dụng vào thực nghiệm mô hình.

4. Phương pháp nghiên cứu

Giải pháp nghiên cứu để kết hợp điều khiển nhiều Drone với nhau để cùng nâng tải. Và khởi đầu là nghiên cứu hệ 2 Drone cùng nâng tải. Chúng em mong là sẽ nghiên cứu thành công, từ đó có thể triển khai mô hình với nhiều Drone hơn.

4.1. Các nội dung nghiên cứu

Đề tài tập trung nghiên cứu vào các nội dung sau:

4.1.1. Tổng quan về lịch sử hình thành thiết bị Drone và nguyên lý hoạt động.

- Lịch sử hình thành Drone
- Nguyên lý hoạt động của Drone
- Nguyên lý hoạt động của hệ Drone kép

4.1.2. Nghiên cứu các trạng thái của Drone kép trên không gian mở.

- Trạng thái cân bằng của hệ Drone kép
- Trạng thái cùng tịnh tiến lên/xuống
- Trạng thái cùng tịnh tiến tới/lui
- Trạng thái nghiêng sang phải/trái
- Trạng thái xoay qua phải/trái

4.1.3. Nghiên cứu đề xuất thiết kế bộ điều khiển PID cho Drone.

- Sơ đồ điều khiển PID được đặt tên theo 3 khâu hiệu chỉnh đó là khâu tỉ lệ, khâu tích phân và khâu vi phân, ngõ ra điều khiển tốc độ của 4 động cơ là tổng của 3 khâu này.
- Ta có:

$$Output(t) = Pout + Iout + Dout \quad (1)$$

- Trong đó:
 - P_{out} : là thành phần đầu ra khâu tỉ lệ.
 - I_{out} : là thành phần đầu ra khâu tích phân.

- D_{out} : là thành phần đầu ra khâu vi phân.
- Vấn đề khó khăn khi sử dụng thuật toán PID để điều khiển mô hình là việc lựa chọn các giá trị K_p, K_i, K_d .
- Ở đây ta sử dụng phương pháp Ziegler-Nichols kết hợp với thực nghiệm.
- Việc xác định các thông số PID của mô hình máy bay rất phức tạp và phải thay đổi liên tục để có thể làm cho máy bay đạt được sự cân bằng và ổn định.

Dạng điều khiển	K_p	K_i	K_d
P	$0,5 * K_u$	-	-
PI	$0,45 * K_u$	$1,2 * \frac{K_p}{P_u}$	-
PID	$0,6 * K_u$	$2 * \frac{K_p}{P_u}$	$\frac{K_p * P_u}{8}$

- Trong đó:
 - K_u : độ lợi tới hạn
 - P_u : thời gian dao động

4.1.4. Đánh giá và mô phỏng kết quả triển khai công trình trên phần mềm Matlab/Simulink

4.2. Phương pháp nghiên cứu

4.2.1. Phương pháp nghiên cứu lý thuyết

Áp dụng điều khiển PID để xây dựng bộ điều khiển cho Drone đơn. Từ đó xây dựng bộ điều khiển cho Drone kép và mô phỏng trên phần mềm Matlab/Simulink để kiểm chứng, đánh giá và đưa ra kết luận.

4.2.2. Phương pháp nghiên cứu thực nghiệm

- Sử dụng Arduino và các module hỗ trợ.
- Nghiên cứu thuật toán điều khiển tích hợp cảm biến vị trí, cảm biến siêu âm.
- Xây dựng mô hình thử nghiệm với Drone, điều khiển vị trí ổn định.
- Lập trình Arduino để điều khiển vị trí, đo khoảng cách, nâng tải lên độ cao mong muốn.
- Kiểm tra, đánh giá độ chính xác, phản hồi và tối ưu hóa hệ thống.

4.2.3. Công nghệ, kỹ thuật sử dụng

- Phần cứng: Drone tự lắp ráp Quadcopter (4 cánh), cảm biến vị trí, cảm biến siêu âm.
- Truyền thông: tín hiệu RF, PPM, Master – Slave

4.2.4. Công cụ, môi trường

- Phần cứng: Drone Quadcopter tự lắp ráp
- Phần mềm: Arduino, Matlab, Visio
- Môi trường lập trình: Arduino IDE

5. Cấu trúc của đồ án tốt nghiệp

Mở đầu – Mục đích thực hiện đề tài, Mục tiêu đề tài, Phạm vi và đối tượng nghiên cứu, Phương pháp nghiên cứu, Cấu trúc của đồ án.

Chương 1 – Tổng quan về điều khiển hệ Drone kép nâng tải.

Chương 2 – Mô hình toán học của Drone và hệ Drone kép.

Chương 3 – Thiết kế hệ thống điều khiển hệ Drone kép.

Chương 4 – Mô phỏng, thực nghiệm và đánh giá.

Kết luận – Kết luận phương pháp, kết quả đạt được và hướng phát triển.

Tài liệu tham khảo – Liệt kê các nguồn tài liệu được sử dụng, tham khảo trong quá trình xây dựng và phát triển hệ thống.

Phụ lục – Hình ảnh linh kiện, sơ đồ của hệ thống, chương trình lập trình.

CHƯƠNG 1. TỔNG QUAN VỀ ĐIỀU KHIỂN HỆ DRONE KÉP NÂNG TẢI

1.1. Lý do chọn đề tài

1.1.1. Tính cần thiết từ đề tài

1.1.1.1. Cơ sở thực tiễn

Công nghệ đã tạo ra một cuộc cách mạng về mặt đổi mới và phát triển trong nhiều lĩnh vực của cuộc sống con người. Các thiết bị công nghệ cao đang được phát triển và ứng dụng rộng rãi nhằm đáp ứng nhu cầu của xã hội. Và trong số đó, Robot đóng vai trò là biểu tượng cho sự tiến bộ của công nghệ. Chính nhờ sự đổi mới của công nghệ và sự xuất hiện của Robot, cuộc sống con người đã được cải thiện và phát triển. Ngành robot học ngày càng phát triển, khiến cho robot tự hành có khả năng hoạt động trong nhiều môi trường khác nhau và được sử dụng để thực hiện nhiều loại nhiệm vụ khác nhau. Sau dần, thiết bị đang dần cải tiến về tầm vóc, tinh gọn và độ mỏng nhẹ được gọi là Drone.

Máy bay không người lái (Drone) là một trong những loại khí cụ bay không người lái có điều khiển. Từ khi ra đời đến nay Drone ngày càng được sử dụng phổ biến trong nhiều lĩnh vực:

- Chụp ảnh trên không
- Giám sát trên bộ, trên biển, kiểm soát, thăm dò.

Tuy nhiên Drone được sử dụng nhiều nhất trong lĩnh vực an ninh quốc phòng.

Về mặt kỹ thuật, Drone có nhiều điểm giống với máy bay có người lái. Tuy nhiên chúng có những đặc điểm khác biệt như:

- Chi phí cho nghiên cứu phát triển, chế tạo, vận hành, đảm bảo kỹ thuật thấp.
- Không bị tổn thất, ảnh hưởng tính mạng trong quá trình vận hành.

Với các ưu điểm trên, cùng với sự tiến bộ nhanh chóng của khoa học kỹ thuật và công nghệ, trong khoảng 10 năm trở lại đây đang diễn ra sự bùng nổ về Drone.

Cũng như các nước trên thế giới, đối với Việt Nam nhu cầu sử dụng Drone trong thời điểm hiện nay rất đa dạng:

- Làm mục tiêu phục vụ diễn tập.
- Sử dụng cho các mục đích trinh sát quân sự, nhiệm vụ an ninh quốc phòng, và kinh tế xã hội khác.

1.1.1.2. Cơ sở khoa học

Nhằm đáp ứng các vấn đề điều khiển Robot tự hành nói chung và bộ điều khiển Drone nói riêng, các nghiên cứu và phát triển với mục đích phục vụ phải được xác định rõ ràng. Trong đề án này, nghiên cứu sẽ tập trung vào hệ Drone với tiêu chí là mang tải trọng nhất định và không có trường hợp quá tải. Điều này nhằm đáp ứng việc hoạt động tốt cho thiết bị lâu dài theo thời gian cũng như việc bảo dưỡng được đảm bảo định kì. Như vậy, để đảm bảo lượng tải nhất định, việc nghiên cứu sẽ thực hiện theo hướng như sau:

- Dùng thiết bị Drone lớn
- Kết hợp các thiết bị Drone cùng kích cỡ

Các bài báo nghiên cứu về Drone theo các thuật toán của mạch điều khiển Logic, bộ điều khiển trượt, bộ điều khiển thích nghi, điều khiển hiện đại (điều khiển mờ, điều khiển nơron, điều khiển thích nghi, ...). Trong đó, hiện vẫn được sử dụng rộng rãi là bộ điều khiển PID, đây như là một giải thuật trong nhóm điều khiển tuyến tính, nó đang được phát triển vượt bậc và ứng dụng trong các hệ chuyển động phi tuyến, vì:

- Đảm bảo được tính khả thi cao.
- Dễ dàng trong việc thiết kế, sử dụng, điều khiển.
- Các tiêu chí kỹ thuật đạt chất lượng: tính bền vững và ổn định cao với biến thiên của nhiễu và tham số, dễ dàng thiết kế, dễ thay đổi, ...

Từ các nghiên cứu, thí nghiệm trước đó vẫn còn nhiều vấn đề, với việc đang là thời đại mà khoa học kỹ thuật càng có bước đột phá, nên chúng em lựa chọn thực hiện đề tài: **“ĐIỀU KHIỂN PHỐI HỢP HAI DRONE”** để làm đề tài cho Đồ Án Tốt Nghiệp.

1.2. Ý nghĩa về khoa học và thực tiễn

1.2.1. Về mặt khoa học

Nội dung đề tài sẽ là nguồn tài liệu tham khảo cho bất kỳ ai dành sự quan tâm đến ngành khoa học Robot tự hành nói chung và bộ điều khiển cho hệ Drone nói riêng. Thiết kế bộ điều khiển dựa trên PID sẽ được mô phỏng trên phần mềm Matlab Simulink như là dẫn chứng đáng tin cậy.

1.2.2. Về mặt thực tiễn

Kết quả nghiên cứu sẽ góp một phần để hoàn thiện công cuộc nghiên cứu khoa học Robot tự hành trong môi trường không khí. Sẽ cố gắng khắc phục các nhược điểm tồn

động trong việc điều khiển Drone về nhiều, khả năng mang tải và nhiều vấn đề khác từ đó phát triển lên hệ nhiều Drone.

1.3. Hiện trạng thực tế

1.3.1. Lịch sử phát triển Drone

Drone có thể thực hiện nhiều công việc hơn ngoài việc cất cánh và bay ổn định trên không. Cuối thế kỷ 19, sau một thời gian dài nghiên cứu, con người đã thành công trong việc chế tạo những thiết bị bay không người lái đầu tiên.

- Năm 1849

Quân đội Áo sử dụng khinh khí cầu không người lái để thả bom vào Venice, đây là một trong những ví dụ đầu tiên về vũ khí không người lái.



Hình 1.1: Máy bay đầu tiên năm 1900 của anh em nhà Wright

- Năm 1915

Kỹ sư người Anh Archibald Low phát triển hệ thống dẫn đường vô tuyến cho chiếc drone quân sự đầu tiên.

- Năm 1930 – 1940

Đức triển khai Bom Buzz - một loại tên lửa không người. Tuy nhiên, chúng không được trang bị nhiều hệ thống dẫn đường và không chính xác trong việc xác định mục tiêu.

- Năm 1986

Sau sự hợp tác giữa Hòa Kỳ và Israel, chiếc UAV đầu tiên RQ2 được ra đời. Đây là sự kết hợp giữa máy bay không người lái của Israel và các cải tiến về tải trọng của Mỹ.

- Năm 1996

General Atomics Corporation phát triển máy bay không người lái MQ1, với sự hỗ trợ của Abraham Karem.

- Năm 2006

UAV đã được sử dụng trong hoạt động cứu hộ thiên tai, được cho phép bay trong các hoạt động dân sự.

- Năm 2010

Amazon công bố kế hoạch sử dụng drone để giao hàng, mở ra một kỷ nguyên mới cho thương mại điện tử.

- Hiện nay

Công nghệ Drone đã tiến bộ vượt bậc so với những năm đầu với các tính năng như bay tự động, tránh va chạm, quay phim chất lượng cao, đo đạc địa hình, cứu hộ, cứu trợ, ...

1.3.2. Các công trình nghiên cứu và cơ sở khoa học ở Việt Nam.

Một bài báo tại hội nghị toàn quốc lần thứ 3 về Điều khiển và Tự động hoá – VCCA – 2015 của TS. Đỗ Quốc Tuấn, Th. S Nguyễn Đình Dũng và Th. S Phạm Hữu Ưông [1] là “Xác định các tham số trong mô hình toán của quadrotor” tiếp cận với các tham số cơ bản trong một mô hình toán của Drone.

Tiếp theo là công trình nghiên cứu của nhóm sinh viên trẻ K63AE của Viện hàng không vũ trụ vào năm 2019-2020 trong bài báo [2] là “Nghiên cứu, thiết kế, chế tạo Drone mang vật phẩm 1kg hạ cánh tự động”.



Hình 1.2: Mô hình đề tài nghiên cứu của nhóm K63AE Học viện Hàng Không

- Kết quả: Hoàn thành được mô hình có hệ thống cân bằng, xác định vị trí tương đối chính xác từ RTK, cảm biến xác định độ cao hoạt động tốt và đặc biệt là cho phép mang tải trọng 1kg với vận tốc là 8 m/s.

- Tuy nhiên, đây là những kết quả thông qua các bài báo chứ không có các video thực tế hay là các đồ thị cụ thể. Với công trình nghiên cứu của Nguyễn Hùng Thái Sơn và Võ Nguyên Phúc của trường Đại Học Lạc Hồng năm 2015 [3] là “Thiết Kế Và Thi Công Mô Hình Bay Quadcopter” đã xây dựng và phát triển mô hình Drone đã nâng cao hơn với bộ lọc DMP (Digital Motion Processor).



Hình 1.3: Mô hình đã hoàn thiện của bài báo nghiên cứu [3]

1.3.3. Thực tế

Loại máy bay không người lái	Trọng lượng	Khả năng tải trọng	Công dụng
Máy bay không người lái tải trọng thấp	Dưới 11 gram (0,011kg)	4 gam đến 100 gram (0,004 đến 0,1 kg)	Giám sát quân sự
Máy bay không người lái tải trọng nhỏ	200-1000 g (0,2-1 kg)	150-270 g (0,15-0,27 kg)	Giải trí và nhiếp ảnh
Máy bay không người lái hạng trung	1-600 kg (2,20-1323 lb)	3 kg Ví dụ: JOUAV CW-15	Ứng dụng chuyên nghiệp. Chụp ảnh trên không.
		40 kg (88 lb) đến 150 kg (331 lb).	
Máy bay không người lái hạng nặng	Hơn 160 kg	550 kg	Khả năng phát hiện và chiến đấu của kẻ thù. Các ứng dụng dân sự như giao hàng bằng máy bay không người lái hoặc làm phim.

Bảng 1.1: Một số dạng máy bay không người lái trong thực tế

- Một số hình ảnh thực tế



Hình 1.4: Drone vận chuyển hàng hóa trong thực tế

1.3.4. Câu hỏi đặt ra

- Một tải trọng vượt quá khả năng tải trọng của máy bay không người lái hạng thấp và hạng nhỏ thì phải làm sao?

Trả lời: Thiết kế, đầu tư một máy bay không người lái (Drone) hạng trung hoặc hạng nặng để nâng tải trọng lớn hơn. Câu trả lời này dẫn đến một câu hỏi khác.

- Chi phí để đầu tư một máy bay không người lái hạng trung hoặc hạng nặng là quá lớn, trong khi đó tải trọng thì không quá lớn, phương án giải quyết như thế nào?

Trả lời: Dường như, đây là vấn đề chung của bất kì tổ chức hay cá nhân nào có nhu cầu sử dụng Drone để nâng tải hoặc vận chuyển hàng hóa. Và đây cũng là lí do quan trọng nhất để chúng em nghiên cứu đề tài này.

1.4. Dự kiến kết quả đạt được

1.4.1. Quá trình thử nghiệm

1.4.1.1. Phương án kết nối hệ Drone kép

Liên kết cứng: Sử dụng 2 dây vải mềm buộc trực tiếp vào mỗi Drone, mỗi đầu còn lại của dây sẽ buộc vào 2 đầu của 1 thanh đỡ (thanh inox).

Thanh đỡ này ngoài chức năng kết nối 2 thiết bị Drone với nhau nhằm giúp hệ Drone kép ổn định, còn tránh sự va đập giữa các thiết bị với nhau. Nó cũng sẽ coi như là một giá treo để mang tải.

Thanh liên kết cứng không có lực căng dây và các drone hoạt động như một khối thống nhất. Phương trình đơn giản hơn.

Tổng khối lượng hệ thống:

$$m_{max} = m_{drone\ 1} + m_{drone\ 2} + m_{thanh\ inox} + m_{tải} \quad (1.1)$$

Mục đích chính là điều khiển hệ Drone kép nâng tải. Do đó, cần giảm thiểu tải trọng hết mức có thể để hệ Drone kép có thể nâng tải ổn định.

- Ưu điểm:

- Bộ cánh quạt được lắp cố định trên trục động cơ
- Có thể dùng hơn 4 động cơ
- Có thể nâng cấp các thành phần riêng biệt
- Hạn chế tác động nhiễu vì việc điều khiển đồng thời 8 động cơ cùng 1 lúc
- Tải trọng tăng lên vì việc kết hợp 2 drone
- Có thể thay đổi các động cơ khác nhau phù hợp với mục đích.

- Nhược điểm:

- Kích thước lớn, chiếm diện tích nhiều hơn
- Thanh liên kết cần chắc chắn để vừa cố định được 2 drone vừa nâng tải.
- Giá thành cao vì mô hình nhiều động cơ, nhiều cánh quạt và các linh kiện liên quan.
- Khả năng nhiễu tác động sẽ nhiều hơn, như gió, sóng, ...

- Khả năng ứng dụng

- Sử dụng trong các hệ thống vận chuyển trên không.
- Có thể mở rộng để vận tải các hàng hóa lớn tùy mục đích.
- Sử dụng trong công tác cứu nạn tại các khu vực khó tiếp cận, hiểm trở.

1.4.1.2. Lý do chọn nguyên vật liệu kết nối

Dây vải mềm: cụ thể là dây buộc giày bata phổ biến.

- Chi phí thấp, khối lượng nhỏ (rất nhẹ).
- Không bị dòn, thích hợp với mô hình.
- Chắc chắn.
- Dễ dàng thay thế.

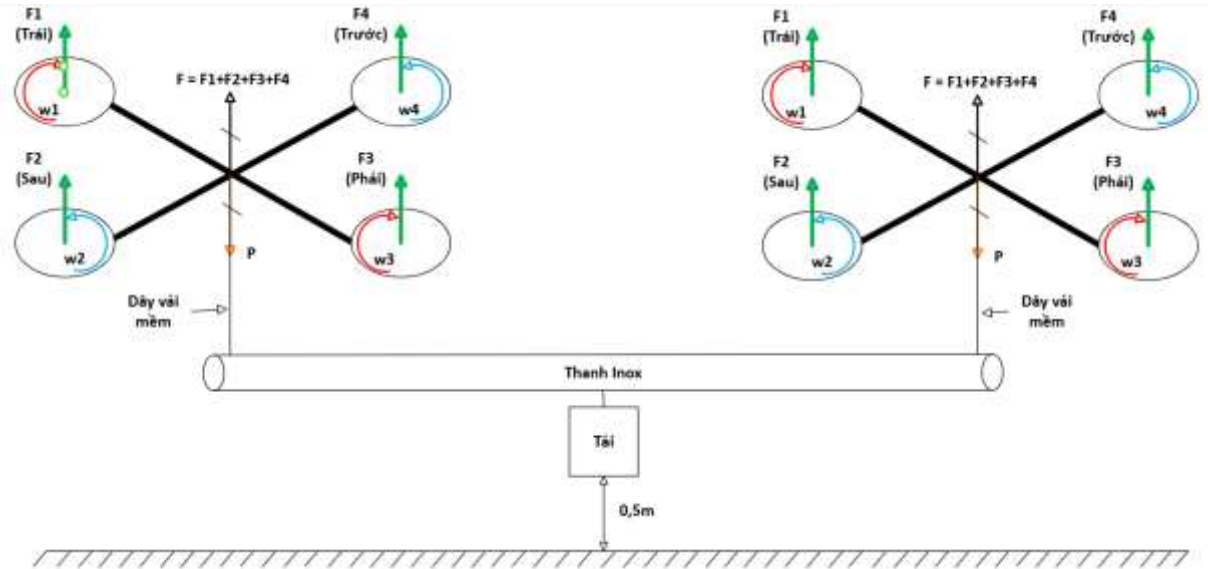
Thanh đỡ: là thanh có chất liệu inox

- Chi phí thấp hoặc tận dụng những vật dụng không còn sử dụng.
- Khối lượng nhỏ.
- Chắc chắn.

- Dễ dàng thay thế.

1.4.2. Kết quả dự kiến:

Mô hình thực nghiệm để kết nối hệ Drone kép được dự kiến như hình dưới đây:



Hình 1.5: Mô hình dự kiến của hệ Drone kép nâng tải

1.5. Kết luận chương

Trong chương 1, thông qua việc tìm hiểu các trạng thái làm việc của Drone cơ bản và ứng dụng để đóng góp cho việc xây dựng trạng thái làm việc cho Drone kép ở chương tiếp theo.

Khi khối lượng tải không thể được nâng bởi 1 Drone, lúc này sự bất cập, hạn chế xảy ra khi tải trọng vượt quá khả năng nâng của Drone. Vì đó, đề án nghiên cứu phương pháp điều khiển đồng thời hệ 2 Drone, giải quyết vấn đề nâng tải. Chi phí là một nguyên nhân lớn nên không thể đầu tư những Drone lớn.

Vì vậy, điều khiển hệ Drone kép nâng tải mang lại nhiều thuận lợi hơn:

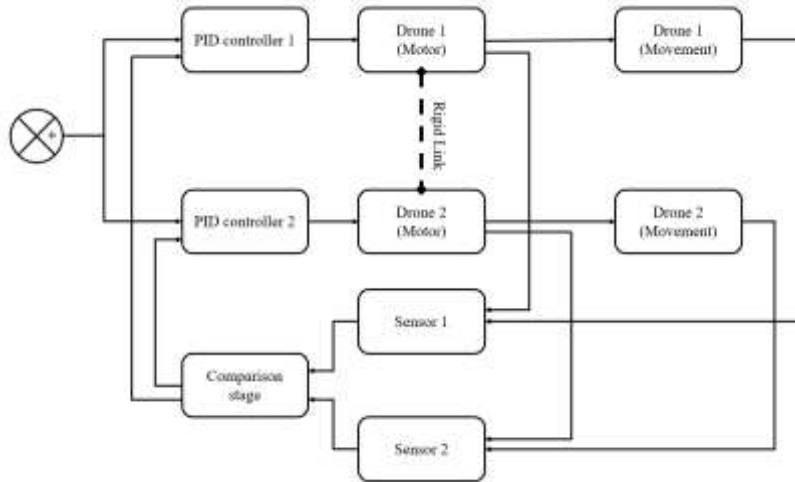
- Có thể thực hiện đối với nhiều tải có khối lượng lớn.
- Phục vụ cho nhiều lĩnh vực khác nhau.
- Được đảm bảo an toàn hơn, tiết kiệm chi phí hơn.
- Giảm tình trạng ùng đống đưa như đối với 1 Drone.

CHƯƠNG 2. MÔ HÌNH TOÁN HỌC CỦA DRONE VÀ HỆ DRONE KÉP

2.1. Cấu trúc

2.1.1. Sơ đồ khối

- Sơ đồ khối điều khiển Drone kép bằng vật liệu cứng (Rigid Link)



Hình 2.1: Sơ đồ khối điều khiển Drone kép bằng vật liệu cứng

Hệ thống điều khiển drone kép được thiết kế với liên kết cơ khí cứng (Rigid Link) giữa hai Drone. Cấu trúc liên kết này đảm bảo rằng cả hai drone luôn di chuyển đồng bộ và duy trì một khoảng cách cố định trong suốt quá trình vận hành.

Do đặc tính không cho phép tách rời hay dao động tương đối giữa hai thiết bị, liên kết cứng loại bỏ hoàn toàn tính độc lập trong chuyển động của từng drone. Cụ thể, mọi thay đổi về vị trí, tốc độ hoặc hướng bay của một drone sẽ tức thời truyền lực và ảnh hưởng trực tiếp đến drone còn lại. Điều này giúp hệ thống hoạt động như một thể thống nhất, đặc biệt phù hợp trong các nhiệm vụ nâng hoặc kéo tải đòi hỏi độ chính xác và cân bằng cao.

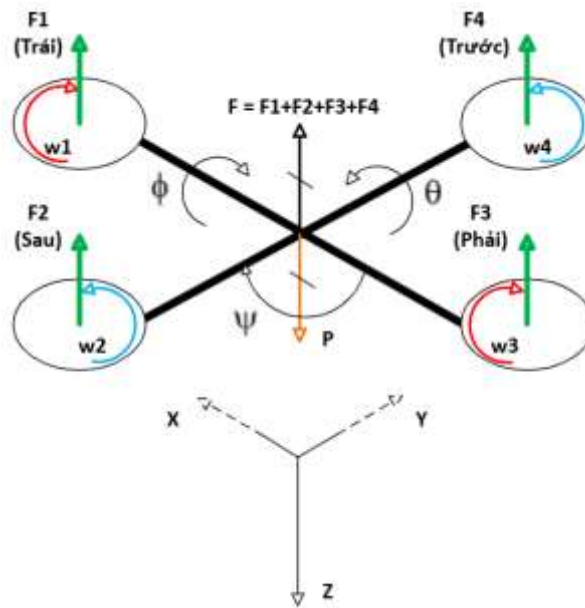
2.2. Nguyên lý hoạt động của drone và hệ drone kép

2.2.1. Nguyên lý hoạt động của Drone

Đối với Drone, kết cấu chỉ gồm 4 motor điện giống nhau có lắp cánh quạt được đặt ở 4 góc hình vuông và cánh quạt có thể gắn trực tiếp hay thông qua truyền động bánh răng, 4 motor này được chia thành 2 cặp: trước – sau, trái – phải.

Chiều quay của 2 motor trong mỗi cặp là giống nhau, nhưng chiều quay của cặp trước – sau và trái – phải là ngược nhau.

Cặp trước – sau quay cùng chiều kim đồng hồ thì cặp trái – phải quay ngược chiều kim đồng hồ. Và cánh quạt dùng trong Drone phải là 2 loại ngược nhau, một loại quay cùng chiều và một loại quay ngược chiều kim đồng hồ.



Hình 2.2: Chiều quay của các động cơ trên 1 Drone

Giải thích một chút về các thuật ngữ được sử dụng khi nó bay về phía trước, ngược, ngang hoặc xoay trong khi lơ lửng. Chúng được gọi là Pitch, Roll và Yaw.

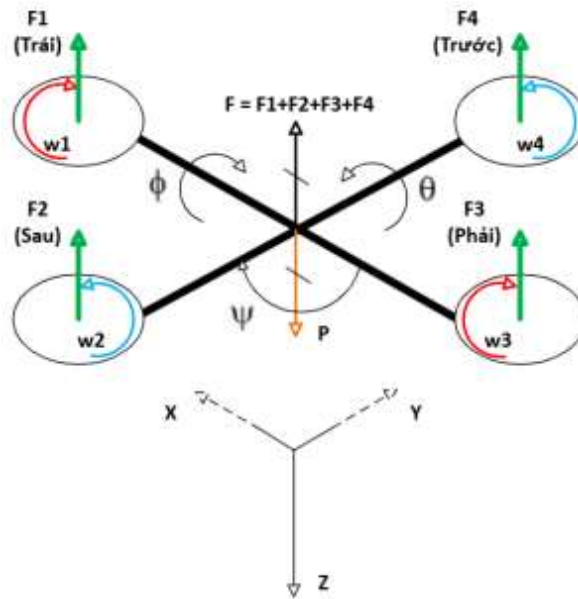
- Yaw - Đây là xoay hoặc xoay đầu của Drone hoặc sang phải hoặc trái. Đó là chuyển động cơ bản để quay Drone quanh trục Z. Trên hầu hết các mục tiêu giả lập, nó là đạt được bằng cách sử dụng thanh ga bên trái hoặc sang trái hoặc phải.
- Pitch - Đây là chuyển động của Drone hoặc về phía trước và phía sau. Chuyển tiếp Pitch đạt được thường bằng cách đẩy thanh ga về phía trước, mà làm cho Drone nghiêng và di chuyển về phía trước, ra xa bạn. Còn phía sau lại đạt được bằng cách di chuyển thanh ga ngược về phía sau.
- Roll - Bay nghiêng về một bên là làm cho bay Drone sang một bên, hoặc sang trái hoặc phải. Bay nhào lộn được điều khiển bởi thanh điều khiển bên phải, làm cho nó bay hoặc trái sang phải.

2.2.1.1. Nguyên lý cân bằng của Drone

Máy bay Drone có dạng chữ thập hay dấu cộng với bốn động cơ đặt ở 4 góc. Hai trục chéo của máy bay được đặt theo hai trục X, Y của hệ trục tọa độ Descartes. Mỗi động cơ kết hợp với cánh quạt trong Drone sẽ tạo ra một lực đẩy và moment xoắn nhất

định, bốn cánh quạt được chia thành hai nhóm có chiều quay ngược nhau, hai cánh đối diện nhau quay cùng chiều.

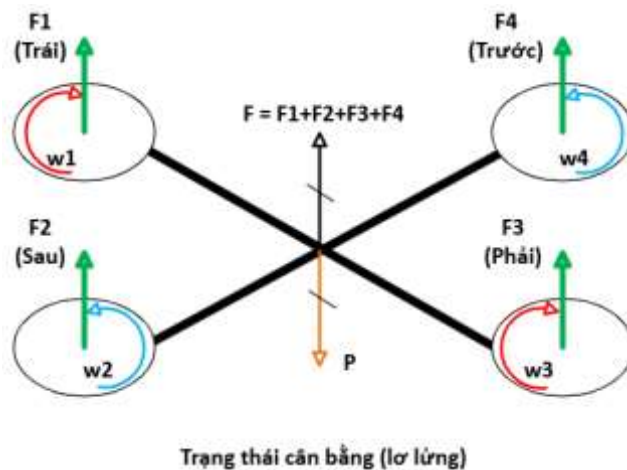
Kết quả là moment xoắn bị triệt tiêu nếu 4 cánh quạt đều có cùng một vận tốc góc, do đó có thể làm cho máy bay không bị xoay tròn khi bay. Để cân bằng mô hình thì các động cơ phải được điều khiển ra sao cho mô hình có góc lệch so với trục chuẩn trong phạm vi cho phép. Các hệ trục tọa độ của Drone thể hiện trong hình.



Hình 2.3: Chiều quay và các góc quay để Drone cân bằng

2.2.1.2. Mô tả các trạng thái làm việc của Drone đơn

2.2.1.2.1. Trạng thái lơ lửng



Hình 2.4: Chiều quay của Drone ở trạng thái lơ lửng

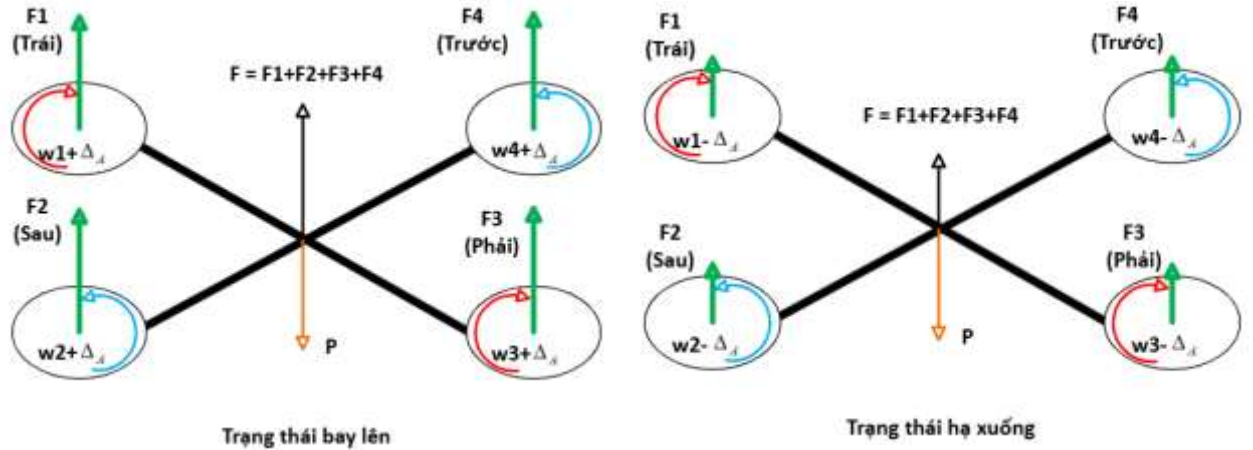
Ở trạng thái này, tất cả các cánh quạt quay cùng một tốc độ không đổi ($\omega_1 = \omega_2 = \omega_3 = \omega_4$).

Trạng thái này sẽ kiểm tra khả năng độ ổn định của Drone ở trên không trung một thời gian dài, nhờ vậy có thể kiểm tra xem khả năng chịu nhiễu của Drone ở thời gian dài.

Tổng lực nâng các động cơ sinh ra: $F = F_1 + F_2 + F_3 + F_4$

- Điều khiển để Drone cân bằng (lơ lửng) tại 1 vị trí: $F = P$

2.2.1.2.2. Trạng thái bay lên/xuống



Hình 2.5: Trạng thái Drone di chuyển lên hoặc xuống

Drone sẽ bay lên hoặc hạ xuống theo phương thẳng đứng. Để bay lên, tốc độ của 4 cánh quạt tăng lên một lượng bằng nhau và hạ xuống thì cả 4 cánh cùng giảm tốc độ một lượng bằng nhau, lúc đó sẽ sinh ra một hợp lực dọc trục theo phương Z điều khiển quadrotor bay lên hoặc bay xuống.

Tổng lực nâng các động cơ sinh ra: $F = F_1 + F_2 + F_3 + F_4$

- Drone bay lên: $F > P$
- Drone hạ xuống: $F < P$

Trong đó:

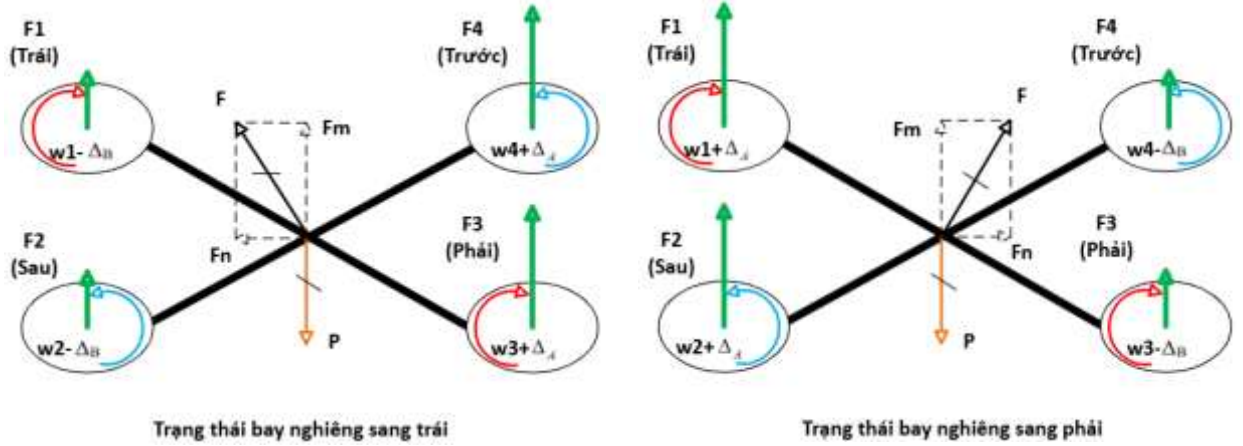
- Z: Gia tốc theo phương Z_b
- $\omega_1, \omega_2, \omega_3, \omega_4$: Tốc độ của các cánh quạt
- Δ : Lượng tăng hoặc giảm vận tốc của cánh quạt để bay lên hay hạ xuống.

2.2.1.2.3. Trạng thái bay nghiêng sang trái/phải " Φ " (Roll)

Drone bay nghiêng sang trái hoặc sang phải:

- Để bay nghiêng bên phải:

- Tăng tốc độ cánh quạt bên trái và cánh quạt phía sau.
- Giảm tốc độ cánh quạt bên phải và cánh quạt phía trước.
- Để bay nghiêng bên trái:
 - Tăng tốc độ cánh quạt bên phải và cánh phía trước.
 - Giảm tốc độ cánh quạt bên trái và cánh phía sau.



Hình 2.6: Trạng thái Drone di chuyển nghiêng sang trái hoặc sang phải

Trong đó:

- Φ : Góc tốc theo phương Y_b
- $\omega_1, \omega_2, \omega_3, \omega_4$: Tốc độ của các cánh quạt
- Δ_A, Δ_B ($\Delta_A \approx \Delta_B$): Lượng tăng hoặc giảm của ω_i

Tổng lực nâng các động cơ sinh ra: $F = \sqrt{(F_m)^2 + (F_n)^2}$ lực này kéo Drone bay nghiêng sang trái hoặc sang phải.

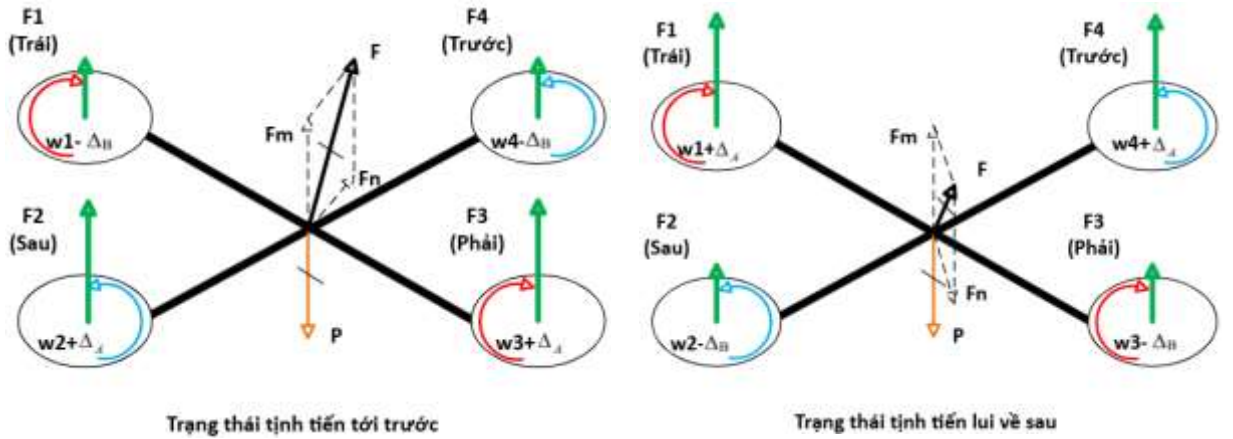
- Để đảm bảo Drone có thể bay nghiêng sang trái hoặc sang phải nhưng vẫn cân bằng thì: $F = P$

2.2.1.2.4. Trạng thái tịnh tiến tới/lui " θ " (Pitch)

Drone bay tịnh tiến tới trước hoặc lui về sau:

- Để bay tịnh tiến tới trước:
 - Tăng tốc độ cánh quạt bên phải và cánh quạt phía sau.
 - Giảm tốc độ cánh quạt bên trái và cánh quạt phía trước.
- Để bay tịnh tiến lui về sau:
 - Tăng tốc độ cánh quạt bên trái và cánh phía trước.

- Giảm tốc độ cánh quạt bên phải và cánh phía sau.



Hình 2.7: Trạng thái Drone di chuyển tịnh tiến tới hoặc lui

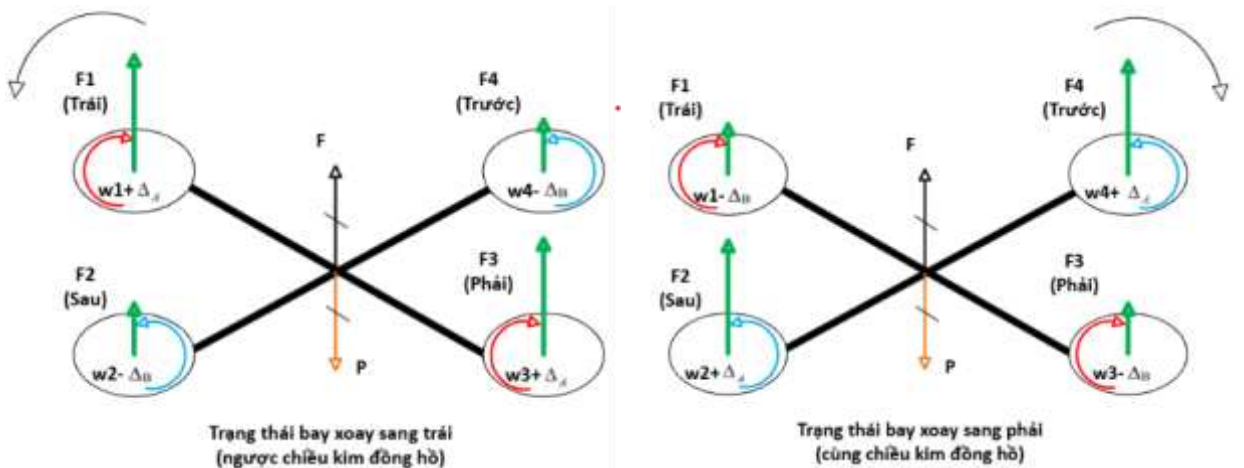
Trong đó:

- θ : Góc tốc theo phương X_b
- $\omega_1, \omega_2, \omega_3, \omega_4$: Tốc độ của các cánh quạt
- Δ_A, Δ_B ($\Delta_A \approx \Delta_B$): Lượng tăng hoặc giảm của ω_i

Tổng lực nâng các động cơ sinh ra: $F = \sqrt{(F_m)^2 + (F_n)^2}$ lực này giúp kéo vật tiến tới trước hoặc lui về sau.

- Để đảm bảo Drone có thể bay tịnh tiến tới trước hoặc lui về sau nhưng vẫn cân bằng thì: $F = P$

2.2.1.2.5. Trạng thái xoay qua trái/phải " Ψ " (Yaw)



Hình 2.8: Trạng thái Drone di chuyển xoay sang trái hoặc sang phải

Drone quay quanh trục Z. Điều khiển tốc độ cánh quạt theo cách sau:

- Tốc độ 2 cánh đối diện thì bằng nhau, nhưng khác với tốc độ 2 cánh đối diện còn lại.
- Nếu chỉ giảm tốc độ của 1 trong 2 cặp cánh thì sẽ làm lực đẩy của Drone. Do đó, chúng ta cần tăng tốc độ cho cặp còn lại để tổng lực đẩy không đổi, giúp Drone vẫn cân bằng tại vị trí nhất định.
- Để Drone xoay sang trái (ngược chiều kim đồng hồ):
 - Tăng tốc độ 2 cánh quạt có chiều cùng chiều kim đồng hồ: cánh trái và cánh phải.
 - Giảm tốc độ 2 cánh quạt có chiều ngược chiều kim đồng hồ: cánh trước và cánh sau.
- Để Drone xoay sang phải (cùng chiều kim đồng hồ):
 - Tăng tốc độ 2 cánh quạt có chiều ngược chiều kim đồng hồ: cánh trước và cánh sau.
 - Giảm tốc độ 2 cánh quạt có chiều cùng chiều kim đồng hồ: cánh trái và cánh phải.

Trong đó:

- Ψ : Gia tốc theo phương Z_b
- $\omega_1, \omega_2, \omega_3, \omega_4$: Tốc độ của các cánh quạt
- Δ_A, Δ_B ($\Delta_A \approx \Delta_B$): Lượng tăng hoặc giảm của ω_i

Tổng lực nâng các động cơ sinh ra: $F = F_1 + F_2 + F_3 + F_4$

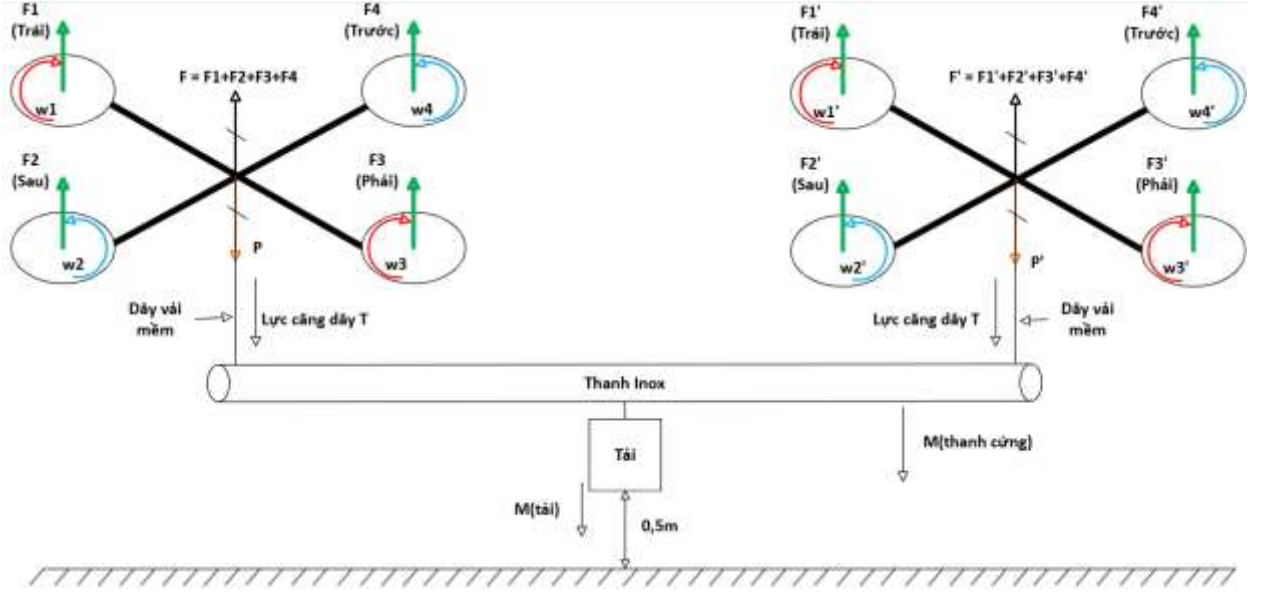
- Các lực nâng F_1, F_2, F_3, F_4 có thay đổi, nhưng tổng lực nâng luôn không đổi.

Qua đó, thấy rằng việc điều khiển bay của Multirotor UAV (Drone) là việc điều khiển tốc độ quay của các động cơ.

2.2.2. Nguyên lý hoạt động của hệ Drone kép

Đầu tiên, đề tài sẽ trình bày hệ kép Drone dựa trên sự kết hợp của 2 thiết bị bay giống nhau về các cấu trúc thành phần bao gồm: động cơ, khối lượng, lực nâng, và các chuyển động cơ bản đã đề cập ở trên.

Dựa vào nguyên lý hoạt động đã trình bày, ta áp dụng vào hệ Drone kép có mang tải theo hình dưới đây. Ta có sơ đồ hình vẽ như sau:



Hình 2.9: Mô phỏng một mô hình hệ Drone kép

2.2.2.1. Nguyên lý cân bằng của hệ Drone kép

Tương tự như thiết bị Drone thứ nhất, thiết bị Drone thứ hai cũng sẽ có các thiết bị thành phần tương đồng về mẫu mã, nguyên liệu và công suất. Các cặp cánh bên phải của 2 thiết bị Drone phải về tương đồng góc quay và cặp cánh bên trái cũng vậy. Khi cất cánh thì tốc độ quay của 8 cánh phải bằng nhau để sinh ra lực đẩy cân bằng.

Ta có phương trình sơ bộ dựa theo khi hệ Drone kép cân bằng như sau:

$$\begin{cases} F_1 = F_2 = F_3 = F_4 \\ F_1' = F_2' = F_3' = F_4' \\ \Sigma F + \Sigma F' = P + P' + P_T + P_{t\grave{a}i} \\ M_{12} + M_{1'2'} = M_{34} + M_{3'4'} \end{cases} \quad (2.1)$$

Trong đó:

- F_i : Lực nâng của cánh quạt thứ i của Drone 1
- F_i' : Lực nâng của cánh quạt thứ i của Drone 2
- P : Trọng lượng của Drone 1
- P' : Trọng lượng của Drone 2
- P_T : Trọng lượng của thanh Inox
- $P_{t\grave{a}i}$: Trọng lượng của tải
- M_{xy} : Moment xoay được tạo ra bởi các cặp cánh quạt

2.2.2.2. Mô tả các trạng thái làm việc của hệ Drone kép

2.2.2.2.1. Trạng thái bay lên/xuống

Hệ Drone kép sẽ bay lên hoặc hạ xuống theo phương thẳng đứng *hình 2.9*. Để bay lên, tốc độ của 8 cánh quạt tăng lên hoặc nếu hạ xuống thì cả 8 cánh cùng giảm tốc độ, lúc đó sẽ sinh ra một hợp lực dọc trục theo phương Z điều khiển hệ Drone kép bay lên hoặc bay xuống.

Ta biểu diễn như sau:

$$\begin{cases} \Sigma F + \Sigma F' > \Sigma P \rightarrow \text{Lên} \\ \Sigma F + \Sigma F' < \Sigma P \rightarrow \text{Xuống} \end{cases} \quad (2.2)$$

2.2.2.2.2. Trạng thái tịnh tiến tới/lui

Hệ Drone kép sẽ tịnh tiến tới hoặc lui về sau như theo *hình 2.9*. Nguyên lý hoạt động cũng tương tự như Drone đơn:

- Hệ Drone kép tịnh tiến tới trước:
 - Giảm tốc độ của 2 cặp cánh quạt trước và cánh quạt trái.
 - Tăng tốc độ của 2 cặp cánh quạt sau và cánh quạt phải.
- Hệ Drone kép tịnh tiến lui về sau:
 - Giảm tốc độ của 2 cặp cánh quạt sau và cánh quạt phải.
 - Tăng tốc độ của 2 cặp cánh quạt trước và cánh quạt trái.

Ta biểu diễn như sau:

$$\begin{aligned} [(\omega_1 + \Delta_1) + (\omega_4 + \Delta_4)] + [(\omega_{1'} + \Delta_{1'}) + (\omega_{4'} + \Delta_{4'})] \\ > [(\omega_2 + \Delta_2) + (\omega_3 + \Delta_3)] + [(\omega_{2'} + \Delta_{2'}) + (\omega_{3'} + \Delta_{3'})] \end{aligned} \quad (2.3)$$

$$\begin{aligned} [(\omega_1 + \Delta_1) + (\omega_4 + \Delta_4)] + [(\omega_{1'} + \Delta_{1'}) + (\omega_{4'} + \Delta_{4'})] \\ < [(\omega_2 + \Delta_2) + (\omega_3 + \Delta_3)] + [(\omega_{2'} + \Delta_{2'}) + (\omega_{3'} + \Delta_{3'})] \end{aligned} \quad (2.4)$$

Trong đó:

- ω_i : Tốc độ của cánh quạt thứ i của Drone 1.
- $\omega_{i'}$: Tốc độ của cánh quạt thứ i của Drone 2.
- Δ_i : Lượng tăng hoặc giảm tốc độ cánh quạt thứ i của Drone 1.
- $\Delta_{i'}$: Lượng tăng hoặc giảm tốc độ cánh quạt thứ i của Drone 2.
- Phương trình (2.3) khiến hệ Drone kép tịnh tiến lùi về sau.
- Phương trình (2.4) khiến hệ Drone kép tịnh tiến tới trước.

2.2.2.2.3. Trạng thái bay nghiêng sang trái/phải

Hệ Drone kép bay nghiêng sang trái hoặc sang phải như theo *hình 2.9*. Nguyên lý hoạt động cũng tương tự như Drone đơn:

- Hệ Drone kép bay nghiêng sang trái:
 - Giảm tốc độ của 2 cặp cánh quạt trái và cánh quạt sau.
 - Tăng tốc độ của 2 cặp cánh quạt phải và cánh quạt trước.
- Hệ Drone kép bay nghiêng sang phải:
 - Giảm tốc độ của 2 cặp cánh quạt phải và cánh quạt trước.
 - Tăng tốc độ của 2 cặp cánh quạt trái và cánh quạt sau.

Ta biểu diễn như sau:

$$[(\omega_1 + \Delta_1) + (\omega_2 + \Delta_2)] + [(\omega_{1'} + \Delta_{1'}) + (\omega_{2'} + \Delta_{2'})] < [(\omega_3 + \Delta_3) + (\omega_4 + \Delta_4)] + [(\omega_{3'} + \Delta_{3'}) + (\omega_{4'} + \Delta_{4'})] \quad (2.5)$$

$$[(\omega_1 + \Delta_1) + (\omega_2 + \Delta_2)] + [(\omega_{1'} + \Delta_{1'}) + (\omega_{2'} + \Delta_{2'})] > [(\omega_3 + \Delta_3) + (\omega_4 + \Delta_4)] + [(\omega_{3'} + \Delta_{3'}) + (\omega_{4'} + \Delta_{4'})] \quad (2.6)$$

Trong đó:

- ω_i : Tốc độ của cánh quạt thứ i của Drone 1.
- $\omega_{i'}$: Tốc độ của cánh quạt thứ i của Drone 2.
- Δ_i : Lượng tăng hoặc giảm tốc độ cánh quạt thứ i của Drone 1.
- $\Delta_{i'}$: Lượng tăng hoặc giảm tốc độ cánh quạt thứ i của Drone 2.
- Phương trình (2.5) khiến hệ Drone kép bay nghiêng sang trái.
- Phương trình (2.6) khiến hệ Drone kép bay nghiêng sang phải.

2.2.2.2.4. *Trạng thái thay đổi quỹ đạo của toàn bộ hệ thống*

Hệ Drone kép ở trạng thái này sẽ xoay toàn bộ 1 góc bất kỳ để phù hợp với các điều kiện địa hình trong thực tế. Ở trạng thái này, các Drone độc lập trong hệ thống sẽ di chuyển tiến và lui một cách ngược nhau cùng với một lực.

Giả sử, nếu Drone 1 hành động tiến lên thì trong lúc đó Drone 2 sẽ hành động lùi lại. Điều này giúp cho hệ Drone kép sẽ thay đổi góc mong muốn mà không có sự cản trở gì.

Ta biểu diễn như sau:

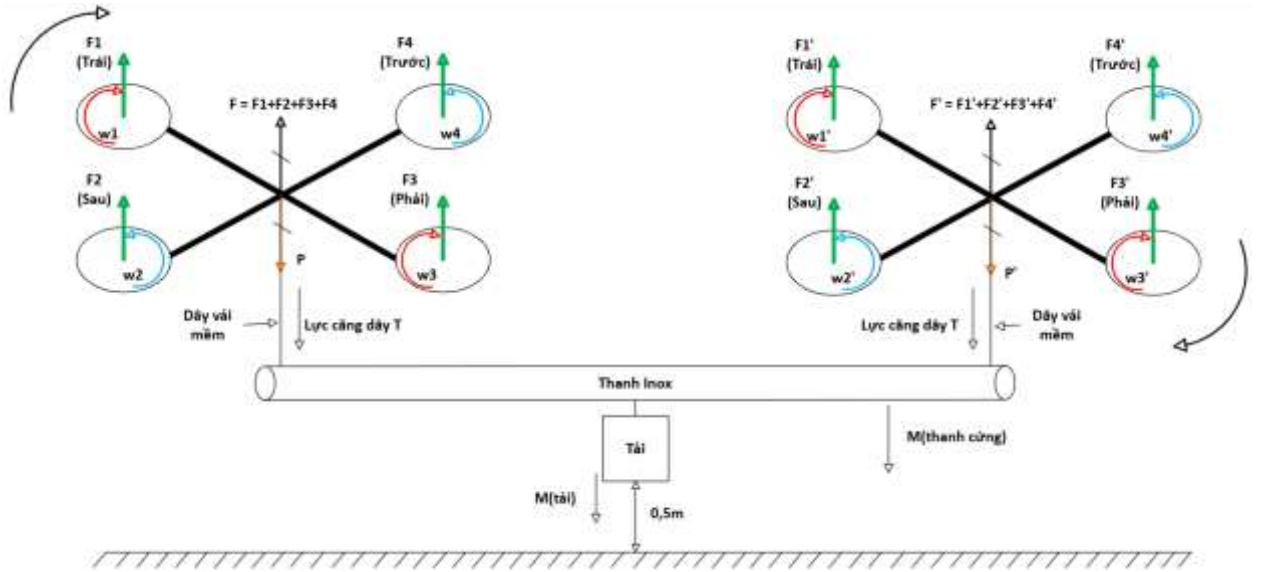
$$\begin{cases} (\omega_1 + \Delta_1) + (\omega_4 + \Delta_4) < (\omega_2 + \Delta_2) + (\omega_3 + \Delta_3) \\ ((\omega_{1'} + \Delta_{1'}) + (\omega_{4'} + \Delta_{4'})) > ((\omega_{2'} + \Delta_{2'}) + (\omega_{3'} + \Delta_{3'})) \end{cases} \quad (2.7)$$

Hoặc là:

$$\begin{cases} (\omega_1 + \Delta_1) + (\omega_4 + \Delta_4) > (\omega_2 + \Delta_2) + (\omega_3 + \Delta_3) \\ (\omega_{1'} + \Delta_{1'}) + (\omega_{4'} + \Delta_{4'}) < (\omega_{2'} + \Delta_{2'}) + (\omega_{3'} + \Delta_{3'}) \end{cases} \quad (2.8)$$

Trong đó:

- ω_i : Tốc độ của cánh quạt thứ i của Drone 1.
- $\omega_{i'}$: Tốc độ của cánh quạt thứ i của Drone 2.
- Δ_i : Lượng tăng hoặc giảm tốc độ cánh quạt thứ i của Drone 1.
- $\Delta_{i'}$: Lượng tăng hoặc giảm tốc độ cánh quạt thứ i của Drone 2.
- Phương trình **Lỗi! Không tìm thấy nguồn tham chiếu.** khiến hệ Drone 1 tịnh tiến tới và Drone 2 tịnh tiến lui.
- Phương trình **Lỗi! Không tìm thấy nguồn tham chiếu.** khiến hệ Drone 1 tịnh tiến lui và Drone 2 tịnh tiến tới.



Hình 2.10: Chiều di chuyển của 2 Drone khi xoay một góc bất kỳ

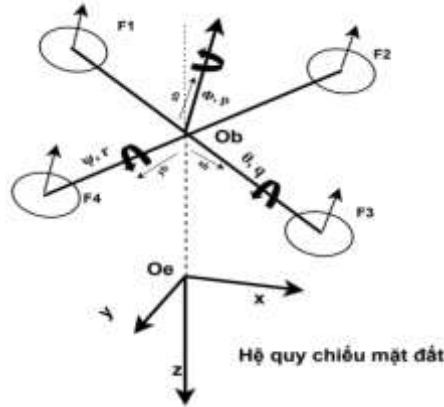
2.3. Phương trình động lực

2.3.1. Tính chất động lực học của Drone

2.3.1.1. Các hệ tọa độ

Từ nội dung chương 1, ta xác nhận lại Multirotor UAV (Drone) là một hệ thống 6 – DOF, trong đó sử dụng 12 biến trạng thái để mô tả. Trong đó, 6 biến trạng thái đầu tiên được sử dụng để mô tả độ cao. Các biến (Φ, θ, Ψ) tương ứng với các góc roll, pitch và

yaw, là góc Euler giữa hệ quy chiếu mặt đất. Trong khi đó (p, q, r) là vận tốc góc roll, pitch và yaw của hệ quy chiếu vật thể và (u, v, w) là vận tốc dài.



Hình 2.11: Mô phỏng các đại lượng trên một mô hình Drone

Các biến x, y, z tương ứng với hệ trục tọa độ mặt đất và các biến u, v, w là vận tốc dài trong hệ quy chiếu vật thể.

2.3.1.2. Các thông số được sử dụng

- Góc Euler đặt trên thân

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \quad (2.9)$$

- Hệ trục tọa độ mặt đất

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2.10)$$

Trong đó:

- x : tọa độ của Drone trên trục O_{ex} của hệ trục quán tính gắn với mặt đất.
- y : tọa độ của Drone trên trục O_{ey} của hệ trục quán tính gắn với mặt đất.
- z : tọa độ của Drone trên trục O_{ez} của hệ trục quán tính gắn với mặt đất.

- Vận tốc dài ứng với hệ trục tọa độ mặt đất góc O

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.11)$$

Trong đó:

- u : vận tốc tịnh tiến trên trục O_{bx} gắn với Drone.

- y: vận tốc tịnh tiến trên trục O_{by} gắn với Drone.
- z: vận tốc tịnh tiến trên trục O_{bz} gắn với Drone.
- Vận tốc ứng với góc Euler

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.12)$$

Trong đó:

- p: vecto vận tốc góc nghiêng nằm trên trục O_{bx} gắn với Drone.
- q: vecto vận tốc góc nghiêng nằm trên trục O_{by} gắn với Drone.
- r: vecto vận tốc góc nghiêng nằm trên trục O_{bz} gắn với Drone.

2.3.1.3. Động lực học của cánh quạt

Trong phần này, bài nghiên cứu sẽ tập trung vào động lực học của cánh quạt và động cơ một chiều kéo quay cánh quạt. Kết quả nghiên cứu được tham khảo từ một số công trình trước đây đã đăng. Người ta đã chỉ ra rằng, lực đẩy và mômen tạo ra bởi cánh quạt có độ lớn tỉ lệ thuận với bình phương của tốc độ quay của cánh quạt. Ta có hệ phương trình các lực và moment dưới đây dưới đây:

$$\begin{cases} U_1 = b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) = F_1 + F_2 + F_3 + F_4 \\ U_2 = Lb(\omega_2^2 - \omega_4^2) = T_x \\ U_3 = Lb(\omega_1^2 - \omega_3^2) = T_y \\ U_4 = d(\omega_2^2 + \omega_4^2 - \omega_1^2 - \omega_3^2) = T_z \end{cases} \quad (2.13)$$

Trong đó:

- U_1 : Tổng lực nâng do 4 cánh quạt tạo ra.
- U_2 : Mômen quay quanh trục O_{bx} của Drone, gây ra góc (Roll).
- U_3 : Mômen quay quanh trục O_{by} của Drone, gây ra góc (Pitch).
- U_4 : Mômen quay quanh trục O_{bz} của Drone, gây ra góc (Yaw).
- ω_i : lần lượt tương ứng với tốc độ quay của cánh thứ 1 đến cánh thứ 4.
- F_i : lần lượt tương ứng với lực nâng tạo ra bởi cánh thứ 1 đến cánh thứ 4.
- b: hệ số lực nâng.
- L: độ dài tay đòn từ vị trí trọng tâm đến mỗi động cơ.

- T_x : tương ứng với momen trên trục O_{bx} gắn với Drone, gây ra góc nghiêng Φ .
- T_y : tương ứng với momen trên trục O_{by} gắn với Drone, gây ra góc nghiêng θ .
- T_z : tương ứng với momen trên trục O_{bz} gắn với Drone, gây ra góc nghiêng Ψ .
- d : hệ số kéo dài.

Để tiện cho việc phân tích động lực học và việc điều khiển Drone, đồ án sẽ tách riêng phần điều khiển động cơ cánh quạt, vì vấn đề này đã được khảo sát ở đầu chương.

Như vậy đối với nhiệm vụ điều khiển Drone thì đồ án sẽ sử dụng trực tiếp các khái niệm lực và momen làm các tín hiệu điều khiển thông qua các thông số đã được trình bày ở phía trên.

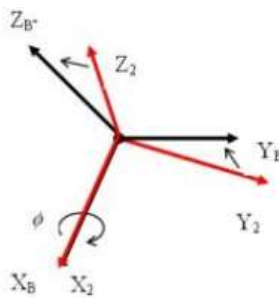
2.3.1.4. Phân tích động lực học của Drone

Trước tiên, ta xem Drone như một vật rắn cứng tuyệt đối chuyển động tự do trong không gian. Từ đó, ta phân tích chuyển động của Drone thành 2 phần cơ bản chính là: Chuyển động tịnh tiến và chuyển động quay.

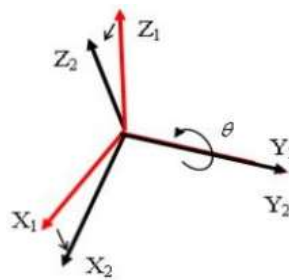
Chuyển động tịnh tiến sẽ ứng với khái niệm vận tốc dài và chuyển động quay ứng với khái niệm vận tốc góc. Đồ án sẽ đi sâu vào từng thành phần cơ bản.

2.3.1.4.1. Ma trận chuyển đổi

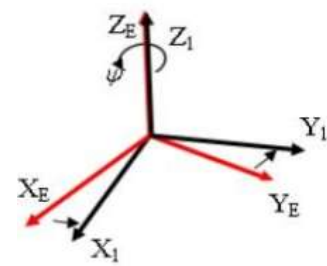
Ma trận xoay R là ma trận chuyển đổi cho phép ta chuyển từ một hệ tọa độ này sang hệ tọa độ khác thông qua 3 góc Euler (Φ – góc nghiêng, θ – góc chúc góc, Ψ – góc hướng) và 3 ma trận cos-sin hướng.



Hình 2.12a



Hình 2.12b



Hình 2.12c

Quay quanh trục Z_E một góc ψ (yaw) ta được ma trận quay $R(\psi, z)$ và thu được hệ tọa độ trung gian $B_{x_1 y_1 z_1}$ (hình 2.12c).

$$R_{(\psi,z)} = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.14)$$

Tiếp tục đem qua hệ tọa độ $B_{x_1y_1z_1}$ xung quanh trục Quay quanh trục Y_2 một góc θ (pitch) ta được ma trận quay $R(\theta, y)$ (hình 2.12b) và thu được hệ tọa độ trung gian thứ hai $B_{x_2y_2z_2}$.

$$R_{(\theta,y)} = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.15)$$

Cuối cùng ta quay $B_{x_2y_2z_2}$ xung quanh trục X_B một góc Φ (roll) ta được ma trận quay $R(\Phi, x)$ (hình 2.12a) và được hệ tọa độ với Drone B_{xyz} .

$$R_{(\Phi,x)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \quad (2.16)$$

Như vậy thông qua 3 phép quay ta có 3 ma trận quay tương ứng, tiếp theo ta tính tích của 3 ma trận vừa tìm được:

$$R_{(\Phi,\theta,\psi)} = R_{(\Phi,x)} \cdot R_{(\theta,y)} \cdot R_{(\psi,z)} \quad (2.17)$$

Thay (2.14), (2.15) và (2.16) vào (2.17) ta thu được:

$$R_{(\Phi,\theta,\psi)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \times \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \times \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.18)$$

Sử dụng phép nhân ma trận ta có:

$$R_{(\Phi,\theta,\psi)} = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \cos \theta \end{bmatrix} \quad (2.19)$$

Ma trận xoay R là ma trận chuyển đổi từ hệ trục tọa độ mặt đất sang hệ tọa độ vật thể.

Phương trình trạng thái mô tả sự thay đổi vị trí vận tốc trong hệ tọa độ vật thể:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = R^{-1} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.20)$$

2.3.1.4.2. Chuyển đổi vận tốc góc

Việc phân tích tốc độ góc $\begin{bmatrix} p \\ q \\ r \end{bmatrix}$ của Drone trong hệ tọa độ quán tính có khác đôi chút.

Vì vecto Ψ nằm trên trục Ob_{z_1} thuộc hệ trục $Ob_{x_1y_1z_1}$, vecto θ nằm trên trục Ob_{z_2} thuộc hệ trục $Ob_{x_2y_2z_2}$ và cuối cùng vecto Φ thuộc hệ trục tọa độ Ob_{xyz} (1)(2)(3).

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \phi \\ 0 \\ 0 \end{bmatrix} + R_{(\Phi)} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_{(\Phi)} \times R_{(\theta)} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \quad (2.21)$$

Thực hiện phép tính toán ma trận, ta được:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & \sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \times \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.22)$$

Ta được phương trình trạng thái mô tả sự thay đổi độ cao khi xoay trong hệ trục tọa độ vật thể:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \times \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.23)$$

2.3.1.4.3. Vận tốc dài

Phương trình mô tả vận tốc dài trong hệ trục tọa độ mặt đất theo định luật II Newton:

$$F = m\vec{V} \quad (2.24)$$

Phương trình Coriolis trong hệ chiếu quay so với hệ quy chiếu quán tính:

$$\vec{F} = m(\vec{\omega} \times \vec{v}) \quad (2.25)$$

Định luật biến thiên động lượng có dạng:

$$\vec{F} \cdot \Delta t = m\vec{v}_2 - m\vec{v}_1 \quad (2.26)$$

Dựa vào phương (11), (12) và (13), khi Drone tịnh tiến ta được:

$$\frac{d\vec{Q}}{dt} = \frac{m d\vec{v}}{dt} = m(\vec{\alpha} + \vec{\omega} \times \vec{v}) = \vec{F} \quad (2.27)$$

Thực hiện phép tính toán đơn giản, ta rút ra được:

$$\vec{F} = m\vec{\alpha} + m\vec{\omega} \times \vec{v} \quad (2.28)$$

Phương trình vận tốc góc có dạng:

$$\vec{\omega} = \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \quad (2.29)$$

Phương trình vận tốc dài có dạng:

$$\vec{\alpha} = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} \quad (2.30)$$

Thay (2.29), (2.30) và (2.11) vào phương trình (2.28) lên các trục tọa độ x, y, z ta thu được:

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = m \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + m \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.31)$$

Thực hiện phép tính toán ma trận, ta được:

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = m \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} \quad (2.32)$$

Thực hiện phép đảo (2.31), thu được:

$$\vec{\alpha} = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} -(qw - rv) \\ +(ru - pw) \\ -(pv - qu) \end{bmatrix} + \frac{1}{m} \times \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \quad (2.33)$$

Ngoại lực tác động lên Drone bao gồm lực nâng của 4 cánh quạt U_1 tại phương trình (2.32) và trọng lực W. Lực nâng U_1 có hướng theo trục z trong khi đó trọng lực W có hướng theo trục z nhưng ngược chiều với U_1 .

$$\begin{bmatrix} W_x \\ W_y \\ W_z - U_1 \end{bmatrix} = m \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} \quad (2.34)$$

2.3.1.4.4. Vận tốc góc

Đối với chuyển động quay, ta áp dụng định luật biến thiên momen động lượng ta được:

$$\vec{T} = \frac{d\vec{H}}{dt} = \frac{m d\vec{\omega}}{dt} = I \left(\frac{d\vec{\omega}}{dt} + \vec{\Omega} \times \vec{\omega} \right) \quad (2.35)$$

Trong đó:

- T: vecto momen ngoại lực
- H: vecto momen động lượng của Drone
- $I(\vec{\Omega} \times \vec{\omega})$: thành phần tương ứng với hiệu ứng con quay

Thiết kế Drone có cấu trúc đối xứng, với tốc độ quay của 2 cặp động cơ cánh quạt bằng nhau về độ lớn, thiết bị và ngược chiều nhau nên thành phần momen con quay sẽ được triệt tiêu.

Do đó, ta chỉ sẽ khảo sát ảnh hưởng của momen lực điều khiển T lên chuyển động góc của Drone, ta có tổng momen ngoại lực như sau:

$$T = T_1 + T_2 + T_3 + T_4 \quad (2.36)$$

Ta cho Drone với hình dạng đối xứng, nên ta sẽ có được momen quán tính của thân Drone có dạng như sau:

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (2.37)$$

Ta có mối quan hệ giữa 2 đại lượng I đặt theo thân tổng thể tại phương trình (2.38) và T tại phương trình (2.37) như sau:

$$\vec{T} = I \times \vec{\omega} \quad (2.38)$$

Thay phương trình (2.38) vào phương trình vận tốc góc (2.29), ta có:

$$\vec{\omega} = \begin{bmatrix} \frac{T_x}{I_{xx}} \\ \frac{T_y}{I_{yy}} \\ \frac{T_z}{I_{zz}} \end{bmatrix} \quad (2.39)$$

Phương trình mô tả vận tốc góc có dạng:

$$U = \dot{T} \quad (2.40)$$

Khi vecto momen góc đổi chiều:

$$U = \dot{T} + \omega T \quad (2.41)$$

Thay phương trình (2.37), (2.38) vào phương trình (2.41) ta có dạng:

$$\begin{cases} U_2 = \dot{p}I_x + qr(I_z - I_y) \\ U_3 = \dot{q}I_y + pr(I_x - I_z) \\ U_4 = \dot{r}I_z + pq(I_y - I_x) \end{cases} \quad (2.42)$$

Mặt khác, xz và yz đối xứng, nên $I_x \approx I_y$:

$$\begin{cases} U_2 = \dot{p}I_x + qr(I_z - I_y) \\ U_3 = \dot{q}I_y + pr(I_x - I_z) \\ U_4 = \dot{r}I_z \end{cases} \quad (2.43)$$

Ngoài ra momen được tạo ra bởi motor (2.13):

$$\begin{cases} U_2 = Lb(\omega_2^2 - \omega_4^2) \\ U_3 = Lb(\omega_1^2 - \omega_3^2) \\ U_4 = d(\omega_2^2 + \omega_4^2 - \omega_1^2 - \omega_3^2) \end{cases} \quad (2.44)$$

Từ phương trình (2.42) vào (2.43) sau đó đổi về ta thu được:

$$\begin{cases} \dot{p} = \frac{Lb}{I_x}(\omega_2^2 - \omega_4^2) - qr \frac{(I_z - I_y)}{I_x} \\ \dot{q} = \frac{Lb}{I_y}(\omega_1^2 - \omega_3^2) - pr \frac{(I_x - I_z)}{I_y} \\ \dot{r} = \frac{d}{I_z}(\omega_2^2 + \omega_4^2 - \omega_1^2 - \omega_3^2) \end{cases} \quad (2.45)$$

Tương ứng với:

- Nếu tổng trọng lượng của Drone là m và có trọng lực đặt tại tâm của vật. Giả sử Drone ở trạng thái bền vững: với d_{cg} là khoảng cách từ trọng tâm đến tâm trục động cơ và h_{cg} chiều cao mô hình. Nếu m_r là trọng lượng của mỗi motor với bán hình cánh quạt là r và chiều dài cánh là c , nếu trọng lượng của mỗi động cơ là m_m , bán kính động cơ r_m và chiều cao động cơ h_m , m_t là trọng lượng của thanh cái nối giữa 2 thiết bị Drone, thì momen quán tính $I \in R^{3 \times 3}$ của Drone:

$$I_x = m \frac{h_{cg}^2 + 3d_{cg}^2}{12} + 2m_m l^2 + 2 \frac{1}{12} m_m (3r_m^2 + 4h_m^2) + 2m_r l^2 + 2 \left(\frac{1}{12} m_r (2r)^2 \right) \quad (2.46)$$

$$I_y = m \frac{h_{cg}^2 + 3d_{cg}^2}{12} + 2m_m l^2 + 2 \frac{1}{12} m_m (3r_m^2 + 4h_m^2) + 2m_r l^2 + 2 \left(\frac{1}{12} m_r c^2 \right) \quad (2.47)$$

$$I_z = \frac{md_{cg}^2}{2} + 4m_m l^2 + 4 \frac{m_m r_m^2}{2} + 4 \frac{m_r ((2r)^2 + c^2)}{12} + 4m_r l^2 \quad (2.48)$$

2.3.1.4.5. Tính chất phi tuyến của động học Drone

Ta có một hệ phương trình mô tả động lực học hoàn chỉnh cho Drone bao gồm:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = R^{-1} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.47)$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi \tan\theta & \cos\phi \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \times \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.48)$$

$$\begin{cases} \dot{u} = rv - qw - s \sin\theta \\ \dot{v} = pw - ru + g \cos\theta \sin\phi \\ \dot{w} = qu - pv + s \cos\phi \cos\theta - \frac{b(\Omega_1 + \Omega_2 + \Omega_3 + \Omega_4)}{m} \end{cases} \quad (2.49)$$

$$\begin{cases} \dot{p} = \frac{Lb}{I_y} (\omega_2^2 - \omega_4^2) - qr \frac{I_z - I_y}{I_x} \\ \dot{q} = \frac{Lb}{I_y} (\omega_1^2 - \omega_3^2) - pr \frac{I_x - I_z}{I_y} \\ \dot{r} = \frac{d}{I_z} (\omega_2^2 + \omega_4^2 - \omega_1^2 - \omega_3^2) \end{cases} \quad (2.50)$$

Như vậy ta thu được hệ phương trình động học phi tuyến toàn phần của Drone được tổng hợp dưới đây:

$$\ddot{\phi} = -\dot{\psi} \times \dot{\theta} \cos\phi + \cos\psi \times \frac{T_x}{i_{xx}} - \sin\psi \times \frac{T_y}{i_{yy}} \quad (2.51)$$

$$\ddot{\theta} = \dot{\psi} \times \frac{\dot{\phi}}{\cos\phi} + \dot{\phi} \times \dot{\theta} \tan\phi + \frac{\sin\psi}{\cos\phi} \frac{T_x}{i_{xx}} + \frac{\cos\psi}{\cos\phi} \frac{T_y}{i_{yy}} \quad (2.52)$$

$$\begin{aligned} \ddot{\psi} = & \dot{\phi} \times \dot{\psi} \tan\phi + \dot{\phi} \times \frac{\dot{\theta}}{\cos\phi} + \sin\psi \tan\phi \frac{T_x}{i_{xx}} \\ & + \cos\psi \tan\phi \frac{T_y}{i_{yy}} \frac{T_z}{i_{zz}} \end{aligned} \quad (2.53)$$

$$\ddot{x} = (-\cos\phi \sin\theta \cos\psi - \sin\phi \sin\psi) \times \frac{1}{m} \times F_L \quad (2.54)$$

$$\ddot{y} = (-\cos\phi \sin\theta \sin\psi + \sin\phi \cos\psi) \times \frac{1}{m} \times F_L \quad (2.55)$$

$$\ddot{z} = g - \cos\phi \cos\theta \times \frac{1}{m} \times F_L \quad (2.56)$$

Trong đó:

- Φ, θ và Ψ : các góc cuộn, ngóc và góc nghiêng tương ứng.
- I_x, I_y, I_z : momen quán tính theo các trục x, y và z.
- J_r : quán tính động cơ.
- Ω : vận tốc góc của rotor.

- L : chiều dài của cánh tay cánh quạt từ gốc của hệ tọa độ.

2.3.1.5. Động lực học của hệ Drone kép

2.3.1.5.1. Vận tốc dài

Trong hệ Drone kép, ta có hai Drone với vận tốc dài v_1, v_2 và lực đẩy tương ứng F_1, F_2 .

Vận tốc dài tổng hợp của hệ là:

$$v_{dual} = \frac{v_1 + v_2}{2} \quad (2.57)$$

Phương trình động lực học của vận tốc dài của hệ:

$$\dot{v}_{dual} = \frac{F_{dual}}{m_{dual}} - g - \omega_{dual} v_{dual} \quad (2.58)$$

Với:

$$F_{dual} = F_1 + F_2$$

$$m_{dual} = m_1 + m_2$$

- Tương tác phi tuyến

Hiệu vận tốc giữa hai Drone:

$$\Delta v = v_1 - v_2 \quad (2.59)$$

Lực tương tác khí động học giữa hai Drone:

$$F_{int} = C_d A \rho (v_1 - v_2) |v_1 - v_2| \quad (2.60)$$

Trong đó:

- F : lực gió (N).
- ρ : mật độ không khí (kg/m^3), thường là giá trị khoảng $1,255 \text{ kg}/\text{m}^3$.
- v : tốc độ gió (m/s).
- C_d : hệ số cản khí động học của vật.
- A : diện tích bề mặt tiếp xúc với gió (m^2).

Phương trình vận tốc dài tổng hợp bao gồm tương tác:

$$\dot{v}_{dual} = \frac{F_{dual}}{m_{dual}} - g - \omega_{dual} v_{dual} + \frac{F_{int}}{m_{dual}} \quad (2.61)$$

- Vận tốc góc

Mở rộng cho hệ Drone kép

Trong hệ Drone kép, vận tốc góc tổng hợp là:

$$\omega_{dual} = \frac{\omega_1 + \omega_2}{2} \quad (2.62)$$

Phương trình động lực học của vận tốc góc tổng hợp:

$$I_{dual}\dot{\omega}_{dual} = T_{dual} - \omega_{dual}(I_{dual}\omega_{dual}) + T_{int} \quad (2.63)$$

Với:

$$T_{dual} = T_1 + T_2 \quad I_{dual} = \frac{I_1 + I_2}{2}$$

Hiệu vận tốc góc giữa hai Drone:

$$\Delta\omega = \omega_1 - \omega_2 \quad (2.64)$$

Momen tương tác phi tuyến:

$$T_{int} = k_T(\omega_1 - \omega_2) \quad (2.65)$$

Phương trình vận tốc góc tổng hợp bao gồm tương tác:

$$I_{dual}\dot{\omega}_{dual} = T_{dual} - \omega_{dual}(I_{dual}\omega_{dual}) + k_T(\omega_1 - \omega_2) \quad (2.66)$$

Ma trận xoay tổng hợp:

$$R_{dual} = \frac{R_1 + R_2}{2} \quad (2.67)$$

Quan hệ giữa R_1 và R_2 :

$$R_2 = k \cdot R_1 \quad (2.68)$$

Ma trận xoay tổng hợp trong hệ:

$$\dot{R}_{dual} = \frac{\dot{R}_1 + \dot{R}_2}{2} \quad (2.69)$$

Đồ án sẽ giả sử cho tọa độ của cả hệ Drone kép đặt tạo trọng tâm của thanh nối, tổng hợp phương trình động lực học của hệ Drone kép.

2.3.1.5.2. Cấu hình hệ Drone kép

Phân phối lực nâng và tải trọng

- Tổng khối lượng toàn hệ:

$$m_{tổng} = m_{drone\ 1} + m_{drone\ 2} + m_{thanh\ inox} + m_{tải} \quad (2.70)$$

- Trọng lực G:

$$G = m_{tong} \cdot g \quad (2.71)$$

Thanh tải nối Drone 1 và Drone 2:

- Drone 1 nằm ở vị trí $(-d_1, 0, 0)$ cách trọng tâm một khoảng d_1 .
- Drone 2 nằm ở vị trí $(+d_2, 0, 0)$ cách trọng tâm một khoảng d_2 .
- Chiều dài của thanh tải: $l_t = d_1 + d_2$

2.3.1.5.3. Phương trình động lực học

- Cân bằng lực toàn hệ

Phương ngang (x): Tổng lực tác động theo phương ngang

$$m_{tong} \times \ddot{x} \times G = (F_{up1} - T_1 \times \sin \alpha_1) + (F_{up2} - T_2 \times \sin \alpha_2) \quad (2.72)$$

Phương ngang (y): Tổng lực tác động theo phương dọc

$$m_{tong} \times \ddot{y} \times G = (F_1 + F_2) - G - T_1 \times \cos \alpha_1 + T_2 \times \cos \alpha_2 \quad (2.73)$$

Phương trình (z): Tổng lực tác động theo phương đứng

$$m_{tong} \times \ddot{z} \times G = F_1 + F_2 - G \quad (2.74)$$

- Cân bằng momen toàn hệ

Quanh trục Roll (x): Momen quanh trục Roll

$$M_x = Lb \times (\omega_2^2 - \omega_4^2) + Lb \times (\omega_2'^2 - \omega_4'^2) \quad (2.75)$$

Quanh trục Pitch (y); Momen quanh trục Pitch

$$M_y = d_1 \times F_1 - d_2 \times F_2 \quad (2.76)$$

Quanh trục Yaw (z): Momen quanh trục Yaw

$$M_z = d(\omega_1^2 + \omega_3^2 - \omega_2^2 - \omega_4^2) + d(\omega_1'^2 + \omega_3'^2 - \omega_2'^2 - \omega_4'^2) \quad (2.77)$$

- Phương trình động lực học tuyến tính

Vận tốc theo phương ngang (x)

$$\dot{x} = (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \frac{U_1}{m_{tong}} \quad (2.78)$$

Vận tốc theo phương dọc (y)

$$\dot{y} = (\cos \phi \sin \theta \sin \psi + \sin \phi \cos \psi) \frac{U_1}{m_{tổng}} \quad (2.79)$$

Vận tốc theo phương đứng (z)

$$\dot{z} = -g + (\cos \phi \cos \theta) \frac{U_1}{m_{tổng}} \quad (2.80)$$

- Phương trình động học góc

Gia tốc Roll

$$\ddot{\phi} = \dot{\theta} \times \psi \left(\frac{I_y - I_z}{I_x} \right) - \frac{J_r}{I_x} \times \dot{\theta} \times \Omega + \frac{L}{I_x} U_2 \quad (2.81)$$

Gia tốc Pitch

$$\ddot{\theta} = \dot{\phi} \times \psi \left(\frac{I_z - I_x}{I_y} \right) - \frac{J_r}{I_y} \times \dot{\phi} \times \Omega + \frac{L}{I_y} U_3 \quad (2.82)$$

Gia tốc Yaw

$$\ddot{\psi} = \dot{\phi} \times \dot{\theta} \left(\frac{I_x - I_y}{I_z} \right) + \frac{L}{I_z} U_4 \quad (2.83)$$

2.3.1.5.4. Liên hệ giữa hai Drone

- Lực truyền giữa 2 Drone:
 - Thanh tải phân phối lực nâng và truyền tải trọng giữa hai Drone.
 - Lực ngang và dọc từ mỗi Drone đóng vai trò cân bằng qua thanh tải.
- Sự phụ thuộc lực nâng:

Lực nâng từ Drone 1 và Drone 2 phải đảm bảo đủ để nâng toàn bộ trọng lượng và duy trì sự ổn định:

$$F_1 + F_2 = (m_{thanh\ inox} + m_{tải}).g \quad (2.84)$$

- Liên hệ qua momen:

Thanh tải chịu momen quay do sự phân bố tải trọng và truyền momen đến hai Drone để duy trì cân bằng góc:

$$m_{tổng} = d_1 \times F_1 - d_2 \times F_2 + m_{tải} \cdot \Delta d \quad (2.85)$$

Điều kiện cân bằng:

$$d_1 \times F_1 = d_2 \times F_2 - m_{tải} \times g \times \Delta d \quad (2.86)$$

2.4. Đề xuất cho hệ thống

- Đối với phần cứng

Ở khối truyền tín hiệu TX, thiết kế một tay cầm điều khiển được kết nối trực tiếp với Arduino.



Hình 2.13: Tay cầm điều khiển thiết kế tích hợp Arduino



Hình 2.14a: Mô hình thực tế Drone 1



Hình 2.14b: Mô hình thực tế Drone 2

- Đối với phần mềm

Sử dụng phần mềm Arduino IDE, thực hiện viết chương trình thiết kế khối nhận tín hiệu Transmitter (TX) và khối nhận tín hiệu Receiver (RX).

2.5. Kết luận chương

Từ các bài báo, nghiên cứu ở chương 1 cũng như các đặc tính cơ bản của 1 Drone. Chương 2 đã xây dựng các phương trình cơ bản của nguyên lý hoạt động, cách di chuyển định các vấn đề cơ bản cần giải quyết cho đề tài vào việc nghiên cứu điều khiển mô hình, sử dụng lại các tài nguyên trước để xây dựng đồ án.

CHƯƠNG 3. THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN HỆ DRONE KÉP

3.1. Tính toán, lựa chọn động cơ, cánh quạt trong hệ thống và thiết kế linh kiện cho phần động lực

3.1.1. Tính toán, lựa chọn động cơ, cánh quạt trong hệ thống

- Mục tiêu:
 - Tải cần nâng: $\leq 2\text{kg}$ (tính cả trọng lượng drone).
 - Loại drone: F450 (quadcopter – 4 động cơ).
 - Mục tiêu: Bay ổn định, có thể treo tại chỗ, bay chở tải nhẹ.
- Tính lực nâng yêu cầu mỗi motor:
 - Tổng tải trọng cần nâng:
 - Tải cần kéo: $m_{\text{tải}} \leq 2\text{kg}$
 - Trọng lượng toàn bộ:

$$W = 2\text{kg} \Rightarrow F = mg = 2 * 9,81 = 19,62 \text{ (N)}$$

- Quy tắc:
 - Lực nâng tổng cần ít nhất gấp 2 lần tổng trọng lượng để bay ổn định, có dư lực để điều khiển và nâng hạ.

⇒ Lực nâng tối thiểu toàn bộ:

$$T_{\text{total}} \geq 2 * 9,81 = 19,62 \text{ (N)}$$

⇒ Mỗi động cơ cần cung cấp lực nâng:

$$T_{\text{motor}} \geq \frac{19,62}{4} = 4,905 \text{ (N)} \approx 500 \text{ (g)}$$

⇒ Tối thiểu mỗi motor phải tạo ra 500g lực nâng

Chú thích: vì trong thực tế và khi tra thông số lực nâng của motor, người ta thường dùng đơn vị "gram lực" (g) để dễ hình dung và so sánh. Lý do:

- Motor được thử nghiệm trên bench test hay datasheet thường cho kết quả: "Motor này sinh ra được 800g", 900g...
- 1 Newton $\approx$ 100g lực. Cụ thể là:

$$1\text{N} = \frac{1\text{kg} * m}{s^2} \approx 9,8m/s^2 \Rightarrow 1\text{N} \approx 100\text{g}$$

- Để đảm bảo an toàn khi bay:

- Trong thực tế, ta không nên thiết kế motor để chạy đạt 100% lực nâng khi bay, mà chọn hệ số an toàn 1.5x hoặc hơn:

⇒ Cần motor sinh ra: $T_{motor} \geq 500 * 1,5 = 750 (g)$

- Tính toán chọn motor, cánh quạt, khung drone:

Tra thông số thực nghiệm của motor A2212 1400KV + cánh 1045:

- A2212 1400KV + cánh 10x4.5 inch + pin 3S:

- Cực đại sinh ra ~ 850g lực nâng/motor
- Dòng tiêu thụ ~13-15A/motor

⇒ Đạt yêu cầu lực nâng an toàn.

- Cánh quạt 1045R

- Dài 10inch, pitch 4.5 inch => hợp lý cho khả năng sinh lực nâng 750-900g khi kết hợp A2212

- Khung F450

- Khoảng cách giữa các motor => đủ khớp với cánh 10 inch (25.4cm) => không chồng lấn nhau
- Chống chọi tốt tới 1.5 – 5kg

- Tính toán pin:

Tổng dòng tiêu thụ:

- Mỗi motor ~15A khi toàn tải
- 4 motor ~ 60A tổng

Pin:

- Dùng 2 pin 3S 2200mAh 75C
- Tốc độ xả: $2200mAh \times 75 = 165A$ (dư sức cung cấp)
- Mỗi pin cung cấp 2 motor => mỗi pin cung ~ 30A, hoàn toàn đủ.

- Tính chọn ESC:

- Cơ sở lựa chọn ESC:
- Chịu được dòng tối đa mà motor tiêu thụ khi hoạt động toàn tải.
- Tương thích với loại pin (3S = 11.1V) và loại motor brushless không chổi than.

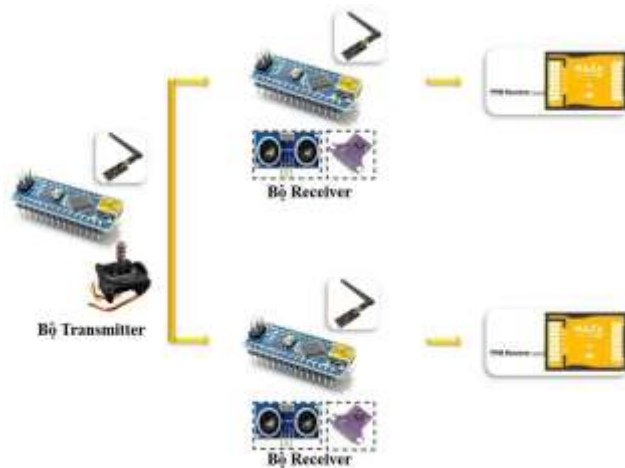
- Dòng tiêu thụ tối đa của motor: $\sim 13\text{--}15\text{A} \rightarrow$ nên sử dụng $\text{ESC} \geq 20\text{A}$

$\rightarrow$ Chọn ESC 30A vì dòng tối đa chịu được: 30A, gấp ~ 2 lần dòng thực tế motor sử dụng.

$\Rightarrow$ Đảm bảo mát, bền và an toàn. Tương thích hoàn toàn với motor A2212 và pin 3S. Dự phòng tốt cho các tình huống tăng ga đột ngột hoặc tải nặng.

3.2. Nguyên lý điều khiển

3.2.1. Phương pháp kết nối hệ Drone kép



Hình 3.1: Phương pháp kết nối – truyền tín hiệu cho hệ Drone kép

Mục đích: Điều khiển đồng thời hệ Drone kép để nâng tải

Sử dụng Arduino Nano để truyền – thu tín hiệu để điều khiển hệ Drone kép. Bao gồm 2 khối chính: khối truyền tín hiệu TX và khối thu nhận tín hiệu RX.

3.2.1.1. Thiết kế hệ truyền tín hiệu (TX)

Sử dụng Arduino Nano thực hiện nhiệm vụ truyền tín hiệu điều khiển. Thiết kế một thiết bị tay cầm điều khiển trên Arduino, kèm theo một thiết bị thu – phát.

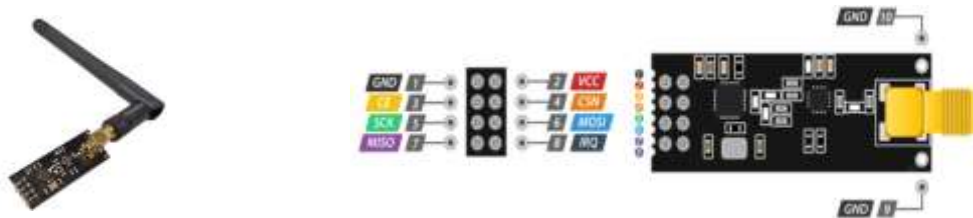
Arduino Nano này đóng vai trò là một khối Transmitter, có thiết kế 2 nút điều khiển thanh ga và hướng của Drone. Đồng thời, nắm giữ vai trò đưa tín hiệu Setpoint, để truyền tín hiệu đến các Arduino khác (vai trò là Receiver) qua tín hiệu RF.

3.2.1.2. Thiết kế hệ nhận tín hiệu (RX)

Sử dụng Arduino Nano được đặt trực tiếp trên mỗi Drone. Các Arduino Nano cũng kèm theo thiết bị thu – phát, qua tín hiệu tần số vô tuyến (RF) được nhận từ Arduino (TX), Arduino thực hiện các nhiệm vụ xử lý, điều khiển, hiệu chỉnh theo mong muốn của đề tài.

Ở khối nhận tín hiệu RX, sử dụng Arduino Nano, thiết kế bộ điều khiển PID, điều khiển các thông số mong muốn trong hệ thống. Tín hiệu được truyền đến bộ xử lý cân bằng Naza (được gắn trên Drone) bằng tín hiệu PPM.

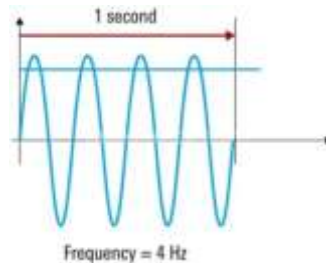
- Sóng vô tuyến RF



Hình 3.2: Thiết bị thu – phát sóng vô tuyến RF

Sóng RF (RadioFrequency) là một dạng sóng điện từ, có tần số nằm trong khoảng từ 3 kHz đến 300 GHz được cho là thấp nhất trong dãy quang phổ. Sóng Radio Frequency là nền tảng của nhiều ngành công nghệ hiện đại, từ truyền thông không dây đến các ứng dụng công nghiệp và thẩm mỹ.

Cấu tạo của sóng Radio Frequency bao gồm các thành phần cơ bản như tần số, biên độ, và pha. Tần số quyết định tính chất của Radio Frequency, ảnh hưởng đến khả năng truyền tải và phạm vi ứng dụng.



- Đặc điểm của sóng RF

Sóng RF truyền được bao xa: Chúng có thể truyền đi trong không gian và vượt qua các vật cản như tường, cây cối, tùy thuộc vào tần số của nó.

Ổn định: Radio Frequency có thể duy trì tín hiệu ổn định trên khoảng cách xa hàng nghìn ki lô mét (dải tần số càng cao thì càng ổn định) mà không bị ảnh hưởng bởi nhiễu hoặc mất mát tín hiệu.

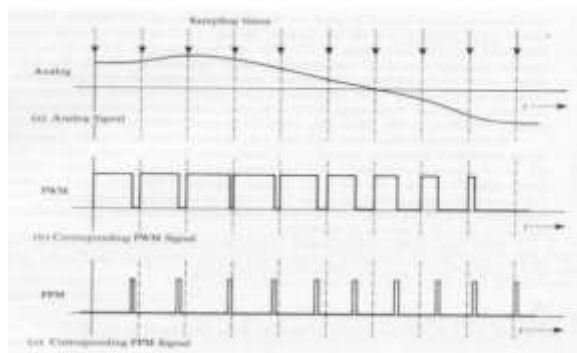
Khả năng điều khiển: Sóng Radio Frequency rất dễ dàng điều khiển và điều chỉnh theo nhu cầu sử dụng, phù hợp với nhiều ứng dụng khác nhau.

- Tín hiệu PPM

Thực chất, tín hiệu PPM cũng tương tự với tín hiệu PWM. Tín hiệu PWM và PPM là hai giao thức thu sóng vô tuyến được sử dụng để truyền dữ liệu từ RX đến bộ điều khiển bay.

Loại tín hiệu	PWM (Pulse Width Modulation – Điều chế độ rộng xung)	PPM (Pulse Position Modulation – Điều chế vị trí xung)
Chức năng	PWM là một kỹ thuật được sử dụng để chuyển tiếp dữ liệu dưới dạng độ rộng xung thay đổi.	Các giá trị mẫu tương tự xác định vị trí của xung hẹp so với thời gian xung nhịp.
Thiết bị RC sử dụng	- Servo, Bộ điều khiển tốc độ điện tử - Máy thu R/C, Bộ ghi dữ liệu - Hệ thống lái tự động/ổn định - Bộ điều khiển Servo	- Máy phát R/C, Máy thu R/C - Hệ thống lái tự động/ổn định - PCTx

Bảng 3.1: So sánh tín hiệu PWM và PPM



- PWM – Điều chế độ rộng xung

Trong PWM, mỗi kênh RC có cáp riêng. Nếu chúng ta muốn có 9 kênh, chúng ta phải ghi cả 9 cáp cùng với nguồn và đất. Giá trị của mỗi kênh được biểu thị dưới dạng tín hiệu “BẬT” từ 1ms đến 2ms và tín hiệu này lặp lại (hoặc cập nhật) sau mỗi 20ms.

Nó lên mức cao trong 1-2ms, sau đó giảm xuống mức Thấp. Khoảng thời gian nó ở mức cao là giá trị cho kênh đó. Toàn bộ khung hình dài 20ms, phần quan trọng của xung là thời gian xung bật; 1-2ms. Mặc dù thời gian giữa các xung không quan trọng bằng nhưng nó đóng vai trò quan trọng. Thông thường, giữ thời gian giữa các xung khoảng 20ms là tốt nhất.

- PPM – Điều chế vị trí xung

Hãy xem PPM như một số tín hiệu PWM được xếp hàng nối tiếp nhau. Trong PPM, cùng một tín hiệu được sử dụng nhưng mỗi kênh được gửi liên tiếp, sau đó là độ trễ, sau

đó nó lặp lại kênh 1. Trong PWM thông thường, có 50 bản cập nhật được gửi mỗi giây (50Hz), nghĩa là mỗi bản cập nhật mất 20ms.

3.2.2. Phương pháp điều khiển hệ Drone kép

- Hệ thống thiết kế gồm: một Drone Master và một Drone Slave. Sử dụng kiểu giao tiếp truyền thông Master – Slave để kết nối và điều khiển 2 Drone. Điều này có nghĩa:

Một Drone giữ nhiệm vụ nhận tín hiệu từ tay cầm điều khiển (bộ truyền tín hiệu TX) thông qua bộ thu – phát RF, Drone này đóng vai trò điều khiển, ra lệnh cho các thiết bị khác. Gọi là Drone Master.

Một Drone thực hiện nhiệm vụ nhận tín hiệu từ Drone Master và đợi Drone Master điều khiển. Gọi là Drone Slave.

3.2.2.1. Mô hình truyền thông Master – Slave

Master là một thiết bị có vai trò điều khiển các thiết bị khác (ví dụ như Slave). Các thiết bị Master sẽ có toàn quyền ghi lại tất cả các bản cập nhật dữ liệu. Ngoài ra, tất cả các dữ liệu khác sẽ được đồng bộ với cơ sở dữ liệu chính. Có thể nói, đây là thiết bị có toàn quyền điều khiển, ra lệnh cho các thiết bị khác.

Slave là một thiết bị được điều khiển bởi thiết bị khác có tên là Master. Các thiết bị Slave sẽ sử dụng các bản cập nhật thông tin được lưu trong hệ thống thông tin dữ liệu Master để đồng bộ với dữ liệu của thiết bị Master.

Mô hình Master – Slave này giúp cho hoạt động truyền thông tin sẽ được diễn ra liên tục và chính xác mà không bị can thiệp bởi bất kỳ thiết bị nào bên ngoài.

- Cách hoạt động của Master – Slave

Trong một mô hình Master-Slave, việc trao đổi là 1 chiều. Chỉ có thiết bị Master mới có thể liên lạc với các Slave, còn Slave không thể chủ động liên lạc với Master. Các thiết bị Slave sẽ thực hiện yêu cầu mà Master đưa ra. Trong mô hình nhiều Slave khác nhau, Master sẽ đánh địa chỉ cho từng Slave riêng biệt để phân biệt và liên lạc khi cần thiết.

- Ưu và nhược điểm của Master – Slave

Ưu điểm của mô hình này là cho phép Master kiểm soát hoàn toàn việc vận hành của hệ thống, điều này giúp quá trình chia sẻ tài nguyên được đơn giản hóa và hạn chế xảy ra việc trao đổi thông tin trái phép.

Nhược điểm là chỉ cho phép kết nối theo hướng 1 chiều. Các thiết bị Slave không thể chủ động liên lạc với Master. Và nếu như Drone Master bị lỗi, thì Drone Slave cũng sẽ không nhận được tín hiệu từ bộ truyền tín hiệu TX.

3.2.2.2. *Giao tiếp giữa các Arduino*

- Trong bộ truyền tín hiệu (TX), bao gồm 2 Arduino (Arduino 1 – Arduino 2).
 - Arduino 1: Truyền tín hiệu setpoint đến Arduino 3 và Arduino 5 qua tín hiệu RF.
 - Arduino 2: Nhận tín hiệu phản từ Arduino 4 và Arduino 6 qua tín hiệu RF và hiển thị về LCD.
 - Arduino 1 và Arduino 2 giao tiếp với nhau theo chuẩn UART.
- Trong bộ nhận tín hiệu (RX):
 - Drone Master gồm 2 Arduino (Arduino 3 – Arduino 4).
 - Arduino 3: Nhận tín hiệu từ Arduino 1 qua tín hiệu RF, chuyển đổi thành xung PPM đưa vào mạch Naza.
 - Arduino 4: Đọc dữ liệu từ cảm biến độ cao và cảm biến vị trí, giao tiếp với Arduino 3 theo UART và phản hồi về Arduino 2.
 - Drone Slave gồm 2 Arduino (Arduino 5 – Arduino 6).
 - Arduino 5: Nhận tín hiệu từ Arduino 1 qua tín hiệu RF, chuyển đổi thành xung PPM đưa vào mạch Naza.
 - Arduino 6: Đọc dữ liệu từ cảm biến độ cao và cảm biến vị trí, giao tiếp với Arduino 5 theo UART và phản hồi về Arduino 2.
- Chuẩn giao tiếp UART

Giao thức UART là một giao thức đơn giản và phổ biến, bao gồm hai đường truyền dữ liệu độc lập là TX (truyền) và RX (nhận).

Dữ liệu được truyền và nhận qua các đường truyền này dưới dạng các khung dữ liệu (data frame) có cấu trúc chuẩn, với một bit bắt đầu (start bit), một số bit dữ liệu (data bits), một bit kiểm tra chẵn lẻ (parity bit) và một hoặc nhiều bit dừng (stop bit).

3.3. **Linh kiện phần động lực**

- Khung Drone Flane Wheet 450 (F450)
 - Số lượng: 2

- Vật liệu được sử dụng làm khung là Flare Wheel 450 (F450), khung có tổng trọng lượng 282g, chiều dài đường chéo 450mm, tổng trọng lượng cất cánh 800-1600g.
- Động cơ phù hợp sử dụng cho khung là các động cơ: 2212 – 1000KV/ 2212 – 1400KV/ 2216 – 880KV/ 2216-1100KV/ 2216-1250KV, ESC: 30A/ 40A.
- Động cơ không chổi than A2212/10T 1400KV
 - Số lượng: 8 động cơ
- Mạch điều tốc (ESC) 30A cho động cơ không chổi than
 - Số lượng: 8
 - Bộ điều khiển tốc độ điện tử (ESC): ESC chịu trách nhiệm kiểm soát tốc độ động cơ dựa trên các lệnh nhận được từ bộ điều khiển máy bay. Chúng điều chỉnh công suất đầu ra của động cơ, đảm bảo chuyển bay trơn tru và ổn định.
- Cánh quạt
 - Số lượng: 8 cánh
 - Cánh quạt 2 cánh được sử dụng rộng rãi cho các Drone, cánh quạt được chọn cũng phải phù hợp với động cơ. Với kích thước là 90mm, loại 1045.
- Pin Ovonix 2200mAh 3S-1P (11.1V) 75C jack XT60
 - Số lượng: 4
- Bộ điều khiển tự cân bằng Naza – M Lite V2
 - Số lượng: 2 bộ
 - Bao gồm:
 - Mạch cân bằng Naza – M Lite (Main Controller)

Mạch cân bằng Naza M lite là loại mạch cân bằng phổ thông tốt bậc nhất hiện nay với nhiều tính năng nổi trội như giữ vị trí, giữ độ cao, bảo vệ nguồn yếu, tự quay về, ... Mạch cân bằng Naza có thể dùng được cho nhiều cấu hình máy bay như X4, +4, hexa, delta.

- Module nguồn PMU

Cung cấp điện ổn định cho hệ thống: PMU lấy nguồn từ pin chính (thường là pin LiPo 3S/4S) và chuyển đổi để cung cấp điện áp ổn định cho các thành phần.

Giám sát điện áp – dòng điện (bảo vệ điện): PMU theo dõi điện áp và dòng điện đầu vào.

Mở rộng chức năng điều khiển: PMU có vai trò mở rộng tính năng bằng cách kết nối các phụ kiện thông minh khác.

- Module GPS + Compass

Bộ thu GPS: Máy thu GPS cung cấp dữ liệu về vị trí địa lý của máy bay không người lái, cho phép các tính năng như giữ vị trí, trở về nhà và điều hướng điểm tham chiếu.

La bàn: La bàn cung cấp thông tin về hướng của máy bay không người lái, cho phép nó duy trì một hướng cụ thể trong suốt chuyến bay và điều hướng chính xác.

- Module LED

Thông báo trạng thái hoạt động của hệ thống.

Cảnh báo lỗi: GPS hỏng, La bàn lỗi, Gyro mất tín hiệu.

3.4. Linh kiện phần điều khiển

- Arduino Nano V3.0 ATmega328P
 - Số lượng: 6 arduino
- Cảm biến siêu âm HC-SR04
 - Số lượng: 2
 - Khi phát tín hiệu từ việc điều khiển trên Arduino (TX), vì khi điều chỉnh thanh ga độ cao của 2 Drone sẽ có sự chênh lệch, chúng ta không thể biết chính xác được.
 - Cảm biến siêu âm sử dụng sóng siêu âm đo khoảng cách của từng Drone so với mặt đất. Từ đó, cho phép điều chỉnh các thông số từ hệ Arduino (RX).
- Cảm biến vị trí Optical Flow PMW3901
 - Số lượng: 2
 - Các cảm biến này giúp duy trì trạng thái lơ lửng ổn định bằng cách cung cấp dữ liệu chính xác về vị trí và độ cao của máy bay không người lái.

3.4.1. Chi phí đầu tư

Loại linh kiện	Giá thành (nghìn đồng)	Số lượng	Thành tiền (nghìn đồng)
Bộ xử lý Naza V2	3500	2	7000

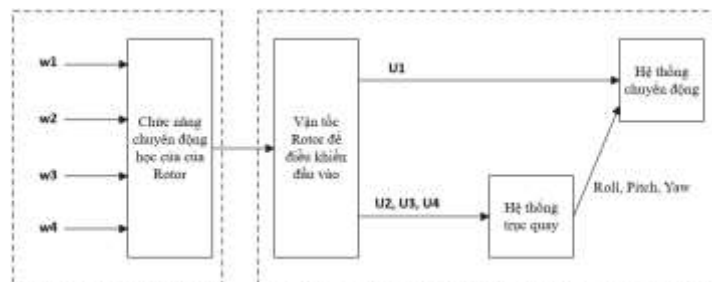
Khung Drone F450	180	2	360
Động cơ không chổi than A2212/10T 1400KV	90	8	720
Bộ điều tốc ESC 30A	100	8	800
Cánh quạt 1045	>18	8	150
Pin Ovonix 2200mAh 3S-1P (11.1v) 75C jack XT60	440	4	1760
Arduino Nano V3.0 Atmega328	75	6	450
Cảm biến siêu âm HC-SR04	22	2	44
Cảm biến Optical Flow PMW3901	210	2	420

Bảng 3.2: Bảng chi phí linh kiện

- Tổng chi phí: 11.704.000 (đồng)

3.5. MÔ HÌNH ĐIỀU KHIỂN CHO HỆ DRONE KÉP

Ta biểu diễn mô hình toán học mở của Drone bao gồm 2 hệ thống con là hệ chuyển động quay và hệ chuyển động tịnh tiến. Tốc độ góc của motor $\omega_1, \omega_2, \omega_3, \omega_4$ qua các chuyển động lực học và tốc độ sẽ cho các tác động điều khiển U_1, U_2, U_3, U_4 .



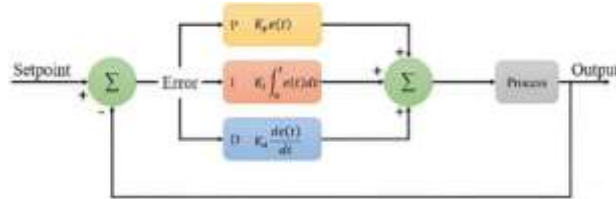
Hình 3.3: Mô hình mô phỏng tổng quát của việc điều khiển hệ thống Drone

3.5.1. Bộ điều khiển PID cơ bản

Để hệ thống hoạt động ổn định, bộ điều khiển đóng vai trò then chốt, tương tự như bộ não – tiếp nhận tín hiệu phản hồi từ hệ thống, so sánh với giá trị đặt (setpoint) và xuất tín hiệu điều khiển phù hợp. Tùy theo đặc tính mô hình và yêu cầu điều khiển, ta lựa chọn thuật toán điều khiển phù hợp như PID, LQR, LQG, điều khiển trượt... Mỗi loại có ưu nhược điểm riêng.

Trong số đó, bộ điều khiển PID (Proportional – Integral – Derivative) vẫn được ứng dụng rộng rãi nhờ tính đơn giản, hiệu quả và dễ triển khai. PID giúp cải thiện chất lượng

điều khiển như: rút ngắn thời gian quá độ, giảm sai lệch tĩnh, hạn chế quá điều chỉnh và nâng cao độ ổn định. Tín hiệu điều khiển đầu ra là tổng hợp của ba thành phần: tỉ lệ (P), tích phân (I) và vi phân (D). Bộ PID số thường được triển khai trong mạch vòng kín để điều khiển chính xác, ví dụ như điều chỉnh tốc độ của 4 động cơ trong hệ quadcopter.



Hình 3.4: Sơ đồ tổng quát của một bộ điều khiển PID cơ bản

Ta có hệ phương trình tổng quát:

$$Output = P_{out} + I_{out} + D_{out} \quad (2.89)$$

Trong đó, các thành phần là các thành phần đầu ra từ ba khâu trong một bộ điều khiển PID.

Trong đó:

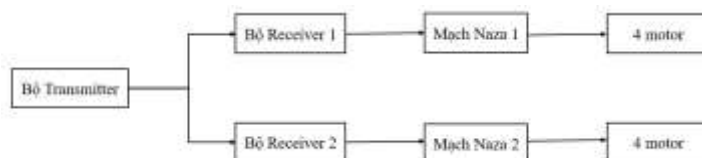
- P_{out} : là thành phần đầu ra khâu tỉ lệ.
- I_{out} : là thành phần đầu ra khâu tích phân.
- D_{out} : là thành phần đầu ra khâu vi phân.

Các cảm biến ứng dụng trong hệ thống:

- Cảm biến siêu âm HC-SR04: Cảm biến sử dụng sóng siêu âm để đo khoảng cách trong khoảng từ 2 → 450 cm của drone so với mặt đất.
- Cảm biến Optical Flow: hỗ trợ định vị và giữ vị trí theo 2 trục không gian (x, y) để làm cho drone ít bị nhiễu bởi các tác động môi trường bên ngoài.

3.5.2. Thiết kế bộ điều khiển PID cho hệ Drone kép

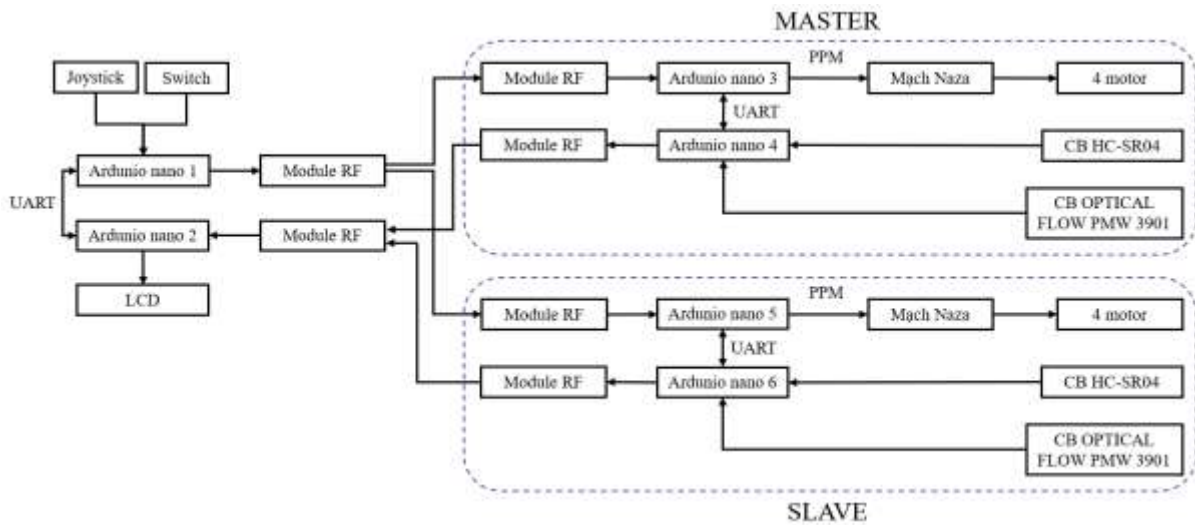
3.5.2.1. Sơ đồ tổng quan của hệ thống điều khiển



Hình 3.5: Sơ đồ tổng quan của hệ thống

Tổng quan vận hành hệ thống: bộ điều khiển sẽ điều khiển 2 drone vận hành giống nhau để cùng nâng tải cân bằng.

3.5.2.2. Sơ đồ chi tiết của hệ thống điều khiển



Hình 3.6: Sơ đồ chi tiết của hệ thống

- Quá trình vận hành hệ thống: hệ thống sẽ có 1 drone master (ardunio 3 và 4) và 1 drone slave (ardunio 5 và 6).

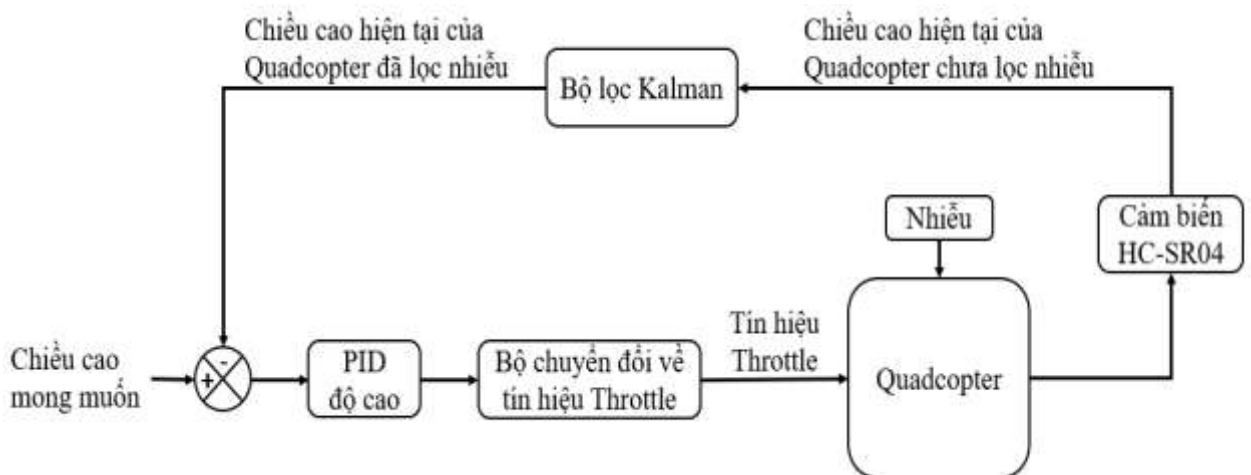
Hai joystick phát sinh tín hiệu điều khiển sẽ được Arduino 1 tiếp nhận và mã hóa thành tín hiệu RF, rồi truyền qua module RF phát. Ở phía nhận, Arduino 3 và Arduino 5 sẽ sử dụng module RF thu để nhận tín hiệu này. Sau khi nhận, tín hiệu sẽ được chuyển đổi thành xung PPM để đưa vào bộ điều khiển trung tâm Naza, từ đó điều khiển hoạt động của 4 động cơ trên mỗi drone. Ở đây drone 2 sẽ bay theo cùng với độ cao của drone 1.

- Khi hai drone cất cánh bắt đầu kéo tải tới một độ cao mong muốn ta sẽ gạt switch để tạo setpoint độ cao, 2 drone sẽ nâng tải giữ ở mức setpoint đó, trong đó:
 - Cảm biến siêu âm HC-SR04 sẽ đo khoảng cách từ drone đến mặt đất, giúp xác định độ cao bay.
 - Cảm biến optical flow sẽ khoá vị trí (x, y) để làm cho drone ít bị nhiễu bởi các tác động môi trường bên ngoài.
- Các cảm biến này được lắp tương tự trên cả hai drone. Dữ liệu độ cao đo được từ cảm biến HC-SR04 sẽ được:
 - Drone 1: Dữ liệu cảm biến độ cao sẽ được Arduino 4 đọc và truyền về Arduino 3 (Arduino 3 và 4 giao tiếp UART), Arduino 3 sẽ thực hiện khâu PID để giữ drone ở mức setpoint; dữ liệu đọc về của Arduino 4 cũng sẽ được truyền về Arduino 2 (sóng RF) để hiển thị ra LCD.

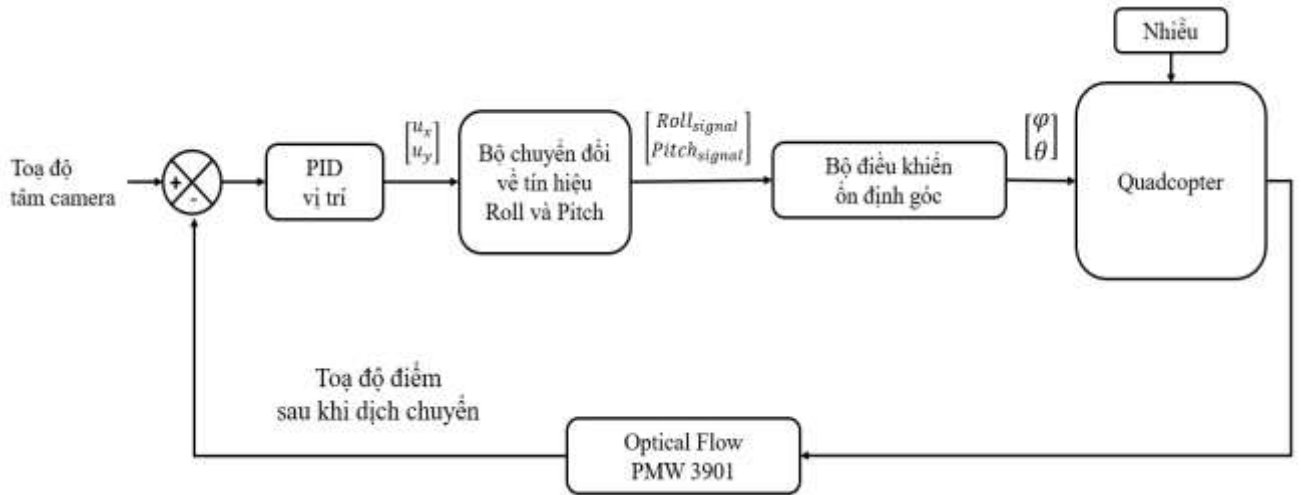
- Drone 2: Dữ liệu độ cao từ drone 1 sẽ được Arduino 2 truyền sang Arduino 1 (UART), từ đó nó sẽ đẩy tín hiệu tới Arduino 5 (Sóng RF) để điều khiển drone 2 theo độ cao drone 1, cảm biến độ cao phản hồi dữ liệu khi bay trả về Arduino 6, A6 đưa giá trị độ cao đó qua A5 để thực hiện khâu PID đáp ứng sao cho drone 2 bay ổn định theo drone 1. Và dữ liệu độ cao hiện tại của drone 2 sẽ được hiển thị về LCD.
- Dữ liệu vị trí x, y đo được từ cảm biến optical flow PMW 3901:
 - Drone 1: Dữ liệu cảm biến vị trí sẽ được Arduino 4 đọc và truyền về Arduino 3, Arduino 3 sẽ thực hiện khâu PID để giữ drone đứng yên không bị trôi khi có tác động nhiễu từ môi trường bên ngoài.
 - Drone 2: Dữ liệu cảm biến vị trí sẽ được Arduino 6 đọc và truyền về Arduino 5, Arduino 5 sẽ thực hiện khâu PID để giữ drone đứng yên không bị trôi khi có tác động nhiễu từ môi trường bên ngoài.
- Ưu điểm của hệ thống:
 - Dễ điều khiển, drone slave dễ đáp ứng theo drone master.
- Nhược điểm của hệ thống:
 - Nếu drone master bị lỗi thì drone slave cũng sẽ lỗi theo hệ thống không hoạt động.

3.5.2.3. Các bộ PID của các cảm biến

Hệ thống sẽ có 2 bộ PID chính, 2 bộ PID của 2 cảm biến (CB HC-SR04, CB Optical Flow PMW 3901).

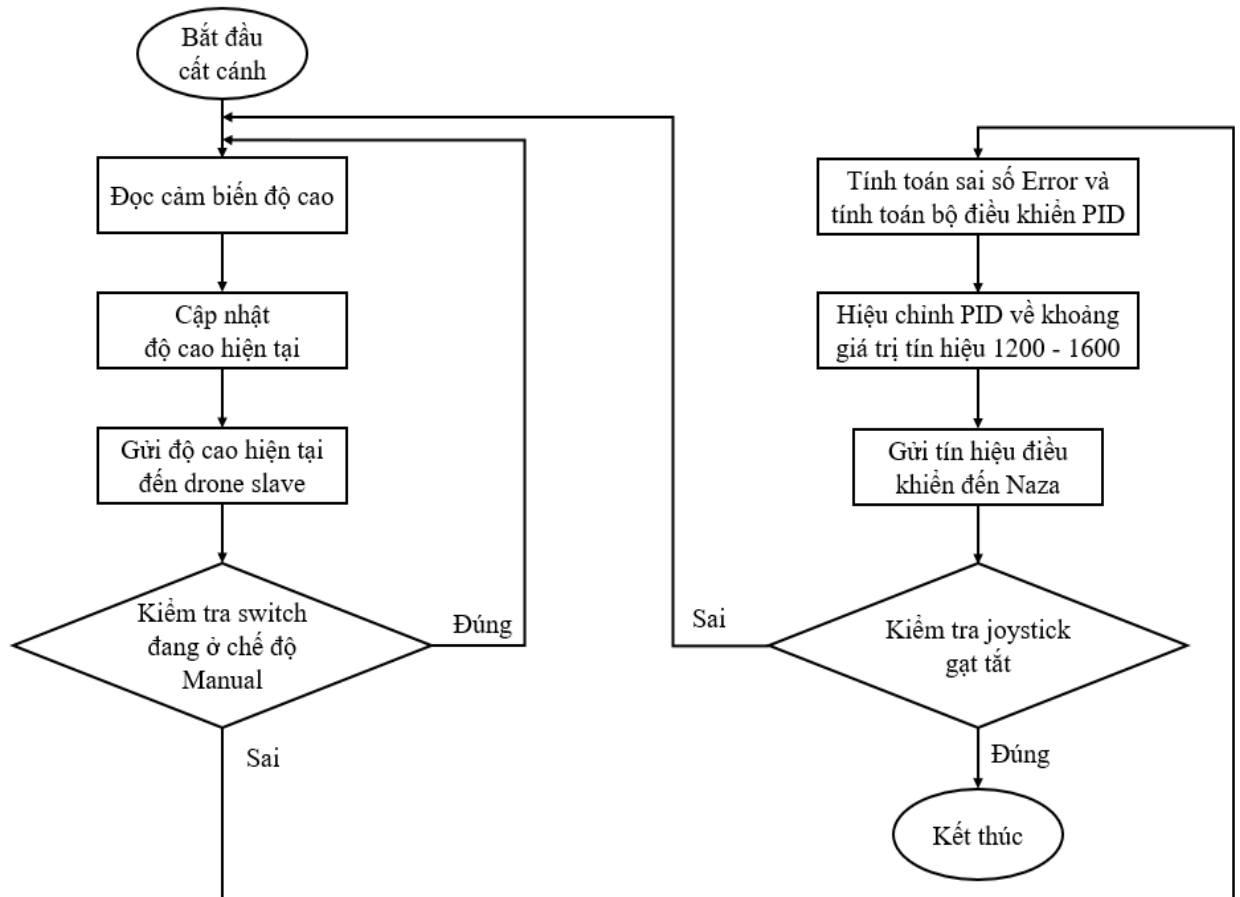


Hình 3.7: Sơ đồ khối điều khiển PID cho cảm biến HC-SR04

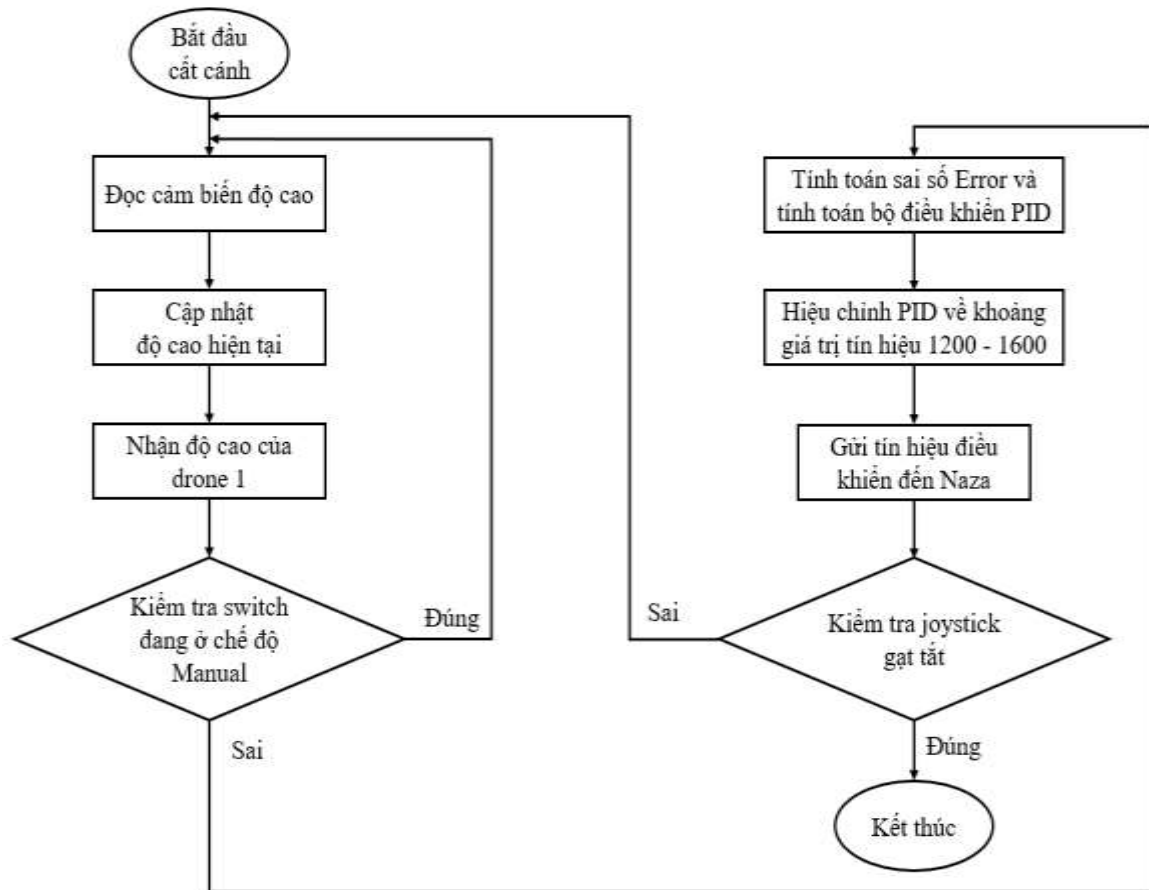


Hình 3.8: Sơ đồ khối điều khiển PID cho cảm biến Optical Flow

3.5.3. Lưu đồ thuật toán



Hình 3.9: Lưu đồ thuật toán của bộ điều khiển PID giữ độ cao cho Drone 1



Hình 3.10: Lưu đồ thuật toán của bộ điều khiển PID giữ độ cao cho Drone 2

3.6. Kết luận chương

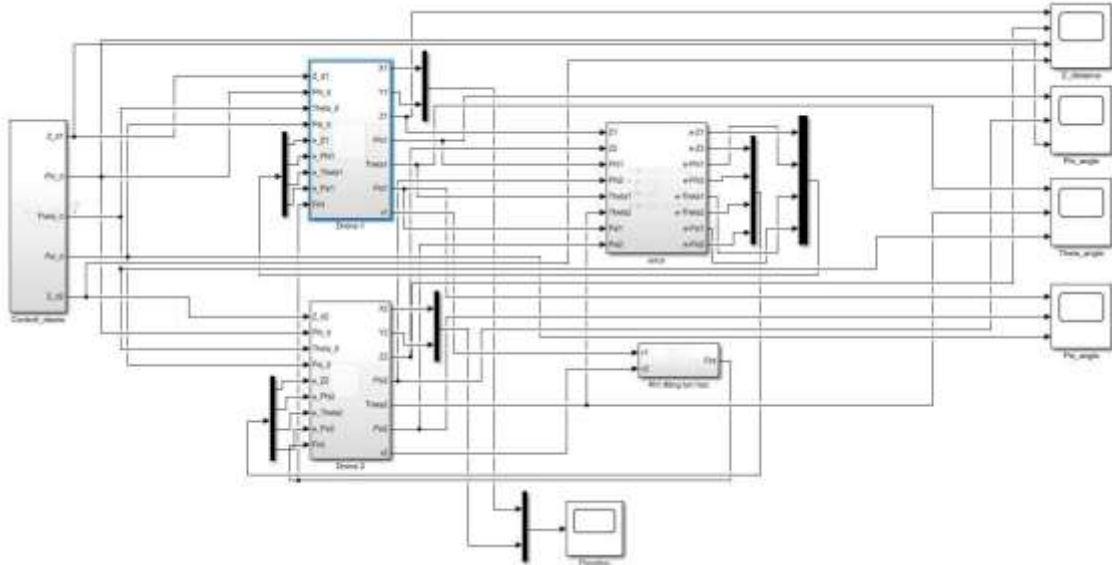
Trong chương 3, thông qua việc tìm hiểu các trạng thái làm việc của Drone cơ bản và ứng dụng cho việc xây dựng trạng thái làm việc cho Drone kép ở chương tiếp theo. Bên cạnh đó, việc phân tích ưu và nhược cũng như khả năng ứng dụng, đồ án đã xác định mục tiêu nghiên cứu và tham khảo một số công trình trong và ngoài nước để xác định các vấn đề cơ bản cần giải quyết cho đề tài.

Qua đó, chương 3 đã nghiên cứu được phương pháp kết nối truyền thông cho hệ Drone kép, từ đó thiết kế được bộ điều khiển PID và suy ra các lưu đồ thuật toán biểu diễn cho hệ thống. Và quá trình thực nghiệm để hiệu chỉnh bộ PID được thực hiện ở chương tiếp theo.

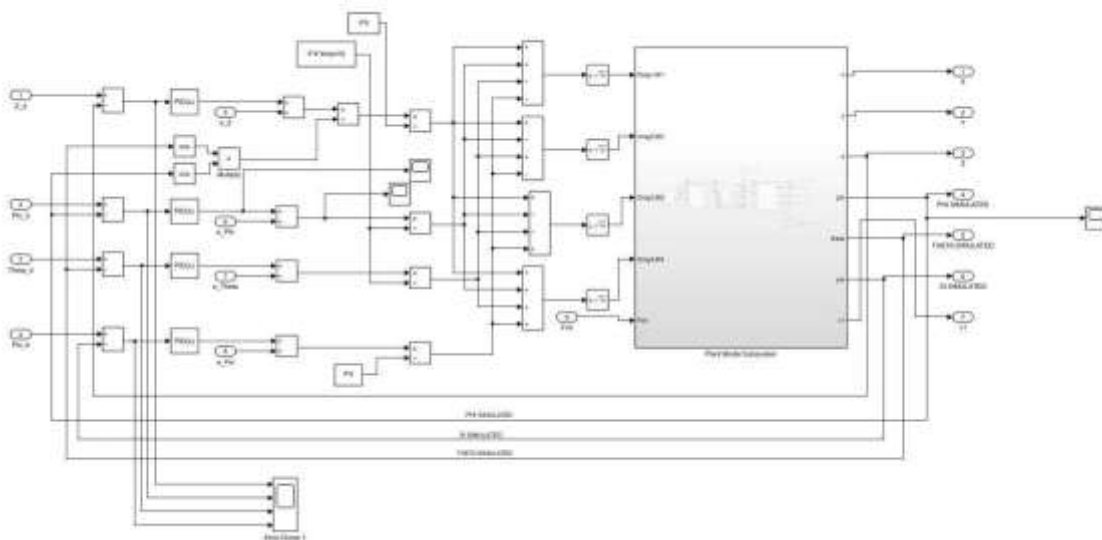
CHƯƠNG 4. MÔ PHỎNG, THỰC NGHIỆM VÀ ĐÁNH GIÁ

4.1. Kết quả mô hình

4.1.1. Mô hình mô phỏng trên Matlab/Simulink (môi trường không có nhiễu)



Hình 4.1: Mô hình mô phỏng Matlab/Simulink của hệ thống



Hình 4.2: Mô hình mô phỏng từng Drone

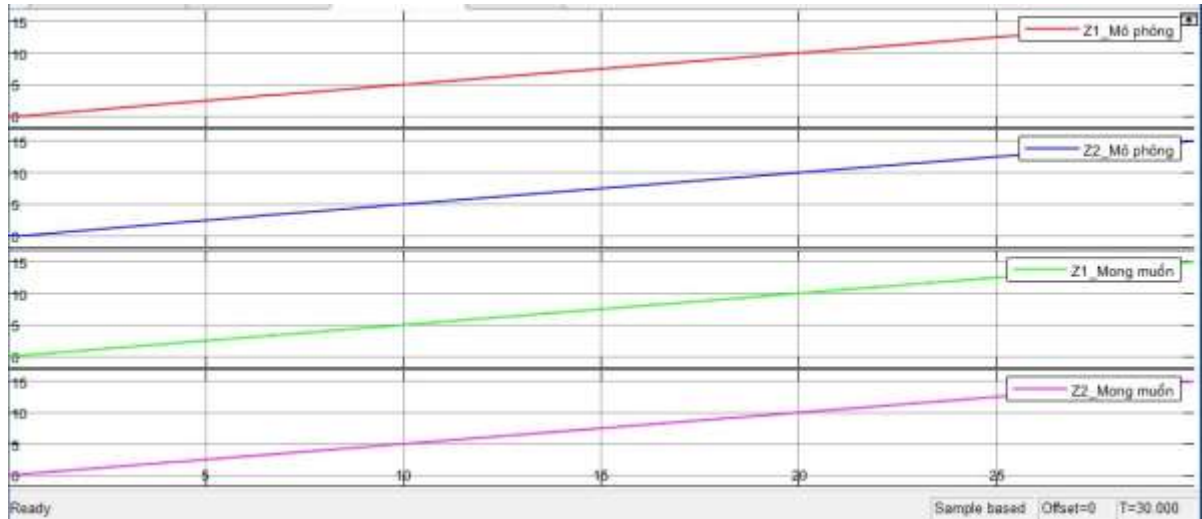
4.1.2. Kết quả mô phỏng trên Matlab/Simulink (môi trường không có nhiễu)

• Mô tả hoạt động

Hệ thống Drone kép đang đứng yên, sau đó khi đưa các tín hiệu điều khiển đầu vào bao gồm lực Z, và các góc Roll, Pitch, Yaw thông qua các khối điều chỉnh. Sau khi chạy

mô phỏng thì đồ án sẽ nhận xét tính ổn định bằng cách dựa vào sự so sánh kết quả mô hình của giá trị tạo ra với giá trị điều khiển mong muốn đưa vào.

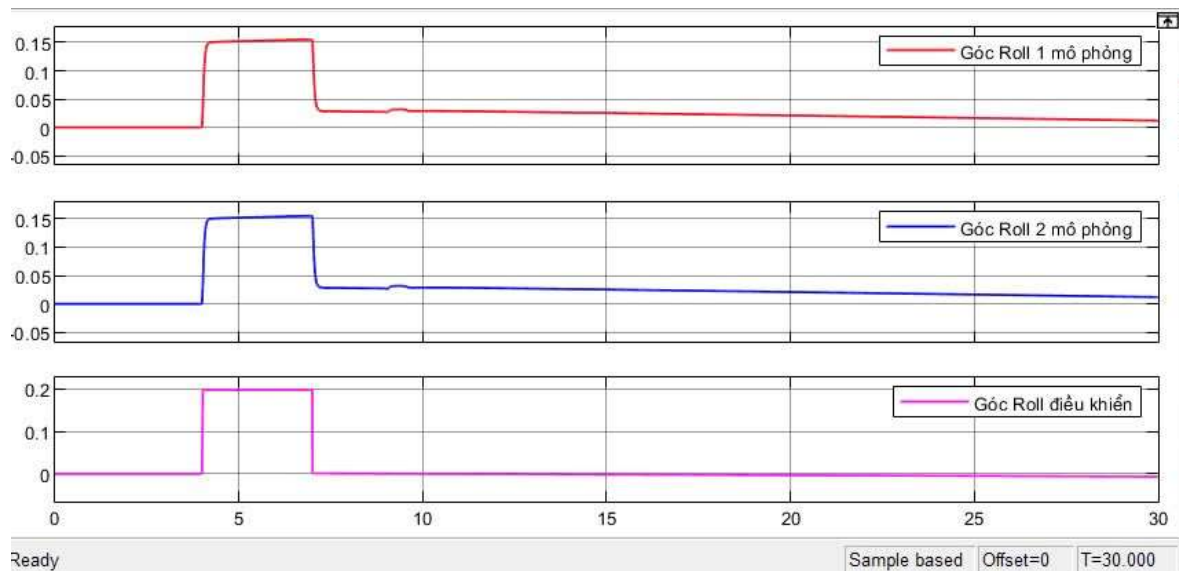
- Kết quả mô phỏng của lực nâng Z đầu vào và lực nâng Z thực tế



Hình 4.3: Lực nâng Z đầu vào và lực nâng Z thực tế của 2 Drone

Nhận xét: Ở cả 2 tín hiệu của Drone 1 và Drone 2, trong môi trường không có nhiễu, có nghĩa là điều kiện thời tiết ổn định, sóng, gió đều lý tưởng thì tín hiệu mô phỏng được từ hệ thống tương tự với tín hiệu mong muốn đưa vào.

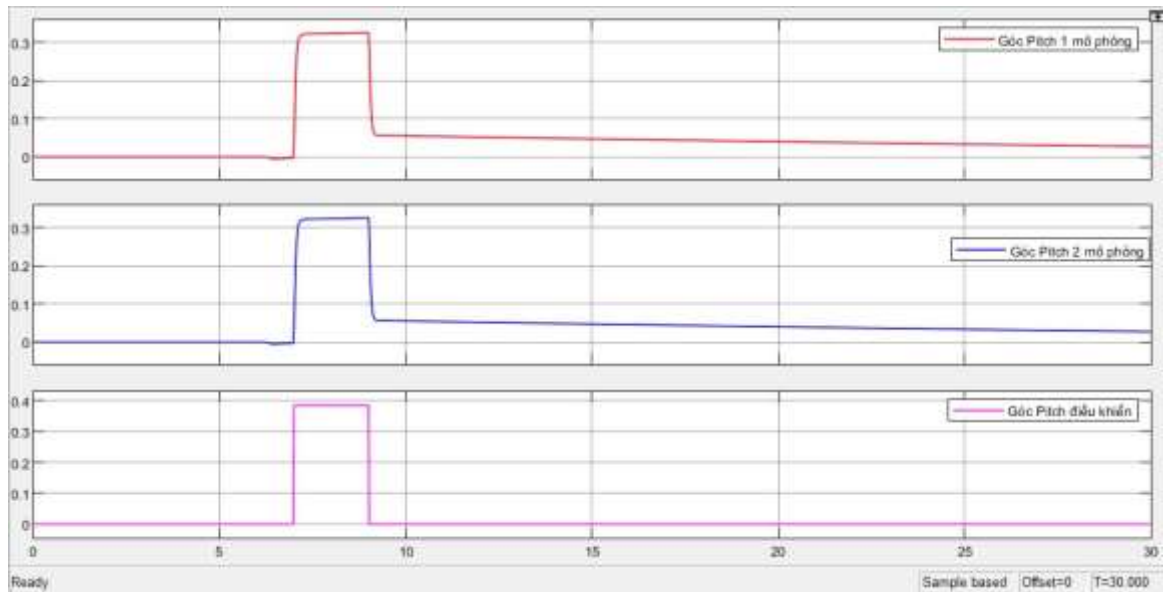
- Kết quả mô phỏng của giá trị góc Roll đầu vào và giá trị góc Roll thực tế



Hình 4.4: Giá trị góc Phi Φ

Nhận xét: Kết quả mô hình cho tín hiệu của góc Roll bám sát tín hiệu đầu vào, cho thấy hệ thống Drone kép khá ổn định cho giai đoạn sau.

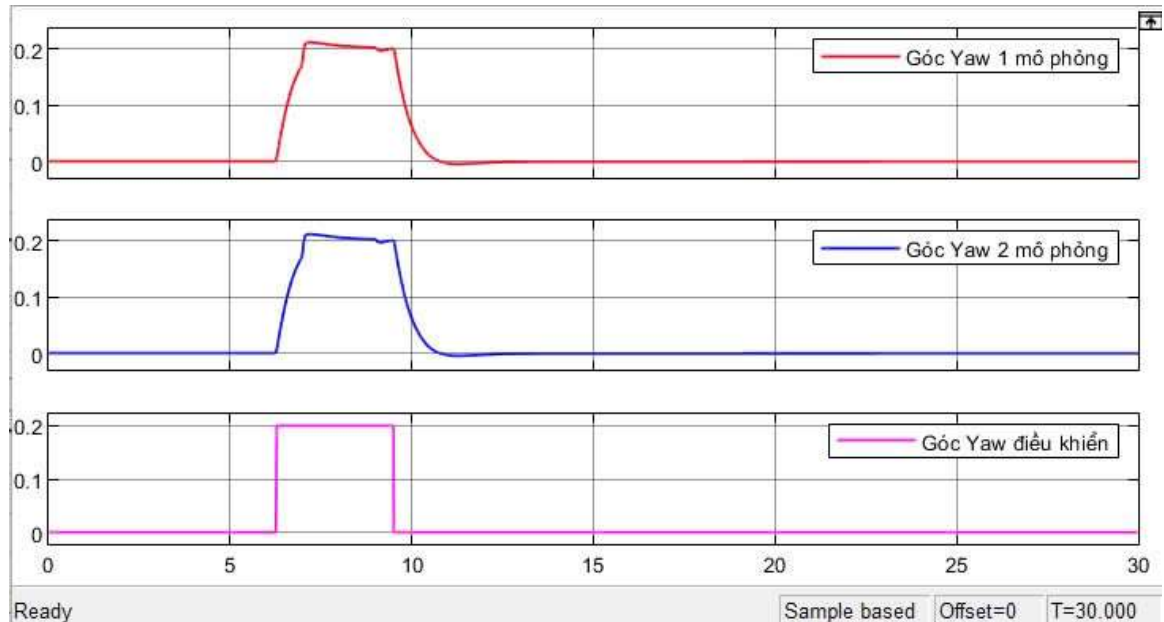
- Kết quả mô phỏng của giá trị góc Pitch đầu vào và giá trị góc Pitch thực tế



Hình 4.5: Giá trị góc Theta θ

Nhận xét: Kết quả mô hình cho tín hiệu của góc Pitch mô phỏng sát với tín hiệu điều khiển mong muốn, chỉ có xuất hiện nhiễu ở giai đoạn đầu khi phát xung.

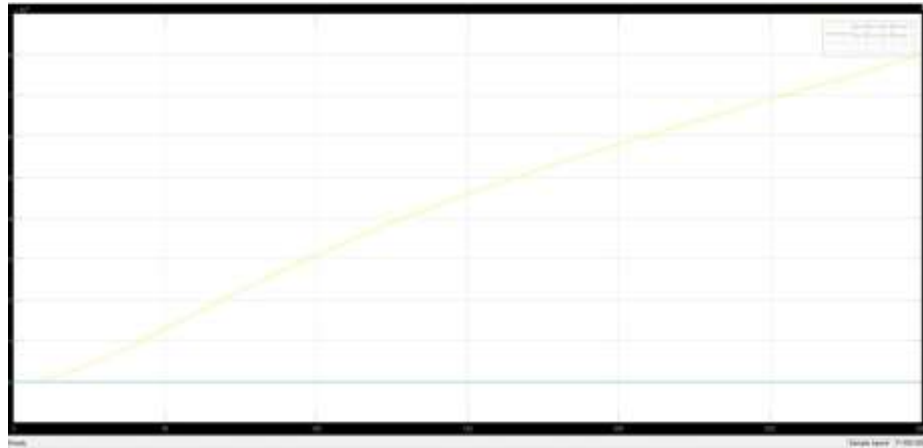
- Kết quả mô phỏng của giá trị góc Yaw đầu vào và giá trị góc Yaw thực tế



Hình 4.6: Giá trị góc Psi Ψ

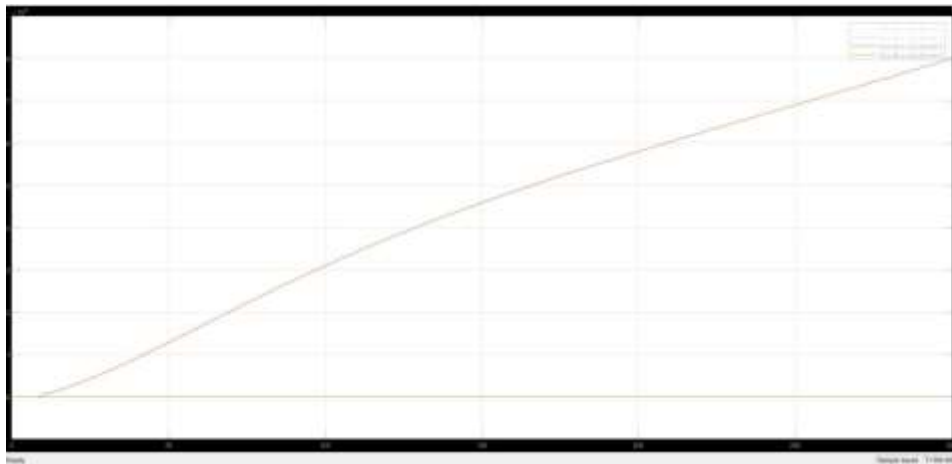
Nhận xét: Mô phỏng cho thấy có sự nhiễu khi kết quả lệch 1 chút so với tín hiệu điều khiển mong muốn đưa vào. Tuy nhiên, lộ trình tín hiệu tạo ra Yaw vẫn bám sát xung.

- Kết quả tọa độ của Drone 1



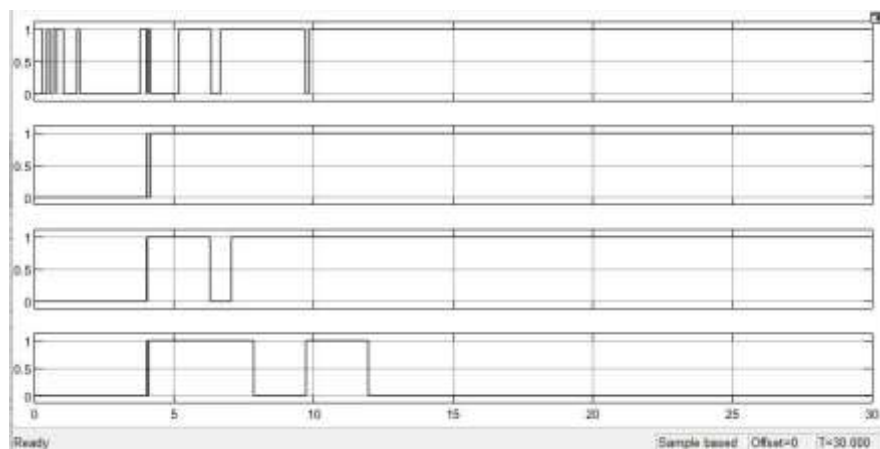
Hình 4.7: Tọa độ của Drone 1

- Kết quả của tọa độ Drone 2



Hình 4.8: Tọa độ của Drone 2

4.1.3. Kết quả của bộ sai lệch



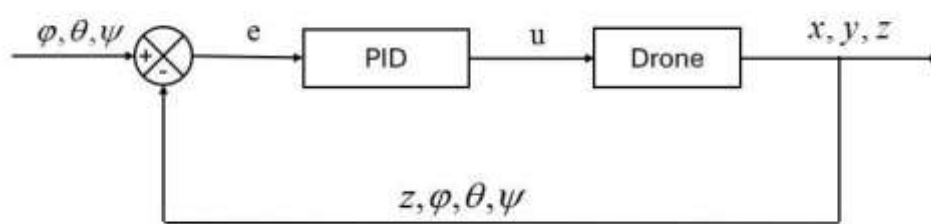
Hình 4.9: Kết quả mô phỏng bù trừ lẫn nhau của 2 thiết bị Drone khi $T=30$

Nhận xét:

- **Sai số lực nâng:** Đồ thị này tiếp tục hiển thị các dao động tuần hoàn với biên độ dao động lớn. Các dao động có tần số cao hơn, điều này có thể cho thấy rằng hệ thống điều khiển đang điều chỉnh lực nâng một cách thường xuyên hơn để duy trì độ cao mong muốn. Mặc dù vậy, hệ thống vẫn duy trì được sự ổn định tương đối.
- **Sai số góc Roll:** Đồ thị này cho thấy rằng sai số góc roll bắt đầu lớn, sau đó giảm về không và duy trì ở mức không trong phần lớn thời gian. Hệ thống điều khiển hoạt động rất tốt trong việc ổn định góc roll sau giai đoạn đầu, duy trì góc roll mong muốn với sai số gần như bằng không.
- **Sai số góc Pitch:** Đồ thị này cũng bắt đầu với một sai số lớn nhưng giảm về mức không trong phần lớn thời gian. Hệ thống điều khiển có khả năng ổn định góc pitch sau giai đoạn đầu, duy trì góc pitch mong muốn với sai số gần như bằng không.
- **Sai số góc Yaw:** Đồ thị này cho thấy các dao động không đều, với một số đoạn có biên độ dao động lớn và tần số cao hơn. Hệ thống điều khiển gặp một số khó khăn trong việc duy trì góc yaw ổn định, với các sai số thay đổi liên tục.
- **Nhận xét chung:** Mô hình đã mô phỏng các tín hiệu đầu ra gần như là bám sát với tín hiệu điều khiển mong muốn đưa vào. 4 tín hiệu điều khiển mong muốn đặt vào hệ thống vẫn được mô phỏng gần như là đúng ở điều kiện thuận lợi – không có bất kỳ nhiễu bên ngoài.

4.2. Quá trình thực nghiệm

4.2.1. Sơ đồ khối PID hệ thống



Hình 4.10: Sơ đồ khối PID của hệ thống

Tín hiệu đầu vào là các giá trị mong muốn về góc (ϕ, θ, ψ), được so sánh với giá trị thực tế để tạo ra sai số e . Bộ PID xử lý sai số này và tạo ra tín hiệu điều khiển u đưa vào drone. Hệ thống phản hồi giúp drone ổn định và đạt trạng thái bay ổn định theo mong muốn.

4.2.2. Thông số PID hiệu chỉnh qua quá trình thực nghiệm

- PID hiệu chỉnh độ cao của Drone 1

$$P = 1.5;$$

$$I = 0.01;$$

$$D = 0.3$$

- PID hiệu chỉnh độ cao của Drone 2

$$P = 1.2;$$

$$I = 0.02;$$

$$D = 0.3$$

4.2.3. Thực nghiệm bay mô hình



Hình 4.11: Bay thực tế cùng độ cao cho 2 Drone



Hình 4.12: Giá trị độ cao của 2 Drone phản hồi về LCD



Hình 4.13: Test bay thực tế có gắn thanh liên kết theo mô hình

KẾT LUẬN

Tổng thể đồ án, qua quá trình thực nghiệm, nhận thấy đồ án có tính khả thi và có thể phát triển được cho các mô hình có quy mô lớn hơn. Nhưng hiện đồ án vẫn đang có nhiều bất cập về việc hiệu chỉnh bộ điều khiển PID, thực nghiệm gặp khó khăn vì các yếu tố môi trường và đang hiệu chỉnh phân lập trình nên các thông số PID thành phần trong hệ thống vẫn chưa đạt được kết quả tối ưu cuối cùng.

Song, vẫn đảm bảo được việc kết nối và truyền thông giữa 2 Drone theo phương pháp mong muốn. Đặc biệt là, điều khiển được hệ Drone đạt độ cao theo giá trị Setpoint và phản hồi giá trị về màn hình. Bên cạnh đó, khi gắn thanh liên kết và tải thì hệ thống chưa thực sự ổn định theo mong muốn. Đồ án vẫn còn thiếu sót, nhưng vẫn bám sát theo mục tiêu thiết kế và cần có thêm thời gian thực nghiệm và hiệu chỉnh.

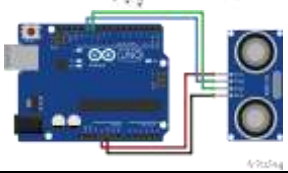
TÀI LIỆU THAM KHẢO

- [1] TS. Đỗ Quốc Tuấn, Th. S Nguyễn Đình Dũng và Th. S Phạm Hữu Ưông (2015), *Xác định các tham số trong mô hình toán của quadrotor*, Hội nghị toàn quốc lần thứ 3 về Điều khiển và Tự động hoá – VCCA.
- [2] Nhóm sinh viên K63AE (2019-2020), *Nghiên cứu, thiết kế, chế tạo Drone mang vật phẩm 1kg hạ cánh tự động*, Viện hàng không vũ trụ.
- [3] *Thiết kế và thi công mô hình bay quadcopter* - Khoa Cơ Điện - Điện Tử Trường Đại học Lạc Hồng, Đồng Nai, Việt Nam.
- [4] *Tạp chí khoa học công nghệ hành hải* – Journal of marine science and technology.
- [5] *Tạp chí Nghiên cứu KH&CN quân sự*, Số 60, 4 – 2019.
- [6] Nguyễn Cao Tuyên, “*Điều khiển cân bằng trên mô hình bay Quadcopter*”, Trường Đại học Lạc Hồng, 2013.
- [7] Lê Công Danh (2009), *Mô hình Quadcopter*, Khoa Cơ khí trường ĐH Giao thông vận tải TP. Hồ Chí Minh.
- [8] Nguyễn Doãn Phước (2007), *Lý thuyết điều khiển nâng cao*, Nhà xuất bản Khoa học và Kỹ thuật.
- [9] Nguyễn Doãn Phước (2012), *Phân tích và điều khiển hệ phi tuyến*, Nhà xuất bản Bách Khoa Khoa Hà Nội.
- [10] Phạm Quốc Vương (2012), *Nghiên cứu ứng dụng mô hình Quadrotor trong giám sát và cứu hộ*, Luận văn thạc sĩ, ĐHKT Công nghệ TP.HCM.

PHỤ LỤC 1

Tên linh kiện	Hình ảnh	Thông tin linh kiện
Khung Drone Flane Wheel 450 (F450)		- Khung cánh tay: PA66 + 30GF thiết kế vật liệu nhựa, các tay arm có độ đàn hồi nhất định. - Bản mạch PCB có độ bền cao: Làm cho hệ thống dây điện của ESC và pin an toàn hơn và dễ sử dụng hơn. - Tối ưu hóa thiết kế khung: Tích hợp mạch hàn PDB trên khung nhằm tiết kiệm diện tích khung - Màu sắc khác nhau khung Arms: Đỏ và trắng hoặc đen giúp dễ dàng phân biệt đầu đuôi.
Động cơ không chổi than A2212/10T 1400KV		- Nguồn là pin 2-3S Lipo - Tốc độ quay: 1400 vòng/phút/vôn - Dòng không tải (10V): 0.7A - Cường độ dòng lớn nhất: 16A - Hiệu suất tối đa: 78% - Cường độ dòng mang lại hiệu suất lớn nhất: 6-12A (hiệu suất đạt trên 75%) - Đường kính trục: 3.17mm - Công suất lớn nhất: 180W - Khối lượng: 50g - Điều khiển bằng ESC tối thiểu 20A (ESC là module điều khiển động cơ không chổi than)
Mạch điều tốc (ESC) 30A cho động cơ không chổi than		- Dòng cực đại: 30A - Dòng vượt tải: 35A (không để quá 10s sẽ làm cháy ESC) - BEC: 2A / 5V (Linear mode). - Điện áp vào: 5.6V - 16.8V. - Pin sử dụng: 2-3S. - Kích Thước: 32mm x 24mm x 7mm. - Trọng Lượng: ~20g.
Cánh quạt 1045		- Đường kính: 25.5cm - Màu sắc: Đen, Đỏ - Bề rộng cánh: khoảng 27mm - Độ dày đế trung tâm: 8MM - Đường kính lỗ: 6mm - [Trung tâm lỗ ngược bề mặt khẩu độ]: 9MM

		- [Nhóm vòng biến đổi]: x 6 (2 nhóm), vòng 3MM, 3.3MM, 4mm, 5mm, 6mm, 7.8mm - Trọng lượng: khoảng 9.3g (cái)
Pin Ovonc 2200mAh 3S-1P (11.1v) 75C jack XT60		- Số cell/ Số Volt: 3S1P / 11.1V / 3Cells - Tốc độ xả: 75C - Tốc độ xả cực đại: 150C - Trọng lượng: 174g - Kích thước: 107mm x 35mm x 24mm (L x W x H) - Jack sạc: JST-XHR - Jack xả: XT60
Bộ điều khiển tự cân bằng Naza – M Lite V2		
Mạch cân bằng Naza – M Lite (Main Controller)		- Hỗ trợ nhiều loại rotor: Quad – rotor I4, X4, Hex – rotor I6, X6, IY6, Y6. - Tần số ESC đầu ra: 400Hz. - Kênh phát đề nghị: PCM hoặc tối thiểu 4 kênh với tần số 2,4GHz. - Điện áp làm việc: Bo mạch Naza M Lite: 4,8V – 5,5V. Đầu vào VU: 7,2V – 26V (nên sử dụng pin 2S ~ 6S). - Nhiệt độ: -10°C ~ 50°C. - Phần mềm hỗ trợ các hệ điều hành: Window XP/7/8/10.
Module nguồn PMU		
Mudule GPS + Compass		

Module LED	 	
Arduino Nano V3.0 ATmega328P		- Điện áp hoạt động: 5V - Điện áp đầu vào (khuyến dùng): 7 – 12V - Điện áp đầu vào (giới hạn): 6 – 20V - Chân Digital I/O: 14 (với 6 chân PWM output) - Chân PWM Digital I/O: 6 - Chân đầu vào Analog: 8 (thêm A6, A7) so với UNO - Dòng sử dụng I/O Pin: 20 mA (tối đa 40 mA) - Bộ nhớ Flash: 32 KB (ATmega328) - SRAM: 2 KB (ATmega328) - EEPROM: 1 KB (ATmega328) - Clock Speed: 16 MHz - Chiều dài: 43.2 mm - Chiều rộng: 18.5 mm Trọng lượng: 5g
Cảm biến siêu âm HC-SR04	 	- Điện áp: 5V DC - Dòng hoạt động: < 2mA - Mức cao: 5V - Mức thấp: 0V - Góc tối đa: 15 độ - Khoảng cách: 2cm – 450cm (4.5m) - Độ chính xác: 3mm
Cảm biến vị trí Optical Flow PMW3901		- Điện áp cung cấp: 1.8 – 3.6V - Phạm vi làm việc: 80mm đến + ∞

PHỤ LỤC 2

1. Phần lập trình cho khối RX – Drone Master

```
#include <SPI.h>

#include <nRF24L01.h>

#include <RF24.h>

//////////////////////////////// PPM CONFIGURATION //////////////////////////////////

#define channel_number 7    // Số kênh

#define sigPin 2            // PPM output

#define PPM_FrLen 22500    // Độ dài frame PPM (μs)

#define PPM_PulseLen 300   // Độ rộng xung PPM (μs)

////////////////////////////////////////////////////////////////////////

String buffer = "";

float distance = 0;

float current_height = 0.0;

float temp1_height = 0.0; float temp2_height = 0.0; float Required_height = 0.0;

unsigned long last_time;

float input_THROTTLE = 1000;

int ppm[channel_number];

const uint64_t pipeIn = 0xE8E8F0F0E1LL;

RF24 radio(9, 10); // CE, CSN

// Dữ liệu truyền từ TX sang RX

struct MyData {
```

```
byte throttle;  
byte yaw;  
byte pitch;  
byte roll;  
byte AUX1;  
byte AUX2;  
byte U;  
byte MODE;  
float H;  
};
```

```
MyData data;
```

```
//PID Variables
```

```
float p=1.5;float p_value=0;//2.75 là ổn
```

```
float I=0.01;float I_value=0;//0.01
```

```
float D=0.3;float D_value=0; //0.3
```

```
float total=0;
```

```
float error=0;float pre_error=0;float derror=0;float ierror=0;
```

```
void resetData() {
```

```
    // Giá trị an toàn khi mất tín hiệu
```

```
    data.throttle = 126;
```

```
    data.yaw = 130;
```

```
    data.pitch = 255;
```

```
    data.roll = 128;
```

```
    data.AUX1 = 0;
```

```
    data.AUX2 = 0;
```

```
    data.U = 132;
```

```

data.MODE = 0;

setPPMValuesFromData();

}

void setPPMValuesFromData() {

  ppm[0] = map(data.throttle, 0, 255, 1000, 2000); //A = throttle
  ppm[1] = map(data.yaw, 0, 255, 1000, 2000); //E
  input_THROTTLE = map(data.pitch, 0, 255, 1000, 2000);
  switch ((data.MODE > 0)) //output of the channel 8
  {
    case 0: //normal mode

      //reset the error values

      //Serial.println("M_mode");

      error=0; pre_error=0; derror=0; ierror=0;

      temp1_height=distance;

      if(temp1_height<500)

        Required_height=100; //temp1_height; //keep updating the required height in
normal mode

        Serial.print("<0:");

        Serial.print(Required_height, 2);

        Serial.println(">");

        ppm[2] = input_THROTTLE; //T

        break;

    case 1: //altitude hold mode

      //Serial.println("AUTO");

      temp2_height=distance;

      if(temp2_height<500)

        current_height=temp2_height;

```

```

//PID implementation //update values

unsigned long now = millis();

float dt = (now - last_time) / 1000.0;

error=100-current_height; //Required_height-current_height;

derror=(error-pre_error)/dt; ierror+=error*dt;

p_value=p*(error); I_value=I*(ierror); D_value=D*(derror);

total=constrain(p_value + I_value + D_value, -250, 250);

pre_error=error;

last_time = now;

//input_THROTTLE-=total;

Serial.print("<1:");

Serial.print(Required_height, 2);

Serial.println(">");

ppm[2] = 1450-total;//input_THROTTLE-total; //T

break;

}

ppm[3] = map(data.roll, 0, 255, 1000, 2000); //R

ppm[4] = map(data.AUX1, 0, 1, 1000, 2000);

ppm[5] = map(data.AUX2, 0, 1, 1000, 2000);

ppm[6] = map(data.U, 0, 255, 1000, 2000); //GPS

}

void setupPPM() {

pinMode(sigPin, OUTPUT);

digitalWrite(sigPin, LOW); // trạng thái mặc định

```

```
cli(); // tắt ngắt toàn cục
```

```
// Reset Timer1
```

```
TCCR1A = 0;
```

```
TCCR1B = 0;
```

```
OCR1A = 100;          // giá trị so sánh ban đầu
```

```
TCCR1B |= (1 << WGM12); // CTC mode
```

```
TCCR1B |= (1 << CS11); // prescaler = 8 → tick = 0.5us @16MHz
```

```
TIMSK1 |= (1 << OCIE1A); // bật ngắt so sánh
```

```
sei(); // bật lại ngắt
```

```
}
```

```
void setup() {
```

```
  Serial.begin(9600);
```

```
  resetData();
```

```
  setupPPM();
```

```
  radio.begin();
```

```
  radio.setDataRate(RF24_250KBPS);
```

```
  radio.setAutoAck(false);
```

```
  radio.openReadingPipe(1, pipeIn);
```

```
  radio.startListening();
```

```
  resetData();
```

```
  //delay(3000);
```

```
}
```

```
unsigned long lastRecvTime = 0;
```

```

void recvData() {
    while (radio.available()) {
        radio.read(&data, sizeof(MyData));
        lastRecvTime = millis();
    }
}

```

```

void loop() {

    recvData();

    if (millis() - lastRecvTime > 1000) {
        // mất tín hiệu
        resetData();
    }
    static String buffer = "";
    while (Serial.available()) {
        char c = Serial.read();
        if (c == '<') {
            buffer = "";
        } else if (c == '>') {
            int firstColon = buffer.indexOf(':');
            int secondColon = buffer.indexOf(':', firstColon + 1);
            if (firstColon != -1 && secondColon != -1) {
                distance    = buffer.substring(0, firstColon).toFloat();
                float pidOutX  = buffer.substring(firstColon + 1, secondColon).toFloat();
                float pidOutY  = buffer.substring(secondColon + 1).toFloat();
                // Xử lý các giá trị height, pidOutX, pidOutY tại đây
            }
        }
    }
}

```

```

        Serial.println(distance);

        Serial.println(pidOutX);

        Serial.println(pidOutY);

    }

    } else {

        buffer += c;

    }

}

/*

while (Serial.available()) {

    char c = Serial.read();

    if (c == '<') {

        buffer = "";

    } else if (c == '>') {

        distance = buffer.toFloat();

        //Serial.println(distance);

    } else {

        buffer += c;

    }

}

*/

setPPMValuesFromData();

//Serial.println(data.MODE);

}

//-----

// Thay đổi clockMultiplier theo tần số MCU:

#define clockMultiplier 2 // dùng 2 nếu Arduino chạy 16MHz

```

```

ISR(TIMER1_COMPA_vect) {

    static boolean state = true;

    static byte cur_chan_numb = 0;

    static unsigned int calc_rest = 0;


    TCNT1 = 0;


    if (state) {
        // kết thúc xung

        digitalWrite(sigPin, LOW);

        OCR1A = PPM_PulseLen * clockMultiplier;

        state = false;
    } else {
        // bắt đầu xung

        digitalWrite(sigPin, HIGH);

        state = true;


        if (cur_chan_numb >= channel_number) {

            cur_chan_numb = 0;

            calc_rest += PPM_PulseLen;

            OCR1A = (PPM_FrLen - calc_rest) * clockMultiplier;

            calc_rest = 0;
        } else {

            OCR1A = (ppm[cur_chan_numb] - PPM_PulseLen) * clockMultiplier;

            calc_rest += ppm[cur_chan_numb];

            cur_chan_numb++;
        }
    }
}

```

2. Phần lập trình cho khối RX – Drone Slave

```
#include <SPI.h>

#include <nRF24L01.h>

#include <RF24.h>

//////////////// PPM CONFIGURATION //////////////////

#define channel_number 7    // Số kênh

#define sigPin 2            // PPM output pin (pin 9 trên Pro Micro)

#define PPM_FrLen 22500     // Độ dài frame PPM (μs)

#define PPM_PulseLen 300    // Độ rộng xung PPM (μs)

////////////////////////////////////

String buffer = "";

float distance = 0; float H = 3;

float current_height = 0.0;

float temp1_height = 0.0; float temp2_height = 0.0; float Required_height=0;

float input_THROTTLE=1000;

unsigned long last_time;

unsigned long last_pid_time = 0;

const unsigned long pid_interval = 50; // 50ms = 20Hz

int ppm[channel_number];

const uint64_t pipeIn = 0xE8E8F0F0E1LL;

RF24 radio(9, 10); // CE, CSN
```

```
// Dữ liệu truyền từ TX sang RX
```

```
struct MyData {
```

```
    byte throttle;
```

```
    byte yaw;
```

```
    byte pitch;
```

```
    byte roll;
```

```
    byte AUX1;
```

```
    byte AUX2;
```

```
    byte U;
```

```
    byte MODE;
```

```
    float H;
```

```
};
```

```
MyData data;
```

```
//PID Variables
```

```
float p=1.2;float p_value=0;
```

```
float I=0.02;float I_value=0;
```

```
float D=0.3;float D_value=0;
```

```
float total=0;
```

```
float error=0;float pre_error=0;float derror=0;float ierror=0;
```

```
void resetData() {
```

```
    // Giá trị an toàn khi mất tín hiệu
```

```
    data.throttle = 126;
```

```
    data.yaw = 130;
```

```
    data.pitch = 255;
```

```
    data.roll = 128;
```

```
    data.AUX1 = 0;
```

```

data.AUX2 = 0;

data.U = 132;

data.MODE = 0;

setPPMValuesFromData();

}

void setPPMValuesFromData() {

  ppm[0] = map(data.throttle, 0, 255, 1000, 2000); //A
  ppm[1] = map(data.yaw, 0, 255, 1000, 2000); //E
  H = data.H;

  // --- PID chỉ cập nhật theo chu kỳ 50ms ---
  if (millis() - last_pid_time >= pid_interval) {
    //Serial.println(H);

    input_THROTTLE = map(data.pitch, 0, 255, 1000, 2000);
    last_pid_time = millis();

    if (H > 2.0 && distance < 300) {
      current_height = distance;

      float dt = pid_interval / 1000.0; // cố định 0.05s

      error = H - current_height;
      derror = (error - pre_error) / dt;
      ierror += error * dt;

      p_value = p * error;
      I_value = I * ierror;
      D_value = D * derror;

      total = constrain(p_value + I_value + D_value, -200, 150);
      pre_error = error;
    }
  }
}

```

```

    if (distance>10){
        input_THROTTLE = 1450-total;
    }else{
        input_THROTTLE -= total;
    }
    //Serial.println(input_THROTTLE);
}
}

ppm[2] = input_THROTTLE;           //T
ppm[3] = map(data.roll, 0, 255, 1000, 2000); //R
ppm[4] = map(data.AUX1, 0, 1, 1000, 2000);
ppm[5] = map(data.AUX2, 0, 1, 1000, 2000);
ppm[6] = map(data.U, 0, 255, 1000, 2000); //GPS
}

void setupPPM() {
    pinMode(sigPin, OUTPUT);
    digitalWrite(sigPin, LOW); // trạng thái mặc định

    cli(); // tắt ngắt toàn cục

    // Reset Timer1
    TCCR1A = 0;
    TCCR1B = 0;

    OCR1A = 100;           // giá trị so sánh ban đầu
    TCCR1B |= (1 << WGM12); // CTC mode
    TCCR1B |= (1 << CS11); // prescaler = 8 → tick = 0.5us @16MHz
    TIMSK1 |= (1 << OCIE1A); // bật ngắt so sánh

```

```
sei(); // bật lại ngắt  
}
```

```
void setup() {  
  Serial.begin(9600);  
  
  resetData();  
  setupPPM();  
  
  radio.begin();  
  radio.setDataRate(RF24_250KBPS);  
  radio.setAutoAck(false);  
  radio.openReadingPipe(1, pipeIn);  
  radio.startListening();  
  resetData();  
  //delay(3000);  
}
```

```
unsigned long lastRecvTime = 0;
```

```
void recvData() {  
  while (radio.available()) {  
    radio.read(&data, sizeof(MyData));  
    lastRecvTime = millis();  
  }  
}
```

```
void loop() {
```

```

recvData();

//H = data.H;

////////////////////

Serial.println(data.H);

////////////////////

if (millis() - lastRecvTime > 1000) {
    // mất tín hiệu
    resetData();
}

while (Serial.available()) {
    char c = Serial.read();
    if (c == '<') {
        buffer = "";
    } else if (c == '>') {
        distance = buffer.toFloat();
        //Serial.print("Khoang cach UART: ");
        //Serial.println(distance);
    } else {
        buffer += c;
    }
}

setPPMValuesFromData();
//Serial.println(data.MODE);
}

//-----

// Thay đổi clockMultiplier theo tần số MCU:
#define clockMultiplier 2 // dùng 2 nếu Arduino chạy 16MHz

```

```

ISR(TIMER1_COMPA_vect) {
    static boolean state = true;
    static byte cur_chan_numb = 0;
    static unsigned int calc_rest = 0;

    TCNT1 = 0;

    if (state) {
        // kết thúc xung
        digitalWrite(sigPin, LOW);
        OCR1A = PPM_PulseLen * clockMultiplier;
        state = false;
    } else {
        // bắt đầu xung
        digitalWrite(sigPin, HIGH);
        state = true;

        if (cur_chan_numb >= channel_number) {
            cur_chan_numb = 0;
            calc_rest += PPM_PulseLen;
            OCR1A = (PPM_FrLen - calc_rest) * clockMultiplier;
            calc_rest = 0;
        } else {
            OCR1A = (ppm[cur_chan_numb] - PPM_PulseLen) * clockMultiplier;
            calc_rest += ppm[cur_chan_numb];
            cur_chan_numb++;
        }
    }
}

```

```
}
```

3. Phần lập trình cho khối RX – màn hình LCD

```
#include <Wire.h>
```

```
#include <LiquidCrystal_I2C.h>
```

```
#include <SPI.h>
```

```
#include "RF24.h"
```

```
LiquidCrystal_I2C lcd(0x27, 20, 4); // hoặc 0x3F tùy màn hình
```

```
RF24 radio(9, 10); // CE, CSN
```

```
const uint64_t pipe = 0xF0F0F0F0E1LL;
```

```
struct Payload {
```

```
    uint8_t nodeID;
```

```
    float requiredHeight;
```

```
    float measuredDistance;
```

```
};
```

```
Payload payload;
```

```
int r = 0;
```

```
void setup() {
```

```
    Serial.begin(9600);
```

```
    lcd.init();
```

```
    lcd.backlight();
```

```
    lcd.setCursor(0, 0);
```

```
    lcd.print("Do cao 1:");
```

```
    lcd.setCursor(0, 1);
```

```

lcd.print("Do cao2:");

lcd.setCursor(0, 2);

lcd.print("Set Point:");

radio.begin();

//radio.setAutoAck(true);

radio.setDataRate(RF24_1MBPS);

radio.setChannel(10);

// Mở 2 pipe đọc tương ứng 2 node TX

radio.openReadingPipe(1, pipe);


radio.startListening();
}

void loop() {
  if (radio.available()) {
    radio.read(&payload, sizeof(payload));
    if (payload.nodeID == 0){
      lcd.setCursor(9, 0);
      lcd.print("      "); // clear vùng cũ
      lcd.setCursor(9, 0);
      lcd.print(payload.measuredDistance, 1); // 1 chữ số sau dấu phẩy
      lcd.print(" cm");
      //////////////////////////////////////
      if (payload.requiredHeight > 0.01){
        Serial.print("<");
        Serial.print(payload.requiredHeight, 2);
        Serial.println(">");
      } else{
        Serial.print("<");

```

```

    Serial.print(payload.measuredDistance, 2);

    Serial.println(">");

}

////////////////////////////////////

} else {

    lcd.setCursor(9, 1);

    lcd.print("    "); // clear vùng cũ

    lcd.setCursor(9, 1);

    lcd.print(payload.measuredDistance, 1); // 1 chữ số sau dấu phẩy

    lcd.print(" cm");

}

    lcd.setCursor(11, 2);

    lcd.print("    "); // clear vùng cũ

    lcd.setCursor(11, 2);

    if(payload.requiredHeight>0.01){

        lcd.print(payload.requiredHeight, 1);

        lcd.setCursor(17, 2);

        lcd.print(" ");

    } else {

        lcd.setCursor(17, 2);

        lcd.print("X");

    }

    //Serial.print("MD ");

    //Serial.print(payload.measuredDistance, 2);

    //Serial.print("RH ");

    //Serial.println(payload.requiredHeight, 2);

}

```

```
}
```

4. Phần lập trình cho cảm biến độ cao Drone 1

```
#include <SPI.h>
```

```
#include "RF24.h"
```

```
#include "Bitcraze_PMW3901.h"
```

```
#include <SimpleKalmanFilter.h> // từ thư viện TKJElectronics
```

```
const int trigPin = 4;
```

```
const int echoPin = 3;
```

```
Bitcraze_PMW3901 flow(7);
```

```
RF24 radio(9, 10); // CE, CSN
```

```
const uint64_t pipe = 0xF0F0F0F0E1LL;
```

```
struct Payload {
```

```
    uint8_t nodeID;
```

```
    float requiredHeight;
```

```
    float measuredDistance;
```

```
};
```

```
Payload payload = {0, 0.0};
```

```
// Tham số: noise_measure, noise_estimate, process_noise
```

```
SimpleKalmanFilter kalmanFilter(4, 4, 0.06);
```

```
// Hệ số EMA: càng gần 1 càng nhạy, càng gần 0 càng mượt
```

```
const float alpha = 0.3;
```

```
float filteredDistance = 0;
```

```

// Biến lưu vị trí
float posX = 0.0;
float posY = 0.0;

unsigned long lastUpdate = 0;
const float focalLength = 35.0; // dùng giá trị thực nghiệm

int16_t deltaX, deltaY;

// PID cho giữ vị trí
float Kp = 0.05;
float Ki = 0.0;
float Kd = 0.01;

float errorX = 0, prevErrorX = 0, integralX = 0;
float errorY = 0, prevErrorY = 0, integralY = 0;

float outputX = 0;
float outputY = 0;

float computePID(float error, float& prevError, float& integral, float dt) {
    integral += error * dt;
    float derivative = (error - prevError) / dt;
    prevError = error;
    return Kp * error + Ki * integral + Kd * derivative;
}

```

```

float getAverageDistance(int samples = 4) {
    float sum = 0;
    int validCount = 0;
    for (int i = 0; i < samples; i++) {
        digitalWrite(trigPin, LOW);
        delayMicroseconds(2);
        digitalWrite(trigPin, HIGH);
        delayMicroseconds(10);
        digitalWrite(trigPin, LOW);

        long duration = pulseIn(echoPin, HIGH); // 30ms timeout
        if (duration == 0) continue;

        float distance = duration * 0.0343 / 2;
        sum += distance;
        validCount++;
        delay(10); // nhỏ thôi để không trễ nhiều
    }
    if (validCount == 0 ) return -1;
    return sum / validCount;
}

```

```

void setup() {
    Serial.begin(9600);
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);

    radio.begin();
    radio.setAutoAck(true);
}

```

```

radio.setDataRate(RF24_1MBPS);

radio.setChannel(10);

radio.openWritingPipe(pipe);

radio.stopListening();


if (!flow.begin()) {
    Serial.println("Initialization of the flow sensor failed");
    while(1) { }
}


filteredDistance = getAverageDistance(2);
}


void loop() {
    // Đo khoảng cách

    float rawDistance = getAverageDistance(4);

    if (rawDistance < 0) return;


    // EMA filter

    //filteredDistance = alpha * rawDistance + (1 - alpha) * filteredDistance;


    filteredDistance = kalmanFilter.updateEstimate(rawDistance);
    //Serial.println(filteredDistance);

    ///////////////////////////////////////////////////

    flow.readMotionCount(&deltaX, &deltaY);

    float deltaTime = (millis() - lastUpdate) / 1000.0; // giây
    lastUpdate = millis();

    // Chuyển pixel thành mm/s

    float velX = (deltaX * filteredDistance) / focalLength;

```



```
radio.write(&payload, sizeof(payload));
```

```
delay(5 + random(3));
```

```
}
```

5. Phần lập trình cho cảm biến độ cao Drone 2

```
#include <SPI.h>
```

```
#include "RF24.h"
```

```
#include <SimpleKalmanFilter.h> // từ thư viện TKJElectronics
```

```
const int trigPin = 4;
```

```
const int echoPin = 3;
```

```
RF24 radio(9, 10); // CE, CSN
```

```
const uint64_t pipe = 0xF0F0F0F0E1LL;
```

```
struct Payload {
```

```
    uint8_t nodeID;
```

```
    float requiredHeight;
```

```
    float measuredDistance;
```

```
};
```

```
Payload payload = {1, 0.0};
```

```
// Tham số: noise_measure, noise_estimate, process_noise
```

```
SimpleKalmanFilter kalmanFilter(4, 4, 0.06);
```

```
// Hệ số EMA: càng gần 1 càng nhạy, càng gần 0 càng mượt
```

```
const float alpha = 0.3;
```

```
float filteredDistance = 0;
```

```

float getAverageDistance(int samples = 4) {
    float sum = 0;
    int validCount = 0;
    for (int i = 0; i < samples; i++) {
        digitalWrite(trigPin, LOW);
        delayMicroseconds(2);
        digitalWrite(trigPin, HIGH);
        delayMicroseconds(10);
        digitalWrite(trigPin, LOW);

        long duration = pulseIn(echoPin, HIGH); // 30ms timeout
        if (duration == 0) continue;

        float distance = duration * 0.0343 / 2;
        sum += distance;
        validCount++;
        delay(10); // nhỏ thôi để không trễ nhiều
    }
    if (validCount == 0) return -1;
    return sum / validCount;
}

void setup() {
    Serial.begin(9600);
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);

    radio.begin();

```

```

//radio.setAutoAck(true);

radio.setDataRate(RF24_1MBPS);

radio.setChannel(10);

radio.openWritingPipe(pipe);

radio.stopListening();


filteredDistance = getAverageDistance(2);
}

void loop() {
    // Đo khoảng cách
    float rawDistance = getAverageDistance(4);
    if (rawDistance < 0) return;

    // EMA filter
    //filteredDistance = alpha * rawDistance + (1 - alpha) * filteredDistance;

    filteredDistance = kalmanFilter.updateEstimate(rawDistance);
    //Serial.println(filteredDistance);

    // Gửi qua UART với dấu phân cách
    Serial.print("<");
    Serial.print(filteredDistance, 2);
    Serial.println(">");

    // Gửi
    payload.measuredDistance = filteredDistance;
    payload.requiredHeight = 0.01;
    radio.write(&payload, sizeof(payload));

```

```
    delay(10+random(3));  
}
```

6. Phần lập trình cho khối TX – Tay cầm điều khiển

```
#include <SPI.h>  
#include <nRF24L01.h>  
#include <RF24.h>
```

```
#define GPS_UP 2  
#define GPS_DOWN 3  
#define AUTO_MODE 4
```

```
const uint64_t pipeOut = 0xE8E8F0F0E1LL; //IMPORTANT: The same as in the receiver
```

```
RF24 radio(9, 10); // select CSN pin
```

```
// The sizeof this struct should not exceed 32 bytes
```

```
// This gives us up to 32 8 bits channels
```

```
struct MyData {  
    byte roll;  
    byte pitch;  
    byte throttle;  
    byte yaw;  
    byte AUX1;  
    byte AUX2;  
    byte U;  
    byte MODE;  
    float H;  
};
```

```

MyData data;

String buffer = "";

float distance = 0;


void resetData()
{
    //This are the start values of each channel

    // Throttle is 0 in order to stop the motors

    //127 is the middle value of the 10ADC.


    data.throttle = 126;

    data.yaw = 130;

    data.pitch = 255;

    data.roll = 128;

    data.AUX1 = 0;

    data.AUX2 = 0;

    data.U = 132;

    data.MODE = 0;
}

void setup()
{
    Serial.begin(9600);

    pinMode(GPS_UP, INPUT_PULLUP); // Công tắc GPS - UP
    pinMode(GPS_DOWN, INPUT_PULLUP); // Công tắc GPS - DOWN
    pinMode(AUTO_MODE, INPUT_PULLUP);

    //Start everything up

    radio.begin();

    radio.setAutoAck(false);

```

```

radio.setDataRate(RF24_250KBPS);

radio.openWritingPipe(pipeOut);

resetData();
}

/*****

// Returns a corrected value for a joystick position that takes into account
// the values of the outer extents and the middle of the joystick range.
int mapJoystickValues(int val, int lower, int middle, int upper, bool reverse)
{
    val = constrain(val, lower, upper);
    if ( val < middle )
        val = map(val, lower, middle, 0, 128);
    else
        val = map(val, middle, upper, 128, 255);
    return ( reverse ? 255 - val : val );
}

int smoothAnalogRead(int pin) {
    int sum = 0;
    for (int i = 0; i < 5; i++) {
        sum += analogRead(pin);
        delayMicroseconds(500); // delay nhỏ để ổn định
    }
    return sum / 5;
}

// Deadzone center: 127, width: ±5

```

```

int applyDeadzone(int value, int center = 127, int threshold = 8) {
    if (abs(value - center) <= threshold) return center;
    return value;
}

```

```

void loop()

```

```

{
    // The calibration numbers used here should be measured
    // for your joysticks till they send the correct values.

    data.throttle = applyDeadzone(mapJoystickValues( smoothAnalogRead(A1), 122, 527, 920,
true ),127);

    data.yaw      = applyDeadzone(mapJoystickValues( smoothAnalogRead(A0), 80, 540, 920,
true),128);

    data.pitch    = applyDeadzone(mapJoystickValues( smoothAnalogRead(A2), 100, 500, 980,
true ),130);

    data.roll     = applyDeadzone(mapJoystickValues( smoothAnalogRead(A3), 100, 460, 980,
true ),126);

    data.AUX1     = 1;
    data.AUX2     = 1;
    //data.U = 220;

    bool gpsUp = digitalRead(GPS_UP) == LOW;
    bool gpsDown = digitalRead(GPS_DOWN) == LOW;
    bool mode = digitalRead(AUTO_MODE) == LOW;

    if (gpsUp && !gpsDown) {
        data.U = 219; // ATTI MODE
    } else if (!gpsUp && gpsDown) {
        data.U = 48; // GPS ON
    } else {

```

```

    data.U = 130; // Gửi
}

if (mode) {
    data.MODE = 1;
} else {
    data.MODE = 0;
}

while (Serial.available()) {
    char c = Serial.read();
    if (c == '<') {
        buffer = "";
    } else if (c == '>') {
        distance = buffer.toFloat();

        //Gửi lên Drone
        data.H = distance;

        //////////////////////////////////////

        Serial.print("Khoang cach UART: ");
        Serial.println(distance);

        //////////////////////////////////////

    } else {
        buffer += c;
    }
}

radio.write(&data, sizeof(MyData));
delay(10);
}

```