

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC CNKT ĐIỆN TỬ - VIỄN THÔNG

**THIẾT KẾ MẠCH ĐIỀU KHIỂN MÁY BAY KHÔNG NGƯỜI LÁI
HIỂN THỊ THÔNG TIN TRÊN WEB**

CBHD: Ths. Đinh Thị Kim Phụng

Sinh viên: Lê Ngọc Hưng

Mã số sinh viên: 1141050436

Hà Nội – 2020

LÊ NGỌC HƯNG

CNKT ĐIỆN TỬ - VIỄN THÔNG

LỜI CẢM ƠN

Trong thời gian thực hiện đề tài đồ án tốt nghiệp, em đã nhận được nhiều sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của các thầy cô bộ môn Điện tử viễn thông cũng như thầy cô trong khoa Điện tử trường Đại học công nghiệp Hà Nội. Đồng thời chúng em đã được tiếp cận các trang thiết bị hiện đại của khoa để phục vụ vào mục đích nghiên cứu, học tập.

Em xin gửi lời cảm ơn chân thành đến cô Đinh Thị Kim Phượng, đã tận tình hướng dẫn, chỉ dạy, giúp đỡ và cung cấp những tài liệu cũng như kinh nghiệm quý báu giúp em hoàn thành các nhiệm vụ được giao trong quá trình thực hiện.

Em cũng xin cảm ơn các thầy cô trong trường Đại học Công nghiệp Hà Nội nói chung, các thầy cô trong khoa Điện tử nói riêng đã chỉ dạy những kiến thức quý báu, giúp em có được cơ sở lý thuyết vững vàng và tạo điều kiện giúp đỡ em trong suốt quá trình học tập.

Cuối cùng, em xin chân thành cảm ơn gia đình và bạn bè đã luôn tạo điều kiện, quan tâm, giúp đỡ, động viên trong suốt quá trình học tập và hoàn thành đề tài đồ án tốt nghiệp.

Hà Nội, Ngày ... tháng ... năm ...

Sinh viên thực hiện

Lê Ngọc Hưng

MỤC LỤC

KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT.....	I
DANH MỤC HÌNH ẢNH	II
DANH MỤC BẢNG BIỂU.....	III
MỞ ĐẦU.....	1
CHƯƠNG 1. TÌM HIỂU VỀ MÁY BAY KHÔNG NGƯỜI LÁI.....	2
1.1. Tổng quan về máy bay không người lái	2
1.2. Nguyên lý hoạt động của quadcopter	3
1.3. Tìm hiểu về mô hình tuyến tính máy bay.....	5
1.3.1. Hệ quy chiếu.....	5
1.3.2. Các góc Euler.....	6
1.3.3. Ma trận chuyển đổi hệ trục tọa độ	7
1.3.4. Hệ phương trình chuyển động.....	8
1.4. Mô hình động cơ và cánh quạt.....	18
1.5. Mô phỏng tìm hệ số PID.....	19
CHƯƠNG 2. CÁC BƯỚC THIẾT KẾ QUADCOPTER	20
2.1. Thiết kế cơ khí và mạch điện tử	20
2.1.1. Thiết kế cơ khí	20
2.1.2. Thiết kế điện tử.....	21
2.2. Sơ đồ các module và mạch nguyên lý	29
2.2.1. Các khối module	29
2.2.2. Mạch nguyên lý khối nhận và xử lý tín hiệu.....	30
2.2.3. Mạch nguyên lý khối truyền tín hiệu.....	31
2.3. Thiết kế bộ điều khiển	31
2.3.1. Tìm hiểu về và các phương pháp thiết kế bộ điều khiển PID. 31	
2.3.2. Phương pháp tuyến tính kết hợp mô phỏng	32
2.3.3. Phương pháp dò thực nghiệm kết hợp Zieler-Nichols	33

2.3.4.	Lưu đồ dải thuật cân bằng.....	33
2.4.	Quá trình thực nghiệm	34
2.4.1.	Quá trình dò các thông số PID	34
2.4.2.	Bay thử nghiệm sản phẩm.....	36
CHƯƠNG 3.	MODULE GỬI THÔNG TIN.	37
3.1.	Đặt vấn đề.....	37
3.2.	Phân cứng sử dụng.....	37
3.3.	Module gửi thông tin.	38
3.3.1.	Chức năng.....	38
3.3.2.	Sơ đồ nguyên lý	39
3.3.3.	Giao diện điều khiển	39
3.3.4.	Kết nối với Module gửi thông tin.....	40
CHƯƠNG 4.	KẾT LUẬN ĐỀ XUẤT	41
4.1.	Kết quả đạt được.....	41
4.2.	Đề xuất	41
TÀI LIỆU THAM KHẢO.....		42
PHỤ LỤC.....		43

KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT

UAV	Máy bay không người lái (Unmanned Aerial Vehicle)
PID	Tỷ lệ - tích phân – vi phân (Proportional – Integral – Derivative)
ESC	Bộ điều tốc (Electronic Speed Controller)
RX	Bộ thu sóng radio (Receiver)
TX	Bộ phát sóng radio (Transmitter)
BLDC	Động cơ một chiều không chổi than (Brushless DC Motor)
PWM	Điều chỉnh độ rộng xung (Pulse Width Modulation)

DANH MỤC HÌNH ẢNH

Hình 1.1 Mô tả các chuyển động của Quadcopter.....	4
Hình 1.2 Hệ tọa độ vật (hình trái) và hệ tọa độ quán tính (hình phải)	6
Hình 1.3 Quy ước các phương chiều	7
Hình 1.4 Sơ đồ mô phỏng để tìm hệ số PID theo sơ đồ sau:	19
Hình 2.1 Thiết kế khung cơ khí	20
Hình 2.2 Arduino nano.....	21
Hình 2.3 Cảm biến MPU-6050	23
Hình 2.4 Esc cho động cơ không chổi than.....	25
Hình 2.5 Pin Li-Po	26
Hình 2.6 Động cơ 3 pha DC không chổi than	26
Hình 2.7 Module NRF24L01.....	27
Hình 2.8 Mô hình truyền nhận của Module NRF24L01	27
Hình 2.9 Joystick.....	28
Hình 2.10 Hardware Diagram.....	29
Hình 2.11 Mạch nguyên lý khối nhận và xử lý tín hiệu	30
Hình 2.12 Mạch nguyên lý khối truyền tín hiệu.....	31
Hình 2.13 Lưu đồ dải thuật cân bằng.....	34
Hình 2.14 Phần mềm điều chỉnh PID	35
Hình 2.15 Bay thử nghiệm sản phẩm.....	36
Hình 3.1 Sơ đồ nguyên lý Module gửi thông tin.....	39
Hình 3.2 Giao diện điều khiển Module gửi thông tin.....	39
Hình 3.3 Kết nối với Modul gửi thông tin.....	40
Hình 4.1 Sản phẩm hoàn thiện.....	41

DANH MỤC BẢNG BIỂU

Bảng 2.1 Thông số của kỹ thuật của Arduino nano.....	22
Bảng 2.2 Thông số kỹ thuật của Cảm biến IMU MPU-6050	24
Bảng 2.3 Thông số kỹ thuật của Pin Li-Po WildScorpion.....	25
Bảng 2.4 Thông số kỹ thuật PID dò được.....	36
Bảng 3.1 Thông số kỹ thuật Module thu phát Wifi NodeMcu	38

MỞ ĐẦU

Lý do chọn đề tài:

Máy bay không người lái (UAV) hiện nay được sử dụng khá phổ biến trong các lĩnh vực, nó cung cấp góc quan sát từ trên cao như trong các lĩnh vực giải trí, thể thao. Ngoài ra UAV còn được sử dụng nhiều trong lĩnh vực quân sự như do thám, cứu hộ và trong nhiệm vụ vận chuyển bưu kiện có khối lượng nhỏ do có ưu điểm linh hoạt, khó bị phát hiện do kích thước và tiếng ồn nhỏ, chi phí vận hành thấp hơn nhiều so với các loại máy bay khác. Do vậy đây là một đề tài cấp thiết phục vụ cho nhiều lĩnh vực khác nhau trong cuộc sống.

Mục đích:

Nghiên cứu và thiết kế thiết bị bay không người lái phục vụ cho quá trình thu thập dữ liệu. củng cố và ứng dụng kiến thức đã được học và thu thập kiến thức trong quá trình làm đề tài. Tìm ra nguyên lý, cơ cấu hoạt động của máy bay. Chế tạo mô hình bay qua quá trình nghiên cứu xây dựng đề tài.

Đối tượng và phạm vi nghiên cứu:

Đối tượng nghiên cứu là thiết bị bay lên thẳng bốn cánh quạt không người lái. Do thời gian nghiên cứu và kiến thức của em còn hạn chế nên phạm vi nghiên cứu nằm trong các thiết bị bay mô hình nhỏ gọn.

Ý nghĩa khoa học và thực tiễn của đề tài:

Đề tài này giúp giải quyết những hạn chế của thiết bị bay khác như: Tính linh hoạt, bảo mật, chi phí chế tạo... Phạm vi ứng dụng thực tiễn đề tài rộng rãi từ lĩnh vực quân sự, dân sự đến giải trí...

Chương 1. Tìm hiểu về máy bay không người lái

1.1. Tổng quan về máy bay không người lái

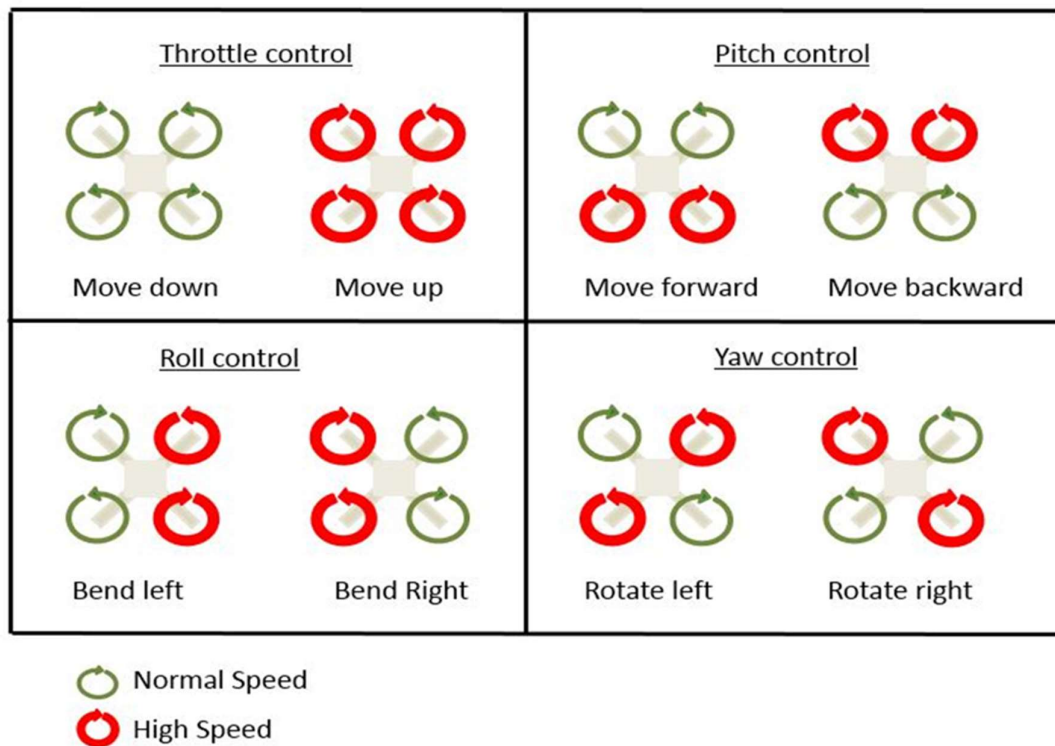
Máy bay không người lái (UAV) là loại máy bay có thể được điều khiển từ xa thông qua hệ thống điều khiển tự động hoặc bán tự động. Trong số các loại máy bay không người lái, quadcopter là một trong những loại máy bay có khả năng cất hạ cánh thẳng đứng mà không cần đường băng. Bộ phận tạo ra lực đẩy và đồng thời cũng làm nhiệm vụ cân bằng cho máy bay là bốn động cơ. Toàn bộ các chuyển động của máy bay được điều khiển thông qua tăng giảm tốc độ quay của động cơ (không thông qua thay đổi góc tấn các cánh), từ đó dẫn đến thay đổi lực đẩy trên các cánh quạt. Do đó, so với các loại máy bay khác, quadcopter có kết cấu cơ khí đơn giản hơn. Điều này đồng nghĩa với việc chế tạo và bảo trì sẽ đơn giản và ít tốn kém so với các loại máy bay khác. Mặt khác, quadcopter tạo lực đẩy từ bốn cánh quạt nên cho phép tạo ra lực đẩy tương đương với các máy bay khác có kích thước công kênh hơn do sử dụng cánh quạt có kích thước lớn hơn. Điều này là ưu thế cho phép quadcopter hoạt động trong các môi trường chật hẹp và nhiều vật cản như trong thành phố, ở các nơi không có quá nhiều không gian cho việc cất hạ cánh. Bên cạnh các ưu điểm, một trong số các hạn chế của quadcopter là hiệu suất lực đẩy của máy bay thấp do toàn bộ lực nâng của máy bay sinh ra từ lực đẩy của động cơ. Mặt khác, sử dụng cánh quạt có đường kính nhỏ cho tổn thất khí động lớn hơn so với cánh quạt có đường kính lớn. Tuy vậy, trong tương lai, việc chế tạo máy bay bằng các vật liệu mới có khối lượng nhẹ, sẽ đóng góp đáng kể vào việc nâng cao hiệu suất tiêu thụ năng lượng của máy bay. Quadcopter đã và đang cho thấy sự hữu ích trong muôn vàn các ứng dụng trong cuộc sống. Với kích thước nhỏ gọn với khả năng mang theo tải trọng lớn, quadcopter có thể làm nhiệm vụ cứu hộ một cách hiệu quả (cứu thương, tìm kiếm cứu hộ sau thảm họa, cứu hỏa...), đặc biệt trong thành phố, nơi mà nguy cơ tắc nghẽn giao thông đường bộ luôn có khả năng làm chậm trễ việc tiếp cận bằng các phương tiện

cứu hộ đường bộ. Bên cạnh đó là rất nhiều ứng dụng trong việc vận chuyển hàng hóa, hay thậm chí cả trong quân sự (trình sát, do thám...).

1.2. Nguyên lý hoạt động của quadcopter

Quadcopter được thiết kế trong đồ án này có dạng dấu nhân “X”. Máy bay có 2 cặp cánh quạt thuận nghịch, với các cánh đặt đối diện nhau là cùng phía. Bên cạnh việc đóng góp vào lực đẩy để lấy độ cao cho máy bay, các cặp cánh còn tạo ra các chuyển động của máy bay xung quanh trọng tâm. Bằng cách tăng và giảm đồng thời tốc độ các động cơ đặt cùng bên, sẽ xuất hiện mô-men lực làm máy bay nghiêng một góc. Từ đó ta có thể điều khiển máy bay di chuyển tịnh tiến theo hướng ngang hoặc dọc so với hướng của máy bay.

Với các cánh quạt quay giống nhau quay ở cùng tốc độ, các mô-men cản do không khí tác dụng lên các cánh quạt sẽ triệt tiêu lẫn nhau. Áp dụng nguyên tắc này, ta có thể ổn định góc hướng của máy bay theo hướng mong muốn, bằng cách thay đổi đồng thời tốc độ (cùng tăng hoặc cùng giảm) của cặp cánh đối diện nhau.



Hình 1.1 Mô tả các chuyển động của Quadcopter

Với trạng thái đứng yên (lơ lửng) của máy bay các cánh quạt của máy bay quay giống nhau quay ở cùng tốc độ, các mô-men cản do không khí tác dụng lên các cánh quạt sẽ triệt tiêu lẫn nhau. Lực nâng của 4 động cơ cân bằng với trọng lượng máy bay giúp máy bay giữ nguyên trạng thái. Khi muốn máy bay di chuyển lên hoặc xuống ta chỉ cần thay đổi tốc độ của 4 động cơ.

Di chuyển tiến về phía trước (Move forward) bằng cách giảm tốc độ cặp động cơ phía trước máy bay, tăng tốc độ 2 động cơ còn lại.

Di chuyển lùi về phía sau (Move backward) bằng cách giảm tốc độ cặp động cơ phía sau và tăng tốc độ 2 động cơ phía trước.

Di chuyển sang bên trái (Bend left) bằng cách giảm tốc độ 2 động cơ bên trái và tăng tốc độ 2 động cơ bên phải.

Di chuyển sang bên phải (Bend right) bằng cách giảm tốc độ 2 động cơ bên phải và tăng tốc độ 2 động cơ bên trái.

Xoay tròn tại chỗ (Yaw control) bằng cách thay đổi tốc độ hai động cơ đối xứng nhau, dẫn đến mô men xoắn thay đổi làm máy bay xoay tại chỗ.

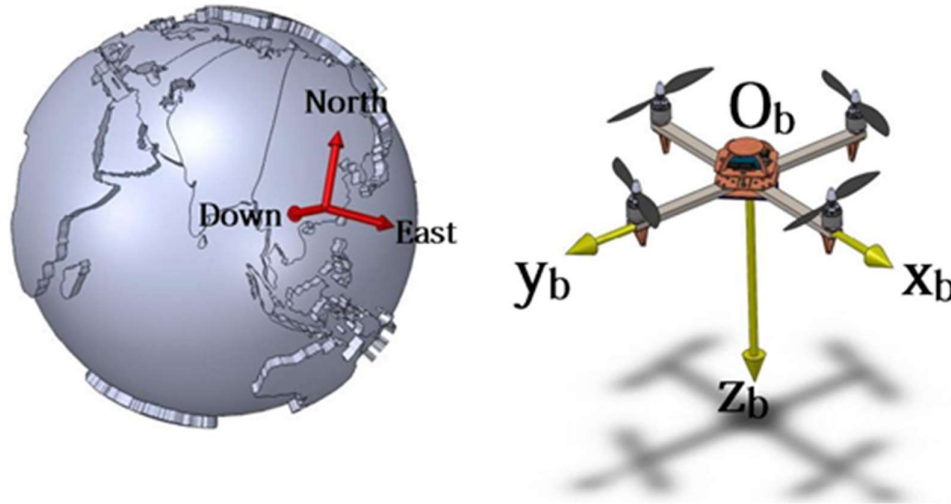
1.3. Tìm hiểu về mô hình tuyến tính máy bay

1.3.1. Hệ quy chiếu

Trước khi phân tích các yếu tố động lực học đối với máy bay ta cần xác định các hệ quy chiếu. Hệ quy chiếu cho phép người quan sát mô tả các chuyển động của máy bay. Những người quan sát khi đứng trong các hệ quy chiếu khác nhau sẽ có cảm nhận khác nhau về chuyển động của cùng một chiếc máy bay tại cùng một thời điểm. Trong các hệ quy chiếu có gắn các hệ trục tọa độ. Các hệ tọa độ này cho phép tính toán những thay đổi mang tính định lượng về trạng thái động học của máy bay đó. Hệ tọa độ được sử dụng phổ biến là hệ Decartes. Để mô tả các chuyển động của máy bay, cần sử dụng hai hệ quy chiếu sau:

- Hệ quy chiếu quán tính: là hệ quy chiếu không có gia tốc, ở đó các định luật Newton được thỏa mãn. Hệ trục tọa độ gắn với hệ này có gốc tọa độ (kí hiệu: OE) được gắn với một vật mốc cố định so với Trái Đất. Ta kí hiệu hệ là NED do ba trục tọa độ gắn với hệ x_E , y_E , z_E được quy ước lần lượt chỉ theo các hướng bắc (North), đông (East) và hướng về tâm Trái Đất.

- Hệ quy chiếu gắn với vật thể bay (Hệ vật): Hệ quy chiếu được gắn cố định và chuyển động cùng với vật thể bay. Gốc OB của hệ tọa độ gắn với hệ vật được lấy trùng với trọng tâm của máy bay (CG). Các trục tọa độ được kí hiệu lần lượt là x_B , y_B , z_B [hình vẽ]

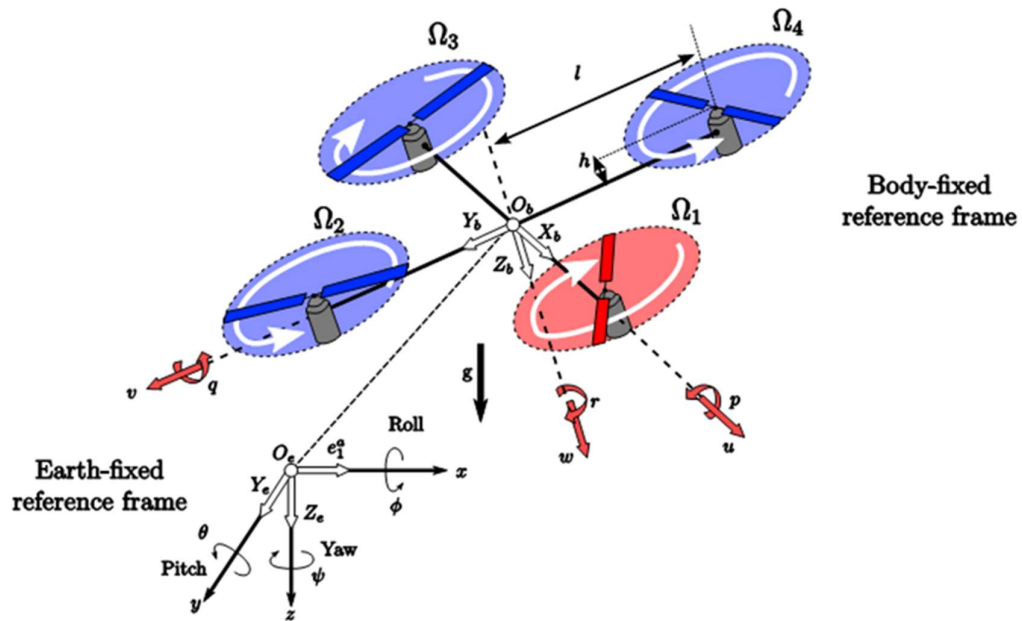


Hình 1.2 Hệ tọa độ vật (hình trái) và hệ tọa độ quán tính (hình phải)

1.3.2. Các góc Euler

Các góc được quay theo quy tắc tay phải xung quanh ba trục tọa độ của một hệ trục thuận được định nghĩa là các góc Euler. Các góc quay không phải là đại lượng vec-tơ, do đó chúng không thỏa mãn tính chất giao hoán. Bởi vậy, thứ tự tiến hành các phép quay là rất quan trọng. Người ta quy ước thực hiện các phép quay theo trình tự sau để đưa hệ trục gắn với máy bay trùng với hệ Trái Đất:

- $O_Bx_By_Bz_B$ quay một góc roll ϕ ($-\pi/2 < \phi < \pi/2$) xung quanh trục O_Bx_B ;
- Quay một góc pitch θ ($-\pi/2 < \theta < \pi/2$) xung quanh trục O_By_B ;
- Quay một góc yaw ψ ($-\pi \leq \psi < \pi$) xung quanh trục O_Bz_B và trùng với $O_Ex_Ey_Ez_E$.



Hình 1.3 Quy ước các phương chiều

1.3.3. Ma trận chuyển đổi hệ trục tọa độ

Theo lý thuyết điều khiển, các ứng xử động học của một vật được mô tả thông qua các biến trạng thái các biến này bao gồm:

- Các góc Euler giữa hệ vật và hệ quán tính: ϕ, θ, ψ .
- Các thành phần vận tốc góc của máy bay chiếu trên các trục tọa độ gắn với hệ vật: $O_b x_b, O_b y_b, O_b z_b$ lần lượt là p, q, r .
- Các thành phần cho biết vị trí của máy bay trong hệ tọa độ O_E, x_E, y_E, z_E : x, y, z .
- Các thành phần vận tốc dài xét trong hệ tọa độ vật được chiếu lên ba trục: $O_b x_b, O_b y_b, O_b z_b$ lần lượt là u, v, w .

Các biến trạng thái đo được trực tiếp bằng các cảm biến điện tử gắn trên máy bay có giá trị đo được trong hệ tọa độ vật. Vì vậy, ta phải sử dụng các ma trận chuyển hệ trục tọa độ, để chuyển đổi các giá trị đo bởi cảm biến về giá trị

xác định trong hệ quán tính. Ma trận này thực chất được xây dựng từ trình tự quay ba góc Euler như đã trình bày trong phần trước.

Đối với các đại lượng dài, ta có mối liên hệ giữa vận tốc dài xét trong 2 hệ trục như sau:

$$\begin{bmatrix} \dot{x}_E \\ \dot{y}_E \\ \dot{z}_E \end{bmatrix} = D^{-1} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

Trong đó:

$$E^{-1} = \begin{bmatrix} 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \\ 0 & \cos \theta & -\sin \theta \\ 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \end{bmatrix}$$

1.3.4. Hệ phương trình chuyển động

- Các giả thiết

Để khảo sát chuyển động của quadcopter, ta cần xây dựng hệ phương trình động học mô tả chuyển động của máy bay. Ta đơn giản hóa việc này bằng cách đưa ra một số giả thiết sau đây:

Bỏ qua các yếu tố biến dạng đàn hồi ở kết cấu khung quadcopter, coi máy bay là cứng tuyệt đối, máy bay là đối xứng, bỏ qua hiệu ứng mặt đất.

Các yếu tố này sẽ có tác động tương đối nhỏ hoặc chỉ trong khoảng thời gian nhỏ so với tổng thời gian máy bay hoạt động. Do đó ta có thể bỏ qua các ảnh hưởng này để đơn giản hóa các phương trình chuyển động. Phương pháp này đã được chứng minh về tính hiệu quả trong thực tiễn thông qua các nghiên cứu trước đó. Việc coi kết cấu khung là cứng tuyệt đối đã bỏ qua ảnh hưởng động học của các mối ghép cơ khí, độ đàn hồi của kết cấu khung và cánh quạt, cho phép đưa bài toán về phân tích một vật thể cứng tuyệt đối có 6 bậc tự do. Việc coi máy bay đối xứng qua các mặt phẳng Obxbyb, Obybzb, Obzbxb cho phép bỏ qua các thành phần mô-men tích quán tính.

- Hệ phương trình động học
 - Các phương trình cân bằng lực

Kí hiệu tổng các ngoại lực tác động lên máy bay là: $\mathbf{F} = [F_x \quad F_y \quad F_z]$ với các thành phần F_x, F_y, F_z xác định bởi hệ tọa độ vật.

Theo định luật 2 Newton ta có:

$$\mathbf{F} = m\dot{\mathbf{V}}$$

Với $\mathbf{V} = (u, v, w)$ là vec-tơ vận tốc tuyệt đối của máy bay xét trong hệ tọa độ vật.

Theo định luật Transport, ta có:

$$\left(\frac{d}{dt}\mathbf{V}\right)_E = \left(\frac{d}{dt}\mathbf{V}\right)_B + \boldsymbol{\omega} \times \mathbf{V}$$

$$m\left(\frac{d}{dt}\mathbf{V}\right)_E = m\left(\frac{d}{dt}\mathbf{V}\right)_B + \boldsymbol{\omega} \times m\mathbf{V}$$

Trong đó: $\boldsymbol{\omega} = (p, q, r)$

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

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

Do máy bay được thiết kế chuyển động với tốc độ chậm, có thể bỏ qua các ảnh hưởng của các lực khí động. Các ngoại lực không thể bỏ qua gồm lực đẩy của bốn động cơ T và trọng lượng của máy bay W.

Do các thành phần F_x, F_y, F_z xác định bởi hệ tọa độ vật, cần đưa các lực về xét trong hệ tọa độ vật thông qua ma trận chuyển hệ trục tọa độ D.

$$\mathbf{W} + \mathbf{T} = m \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix}$$

$$D \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} = m \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix}$$

Sau khi tính toán với ma trận, ta viết lại dưới dạng hệ phương trình như sau:

$$\dot{u} = rv - qw - g \sin \phi$$

$$\dot{v} = pw - ru + g \cos \theta \sin \phi$$

$$\dot{w} = qu - pv + g \cos \theta \cos \phi - \frac{U_1}{m}$$

Trong đó:

$$U_1 = \sum_{i=1}^4 T_i = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2)$$

T_i (i=1,4): là lực đẩy của lần lượt 4 động cơ.

$\Omega_1, \Omega_2, \Omega_3, \Omega_4$: là tốc độ góc của 4 động cơ.

(Theo lý thuyết phân tử cánh, lực đẩy tạo ra do cánh quạt tỉ lệ với tốc độ góc của cánh theo một hằng số nếu coi biên dạng cánh và điều kiện môi trường xung quanh là không đổi.)

- Các phương trình cân bằng mô-men

Gọi tổng các mô-men ngoại lực tác động lên máy bay là: $\mathbf{M} = [M_x \quad M_y \quad M_z]$

với các thành phần M_x, M_y, M_z xác định bởi hệ tọa độ vật.

Theo định luật 2 Newton ta có:

$$\mathbf{M} = I \dot{\boldsymbol{\omega}}$$

Trong đó:

$\boldsymbol{\omega} = (p, q, r)$ là vec-tơ vận tốc góc tuyệt đối của máy bay xét trong hệ tọa độ vật.

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}, \text{ tensor mô-men quán tính của máy bay.}$$

Theo định luật Transport [12,12], ta có:

$$\left(\frac{d}{dt} \boldsymbol{\omega} \right)_E = \left(\frac{d}{dt} \boldsymbol{\omega} \right)_B + \boldsymbol{\omega} \times \boldsymbol{\omega}$$

$$I \left(\frac{d}{dt} \boldsymbol{\omega} \right)_E = I \left(\frac{d}{dt} \boldsymbol{\omega} \right)_B + \boldsymbol{\omega} \times I \boldsymbol{\omega}$$

Trong đó: $\boldsymbol{\omega} = (p, q, r)$

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} I_{xx}\dot{p} - (I_{yy} - I_{zz})qr \\ I_{yy}\dot{q} - (I_{zz} - I_{xx})rp \\ I_{zz}\dot{r} - (I_{xx} - I_{yy})pq \end{bmatrix}$$

Coi máy bay đối xứng qua các mặt phẳng $O_b x_b z_b$, $O_b y_b z_b$ nên ta có $I_{xx} \approx I_{yy}$, do đó phương trình được rút gọn thành:

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} I_{xx}\dot{p} - (I_{yy} - I_{zz})qr \\ I_{yy}\dot{q} - (I_{zz} - I_{xx})rp \\ I_{zz}\dot{r} \end{bmatrix}$$

Mô-men ngoại lực gây ra chủ yếu do lực đẩy tạo ra bởi cánh quạt và lực cản của không khí. Một cách đơn giản, ta có thể coi các thành phần của vec-tơ mô-men ngoại lực như sau:

$$M_x = (-T_2 + T_4)l = bl(\Omega_4^2 - \Omega_2^2)$$

$$M_y = (T_1 - T_3)l = bl(\Omega_1^2 - \Omega_3^2)$$

$$M_z = ld(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2)$$

Trong đó:

l: chiều dài cánh tay đòn, tính bằng khoảng cách từ điểm đặt lực trên cánh tay đòn đến vị trí trọng tâm máy bay.

b: hệ số lực đẩy

d: hệ số mô-men cản. (Theo lý thuyết phần tử cánh, mô-men cản do lực khí động gây ra trên cánh quạt tỉ lệ với tốc độ góc của cánh theo một hằng số nếu coi biên dạng cánh và điều kiện môi trường xung quanh là không đổi)

Kí hiệu:

$$U_2 = -T_2 + T_4 = b(\Omega_4^2 - \Omega_2^2)$$

$$U_3 = T_1 - T_3 = b(\Omega_1^2 - \Omega_3^2)$$

$$U_4 (= T_1 - T_2 + T_3 - T_4) = d(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2)$$

Cuối cùng, ta thu được hệ phương trình cân bằng mô-men:

$$\dot{p} = \frac{I_{yy} - I_{zz}}{I_{xx}} qr + \frac{l}{I_{xx}} U_2$$

$$\dot{q} = \frac{I_{zz} - I_{xx}}{I_{yy}} pr + \frac{l}{I_{yy}} U_3$$

$$\dot{r} = \frac{U_4}{I_{zz}}$$

- Bổ sung hiệu ứng gyroscope của cánh quạt

Việc xây dựng hệ phương trình cân bằng mô-men cho quadcopter dựa trên giả thiết toàn bộ máy bay là vật rắn tuyệt đối và bỏ qua các chuyển động tương đối giữa các thành phần trong hệ thống. Trong thực tế, các cánh quạt khi quay tạo ra các mô-men gyroscope trên các trục còn lại của máy bay. Vì vậy, ta xét thêm các thành phần này vào tổng mô-men ngoại lực của quadcopter. Hệ phương trình cân bằng mô-men cuối cùng thu được là:

$$\dot{p} = \frac{I_{yy} - I_{zz}}{I_{xx}} qr + \frac{l}{I_{xx}} U_2 - \frac{J_r \Omega_r}{I_{xx}} q$$

$$\dot{q} = \frac{I_{zz} - I_{xx}}{I_{yy}} pr + \frac{l}{I_{yy}} U_3 - \frac{J_r \Omega_r}{I_{yy}} p$$

$$\dot{r} = \frac{U_4}{I_{zz}}$$

Trong đó:

Ω_r : tốc độ quay của cánh quạt.

J_r : mô-men quán tính của cánh quạt so với trục quay.

Mô hình tuyến tính của quacopter

Để xây dựng phương trình trạng thái cho 12 biến trạng thái của Quadcopter, bên cạnh 3 phương trình cân bằng lực và 3 phương trình cân bằng mô-men, ta cần bổ sung thêm 6 phương trình, được xây dựng từ các ma trận chuyển hệ trục tọa độ:

$$\dot{\phi} = p + q \sin \phi \tan \theta + r \cos \phi \tan \theta$$

$$\dot{\theta} = q \cos \phi - r \sin \phi$$

$$\dot{\psi} = q \sin \phi \frac{1}{\cos \theta} + r \frac{\cos \phi}{\cos \theta}$$

$$\dot{x} = \cos \psi \cos \theta u + (\cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi) v + (\cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi) w$$

$$\dot{y} = \sin \psi \cos \theta u + (\sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi) v + (\sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi) w$$

$$\dot{z} = -\sin \theta u + \cos \theta \sin \phi v + \cos \theta \cos \phi w$$

Với mục đích giới hạn các chuyển động của máy bay chỉ trong phạm vi các di chuyển đơn giản và xác lập, bằng cách xác định các chế độ hoạt động cơ bản của Quadcopter, ta có thể tuyến tính hóa hệ phương trình động học tổng quát về hệ tuyến tính quanh các điểm làm việc chính. Việc đưa hệ phi tuyến thay đổi theo thời gian về hệ tuyến tính không đổi theo thời gian cho phép ta tiếp cận hệ thống bằng nhiều công cụ khảo sát ổn định trong miền tần số, giúp người thiết kế hiểu sâu hơn bản chất của hệ thống. Đồng thời cách tiến hành này cho phép tách bài toán điều khiển hệ MIMO thành các bài toán điều khiển các kênh SISO riêng rẽ. Trên mỗi hệ SISO, ta có thể quan sát sự tác động của các thông số động học và tín hiệu điều khiển đến từng trạng thái một cách độc lập. Tóm lại, việc tuyến tính hóa hệ phương trình mô tả động học của Quadcopter giúp giảm độ phức tạp và khối lượng tính toán trong việc thiết kế bộ điều khiển cho hệ thống, cho phép người thiết kế có cái nhìn sâu sắc hơn vào bản chất của hệ thống. Tuy nhiên, phương pháp chỉ ứng dụng được hiệu quả trong các môi trường không có nhiễu khí động lớn...

Việc tiến hành tuyến tính hóa hệ phương trình phi tuyến dựa trên lý thuyết xấp xỉ một hàm số bằng khai triển Taylor và lý thuyết nhiễu nhỏ. Dựa trên thực tế rằng các yếu tố khí động tác động chính tới máy bay là hàm tuyến tính của các nhiễu nhỏ, người ta giả thiết máy bay chịu tác động bởi những thay đổi nhỏ quanh trạng thái dừng ổn định. Phương pháp cho độ chính xác tốt trong các ứng dụng cho máy bay chuyển động đều.

Các trạng thái và tín hiệu điều khiển ở điểm cân bằng là các hằng số, được kí hiệu:

$$u_o, v_o, w_o, p_o, q_o, r_o, \phi_o, \theta_o, \psi_o, x_o, y_o, z_o, U_{1o}, U_{2o}, U_{3o}, U_{4o}$$

Mối liên hệ giữa các biến trạng thái, các tín hiệu điều khiển với các biến nhiễu nhỏ:

$$\begin{aligned}
 u &= u_o + \Delta u & \psi &= \psi_o + \Delta \psi \\
 v &= v_o + \Delta v & x &= x_o + \Delta x \\
 w &= w_o + \Delta w & y &= y_o + \Delta y \\
 p &= p_o + \Delta p & z &= z_o + \Delta z \\
 q &= q_o + \Delta q & U_1 &= U_{1o} + \Delta U_1 \\
 r &= r_o + \Delta r & U_2 &= U_{2o} + \Delta U_2 \\
 \phi &= \phi_o + \Delta \phi & U_3 &= U_{3o} + \Delta U_3 \\
 \theta &= \theta_o + \Delta \theta & U_4 &= U_{4o} + \Delta U_4
 \end{aligned}$$

Do giả sử rằng các đại lượng nhiễu nhỏ và đạo hàm của chúng là bé, ta coi tích của hai đại lượng hoặc bình phương của một đại lượng xấp xỉ bằng 0. Ngoài ra, đối với các thành phần lượng giác, ta coi $\sin \Delta \theta = 0$, $\cos \Delta \theta = 1$. Giả thiết áp dụng tương tự với các góc ϕ và ψ .

Xét khai triển Taylor của một hàm gồm biến trạng thái v và biến điều khiển U $f(v, U)$ xung quanh trạng thái ổn định (v_o, U_o) (tại đây, việc xét hàm số cho hai biến vẫn đảm bảo tính tổng quát của khai triển đối với hàm có số biến bất kì):

$$\dot{v} = \frac{dv}{dt} = f(v, U)$$

Áp dụng khai triển Taylor cho các hàm trên, bỏ qua các đại lượng có bậc ≥ 2 , ta có:

$$f(v, U) \approx f(v_o, U_o) + \frac{\partial f}{\partial v}_{v=v_o} (v - v_o) + \frac{\partial f}{\partial U}_{U=U_o} (U - U_o)$$

Từ giả thiết của phương pháp nhiễu nhỏ, ta có:

$$v = v_o + \Delta v$$

$$U = U_o + \Delta U$$

và

$$f(v_o, U_o) = \frac{dv_o}{dt} = 0$$

suy ra

$$f(v, U) = \frac{\partial f}{\partial v_{v=v_o}} \Delta v + \frac{\partial f}{\partial U_{U=U_o}} \Delta U$$

Mặt khác, do

$$\frac{dv_o}{dt} = 0$$

Nên

$$\frac{dv}{dt} = \frac{d(v - v_o)}{dt} = \frac{d\Delta v}{dt}$$

Vậy, dựa trên lý thuyết nhiễu nhỏ, ta thu được hàm Taylor khai triển cho hàm $f(v, U)$ như sau:

$$\Delta \dot{v} = \frac{\partial f}{\partial v_{v=v_o}} \Delta v + \frac{\partial f}{\partial U_{U=U_o}} \Delta U$$

Áp dụng lý thuyết nhiễu nhỏ và khai triển Taylor, ta có thể tuyến tính hóa 12 phương trạng thái của quadcopter về dạng tuyến tính:

$$\Delta \dot{u} = r_o \Delta v - q_o \Delta w - w_o \Delta q + v_o \Delta r - g \Delta \theta \cos \theta_o$$

$$\begin{aligned} \Delta \dot{v} = & -r_o \Delta u + p_o \Delta w + w_o \Delta p - u_o \Delta r + g \Delta \phi \cos \theta_o \cos \phi_o \\ & - g \Delta \theta \sin \theta_o \sin \phi_o \end{aligned}$$

$$\Delta \dot{w} = q_o \Delta u - p_o \Delta v - v_o \Delta p + u_o \Delta q - g \Delta \phi \cos \theta_o \sin \phi_o - g \Delta \theta \sin \theta_o \cos \phi_o - \frac{\Delta U_1}{m}$$

$$\Delta \dot{p} = \frac{I_{yy} - I_{zz}}{I_{xx}} r_o \Delta q + \frac{I_{yy} - I_{zz}}{I_{xx}} q_o \Delta r + \frac{l}{I_{xx}} \Delta U_2 - \frac{J_r \Omega_r}{I_{xx}} \Delta q$$

$$\Delta \dot{q} = \frac{I_{zz} - I_{xx}}{I_{yy}} r_o \Delta p + \frac{I_{zz} - I_{xx}}{I_{yy}} p_o \Delta r + \frac{l}{I_{yy}} \Delta U_3 - \frac{J_r \Omega_r}{I_{yy}} \Delta p$$

$$\dot{\Delta} \mathbf{r} = \frac{1}{I_{zz}} \Delta U_4$$

$$\begin{aligned} \dot{\Delta} \phi = & \Delta \mathbf{p} + \sin \phi_o \tan \theta_o \Delta \mathbf{q} + \cos \phi_o \tan \theta_o \Delta \mathbf{r} + (\tan \theta_o \cos \phi_o q_o \\ & - \tan \theta_o \sin \phi_o r_o) \Delta \phi \\ & + (\sin \phi_o \sec^2 \theta_o q_o + r_o \cos \phi_o \sec^2 \theta_o) \Delta \theta \end{aligned}$$

$$\dot{\Delta} \theta = \cos \phi_o \Delta \mathbf{q} - \sin \phi_o \Delta \mathbf{r} - (q_o \sin \phi_o + r_o \cos \phi_o) \Delta \phi$$

$$\begin{aligned} \dot{\Delta} \psi = & \sec \theta_o \sin \phi_o \Delta \mathbf{q} + \cos \phi_o \sec \theta_o \Delta \mathbf{r} + (q_o \sec \theta_o \\ & - r_o \sec \theta_o \sin \phi_o) \Delta \phi + (q_o \sin \phi_o \sec \theta_o \tan \theta_o \\ & + r_o \cos \phi_o \sec \theta_o \tan \theta_o) \Delta \theta \end{aligned}$$

$$\begin{aligned} \dot{\Delta} x = & \cos \psi_o \cos \theta_o \Delta \mathbf{u} + (\cos \psi_o \sin \theta_o \sin \phi_o - \sin \psi_o \cos \phi_o) \Delta \mathbf{v} \\ & + (\cos \psi_o \sin \theta_o \cos \phi_o + \sin \psi_o \sin \phi_o) \Delta \mathbf{w} \\ & + (\cos \psi_o \sin \theta_o \cos \phi_o v_o + \sin \psi_o v_o \sin \phi_o \\ & - \cos \psi_o \sin \theta_o \sin \phi_o w_o + \sin \psi_o w_o \cos \phi_o) \Delta \phi \\ & + (-\cos \psi_o u_o \sin \theta_o + \cos \psi_o \sin \phi_o v_o \cos \theta_o \\ & + \cos \psi_o \cos \phi_o \cos \theta_o w_o) \Delta \theta + (-\sin \psi_o \cos \theta_o u_o \\ & - \sin \psi_o \sin \theta_o \sin \phi_o v_o - \cos \psi_o \cos \phi_o v_o \\ & - \sin \psi_o \sin \theta_o \cos \phi_o w_o + \cos \psi_o \sin \phi_o w_o) \Delta \psi \end{aligned}$$

$$\begin{aligned} \dot{\Delta} y = & \sin \psi_o \cos \theta_o \Delta \mathbf{u} + (\sin \psi_o \sin \theta_o \sin \phi_o + \cos \psi_o \cos \phi_o) \Delta \mathbf{v} \\ & + (\sin \psi_o \sin \theta_o \cos \phi_o - \cos \psi_o \sin \phi_o) \Delta \mathbf{w} \\ & + (\sin \psi_o \sin \theta_o \cos \phi_o v_o - \cos \psi_o \sin \phi_o v_o \\ & - \sin \psi_o \sin \theta_o \sin \phi_o w_o - \cos \psi_o \cos \phi_o w_o) \Delta \phi \\ & + (-\sin \psi_o \sin \theta_o u_o + \sin \psi_o \sin \phi_o \cos \theta_o v_o \\ & + \sin \psi_o \cos \theta_o \cos \phi_o w_o) \Delta \theta + (\cos \psi_o \cos \theta_o u_o \\ & + \cos \psi_o \sin \theta_o \sin \phi_o v_o - \sin \psi_o \cos \phi_o v_o \\ & + \cos \psi_o \sin \theta_o \cos \phi_o w_o + \sin \psi_o \sin \phi_o w_o) \Delta \psi \end{aligned}$$

$$\begin{aligned}\dot{\Delta z} = & -\sin \theta_o \Delta u + \cos \theta_o \sin \phi_o \Delta v + \cos \theta_o \cos \phi_o \Delta w \\ & + (\cos \theta_o v_o \cos \phi_o - \cos \theta_o w_o \sin \phi_o) \Delta \phi + (-\cos \theta_o u_o \\ & - \sin \phi_o \sin \theta_o v_o - \sin \theta_o \cos \phi_o w_o) \Delta\end{aligned}$$

1.4. Mô hình động cơ và cánh quạt

Thiết bị tạo lực đẩy được sử dụng cho mô hình là động cơ một chiều không chổi than (Brushless DC motor - BLDC). Mặc dù việc chế tạo phức tạp và có giá thành cao hơn so với động cơ một chiều có chổi than, động cơ BLDC vẫn thường được lựa chọn để chế tạo mô hình máy bay bởi nhiều ưu điểm vượt trội như sau:

- Cho mô-men xoắn lớn hơn so với động cơ DC cùng khối lượng.
- Cho hiệu suất mô-men xoắn so với công suất tiêu thụ lớn hơn.
- Độ tin cậy cao, tuổi thọ cao và ít phải bảo trì do không phải hoạt động thông qua cơ cấu tiếp xúc trực tiếp chổi than.
- Dễ tích hợp với hệ thống điều khiển số.

Để đơn giản cho việc điều khiển động cơ, ta sử dụng bộ điều tốc điện tử (Electronic Speed Controller - ESC), cho phép điều khiển động cơ bằng vi điều khiển thông qua điều chỉnh độ rộng xung (Pulse width modulation/PWM). Thông thường, các ESC nhận các xung điều khiển có tần số 50Hz với các độ rộng xung từ 1ms đến 2ms để điều khiển động cơ lần lượt ứng với mức tốc độ từ thấp nhất tới cao nhất.

Tổ hợp ESC, động cơ và cánh quạt cho phép điều khiển lực đẩy tác dụng lên máy bay theo tín hiệu xung cấp từ bộ điều khiển thông qua việc biến đổi tốc độ của động cơ.

Để mô phỏng các ứng xử động học của quadrotor, đầu tiên cần xây dựng mô hình toán học của cụm ESC, động cơ và cánh quạt. Hàm truyền của động cơ một chiều bao gồm một khâu tỉ lệ và một khâu quán tính bậc hai với hai điểm cực lần lượt liên quan tới yếu tố cơ khí và điện của động cơ. Do yếu tố

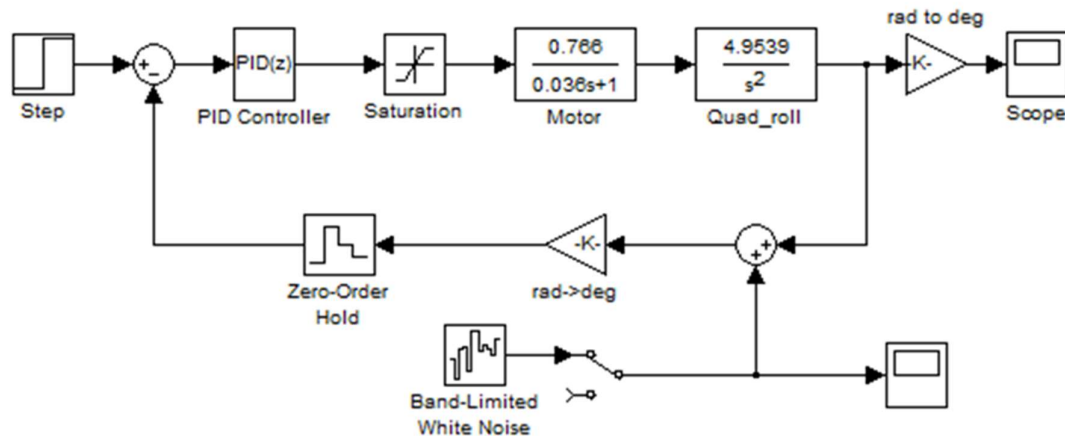
điện có ảnh hưởng không đáng kể (nhỏ hơn so với ảnh hưởng của mô-men quán tính của động cơ), ta có thể loại bỏ một điểm cực và coi gần đúng hàm truyền động cơ là khâu quán tính bậc nhất. Như vậy, hàm truyền cần xác định có dạng:

$$G = \frac{K}{1 + \tau s}$$

Việc nhận dạng mô hình động cơ được tiến hành thông qua phương pháp thực nghiệm. Các thí nghiệm được tiến hành để xác định các hằng số K và τ của hàm truyền G. Trong đó K cho biết tỷ lệ giữa lực đẩy sinh ra do cánh quạt và độ rộng xung tín hiệu cấp cho ESC, hằng số thời gian τ cho biết tốc độ đáp ứng của cụm ESC và động cơ, là khoảng thời gian để đạt được tín hiệu đầu ra mong muốn từ khi có tín hiệu điều khiển.

1.5. Mô phỏng tìm hệ số PID

Sau khi đã tìm được các hàm truyền đạt, ta có thể thiết lập mô phỏng để tìm hệ số PID theo sơ đồ sau:



Hình 1.4 Sơ đồ mô phỏng để tìm hệ số PID theo sơ đồ sau:

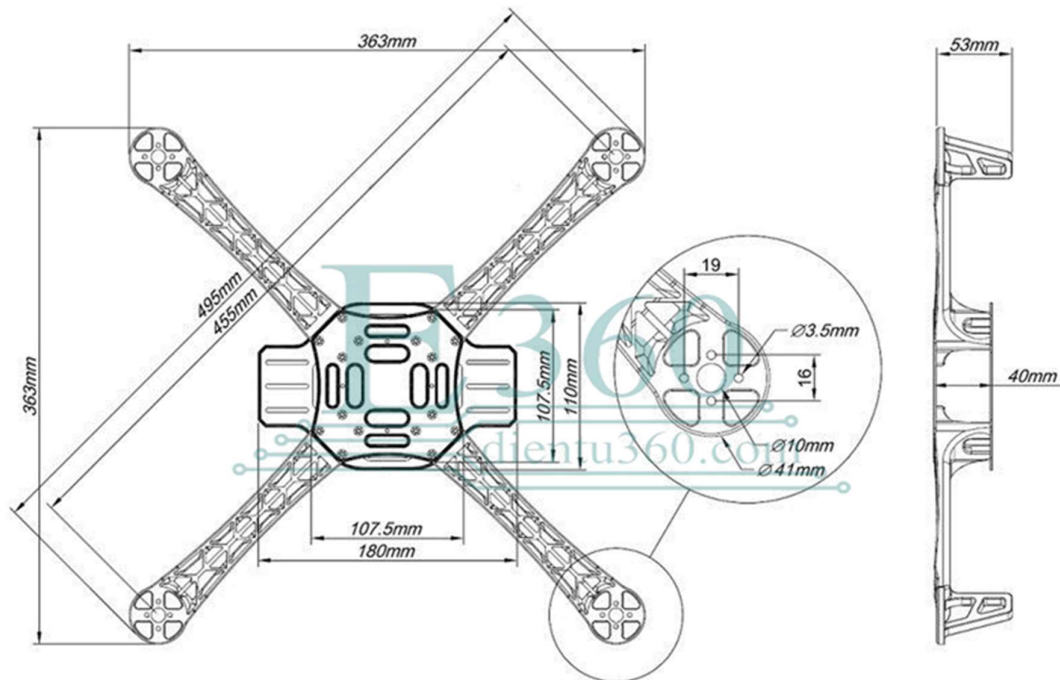
Chương 2. Các bước thiết kế quadcopter

2.1. Thiết kế cơ khí và mạch điện tử

2.1.1. Thiết kế cơ khí

Khung máy bay được thiết kế với hình dạng chữ “X”, cân đối với nhau để trọng tâm của máy bay ở chính giữa khung. Bên cạnh đó, một thông số thiết kế quan trọng khác là khoảng cách từ trục động cơ tới trọng tâm máy bay cũng cần được thiết kế chính xác. Giúp cảm biến có thể gửi dữ liệu trạng thái máy bay với sai số nhỏ nhất.

Do vậy chúng em lựa chọn khung máy bay đã được thiết kế sẵn. Với nhiều ưu điểm như: Chất liệu bền, nhẹ, sai số nhỏ, biến dạng lực nhỏ.



Hình 2.1 Thiết kế khung cơ khí

Bỏ qua các sai số do chế tạo, ta coi lực đẩy của động cơ vuông góc với mặt phẳng xy của máy bay và có phương trùng với trục động cơ. Khi đó mô-men do lực đẩy động cơ tác động lên máy bay có thể tính toán được và bằng tích của trị số lực đẩy nhân với khoảng cách từ trục động cơ tới trọng tâm máy bay.

2.1.2. Thiết kế điện tử

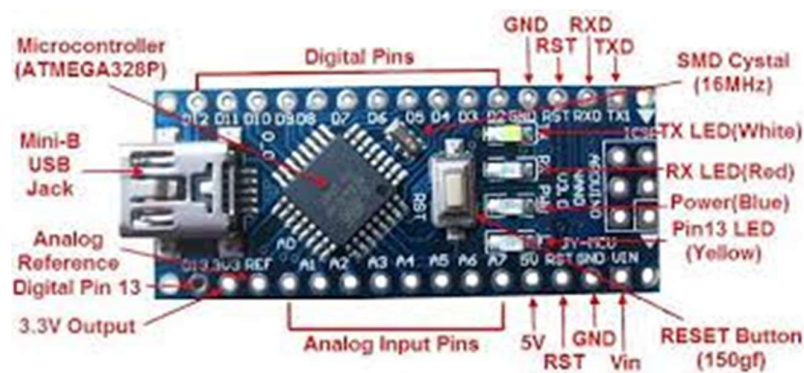
Các thiết bị điện tử được lựa chọn sử dụng trên mô hình:

- **Arduino nano**

Việc cân bằng quadcopter trong không gian rất khó vì nó trôi nổi trong không gian không có điểm tựa vì vậy luôn có xu hướng mất ổn định, vì vậy máy bay cần một hệ thống tự động cân bằng để ổn định.

Vi điều khiển Atmega 328P là vi điều khiển 8 bit có 4x16 ngõ vào, và có 4 bộ phát xung PWM, cho phép xử lý các tác vụ nhanh chóng. Đồng thời, nó cho phép giao tiếp qua các giao thức phổ biến như UART, SPI, I2C, Serial... cho phép kết nối với các cảm biến và giao tiếp với máy tính và các vi điều khiển khác một cách thuận tiện.

Arduino nano là sự tiện dụng, đơn giản có thể lập trình trực tiếp bằng máy tính, tích hợp sẵn vi điều khiển Atmega 328P và đặc biệt hơn cả là kích thước của nó. Kích thước arduino cực kì nhỏ (1.85cm x4.3 cm), giá thành rẻ hơn arduino uno nhưng dùng được tất cả các thư viện của mạch này.



Hình 2.2 Arduino nano

Vi điều khiển	Atmega 328 (họ 8 bit)
Điện áp hoạt động	5V-DC
Tần số hoạt động	16 MHz
Dòng điện tiêu thụ	30mA
Điện áp khuyến áp khuyến dùng	7-12 V DC
Điện áp giới hạn	6-20 V DC
Số chân digital	14 chân (6 chân PWN)
Số chân analog	8 chân (độ phân giải 10bit)
Dòng điện tối đa trên mỗi chân I/O	40 mA
Dòng ra tối đa 5V	500mA
Dòng ra tối đa (3,3 v)	50mA
Bộ nhớ flash	32 KB (ATmega328) với 2KB dùng bởi bootloade
SRAM	2KB (Atmega328)
EEPROM	1KB (Atmega328)
Kích thước	1.85x4.3 cm

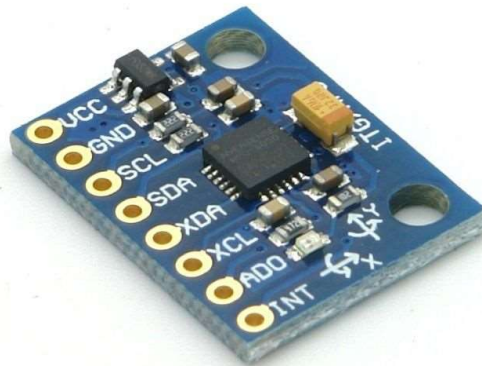
Bảng 2.1 Thông số của kỹ thuật của Arduino nano

- **Cảm biến IMU MPU-6050**

Trong hệ thống điều khiển, khối đo lường quán tính (IMU) bao gồm các cảm biến: gyro, gia tốc kế, từ kế... đóng vai trò đặc biệt quan trọng. Chúng cho phép xác định trạng thái (vị trí, độ cao, vận tốc, gia tốc, góc nghiêng...) của máy bay theo thời gian và không gian. Độ chính xác của các cảm biến đóng vai trò quan trọng bởi đây là nguồn tín hiệu đầu vào cho bộ vi xử lý. Công nghệ MEMS (Hệ thống vi cơ điện tử) cho ra đời các cảm biến quán tính có kích thước nhỏ, với giá thành rẻ, dễ dàng tích hợp với các UAV nói chung và quadcopter nói riêng.

Cảm biến MPU-6050 có 6 bậc tự do gồm một gyroscope 3 trục và một gia tốc kế 3 trục), cho phép xác định 6 thông số quán tính độc lập bao gồm: các gia tốc dài theo các phương và các thành phần vận tốc góc xoay quanh các trục của máy bay. Bên trong cảm biến có một vi xử lý, thực hiện việc tính toán và chuyển đổi ngay từ giá trị đo tương tự sang giá trị số, làm giảm khối lượng công việc cho vi điều khiển. Các giá trị tính toán được truyền về vi điều khiển thông qua giao tiếp I2C.

Giao tiếp I2C có tốc độ truyền gửi dữ liệu thấp hơn so với giao tiếp SPI nhưng yêu cầu ít dây hơn để kết nối (một dây SCK giữ xung nhịp để đồng bộ tốc độ giữa các vi xử lý và một dây SDA để truyền dữ liệu) và số dây này không tăng lên khi số lượng cảm biến kết nối với vi điều khiển tăng.



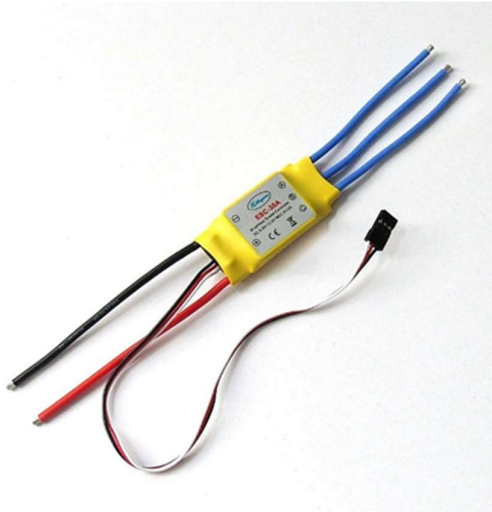
Hình 2.3 Cảm biến MPU-6050

Giao tiếp	I2C
Độ phân giải	16 bits
Giải đo gyroscope	$\pm 250^\circ/\text{giây}$ $\pm 500^\circ/\text{giây}$ $\pm 1000^\circ/\text{giây}$ $\pm 2000^\circ/\text{giây}$
Tốc độ lấy mẫu (gyroscope)	8000 Hz
Giải đo gia tốc kế	$\pm 2g$, $\pm 4g$, $\pm 8g$ and $\pm 16g$
Tốc độ lấy mẫu (gia tốc kế)	1000 Hz
Hệ số độ nhạy	16 384 LSB/g
Thay đổi độ nhạy với nhiệt độ (trong khoảng -40°C tới 85°C)	$\pm 0,02\ \%/^{\circ}\text{C}$
Ảnh hưởng giữa các trục	$\pm 2\ \%$
Sai lệch hiệu chỉnh ban đầu	Trục X và Y: $\pm 50\ \text{mg}$ Trục Z : $\pm 80\ \text{mg}$
Mật độ nhiễu	$400\ \mu\text{g}/\sqrt{\text{Hz}}$

Bảng 2.2 Thông số kỹ thuật của Cảm biến IMU MPU-6050

- **ESC (Electronic speed controller)**

ESC không chổi than là một trong những phụ kiện của mô hình máy bay, cần có mạch điều khiển để hoạt động. Chịu trách nhiệm nhận tín hiệu điều khiển và cung cấp năng lượng cho motor.



Hình 2.4 Esc cho động cơ không chổi than

- **Pin Li-Po WildScorpion**

Trong số các loại pin được sử dụng cho mô hình máy bay, pin Li-Po (Lithium-ion Polymer) cho dung lượng lớn với dòng ra ổn định và có kích thước nhẹ, phù hợp với ứng dụng trên các máy bay mô hình cỡ nhỏ.

Kích thước	23mm x 33mm x 107mm
Khối lượng	184g
Dung lượng	2200mAh
Điện áp	11.1V
Tốc độ xả liên tục	35C
Tốc độ sạc Max	5C
Số cell	3

Bảng 2.3 Thông số kỹ thuật của Pin Li-Po WildScorpion



Hình 2.5 Pin Li-Po

- **Động cơ 3 pha DC không chổi than**

Thuộc nhóm động cơ đồng bộ nam châm vĩnh cửu. Là loại động cơ song hình thang, những động cơ còn lại là động cơ sóng hình Sin. So với động cơ DC có chổi than, động cơ 3 pha DC không chổi than có kết cấu cơ khí tin cậy hơn, đồng thời cho hiệu suất lực đẩy so với khối lượng cao hơn.

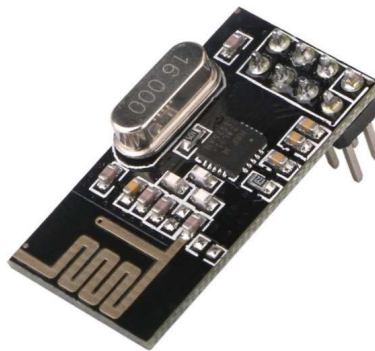


Hình 2.6 Động cơ 3 pha DC không chổi than

- **Module NRF24L01**

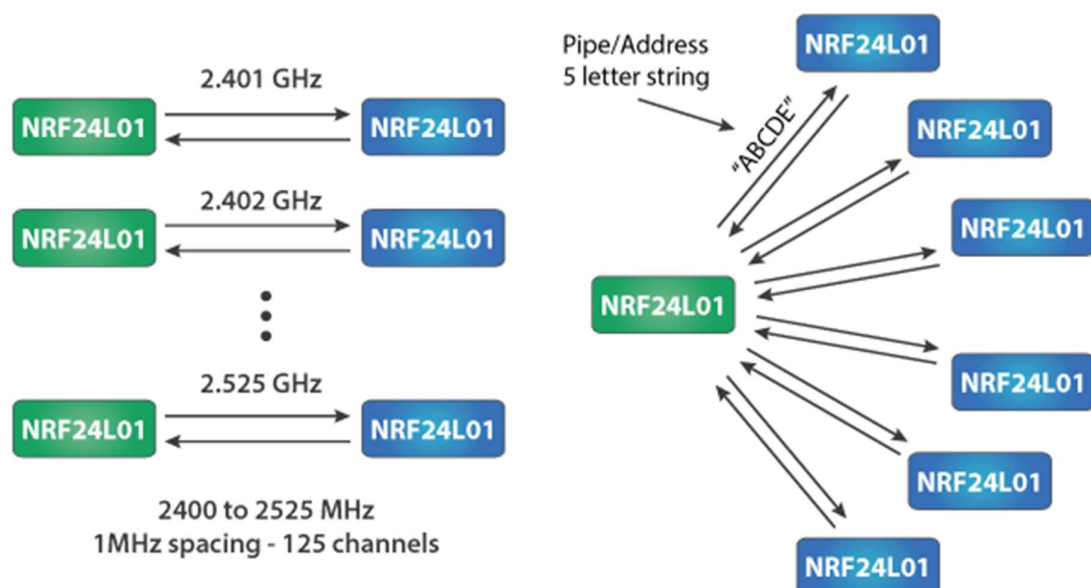
Là module truyền tải tín hiệu không dây, khoảng cách thu phát khoảng 100m.

Thích hợp cho bộ điều khiển cầm tay. Có khả năng truyền tín hiệu 2 chiều.



Hình 2.7 Module NRF24L01

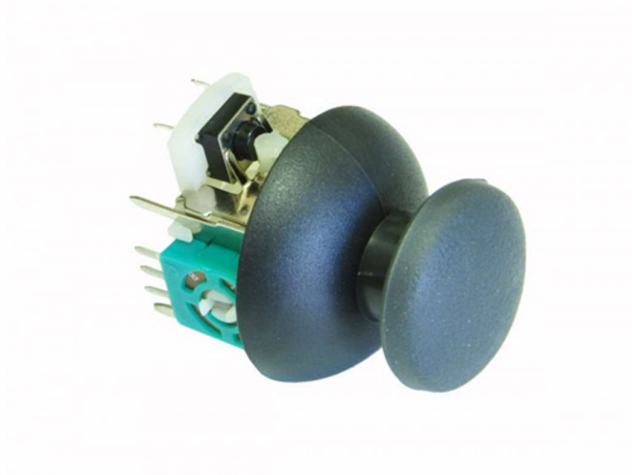
Module sử dụng 125 kênh khác nhau mang lại khả năng có mạng lưới 125 modem hoạt động độc lập ở một nơi. Mỗi kênh có tối đa 6 địa chỉ hoặc mỗi đơn vị có thể liên lạc với tối đa 6 đơn vị khác cùng một lúc.



Hình 2.8 Mô hình truyền nhận của Module NRF24L01

- **Joystick**

Là một thiết bị đầu vào bao gồm một gậy quay và bộ lọc hoặc hướng điều khiển truyền tới thiết bị mà nó đang kiểm soát. Cần điều khiển thường được sử dụng để điều khiển trò chơi điện tử hoặc robot.

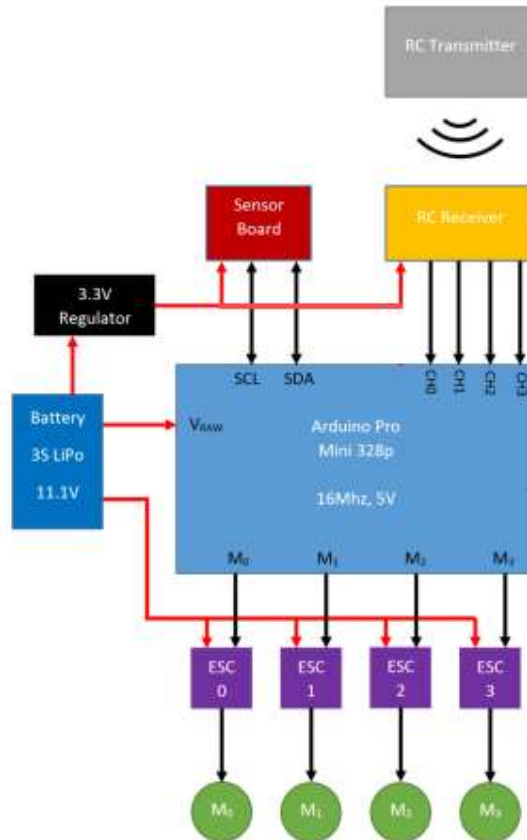


Hình 2.9 Joystick

2.2. Sơ đồ các module và mạch nguyên lý

2.2.1. Các khối module

Hardware Diagram



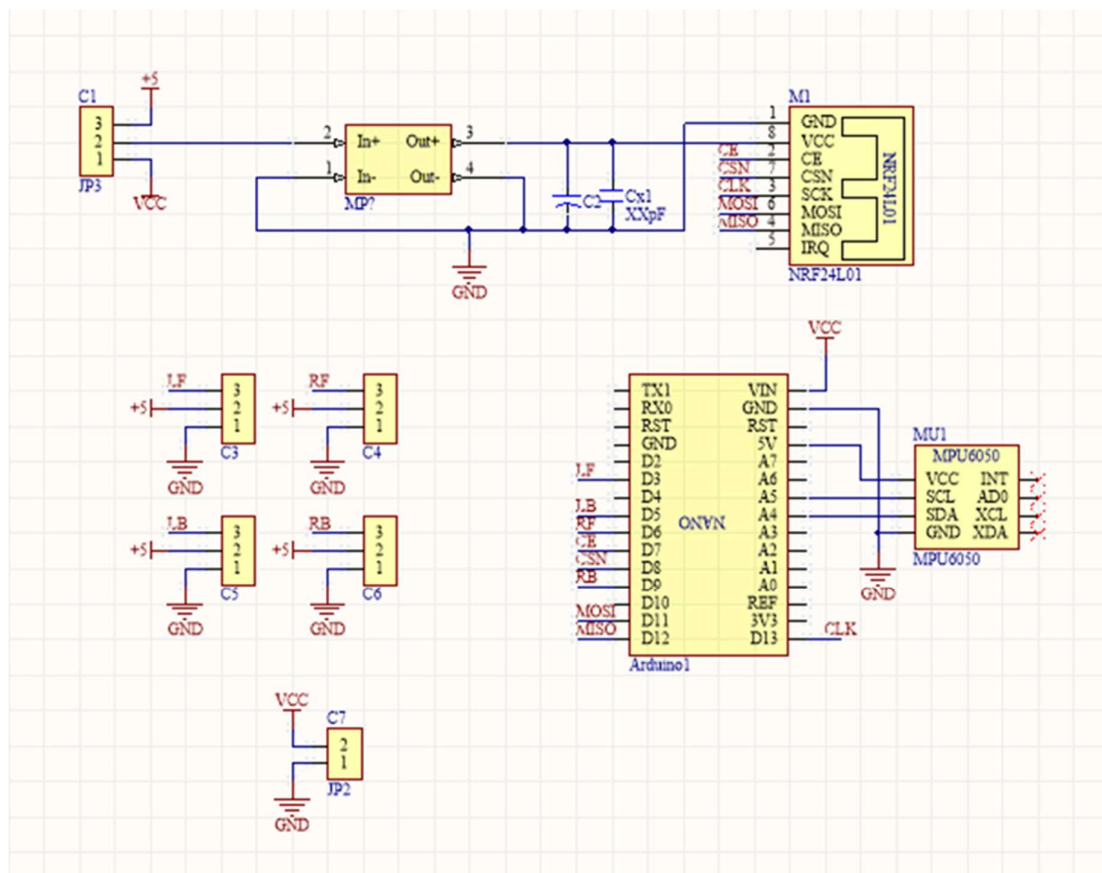
Hình 2.10 Hardware Diagram

Các module chính gồm có:

- Pin: Cung cấp năng lượng cho toàn bộ hệ thống
- RC Transmitter: Gửi tín hiệu điều khiển từ người dùng đến vi điều khiển
- RC Receiver: Nhận tín hiệu không dây từ RC Transmitter và gửi về vi điều khiển

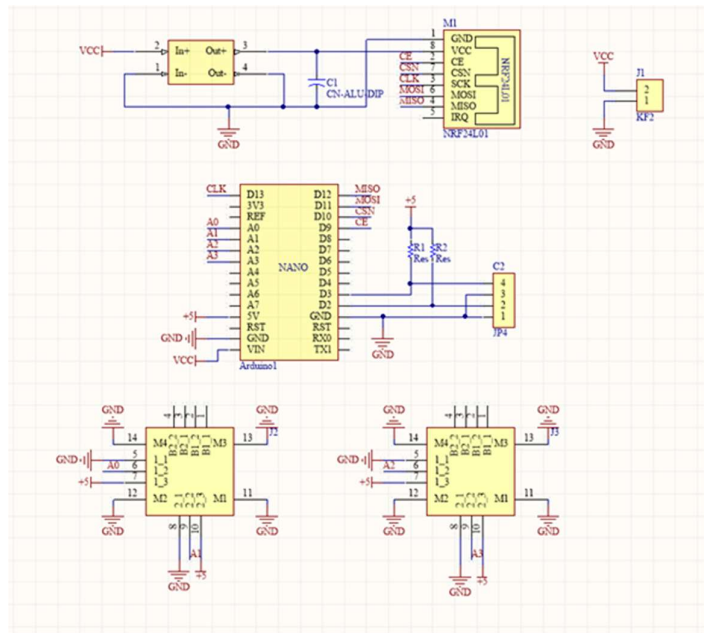
- Arduino Nano: Nhận giữ liệu từ cảm biến và xử lý qua thuật toán PID để đưa ra tín hiệu cho các ESC
- Cảm biến MPU: Gửi dữ liệu giúp vi điều khiển nhận biết được trạng thái máy bay
- ESC: Nhận tín hiệu xung từ vi điều khiển và cung cấp năng lượng cho motor

2.2.2. Mạch nguyên lý khối nhận và xử lý tín hiệu



Hình 2.11 Mạch nguyên lý khối nhận và xử lý tín hiệu

2.2.3. Mạch nguyên lý khối truyền tín hiệu



Hình 2.12 Mạch nguyên lý khối truyền tín hiệu

2.3. Thiết kế bộ điều khiển

2.3.1. Tìm hiểu về và các phương pháp thiết kế bộ điều khiển PID

Trong phần này, ta tiến hành xây dựng các bộ điều khiển cho UAV. Các công trình được nghiên cứu trên thế giới trước đây đã áp dụng thành công nhiều bộ điều khiển khác nhau cho việc ổn định quadcopter

Trong số đó, thực nghiệm đã chứng tỏ bộ điều khiển PID (Proportional – Integral – Derivative) cho chất lượng điều khiển tương đối tốt trong môi trường nhiễu khí động không quá lớn.

- P: Là phương pháp điều chỉnh tỷ lệ, giúp tạo ra tín hiệu điều chỉnh tỉ lệ với sai lệch đầu vào theo thời gian lấy mẫu.
- I: Là tích phân của sai lệch theo thời gian lấy mẫu. Điều khiển tích phân là phương pháp điều chỉnh để tạo ra các tín hiệu điều chỉnh sao cho độ sai lệch giảm về 0. Từ đó cho ta biết tổng sai số tức thời theo thời gian hay sai số tích lũy trong quá khứ. Khi thời gian càng nhỏ thể

hiệu tác động điều chỉnh tích phân càng mạnh, tương ứng với độ lệch càng nhỏ.

- D: Là vi phân của sai lệch. Điều khiển vi phân tạo ra tín hiệu điều chỉnh sao cho tỉ lệ với tốc độ thay đổi sai lệch đầu vào. Thời gian càng lớn thì phạm vi điều chỉnh vi phân càng mạnh, tương ứng với bộ điều chỉnh đáp ứng với thay đổi đầu vào càng nhanh.

Một trong số các ưu điểm của bộ điều khiển PID là khả năng tích hợp dễ dàng với chi phí thấp lên các phần cứng thương mại được bán phổ biến trên thị trường. Giảm sai số xác lập đến mức tối thiểu nhất, hạn chế dao động, giảm thời gian xác lập và độ vọt lố. Tuy nhiên, việc tính toán và dò thực nghiệm các thông số PID trong thực tế cần nhiều kinh nghiệm và có thể mất khá nhiều thời gian. Việc đạt được một yêu cầu này thường kéo theo việc phải hạ thấp các tiêu chuẩn khác. Ví dụ: để tăng tốc độ phản hồi thường kéo theo xuất hiện vọt lố ở đầu ra so với tín hiệu đáp ứng mong muốn. Người thiết kế phải căn cứ vào các yêu cầu của hệ thống (về tốc độ phản hồi, độ bền vững của hệ thống...) để cân đối các đặc tính của hệ thống.

2.3.2. Phương pháp tuyến tính kết hợp mô phỏng

Để thiết kế bộ điều khiển PID, ta có thể sử dụng mô hình quadcopter đơn giản, chỉ xét đến những yếu tố tác động chính lên ứng xử động học của máy bay. Ở đây, ta bỏ qua hiện tượng gyro của các cánh quạt do các mô-men này không đáng kể so với mô-men gây ra bởi lực đẩy từ các động cơ. Bằng cách đó, để đơn giản bài toán, ta có thể xét riêng các bài toán thiết kế bộ điều khiển cho từng chuyển động và coi chúng độc lập với nhau. Các bài toán xét đến bao gồm:

- Ổn định góc roll
- Ổn định góc pitch
- Ổn định góc yaw

Sau khi áp dụng giả thiết để đơn giản mô hình toán học, ta tiến hành mô phỏng và thiết kế bộ điều khiển bằng phần mềm MATLAB-Simulink. Để đánh giá chất lượng của các bộ điều khiển, ta căn cứ vào các tiêu chuẩn: Vọt lố xuất hiện trên đáp ứng trong miền thời gian cần không vượt quá 10% tín hiệu mong muốn.

Độ bền vững đánh giá thông qua độ dư pha và độ dư biên độ, thông qua đánh giá biểu đồ Bode của hệ hở. Hệ thống cần đạt độ dư trữ biên độ tối thiểu là 3dB và độ dư trữ pha tối thiểu là 30° là yêu cầu chung đối với phần lớn các hệ thống điều khiển trong công nghiệp công nghiệp.

2.3.3. Phương pháp dò thực nghiệm kết hợp Ziegler-Nichols

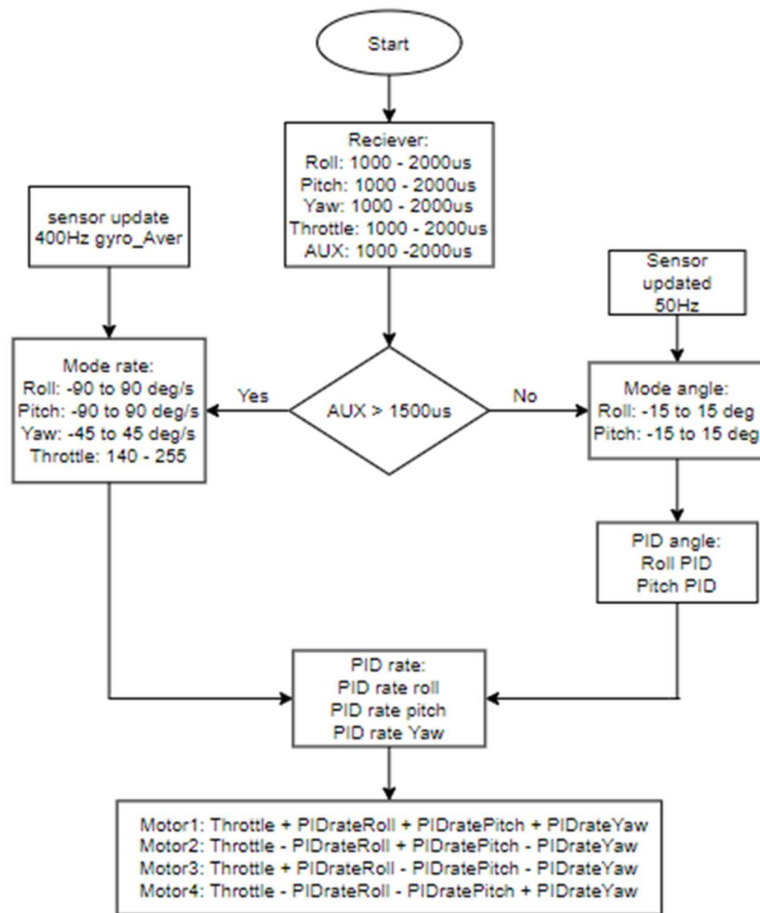
Việc thiết kế bộ điều khiển PID theo phương pháp tuyến tính có nhiều hạn chế như các tác động nhiễu, hiệu ứng vật lý tác động máy bay sẽ không sát với quá trình mô phỏng nên dẫn đến bộ PID không sát với yêu cầu thực tế. Vì vậy phương pháp dò thực nghiệm tuy khó khăn hơn nhưng cho kết quả sát với thực tế, tính ổn định tốt hơn.

Trong đề tài này, chúng em tìm các thông số của PID bằng cách kết hợp phương pháp Ziegler-Nichols và thực nghiệm.

2.3.4. Lưu đồ dải thuật cân bằng

Đầu tiên, bộ xử lý nhận giá trị điều khiển từ bộ điều khiển từ xa với giá trị ban đầu từ 1000us – 2000us, sau đó so sánh các giá trị nhận được với giá trị mẫu là 1500us.

Nếu giá trị nhận được nhỏ hơn giá trị mẫu thì bộ xử lý thực hiện chế độ cân bằng góc với tần số lấy mẫu là 50Hz và giới hạn góc nghiêng trong khoảng -15 đến 15 độ. Sau đó tiếp tục thực hiện xử lý giá trị cảm biến thông qua PID. Giá trị sau khi xử lý trong PID góc sẽ được lấy để làm giá trị đầu vào của PID cân bằng di chuyển.



Hình 2.13 Lưu đồ dải thuật cân bằng

2.4. Quá trình thực nghiệm

2.4.1. Quá trình dò các thông số PID

Từ những phần trên ta tiến hành hành dò PID sử dụng phần mềm MultiwiiConf hỗ trợ cho việc dò tìm thông số và quan sát trạng thái máy bay.

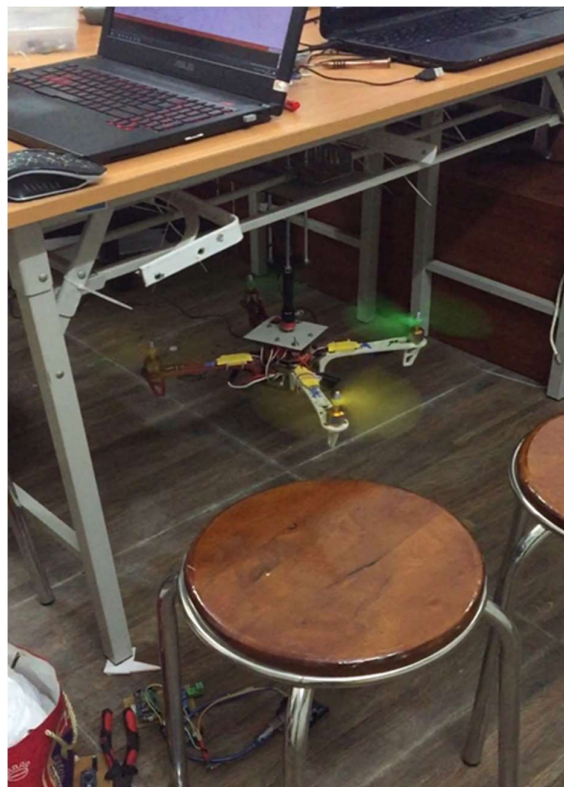
Thông số PID dò được:

	K_p	K_i	K_d
Roll	0.3	0.005	2
Pitch	0.3	0.005	2
Yaw	2.8	0.045	5

Bảng 2.4 Thông số kỹ thuật PID dò được

2.4.2. Bay thử nghiệm sản phẩm

Việc dò các thông số PID rất phức tạp và cần nhiều kinh nghiệm và thời gian, nhất là đối với máy bay không người lái, cần cân bằng 3 trục cùng lúc. Do vậy mô hình chưa thể cân bằng tốt trước những tác động bên ngoài.



Hình 2.15 Bay thử nghiệm sản phẩm

Chương 3. Module gửi thông tin.

3.1. Đặt vấn đề

Trong quá trình bay có rất nhiều thông tin ảnh hưởng đến máy bay như hướng gió, tốc độ gió, vật cản... Ngoài ra còn có những thông tin khác tùy theo mục đích sử dụng của máy bay. Những thông tin đó sẽ cần hiển thị theo thời gian thực. Việc gửi thông tin có thể tích hợp vào thiết bị điều khiển bay nhưng nó sẽ ảnh hưởng tới quá trình xử lý. Ngoài ra việc hiển thị thông tin sẽ cần thêm phần cứng thì gây ồng kênh cho thiết bị điều khiển. Vì vậy sử dụng thêm module gửi thông tin là một lựa chọn hợp lý.

Hiện nay với sự phát triển của công nghệ các thiết bị có thể hiển thị thông tin từ web rất phổ biến đặc biệt như smartphone. Vì vậy việc đưa thông tin lên web sẽ dễ dàng cho việc hiển thị đặc biệt là trong thời gian thực.

Module thu phát Wifi NodeMcu Lua là một lựa chọn phù hợp vì nó có thể đưa thông tin lên hiển thị trên web, nó có giá thành rẻ, sử dụng mã nguồn mở, dễ dàng giao tiếp kết nối với các cảm biến.

3.2. Phần cứng sử dụng

Module thu phát Wifi NodeMcu Lua là phiên bản NodeMCU sử dụng IC nạp giá rẻ CH340 từ Lolin với bộ xử lý trung tâm là module Wifi SoC ESP8266, kit có thiết kế dễ sử dụng và đặc biệt là có thể sử dụng trực tiếp trình biên dịch của Arduino để lập trình và nạp code, điều này khiến việc sử dụng và lập trình các ứng dụng trên ESP8266 trở nên rất đơn giản.

Thông số kỹ thuật:

IC chính:	ESP8266 Wifi SoC.
Phiên bản firmware:	NodeMCU Lua
Chip nạp và giao tiếp UART	: CH340
GPIO	tương thích hoàn toàn với firmware Node MCU
Cấp nguồn:	5VDC MicroUSB hoặc Vin.
GPIO giao tiếp mức	3.3VDC
Tích hợp	Led báo trạng thái, nút Reset, Flash.
Kích thước:	59 x 32mm

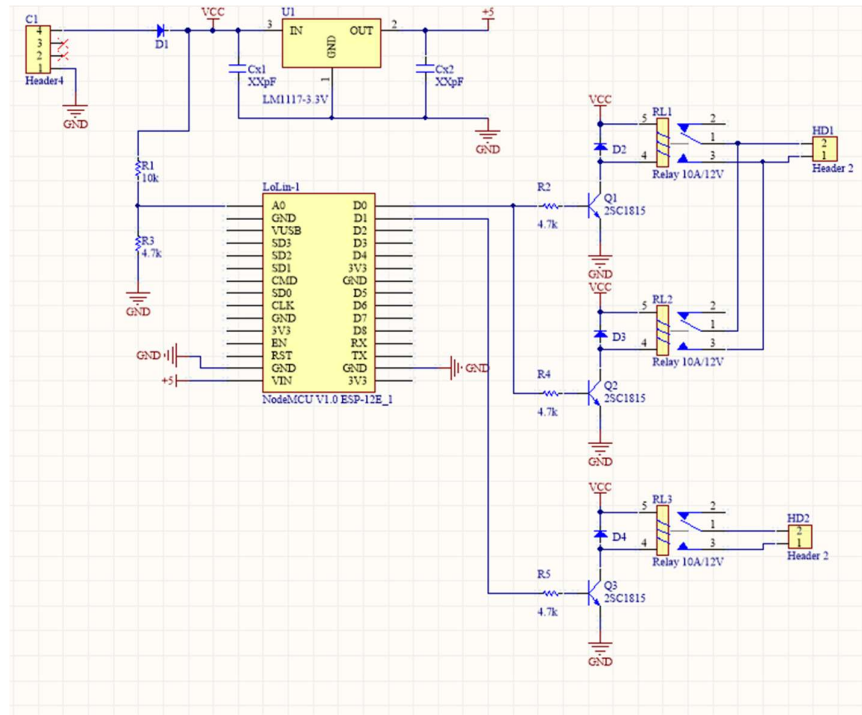
Bảng 3.1 Thông số kỹ thuật Module thu phát Wifi NodeMcu

3.3. Module gửi thông tin.

3.3.1. Chức năng

Module gửi thông tin có nhiệm vụ đo dung lượng pin còn lại của máy bay và điều khiển nguồn điện cho máy bay từ xa. Việc này rất có lợi khi máy bay xảy ra sự cố chúng ta có thể tắt máy bay từ xa để tránh nguy hiểm.

3.3.2. Sơ đồ nguyên lý



Hình 3.1 Sơ đồ nguyên lý Module gửi thông tin

3.3.3. Giao diện điều khiển

ESP8266 Web Server

Dung lượng pin hiện tại

Mạch động lực

ON

OFF

Mạch điều khiển

ON

OFF

Điều khiển tất cả

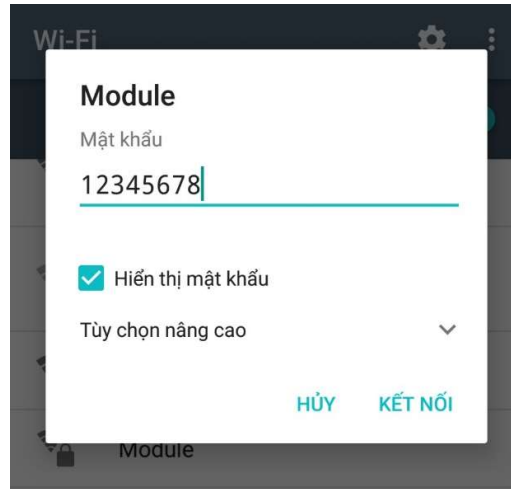
ON

OFF

Hình 3.2 Giao diện điều khiển Module gửi thông tin

3.3.4. Kết nối với Module gửi thông tin

Sau khi cấp điện cho Module gửi thông tin. Modul sẽ tạo một access point (AP) cho phép các thiết bị có kết nối wifi có thể kết nối.



Hình 3.3 Kết nối với Modul gửi thông tin

Sau khi kết nối với modul nhập đường dẫn <http://192.168.4.1/index> vào trình duyệt web của thiết bị để truy cập vào trang giao diện của modul.

Chương 4. Kết luận đề xuất

4.1. Kết quả đạt được

Đề tài đã thực hiện tốt phần thiết kế về điện tử: Gửi, nhận tín hiệu điều khiển không dây và phân tích tín hiệu phản hồi từ cảm biến sử dụng thuật toán PID.



Hình 4.1 Sản phẩm hoàn thiện

Tuy nhiên phần cài đặt thông số PID chưa mang lại hiệu quả cao do còn thiếu kinh nghiệm dẫn đến máy bay chưa có độ ổn định tốt. Việc điều chỉnh thông số PID cần có nhiều kinh nghiệm và thời gian.

4.2. Đề xuất

Cần điều chỉnh các thông số PID để cho máy bay có thể đạt được sự ổn định hơn. Khi có tác động từ bên ngoài dẫn đến mất cân bằng, máy bay có thể đạt trạng thái ổn định

Cần tích hợp thêm module GPS để có thể lập trình cho máy bay, bay tự động từ vị trí A đến vị trí B được thiết đặt sẵn, ứng dụng vào việc giao hàng tự động.

TÀI LIỆU THAM KHẢO

- [1] N. H .T. Son, V. N. Phúc “Thiết kế và thi công mô hình máy bay” Đại học Lạc Hồng, 2015
- [2] V. Thanh and H. Tung, “Optinizing PID Parameters in Control Accelerometers and Gyroscopes in Self-Balancing Quadrotors” Journal of Science and Technology, 2015
- [3] M. Hehn and R.D’Andrea, “A flying inverted pendulum” in IEEE International Conference on Robotics and Automation (ICRA), 2011
- [4] Invensense, “MPU – 6050 Datasheet”
- [5] Arrow Electronic, “NRF24L104 Datasheet”

PHỤ LỤC

- Code mạch điều khiển:

```
#include <Wire.h>

#include <Servo.h>

#include <SPI.h>

#include <nRF24L01.h>

#include <RF24.h>

Servo L_F_prop;

Servo L_B_prop;

Servo R_F_prop;

Servo R_B_prop;

const uint64_t pipeIn = 0xE8E8F0F0E1LL;

RF24 radio(7, 8);

struct MyData {

byte throttle;

byte yaw;

byte pitch;

byte roll;

byte AUX1;

byte AUX2;

};

MyData data;
```

```

void resetData(){
data.throttle = 0;
data.yaw      = 127;
data.pitch    = 127;
data.roll     = 127;
data.AUX1     = 0;
data.AUX2     = 0;
}

unsigned long counter_1, counter_2, counter_3, counter_4, current_count;
byte last_CH1_state, last_CH2_state, last_CH3_state, last_CH4_state;
int input_YAW;
int input_PITCH;
int input_ROLL;
int input_THROTTLE;

//Gyro Variables
float elapsedTime, time, timePrev;
int gyro_error=0;
float Gyr_rawX, Gyr_rawY, Gyr_rawZ;
float Gyro_angle_x, Gyro_angle_y;
float Gyro_raw_error_x, Gyro_raw_error_y;

//Acc Variables
int acc_error=0;
float rad_to_deg = 180/3.141592654;

```

```

float Acc_rawX, Acc_rawY, Acc_rawZ;

float Acc_angle_x, Acc_angle_y;

float Acc_angle_error_x, Acc_angle_error_y;

float Total_angle_x, Total_angle_y;

//More variables for the code

int i;

int mot_activated=0;

long activate_count=0;

long des_activate_count=0;

////////////////////////////////PID FOR ROLL////////////////////////////////

float roll_PID, pwm_L_F, pwm_L_B, pwm_R_F, pwm_R_B, roll_error,
roll_previous_error;

float roll_pid_p=0;

float roll_pid_i=0;

float roll_pid_d=0;

////////////////////////////////ROLL PID CONSTANTS////////////////////////////////

double roll_kp=0.3;//3.55

double roll_ki=0;//0.003

double roll_kd=0;//2.05

float roll_desired_angle = 0;

////////////////////////////////PID FOR PITCH////////////////////////////////

float pitch_PID, pitch_error, pitch_previous_error;

float pitch_pid_p=0;

```

```

float pitch_pid_i=0;

float pitch_pid_d=0;

////////////////////////////////PITCH PID CONSTANTS////////////////////////////////

double pitch_kp=0.3;//3.55

double pitch_ki=0;//0.003

double pitch_kd=0;//2.05

float pitch_desired_angle = 0;

void setup() {

  resetData();

  radio.begin();

  radio.setAutoAck(false);

  radio.setDataRate(RF24_250KBPS);

  radio.openReadingPipe(1,pipeIn);

  radio.startListening();

  L_F_prop.attach(4);

  L_B_prop.attach(5);

  R_F_prop.attach(7);

  R_B_prop.attach(6);

  L_F_prop.writeMicroseconds(1000);

  L_B_prop.writeMicroseconds(1000);

  R_F_prop.writeMicroseconds(1000);

  R_B_prop.writeMicroseconds(1000);

  Wire.begin();

```

```
Wire.beginTransmission(0x68);  
  
Wire.write(0x6B);  
  
Wire.write(0x00);  
  
Wire.endTransmission(true);  
  
Wire.beginTransmission(0x68);  
  
Wire.write(0x1B);  
  
Wire.write(0x10);  
  
Wire.endTransmission(true);  
  
Wire.beginTransmission(0x68);  
  
Wire.write(0x1C);  
  
Wire.write(0x10);  
  
Wire.endTransmission(true);  
  
Serial.begin(9600);  
  
delay(1000);  
  
time = millis();  
  
if(gyro_error==0) {  
    for(int i=0; i<200; i++){  
        Wire.beginTransmission(0x68);  
  
        Wire.write(0x43);  
  
        Wire.endTransmission(false);  
  
        Wire.requestFrom(0x68,4,true);  
  
        Gyr_rawX=Wire.read()<<8|Wire.read();  
  
        Gyr_rawY=Wire.read()<<8|Wire.read()
```

```

/*---X---*/

Gyro_raw_error_x = Gyro_raw_error_x + (Gyr_rawX/32.8);

/*---Y---*/

Gyro_raw_error_y = Gyro_raw_error_y + (Gyr_rawY/32.8);

if(i==199){

    Gyro_raw_error_x = Gyro_raw_error_x/200;

    Gyro_raw_error_y = Gyro_raw_error_y/200;

    gyro_error=1;

}

}

}

if(acc_error==0){

for(int a=0; a<200; a++){

    Wire.beginTransaction(0x68);

    Wire.write(0x3B);

    Wire.endTransmission(false);

    Wire.requestFrom(0x68,6,true);

    Acc_rawX=(Wire.read()<<8|Wire.read())/4096.0 ;

    Acc_rawY=(Wire.read()<<8|Wire.read())/4096.0 ;

    Acc_rawZ=(Wire.read()<<8|Wire.read())/4096.0 ;

    /*---X---*/

    Acc_angle_error_x=Acc_angle_error_x+((atan((Acc_rawY)/sqrt(pow((Acc_r
awX),2) + pow((Acc_rawZ),2)))*rad_to_deg));

```

```
/*---Y---*/
```

```
Acc_angle_error_y=Acc_angle_error_y+((atan(1*(Acc_rawX)/sqrt(pow((Acc_rawY),2) + pow((Acc_rawZ),2)))*rad_to_deg));
```

```
if(a==199)
```

```
{
```

```
Acc_angle_error_x = Acc_angle_error_x/200;
```

```
Acc_angle_error_y = Acc_angle_error_y/200;
```

```
acc_error=1;
```

```
}
```

```
}
```

```
}
```

```
}
```

```
void recvData(){
```

```
while ( radio.available() ) {
```

```
radio.read(&data, sizeof(MyData));
```

```
lastRecvTime = millis(); //here we receive the data
```

```
}
```

```
}
```

```
void loop() {
```

```
recvData();
```

```
unsigned long now = millis();
```

```
if ( now - lastRecvTime > 1000 ) {
```



```

resetData();

//////////////////////////////////I M U//////////////////////////////////

timePrev = time;

time = millis();

elapsedTime = (time - timePrev) / 1000;

//////////////////////////////////Gyro read//////////////////////////////////

Wire.beginTransmission(0x68);

Wire.write(0x43);

Wire.endTransmission(false);

Wire.requestFrom(0x68,4,true);

Gyr_rawX=Wire.read()<<8|Wire.read();

Gyr_rawY=Wire.read()<<8|Wire.read();

/*---X---*/

Gyr_rawX = (Gyr_rawX/32.8) - Gyro_raw_error_x;

/*---Y---*/

Gyr_rawY = (Gyr_rawY/32.8) - Gyro_raw_error_y;

/*---X---*/

Gyro_angle_x = Gyr_rawX*elapsedTime;

/*---X---*/

Gyro_angle_y = Gyr_rawY*elapsedTime;

//////////////////////////////////Acc read//////////////////////////////////

Wire.beginTransmission(0x68);

Wire.write(0x3B);

```

```

Wire.endTransmission(false);

Wire.requestFrom(0x68,6,true);

Acc_rawX=(Wire.read()<<8|Wire.read())/4096.0 ; //each value needs two
registres

Acc_rawY=(Wire.read()<<8|Wire.read())/4096.0 ;

Acc_rawZ=(Wire.read()<<8|Wire.read())/4096.0 ;

/*---X---*/

Acc_angle_x=(atan((Acc_rawY)/sqrt(pow((Acc_rawX),2)+pow((Acc_rawZ),
2))))*rad_to_deg) - Acc_angle_error_x;

/*---Y---*/

Acc_angle_y=(atan(-
1*(Acc_rawX)/sqrt(pow((Acc_rawY),2)+pow((Acc_rawZ),2))))*rad_to_deg) -
Acc_angle_error_y;

////////////////////Total angle and filter////////////////////

/*---X axis angle---*/

Total_angle_x = 0.98 *(Total_angle_x + Gyro_angle_x) + 0.02*Acc_angle_x;

/*---Y axis angle---*/

Total_angle_y = 0.98 *(Total_angle_y + Gyro_angle_y) + 0.02*Acc_angle_y;

/*////////////////////P I D////////////////////*/

roll_desired_angle = map(input_ROLL,1000,2000,-10,10);

pitch_desired_angle = map(input_PITCH,1000,2000,-10,10);

roll_error = Total_angle_y - roll_desired_angle;

pitch_error = Total_angle_x - pitch_desired_angle;

```

```

roll_pid_p = roll_kp*roll_error;
pitch_pid_p = pitch_kp*pitch_error
if(-3 < roll_error <3)
{
    roll_pid_i = roll_pid_i+(roll_ki*roll_error);
}
if(-3 < pitch_error <3)
{
    pitch_pid_i = pitch_pid_i+(pitch_ki*pitch_error);
}
roll_pid_d = roll_kd*((roll_error - roll_previous_error)/elapsedTime);
pitch_pid_d = pitch_kd*((pitch_error - pitch_previous_error)/elapsedTime);
roll_PID = roll_pid_p + roll_pid_i + roll_pid_d;
pitch_PID = pitch_pid_p + pitch_pid_i + pitch_pid_d;
if(roll_PID < -400){roll_PID=-400;}
if(roll_PID > 400) {roll_PID=400; }
if(pitch_PID < -400){pitch_PID=-400;}
if(pitch_PID > 400) {pitch_PID=400;}
pwm_R_F = 115 + input_THROTTLE - roll_PID - pitch_PID;
pwm_R_B = 115 + input_THROTTLE - roll_PID + pitch_PID;
pwm_L_B = 115 + input_THROTTLE + roll_PID + pitch_PID;
pwm_L_F = 115 + input_THROTTLE + roll_PID - pitch_PID;
//Right front

```

```
if(pwm_R_F < 1100)
```

```
{
```

```
    pwm_R_F= 1100;
```

```
}
```

```
if(pwm_R_F > 2000)
```

```
{
```

```
    pwm_R_F=2000;
```

```
}
```

```
//Left front
```

```
if(pwm_L_F < 1100)
```

```
{
```

```
    pwm_L_F= 1100;
```

```
}
```

```
if(pwm_L_F > 2000)
```

```
{
```

```
    pwm_L_F=2000;
```

```
}
```

```
if(pwm_R_B < 1100)
```

```
{
```

```
    pwm_R_B= 1100;
```

```
}
```

```
if(pwm_R_B > 2000)
```

```

{
    pwm_R_B=2000;
}

if(pwm_L_B < 1100)
{
    pwm_L_B= 1100;
}

if(pwm_L_B > 2000)
{
    pwm_L_B=2000;
}

roll_previous_error = roll_error;
pitch_previous_error = pitch_error;

if(mot_activated)
{
    L_F_prop.writeMicroseconds(pwm_L_F);
    L_B_prop.writeMicroseconds(pwm_L_B);
    R_F_prop.writeMicroseconds(pwm_R_F);
    R_B_prop.writeMicroseconds(pwm_R_B);
}

if(!mot_activated)
{
    L_F_prop.writeMicroseconds(1000);

```

```

L_B_prop.writeMicroseconds(1000);
R_F_prop.writeMicroseconds(1000);
R_B_prop.writeMicroseconds(1000);
}

if(input_THROTTLE < 1100 && input_YAW > 1800 && !mot_activated)
{
    if(activate_count==200)
    {
        mot_activated=1;
        PORTB |= B00100000;
    }
    activate_count=activate_count+1;
}

if(!(input_THROTTLE < 1100 && input_YAW > 1800) && !mot_activated)
{
    activate_count=0;
}

if(input_THROTTLE < 1100 && input_YAW < 1100 && mot_activated){
    if(des_activate_count==300) {
        mot_activated=0;
        PORTB &= B11011111;
    }
}

```

```

    des_activate_count=des_activate_count+1;
}

if(!(input_THROTTLE < 1100 && input_YAW < 1100) && mot_activated){
    des_activate_count=0;
}

input_THROTTLE = data.throttl);
input_YAW = data.yaw;
input_PITCH = data.pitch;
input_ROLL = data.roll;
}

```

- Code mạch tay điều khiển:

```

#include <SPI.h>

#include <nRF24L01.h>

#include <RF24.h>

const uint64_t pipeOut = 0xE8E8F0F0E1LL;

RF24 radio(9, 10);

struct MyData {
    byte throttle;
    byte yaw;
    byte pitch;
    byte roll;
    byte AUX1;
    byte AUX2;
}

```

```

};

MyData data;

void resetData() {
    data.throttle = 0;
    data.yaw = 127;
    data.pitch = 127;
    data.roll = 127;
    data.AUX1 = 0;
    data.AUX2 = 0;
}

void setup(){
    Serial.begin(9600);
    radio.begin();
    radio.setAutoAck(false);
    radio.setDataRate(RF24_250KBPS);
    radio.openWritingPipe(pipeOut);
    resetData();
}

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 ? val : 255 );

}

void loop(){

    data.throttle = mapJoystickValues( analogRead(A1), 0, 511, 1023, true );

    data.yaw      = mapJoystickValues( analogRead(A2), 0, 510, 1023, true );

    data.pitch    = mapJoystickValues( analogRead(A3), 0, 521, 1023, true );

    data.roll     = mapJoystickValues( analogRead(A0), 0, 510, 1023, true );

    data.AUX1     = digitalRead(4); //The 2 toggle switches

    data.AUX2     = digitalRead(5);

    radio.write(&data, sizeof(MyData));

    Serial.print("Throttle: "); Serial.print(analogRead(A1)); Serial.print(" ");

    Serial.print("Yaw: ");    Serial.print(analogRead(A2)); Serial.print(" ");

    Serial.print("Pitch: ");  Serial.print(analogRead(A3)); Serial.print(" ");

    Serial.print("Roll: ");   Serial.print(analogRead(A0)); Serial.print("\n");

}

```