**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KỸ THUẬT MÁY TÍNH**



**ĐỀ TÀI NGHIÊN CỨU KHOA HỌC SINH VIÊN**

**THIẾT KẾ VÀ CHẾ TẠO**

**HỆ THỐNG CÂN BẰNG MÁY ẢNH**

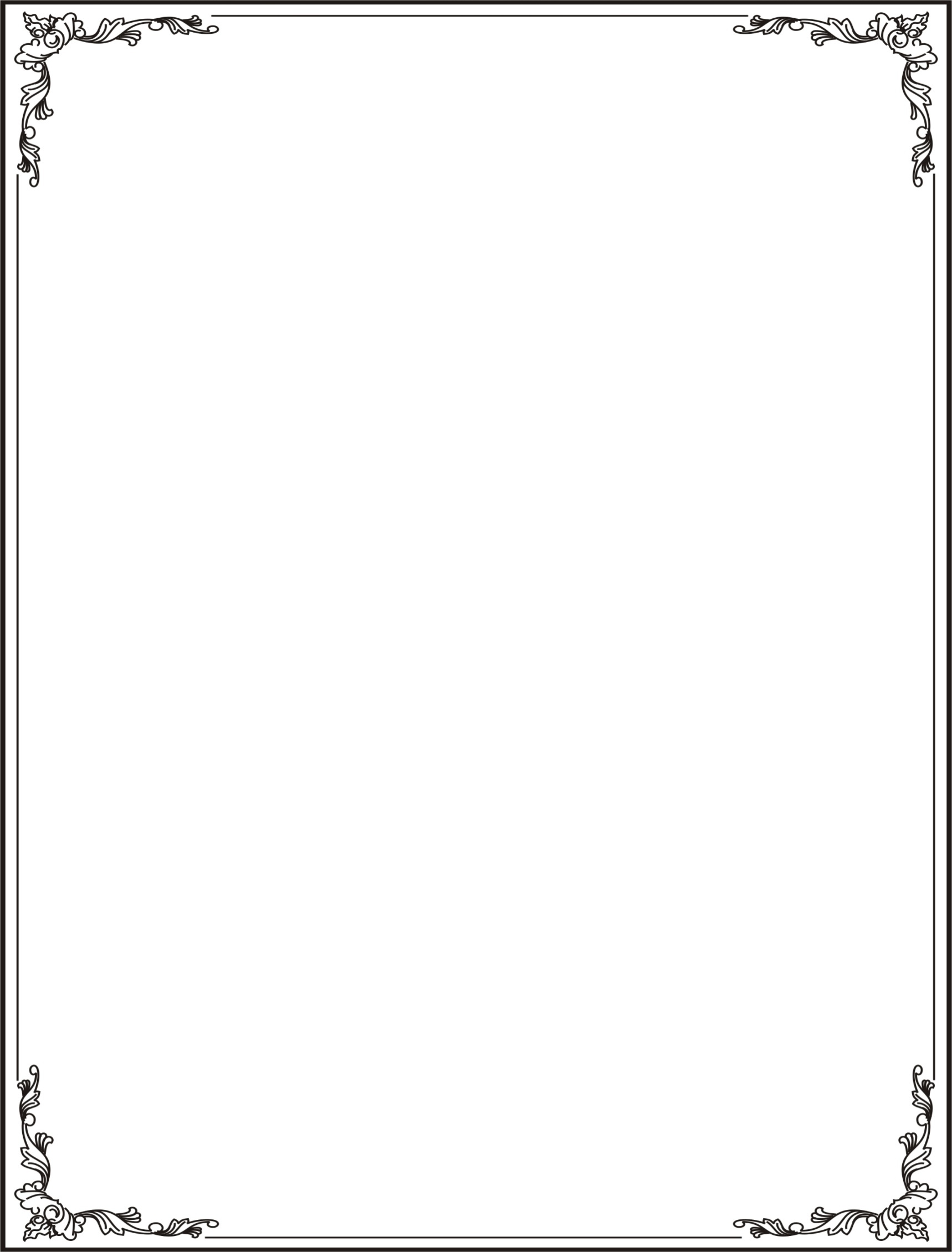
Giảng viên hướng dẫn: **ThS. Phan Đình Duy**

Sinh viên thực hiện: **Trần Quốc Tuấn**

**Võ Hữu Tài**

Lớp: **KTMT2011**

**TP. Hồ Chí Minh, tháng 11 năm 2015**



**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KỸ THUẬT MÁY TÍNH**



**ĐỀ TÀI NGHIÊN CỨU KHOA HỌC SINH VIÊN**

**THIẾT KẾ VÀ CHẾ TẠO**

**HỆ THỐNG CÂN BẰNG MÁY ẢNH**

Giảng viên hướng dẫn: **ThS. Phan Đình Duy**

Sinh viên thực hiện: **Trần Quốc Tuấn**

**Võ Hữu Tài**

Lớp: **KTMT2011**

**TP. Hồ Chí Minh, tháng 11 năm 2015**

# LỜI CẢM ƠN

Để hoàn thành được đề tài nghiên cứu khoa học này, chúng tôi xin chân thành cảm ơn quý thầy cô Trường Đại học Công nghệ thông tin nói chung và quý thầy cô khoa Kỹ thuật máy tính nói riêng đã truyền đạt cho chúng tôi những kiến thức và những kinh nghiệm quý báu.

Xin gửi lời cảm ơn chân thành đến thầy Ths. Phan Đình Duy, người đã dành những thời gian quý báu của mình, trực tiếp hướng dẫn tận tình cho chúng tôi hoàn thành đề tài nghiên cứu.

Xin chân thành cảm ơn sự giúp đỡ của các bạn bè, anh chị, những người đã giúp đỡ chúng tôi tìm kiếm thông tin và động viên tinh thần cho chúng tôi trong suốt quá trình thực hiện nghiên cứu.

Đặc biệt, chúng con xin cảm ơn Cha Mẹ, anh chị em trong gia đình, những người đã sinh thành và nuôi dưỡng chúng con đến ngày hôm nay. Cảm ơn những lời động viên của Cha Mẹ dành cho chúng con trong suốt quá trình học tập và làm việc.

Một lần nữa xin chân thành cảm ơn đến tất cả những người đã quan tâm đến đề tài nghiên cứu của chúng tôi. Tuy nhiên, trong quá trình làm việc không thể tránh khỏi những sai sót, rất mong sự đóng góp ý kiến nhiệt tình của Thầy Cô và các bạn.

Sinh viên thực hiện

**Trần Quốc Tuấn**

**Võ Hữu Tài**

Khoa Kỹ thuật máy tính, lớp KTMT2011

# NHẬN XÉT

# MỤC LỤC

Trang

[LỜI CẢM ƠN i](#_Toc435894211)

[NHẬN XÉT ii](#_Toc435894212)

[MỤC LỤC iii](#_Toc435894213)

[DANH MỤC HÌNH ẢNH vi](#_Toc435894214)

[TÓM TẮT 1](#_Toc435894215)

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 2](#_Toc435894216)

[1.1 GIỚI THIỆU ĐỀ TÀI 2](#_Toc435894217)

[1.2 TÌNH HÌNH NGHIÊN CỨU TRONG VÀ NGOÀI NƯỚC 3](#_Toc435894218)

[1.2.1 Trong nước 3](#_Toc435894219)

[1.2.2 Ngoài nước 3](#_Toc435894220)

[1.3 MỤC TIÊU NGHIÊN CỨU VÀ GIỚI HẠN CỦA ĐỀ TÀI 3](#_Toc435894221)

[1.3.1 Mục tiêu tổng quát 3](#_Toc435894222)

[1.3.2 Mục tiêu cụ thể 3](#_Toc435894223)

[1.4 KHÁCH THỂ VÀ ĐỐI TƯỢNG NGHIÊN CỨU 4](#_Toc435894224)

[1.4.1 Khách thể nghiên cứu 4](#_Toc435894225)

[1.4.2 Đối tượng nghiên cứu: 4](#_Toc435894226)

[1.5 Ý NGHĨA LÝ LUẬN VÀ THỰC TIỄN CỦA ĐỀ TÀI 4](#_Toc435894227)

[1.6 THUẬN LỢI VÀ KHÓ KHĂN 4](#_Toc435894228)

[1.6.1 Thuận lợi 5](#_Toc435894229)

[1.6.2 Khó khăn 5](#_Toc435894230)

[CHƯƠNG 2: CƠ SỞ LÝ LUẬN VÀ PHƯƠNG PHÁP NGHIÊN CỨU 6](#_Toc435894231)

[2.1 CƠ SỞ LÝ LUẬN 6](#_Toc435894232)

[2.1.1 Sử dụng cảm biến 6](#_Toc435894233)

[2.1.2 Con quay hồi chuyển và gia tốc kế 7](#_Toc435894234)

[2.1.3 Board mạch điều khiển hệ thống 14](#_Toc435894235)

[2.1.4 Động cơ Servo KST DS521MG 15](#_Toc435894236)

[2.1.5 Bộ lọc Kalman 15](#_Toc435894237)

[2.1.6 Bộ điều khiển PID 19](#_Toc435894238)

[2.1.7 Các chuẩn giao tiếp được sử dụng 20](#_Toc435894239)

[2.1.8 Tối ưu hiệu năng hệ thống nhúng 27](#_Toc435894240)

[2.2 PHƯƠNG PHÁP NGHIÊN CỨU 29](#_Toc435894241)

[2.2.1 Phương pháp nghiên cứu trực tiếp 29](#_Toc435894242)

[2.2.2 Phương pháp nghiên cứu gián tiếp 29](#_Toc435894243)

[2.3 CÁC KHÁI NIỆM LIÊN QUAN 30](#_Toc435894244)

[2.3.1 Khái niệm điều khiển 30](#_Toc435894245)

[2.3.2 Các thành phần cơ bản của hệ thống điều khiển 30](#_Toc435894246)

[2.3.3 Các bài toán cơ bản trong lĩnh vực điều khiển tự động 31](#_Toc435894247)

[2.4 GIẢ THUYẾT NGHIÊN CỨU 31](#_Toc435894248)

[CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 32](#_Toc435894249)

[3.1 MÔ HÌNH TỔNG QUÁT CỦA HỆ THỐNG 32](#_Toc435894250)

[3.2 MÔ HÌNH TOÁN HỌC CỦA HỆ THỐNG 33](#_Toc435894251)

[3.2.1 Ma trận xoay 33](#_Toc435894252)

[3.2.2 Động cơ và sức kéo 35](#_Toc435894253)

[3.3 NGUYÊN LÝ ĐIỀU KHIỂN GIMBAL 35](#_Toc435894254)

[3.3.1 Nguyên lý điều khiển 35](#_Toc435894255)

[3.3.2 Các trạng thái cơ bản 36](#_Toc435894256)

[3.4 HIỆN THỰC HÓA SẢN PHẨM 36](#_Toc435894257)

[3.4.1 Tài nguyên sử dụng 36](#_Toc435894258)

[3.4.2 Mô hình phần cứng 37](#_Toc435894259)

[3.4.3 Mô hình phần mềm 38](#_Toc435894260)

[3.4.4 Điều khiển 3 trục 40](#_Toc435894261)

[CHƯƠNG 4: KẾT QUẢ THỬ NGHIỆM VÀ ĐÁNH GIÁ 41](#_Toc435894262)

[4.1 QUY TRÌNH THỬ NGHIỆM 41](#_Toc435894263)

[4.1.1 Các thành phần của hệ thống thử nghiệm 41](#_Toc435894264)

[4.1.2 Tiến hành thử nghiệm 41](#_Toc435894265)

[4.2 CÁC KẾT QUẢ THỬ NGHIỆM 41](#_Toc435894266)

[4.2.1 Kết quả sử dụng bộ lọc 41](#_Toc435894267)

[4.2.2 Tính ổn định của hệ thống 41](#_Toc435894268)

[CHƯƠNG 5: KẾT LUẬN VÀ KIẾN NGHỊ 43](#_Toc435894269)

[5.1 KẾT LUẬN 43](#_Toc435894270)

[5.1.1 Các kết quả đạt được 43](#_Toc435894271)

[5.1.2 Chưa đạt được 43](#_Toc435894272)

[5.2 KIẾN NGHỊ 44](#_Toc435894273)

[DANH MỤC TÀI LIỆU THAM KHẢO 45](#_Toc435894274)

# DANH MỤC HÌNH ẢNH

[Hình 1‑1. Các gimbal thực tế trên thị trường 1](#_Toc435892368)

[Hình 2‑1. Gia tốc kế trạng thái tĩnh 6](#_Toc435892369)

[Hình 2‑2. Gia tốc kế di chuyển sang trái 7](#_Toc435892370)

[Hình 2‑3. Ảnh hưởng của lực hấp dẫn 7](#_Toc435892371)

[Hình 2‑4. Tác dụng vào 2 hướng 8](#_Toc435892372)

[Hình 2‑5. Hệ trục tọa độ gia tốc kế trong không gian 8](#_Toc435892373)

[Hình 2‑6. Hệ trục tọa độ và góc nghiêng 11](#_Toc435892374)

[Hình 2‑7. Hệ trục biểu diễn con quay hồi chuyển 12](#_Toc435892375)

[Hình 2‑8. Board điều khiển 13](#_Toc435892376)

[Hình 2‑9. Servo KST DS521MG 14](#_Toc435892377)

[Hình 2‑10. Bộ điều khiển PID 18](#_Toc435892378)

[Hình 2‑11. Sơ đồ kết nối với máy tính 20](file:///D:\Hoc_Ky_9\NCKH%20FINAL.docx#_Toc435892379)

[Hình 2‑12. Mạch chuyển đổi USB sang COM 20](#_Toc435892380)

[Hình 2‑13. Kết nối cho các thiết bị sử dụng I2C 22](#_Toc435892381)

[Hình 2‑14. Hoạt động truyền nhận dữ liệu I2C 23](#_Toc435892382)

[Hình 2‑15. Kết quả tối ưu việc biên dịch 27](#_Toc435892383)

[Hình 2‑16. Các thành phần cơ bản của hệ thống điều khiển 30](#_Toc435892384)

[Hình 3‑1. Mô hình tổng quát của hệ thống 31](#_Toc435892385)

[Hình 3‑2. Bản vẽ trên máy tính 36](#_Toc435892386)

[Hình 3‑3. Các chi tiết sau khi cắt 37](#_Toc435892387)

[Hình 3‑4. Mô hình giải thuật điều khiển 37](#_Toc435892388)

[Hình 4‑1. Kết quả sử dụng bộ lọc Kalman 40](#_Toc435892389)

# TÓM TẮT

Khi nền công nghiệp điện ảnh bùng nổ, các nhà làm điện ảnh luôn muốn có các cảnh quay tuyệt đẹp để thu hút khán giả. Bên cạnh đó các hình ảnh phải có độ trung thực cao, rõ nét không bị nhòe. Từ đó hệ thống cân bằng máy ảnh cũng được phát triển mạnh để đáp ứng cho các nhu cầu trên. Phát triển song song nó là nền công nghiệp máy bay không người lái phục vụ điện ảnh. Một số cảnh quay chỉ có thể thực hiện được bằng máy bay không người lái. Do đó hệ thống cân bằng máy ảnh lại càng phát triển, để tạo ra được các sản phẩm nhỏ gọn, khối lượng tối thiểu để có thể tích hợp vào máy bay không người lái.

Tuy nhiên nó không chỉ dừng lại ở các nhà làm điện ảnh, hiện nay hệ thống này còn được phát triển cho những người nghiệp dư thích quay phim chụp ảnh.

Do đó hệ thống cân bằng máy ảnh là một đề tài khá hấp dẫn để nghiên cứu và chế tạo.

Đề tài nghiên cứu chủ yếu dựa trên dòng vi điều khiển phổ biến trên thị trường: AVR, kết hợp với cảm biến gia tốc kế và con quay hồi chuyển để đo độ nghiêng lệch, từ đó điều khiển bộ phận giữ máy ảnh đáp ứng lại với độ nghiêng lệch này, làm cân bằng máy ảnh.

Nội dung được trình bày trong năm chương. Cụ thể, Chương 1 sẽ trình bày một cách tổng quan về đề tài, cũng như tình hình nghiên cứu trong và ngoài nước. Chương 2 là các cơ sở lý luận và phương pháp nghiên cứu. Chương 3 là phân tích và thiết kế hệ thống, đây là chương quan trọng, hiện thực hóa sản phẩm. Chương 4 là kết quả thử nghiệm và đánh giá kết quả thử nghiệm. Chương 5 là kết luận và kiến nghị cho các sản phẩm tương tự trong tương lai.

# TỔNG QUAN VỀ ĐỀ TÀI

## GIỚI THIỆU ĐỀ TÀI

Đề tài nghiên cứu bộ điều khiển cân bằng máy ảnh ba trục được lấy ý tưởng từ việc quan sát các máy bay bốn cánh có gắn máy quay phim, việc gắn máy quay phim có thể ghi lại hình ảnh từ trên cao ứng dụng cho việc giám sát hay quay phim, chụp hình. Khi gắn máy quay vào máy bay vì do ảnh hưởng trong quá trình hoạt động: rung động tạo ra từ động cơ, ảnh hưởng trong khi di chuyển, ảnh hưởng các yếu tố bên ngoài như gió sẽ làm cho hình ảnh quay về được không được đẹp và bị rung.

Một điểm nữa là khi quay phim ở những môi trường không ổn định, như quay phim khi đi cano trên biển, quay phim lúc cano chạy thì hình ảnh quay được sẽ bị lắc vì người quay phim rất khó để ổn định máy quay trong các trường hợp như thế.

Trên thực tế đã có nhiều thiết bị ổn định máy quay chuyên nghiệp giành cho ngành điện ảnh, các thiết bị này được biết đến nhiều hơn với tên gọi gimbal (từ đây sẽ gọi bộ điều khiển/thiết bị cân bằng máy ảnh ba trục là gimbal). Một số hình ảnh của những bộ gimbal dùng trong điện ảnh và cho máy bay bốn cánh điều khiển từ xa.



Hình ‑. Các gimbal thực tế trên thị trường

Giá của những bộ cân bằng này giao động từ $1000 đến trên $3000[[[1]](#footnote-1)], vì giá thành tương đối cao nên nhóm quyết định tự nghiên cứu cho mình một bộ cân bằng như thế, để có thể quay phim và chụp hình với hình ảnh đẹp nhất.

Một nguyên nhân khác cho việc chọn đề tài này là nhóm mong muốn có thể hiện thực nên một hệ thống có thể tự cân bằng và áp dụng những kiến thức này để xây dụng những ứng dụng thú vị khác như là máy bay cân bằng bốn cánh hoặc bàn chơi bida dành cho các thuyền lớn trên biển.

## TÌNH HÌNH NGHIÊN CỨU TRONG VÀ NGOÀI NƯỚC

### Trong nước

Ở Việt Nam cũng có một số nhóm nhỏ nghiên cứu về hệ thống cân bằng máy ảnh, nhưng đa phần chỉ dừng lại ở mức độ tìm hiểu lý thuyết cho các đồ án môn học chứ chưa mang tính nghiên cứu và chế tạo ra được sản phẩm thực sự.

Trong khi đó, vẫn có nhiều nhóm nghiên cứu hệ thống cân bằng theo cơ cấu cơ học hoặc các hệ thống giảm xóc đơn giản, dẫn đến hệ thống có khối lượng tương đối nặng, khó có thể tích hợp vào cá thiết bị máy bay không người lái.

### Ngoài nước

Hoàn toàn ngược lại với trong nước, hệ thống cân bằng máy ảnh ở các nước tiên tiến trên thế giới được phát triển rất mạnh từ các đề tài nhỏ đến các dự án lớn thu được lợi nhuận.

Các hệ thống khá ổn định, nhưng với giá thành tương đối cao.

## MỤC TIÊU NGHIÊN CỨU VÀ GIỚI HẠN CỦA ĐỀ TÀI

### Mục tiêu tổng quát

Nghiên cứu và chế tạo được một hệ thống cân bằng máy ảnh đơn giản, hoạt động ổn định một cách tương đối.

### Mục tiêu cụ thể

Tìm hiểu các phương pháp đọc cảm biến nghiêng lệch, lọc nhiễu tín hiệu để cho ra độ nghiêng lệch chính xác với thời gian nhanh nhất.

Thiết kế bộ khung nâng đỡ cho hệ thống.

Tìm hiểu các phương pháp điều khiển động cơ với vòng hồi tiếp kín.

## KHÁCH THỂ VÀ ĐỐI TƯỢNG NGHIÊN CỨU

### Khách thể nghiên cứu

* Chip Atmega644p
* Cảm biến MPU6050
* Servo Digital
* Phần mềm thiết kế kỹ thuật

### Đối tượng nghiên cứu:

* Bộ lọc số Kalman Filter.
* Bộ điểu khiển PID.
* Hệ thống thời gian thực.
* Giao tiếp truyền dữ liệu từ chip vi điều khiển lên máy tính.

## Ý NGHĨA LÝ LUẬN VÀ THỰC TIỄN CỦA ĐỀ TÀI

Việc nghiên cứu phát triển hệ thống cân bằng máy ảnh có ý nghĩa rất to lớn, góp phần vào sự phát triển của xã hội.

Trong thời đại hiện nay khi các thiết bị ghi hình, máy bay không người lái ngày càng phát triển, thì nhu cầu sử dụng hệ thống cân bằng máy ảnh cũng tăng mạnh. Việc nghiên cứu chế tạo giúp ta làm chủ được công nghệ hiện tại đang có của cá nước tiên tiến, từ đó phát triển lên ở mức độ cao hơn, giá thành rẻ hơn để cạnh tranh.

Qua kết quả nghiên cứu của đề tài sẽ tạo ra cái nhìn mới mẻ hơn về hệ thống cân bằng máy ảnh, làm tiền đề cho sinh viên khoa Kỹ thuật máy tính có thể phát triển lên và có ứng dụng vào thực tế.

## THUẬN LỢI VÀ KHÓ KHĂN

### Thuận lợi

* Được trường Đại học công nghệ thông tin tài trợ kinh phí mua thiết bị phục vụ nghiên cứu.
* Được sự chỉ bảo tận tình của các thầy cô trong khoa Kỹ thuật máy tính.
* Kế thừa được các kiến thức thực tiễn về cảm biến, động cơ, board mạch… từ các cuộc thi học thuật trong trường.
* Biết sử dụng các công cụ quản lý source code, quản lý dự án, quản lý thời gian… giúp cho quá trình làm việc nhóm diễn tra trôi chảy hơn.

### Khó khăn

Tuy quá trình làm đề tài khá suôn sẻ nhưng cũng không tránh được một số khó khăn nhất định.

Khó khăn lớn nhất gặp phải trong quá trình nghiên cứu là tài liệu tham khảo tương đối ít, chủ yếu bằng tiếng nước ngoài.

# CƠ SỞ LÝ LUẬN VÀ PHƯƠNG PHÁP NGHIÊN CỨU

## CƠ SỞ LÝ LUẬN

### Sử dụng cảm biến

Để tìm được sự định hướng (góc roll, pitch) của một đối tượng không chuyển động có thể sử dụng cảm biến gia tốc kế. Đối với một đối tượng tĩnh, cảm biến gia tốc kế đo “lực hấp dẫn” ở ba trục, sau đó trả về hướng của đối tượng.

Phương pháp này cho kết quả khá chính xác nếu đối tượng đo không di chuyển (thường được sử dụng nhiều nhất trên các điện thoại thông minh), còn khi di chuyển vật thể cần đo thì đã tạo một lực tác động lên nó. Điều này làm thay đổi giá trị của gia tốc kế, nên sẽ có một phương pháp khác chính xác hơn.

Thêm nữa, cảm biến gia tốc kế rất nhạy với những những lực tác động bên ngoài (ảnh hưởng của rung lắc khi sản phẩm hoạt động) vì vậy để cho hệ thống có thể hoạt động được, nhất thiết phải tìm được chính xác góc (smooth angle) của đối tượng. Đây là bước quan trọng nhất, vì khi góc của đối tượng không chính xác hay không ổn định thì việc điều khiển cân bằng cho đối tượng đố là không thể thực hiện được.

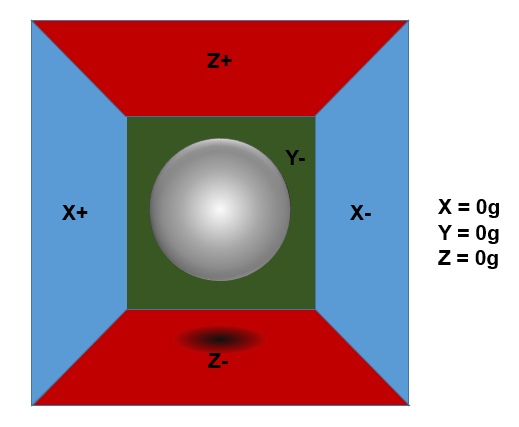
Như đã nói ở trên chỉ sử dụng gia tốc kế để tính góc quay là chưa đủ, cần phải sử dụng thêm một bộ lọc tín hiệu để giảm đi sự rung lắc tác động lên đối tượng. Bộ lọc tín hiệu ở đây là những bộ lọc số (bộ lọc mềm) được nhúng vào trong mã nguồn của hệ thống, có nhiều bộ lọc số khác nhau đã được ứng dụng như Kalman, Complementary, Ballistic…

Các bộ kể trên điều có ưu nhược điểm riêng và được ứng dụng cho các hệ thống khác nhau, Kalman được ứng dụng cho máy bay với sự kết hợp của cảm biến gia tốc và con quay hồi chuyển (hoặc kết hợp thêm cảm biến từ trường) ứng dụng cho máy bay bốn cánh. Ballistic thường kết dùng trong các máy bay bốn cánh có thêm GPS. Riêng bộ lọc Complementary được ứng dụng nhiều vào các ứng dụng trong mô hình tự cân bằng (xe cân bằng, glimbal) vì tính dể hiện thực và hiểu quả sử dụng nguồn tài nguyên trong các hệ thống nhúng nhỏ. Phần tiếp theo sẽ trình bày chi tiết hơn về hoạt động của cảm biến gia tốc, con quay hồi chuyển, hoạt động của bộ lọc Complementary, so sánh hiệu quả của nó so với các bộ lọc Kalman.

### Con quay hồi chuyển và gia tốc kế

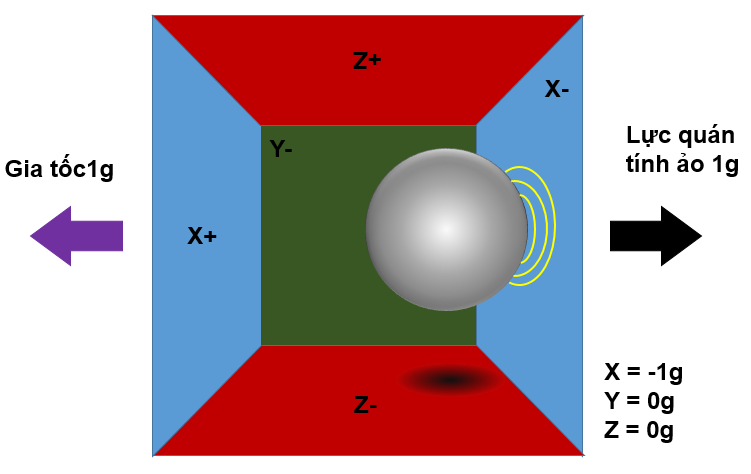
#### Gia tốc kế

Để có thể hình dung về gia tốc kế một cách rõ ràng, phương pháp hiệu quả nhất là tưởng tượng một cái hộp hình vuông và một quả bóng nhỏ ở bên trong, như hình mô phỏng bên dưới.



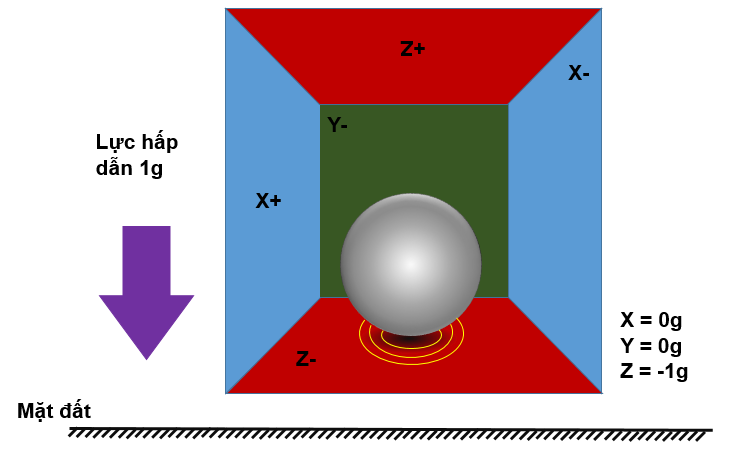
Hình 2‑1. Gia tốc kế trạng thái tĩnh

Nếu để cái hộp đó ở một nơi không có trọng lực, có nghĩa là không có bất cứ “trường” nào tác động vào quả bóng, lúc đó quả bóng sẽ ở vị trí giữa hộp. Hình trên đã bỏ đi mặt phẳng Y+, tưởng tượng mỗi mặt phẳng (vách của chiếc hộp) có thể đo được lực va chạm của quả bóng vào nó (đo được độ nhạy – sensitive). Nếu bất thình lình duy chuyển cái hộp về phía bên trái (giả sử với gia tốc 1g = 9.8 m/s2), lúc này quả bóng sẽ duy chuyển rồi chạm vào vách X-, bằng cách đo lực tác dụng của quả bóng vào vách của hộp ta được kết quả lực (như giả sử là -1g) trên trục X. Hình tiếp theo sẽ mô tả cho quá trình trên.



Hình 2‑2. Gia tốc kế di chuyển sang trái

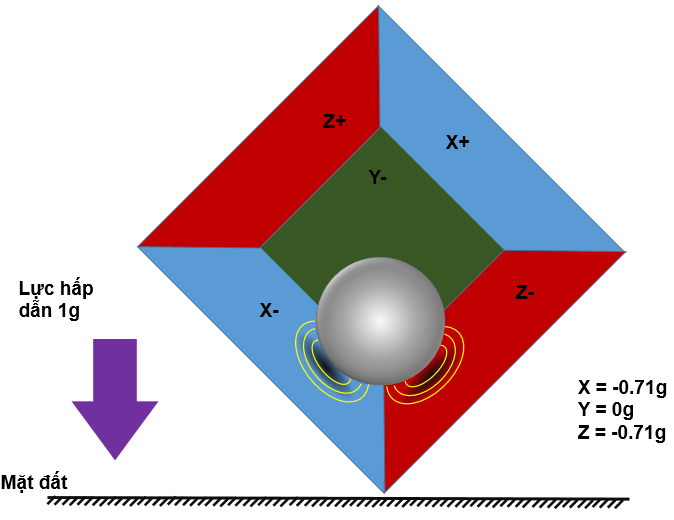
Chú ý rằng gia tốc kế sẽ lấy lực tác dụng có hướng ngược lại với hướng vector gia tốc, lực này gọi là lực khả dĩ (Fictitious Force). Gia tốc kế đo được gia tốc thông qua lực gián tiếp tác dụng vào vách ngăn của nó, lực này có thể được gây ra bởi gia tốc nhưng trong thực tế thì không phải chỉ có lực gia tốc. Nguyên nhân vì gia tốc kế được sử dụng ở Trái Đất, nghĩa là nó có lực hấp dẫn tác dụng lên nó, bình thường thì quả bóng sẽ chạm vào một phía nào đó của vách ngăn (mặt phẳng Z- chẳng hạn).



Hình 2‑3. Ảnh hưởng của lực hấp dẫn

Đó là lí do tại sao chiếc hộp mặc dù không có di chuyển nhưng ta vẫn đọc được giá trị -1g ở trục Z. Kết quả đo này là do lực hấp dẫn tác dụng vào quả bóng chứ không phải ta duy chuyển chiếc hộp. Trong thực tế có thể có nhiều lực khác nhau tác dụng lên gia tốc kế, ví dụ, nếu quả bóng trong chiếc hộp được làm từ kim loại, vậy thì trong môi trường có từ trường, chắc chắn kết quả đo sẽ bị ảnh hưởng.

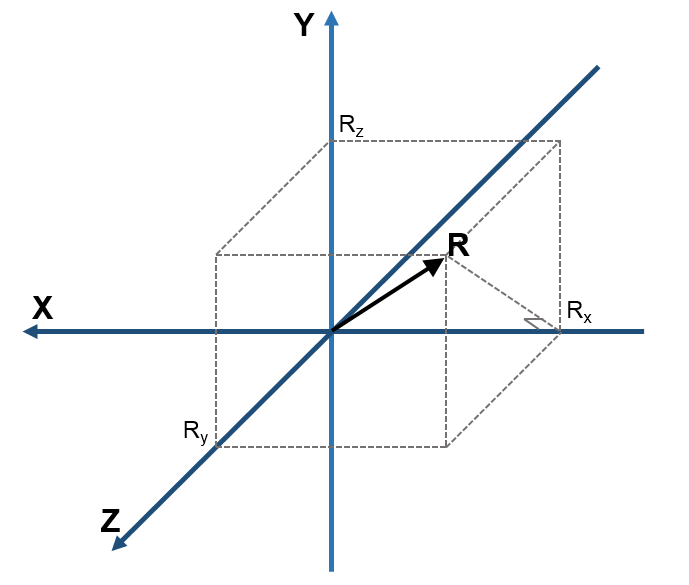
Giá trị thực của gia tốc kế đến từ phép đo của cả 3 trục, ví dụ nếu xoay chiếc hộp một góc 45 độ, cùng chiều kim đồng hồ, lúc này quả bóng chạm vào 2 vách ngăn là Z- và X- như hình bên dưới.



Hình 2‑4. Tác dụng vào 2 hướng

Kết quả đo được là: X = -0.71g, Y = 0g và Z = -0.71g

Bây giờ chuyển chiếc hộp thành hệ trục tọa độ và bắt đầu tìm hiểu mối liên hệ giữa các đại lượng.



Hình 2‑5. Hệ trục tọa độ gia tốc kế trong không gian

Mỗi trục tọa độ tương ứng với bề mặt của chiếc hộp và vector R thể hiện giá trị gia tốc mà gia tốc kế đo được. Rx, Ry, Rz là hình chiếu của vector R lên trục X, Y và Z. Mối liên hệ của R và Rx, Ry, Rz như sau:



Phương Trình: ‑

Áp dụng Phương Trình: ‑ với R = 1, RX = 1/2, RY = 1/2 và RZ = 1/2 ta được:



=> 

Trước khi trình bày tiếp về hình học 3 chiều cần phải nói đến một chút về cảm biển nghiêng, làm thế nào thu được giá trị của cảm biến rồi sau đó chuyển đổi nó thành đại lượng hình học. Cảm biến gia tốc chủ yếu chia làm 2 loại, một loại là digital và một loại là analog, Đối với cảm biến gia tốc analog, giá trị ngõ ra của cảm biến là mức của điện áp, giá trị điện áp này cần được tham chiếu và đọc về bằng bộ ADC của vi điều khiển. Trong khi đó các loại cảm biến gia tốc digital cho ra đại lượng nhị phân trực tiếp mà không cần bộ chuyển đổi ADC, tuy nhiên cần phải sử dụng các chuẩn giao tiếp như I2C, SPI hay UART (tùy theo cảm biến hỗ trợ chuẩn giao tiếp nào) để đọc giá các trị này về.

Ví dụ với bộ ADC 10 bit (khoảng giá trị ADC là 0 đến 210 -1), giá trị ADC đọc được từ các trục của cảm biến lần lượt là:

valADC\_Rx = 586

valADC\_Ry = 630

valADC\_Rz = 561

Mỗi bộ ADC đều có điện áp tham chiếu, giả sử điện áp tham chiếu trong ví dụ này là Vref = 3.3V. Việc chuyển đổi giữa giá trị điện áp và giá trị ADC đọc được ta dùng phương trình sau:

VoltsRx = valADC\_Rx \* Vref/1023

Vậy ta được:

VoltsRx = 586\*3.3V/1023 = ~1.89V

VoltsRy = 630\*3.3V/1023 = ~2.03V

VoltsRz = 561\*3.3V/1023 = ~1.81V

Cảm biến giá tốc đều có giá trị điện áp ở zero-g, giá trị này thường có trong datasheet của chip, giả sử ở 0g có VzeroG = 1.65V ta được:

DeltaVoltsRx = 1.89 – 1.65 = 0.24V

DeltaVoltsRy = 2.03 – 1.65 = 0.38V

DeltaVoltsRz = 1.81 – 1.65 = 0.16V

Một giá trị khác cần được biết đến ở các chip cảm biến nghiêng là độ nhạy của nó, độ nhạy là giá trị thay đổi điện áp trên mỗi thay đổi của g, thường đơn vị này là mV. Giả sử với độ nhạy của chip là: Sensitivity = 478.5mV/g = 0.4785V/g Giá trị cuối cùng cần tìm là lực tác dụng có đơn vị g là:

R = DeltaVoltsR/Sensitivity

Rx = 0.24V/0.4785V/g = ~0.5g

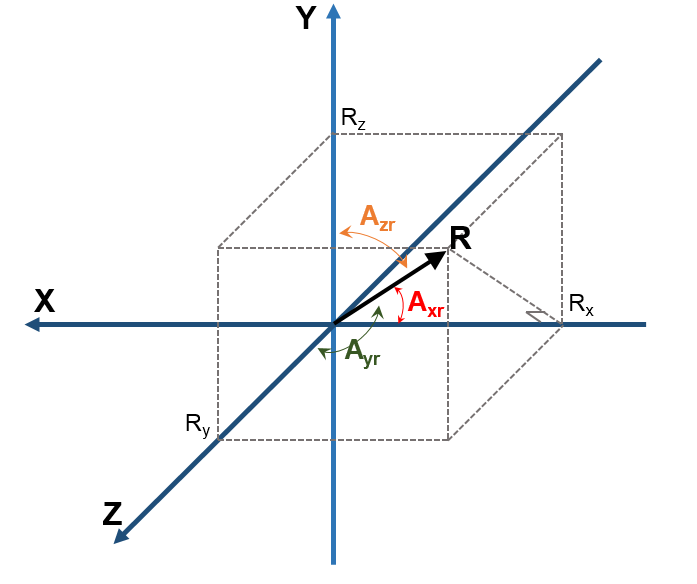
Ry = 0.38V/0.4785V/g = ~0.79g

Rz = 0.16V/0.4785V/g = ~ 0.33g

Công thức cuối cùng thu được là:



Phương Trình: ‑



Hình 2‑6. Hệ trục tọa độ và góc nghiêng

Với các giá trị R vừa tìm được, có thể tìm được độ dốc của cảm biến so với mặt đất, theo như công thức trên thì chỉ cần tìm góc lệch của vector R với trục Z.

Gọi góc Axr là góc giữa R và trục X, góc Ayr là góc giữa R và trục Y và góc Azr là góc giữa R và trục Z. Ta được:

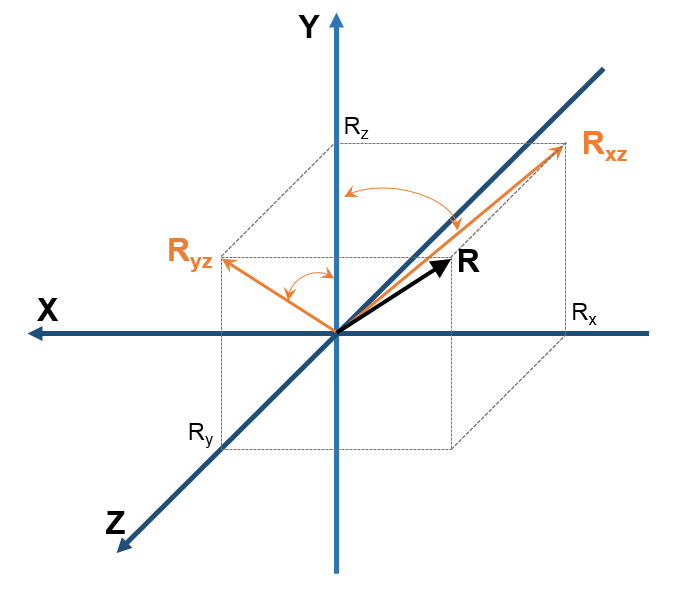
  



Phương Trình: 2‑4

#### Con quay hồi chuyển

Hệ trục để diễn ta con quay hồi chuyển như hình sau:



Hình 2‑7. Hệ trục biểu diễn con quay hồi chuyển

Mỗi kênh của con quay hồi chuyển đo góc quay quanh trục 1 trục của kênh đó. Với con quay hồi chuyển 2 trục ta sẽ có được góc quay của cảm biến tương ứng với cả 2 trục là X, Y.

Gọi Rxz là hình chiếu của vector R lên mặt phẳng XZ.

Ryz là hình chiếu của vector R lên mặt phẳng YZ.

Sử dụng định lý Pitago trong tam giác vuông ta được:

 và 

Vì con quay hồi chuyển đo tỉ lệ thay đổi của góc theo thời gian nên ta ví dụ ở thời điểm t0 có được góc và tại thời điểm t1 có góc, tỉ lệ thay đổi đó là:



Nếu có giá trị là độ và thời gian t là giây, thì giá trị của  sẽ là deg/s. Đây là những gì mà con quay hồi chuyển đo được cho chúng ta.

Trong thực tế (nếu không phải là con quay hồi chuyển digital) thì giá trị đọc được từ cảm biến là giá trị điện áp, vì vậy cần chuyển thiết để chuyển nó về giá trị deg/s theo mong muốn. Tương tự như gia tốc kế ta được phương trình chuyển đổi của con quay hồi chuyển như sau:

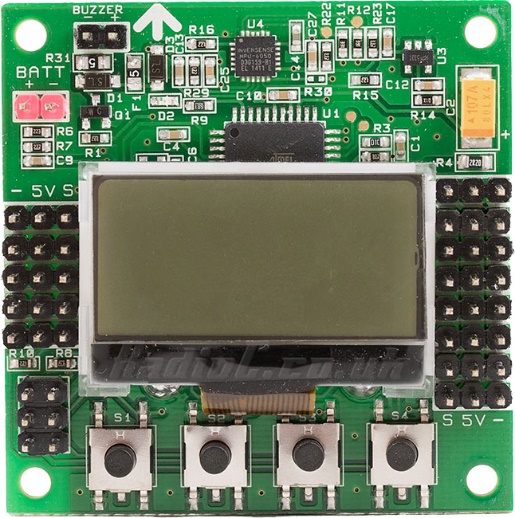




Phương Trình: ‑

### Board mạch điều khiển hệ thống

Board cung cấp khá nhiều tính năng hỗ trợ cho việc nghiên cứu và phát triển.



Hình 2‑8. Board điều khiển

Dưới đây là các thông số chi tiết:

* Chip xử lý trung tâm: Atmega644PA
* Chip vi điều khiển 8 bits đa dụng
* Tần số hoạt động: 20MHz
* Flash ROM 64 Kbytes
* EEPROM 2 Kbytes
* SRAM 16 Kbytes.
* Có 3 bộ Timer, 6 bộ PWM.
* Các chuẩn giao tiếp: I2C, SPI, USART.
* Tích hợp con quay hồi chuyển và gia tốc kế: MPU6050
* MPU-6050 là phần không thể thiếu trong công nghệ xử lý chuyển động bằng cách kết hợp một MEMS 3 trục con quay hồi chuyển và cảm biến gia tốc trên một IC, cùng vớ bộ xử lý chuyển động kỹ thuật số (Digital Motion Processor™ - DMP™), có khả năng xử lý phức tạp thuật toán MotionFusion 9 trục.
* Tích hợp sẵn bộ ADC 16 bits cho mỗi trục.
* Giao tiếp thông qua chuẩn I2C.
* Tầm đo của con quay hồi chuyển: +250, 500, 1000, 2000°/s.
* Tầm đo của gia tốc kế: ±2, ±4, ±8, ±16g.
* Nút nhấn, đèn led, LCD
* Bốn nút nhấn, một đèn led báo hiệu.
* LCD 168x64 theo chuẩn gia tiếp I2C.

### Động cơ Servo KST DS521MG



Hình 2‑9. Servo KST DS521MG

Thông số hoạt động của servo:

* Điện áp hoạt động: 4.5 – 6V
* Lực nâng: 3.18kg/cm ở điện áp 6V
* Tốc độ: 0.06s/600 ở điện áp 6v
* Kích thước: 22.90 x 12.27 x 27.30 mm
* Tần số hoạt động: 1520us/333Hz

### Bộ lọc Kalman

#### Khái niệm bộ lọc số

#### Lý thuyết bộ lọc Kalman

Kalman là một thuật toán sử dụng liên tiếp các phép do theo thời gian, phép đo lấy cả giá trị của gia tốc kế và con quay hồi chuyển. Giá trị của cảm biến thường bao gồm theo giá trị nhiễu của hệ thống và đó là nguyên nhân gây ra lỗi khi đọc giá trị của cảm biến về. Bộ lọc Kalman sẽ cố gắng dự đoán trạng thái của hệ thống bằng cách dựa vào trạng thái hiện tại và trạng thái trước đó của hệ thống, cách này sẽ cho kết quả tốt hơn nếu chỉ đơn thuần đo một lần tại thời điểm cần. Như đã nói vấn đề lỗi của cảm biến gia tốc khi có rung lắc, có sự ảnh hưởng của môi trường xung quanh, vấn đề trôi của cảm biến con quay. Các kết quả trả về là chính xác nhất khi cảm biến gia tốc được đo với tần số cao trong con quay hồi chuyển ở điều kiện đo lâu dài.

Bộ lọc Kalman hoạt động bằng cách tạo ra những thống kê tối ưu của trạng thái hệ thống dựa vào các phép đo liên tiếp. Làm được điều này thì bộ lọc cần phải biết nhiễu của ngõ vào thông qua phép đo nhiễu và nhiễu gây ra bởi hễ thống gọi là nhiễu tiến trình.

#### Trạng thái của hệ thống.

Xk là trạng thái của hệ thống ở thời điểm k. Nếu tại thời điểm k ta có trạng thái của hệ thống là

xk = Fxk-1 + Buk + wk  **(1)**

* Ma trận trạng thái của xk­:



Giá trị  là ngõ ra của bộ lọc và độ chênh dựa vào kết quả đo của cảm biến gia tốc và con quay. Độ chênh nó là đại lượng mà con quay đã bị trôi. Điều đó có nghĩa là có thể lấy được giá trị thực bằng cách trừ đi giá trị chênh lệch .

* Tiếp theo là ma trận F, ma trận F là trạng thái của sử chuyển đổi được áp dụng cho trạng thái trước đó xk-1. Định nghĩa của F như sau:



* Giá trị ngõ vào uk , trong trường hợp này là kết quả đo của con quay hồi chuyển với đơn vị là độ/giây ở thời điểm k. Nó còn được gọi là , vậy phương trình (1) trở thành

 **(2)**

* Ma trận B được gọi là ma trận giá trị ngõ vào điều khiển. Được định nghĩa như sau:



Ta sẽ có được góc , bằng cách nhân tỉ lệ thay đổi  cho .

* wk là nhiễu của tiến trình xử lý, w­k ~ N(0,Qk)

Qk là ma trận hiệp phương sai dựa trên thời điểm hiện tại k, và trong trường hợp này là ma trận hiệp phương sai của cửa trạng thái gia tốc dự đoán và độ lệch. Trong trường hợp này cần xem xét đến dự đoán cảu độ lệch và giá trị của gia tốc mà nó dựa vào. Định nghĩa của ma trận Qk như sau:



Phép đo zk: Bây giờ hãy khảo sát giá trị của zk ở trạng thái xk. Phương trình của zk như sau:



Trong đó:

* Ma trận H được gọi ra để theo dõi quá trình và sử dụng để ánh xạ vào giá trị thật sự của vùng giá trị cần theo dõi, giá trị thuật không thể không thể theo dõi trực tiếp được. Phương trình của H như sau:



* Vk ~ N(0,R)

Giá trị của R không phải là một ma trận nó chỉ là biến của phép đo, định nghĩa của R: R = var(vk).

#### Phương trình của bộ lọc Kalman

Bây giờ hãy sử dụng trạng thái đánh giá giá trị thật của hệ thống tại thời điểm k, . Phương trình có thể viết thành nhiều bước khác nhau.

Bước 1: Dự đoán (predict)

Trong phần đầu của 2 phương trình sẽ cố gắng dự đoán cho trạng thái hiện tại và ma trận lỗi hiệp phương sai ở thời điểm k. Phần đầu cảu bộ lọc cố gắng đánh giá trạng thái hiện tại dựa trên tất cả trạng thái của hệ thống trước đó và giá trị của con quay hồi chuyển.



Đây còn được gọi là ngõ vào điều khiển. Thành phần tiếp theo đánh giá lỗi hiệp phương sai của ma trận Pk|k-1 thành phần đó như sau:



Ma trận hiệp phương lỗi P là ma trận 2x2:



Bước 2: Cập nhật (update)

Đầu tiên là tính các giá trị khác nhau của zk và trạng thái xk|k-1, nó còn được gọi là sự cải tiến



H dùng để ánh xạ đến trạng thái xk|k-1 bên trong vùng quan sát với giá trị mà gia tốc kế đo được.

Phần tiếp theo cần phải tính toán đó là hiệp phương sai mới Sk và Kalman gain Kk

, 

Phương trình của trạng thái hiện tại sẽ là:



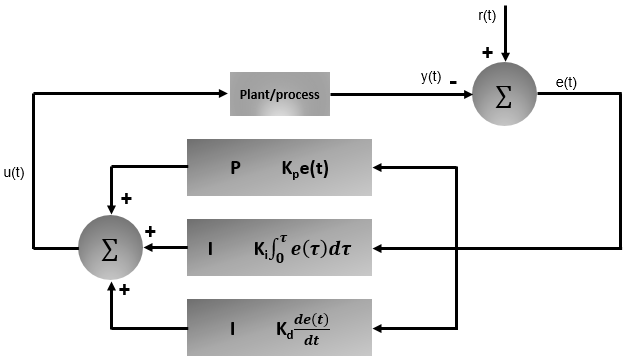
Cập nhật giá trị mới cho lỗi sau khi có hiệp phương sai ma trận.

, I còn được gọi là ma trận đồng nhất được định nghĩ như sau:



### Bộ điều khiển PID

PID (Propotional-Integral-Derivative) là hệ thống điều khiển vòng kín làm cho kết quả thực tế gần hơn với kết quả mong muốn bằng cách điều chỉnh đầu vào.



Hình 2‑10. Bộ điều khiển PID

Đầu ra u(t) của giải thuật PID là:

Có ba thông số trong PID (không âm) tương ứng là: P (tỉ lệ), I (tích phân), D (đạo hàm). P phụ thuộc vào sai lệch hiện tại. I dựa trên sự tích luỹ sai lệch trong quá khứ. D là dự đoán sai lệch tương lai dựa trên tỉ lệ sai lệch hiện tại.

Cần đo đầu ra từ các cảm biến (như góc nghiêng) để có thể ước lượng sai lệch (thực tế chênh lệch bao nhiêu so với góc nghiêng mong muốn)

Ảnh hưởng của ba tham số:

* **Hệ số P:** Hệ số này được xác định bởi người điều khiển. Khi giá trị P tăng cao, hệ thống sẽ trở nên nhạy hơn với sự thay đổi góc và khi P quá cao, hệ thống bắt đầu dao dộng. Khi P quá thấp hệ thống sẽ có đáp ứng chậm và khó khăn hơn để giữ ổn định.
* **Hệ số D:** Hệ số này tăng độ chính xác của vị trí góc. P là cho sai lệch giảm dần, nhưng do hệ thống có tồn tại lực quán tính nên P sẽ gây ra hiện tượng dao động quanh điểm cân bằng. Mục tiêu của D là loại bỏ điều này. Khi càng gần điểm cân bằng, thành phần D sẽ tăng lên, giúp hệ thống không vượt quá xa điểm cần bằng.
* **Hệ số I:** Hệ số này cho hệ thống nhanh chóng đạt được trạng thái mong muốn. Trên thực tế nó tăng tốc độ phản ứng và tăng hiệu quả của hệ số P.

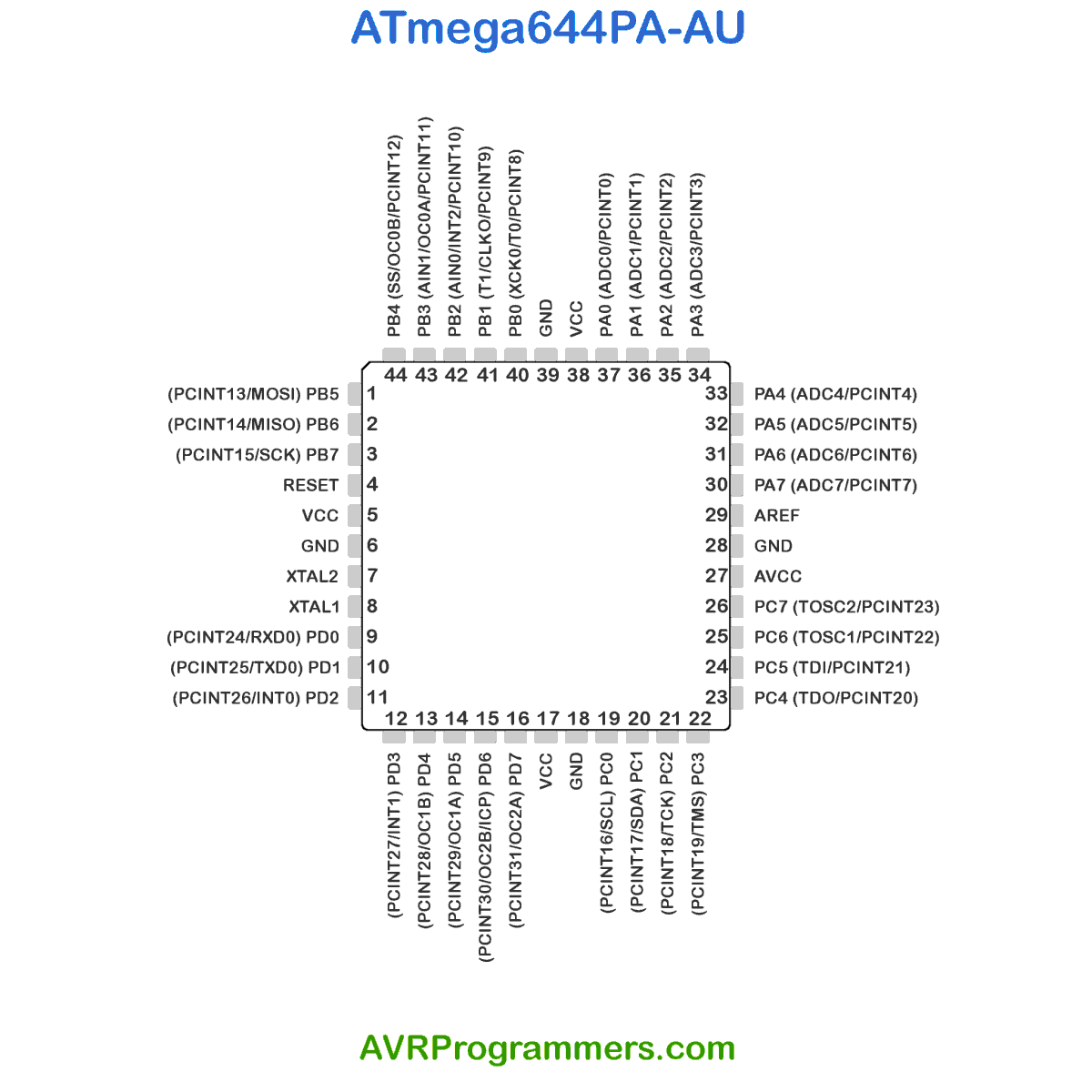
### Các chuẩn giao tiếp được sử dụng

#### USART

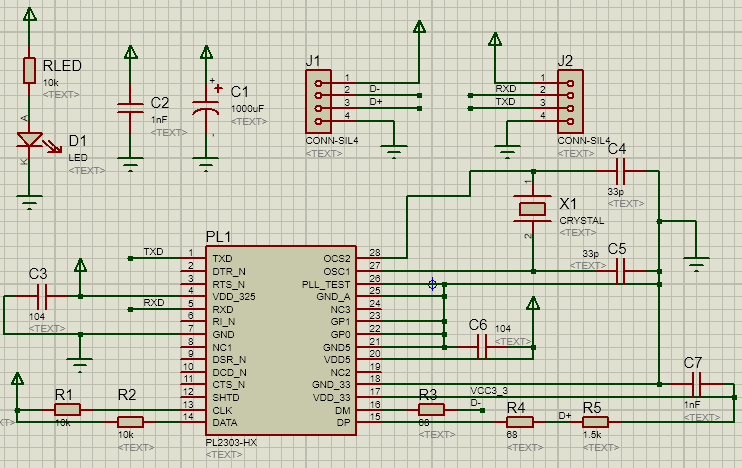
Chuẩn truyền nhận nối tiếp RS232 được giới thiệu vào năm 1962 bởi Radio Sector của EIA[6], mục đích ban đầu của chuẩn giao tiếp này là để kết nối giữa terminal với các modem và hiện nay vẫn được dùng rất phổ biến trong các hệ thống nhúng, một số đặc điểm của chuẩn giao tiếp RS232:

* Có 9 chân kết nối, gọi là DB-9, đối với các máy tính cũ hơn thì sử dụng 25 chân kết nối gọi là DB-25.
* Có thể cho phép truyền song công, nghĩa là việc truyền và nhận dữ liệu được thực hiện cùng một lúc.
* Tốc độ giao tiếp tối đa: 1.5Mbps.
* Nhóm sử dụng 3 chân là GND, Rx và Tx để thực hiện kết nối. Tx của thiết bị này sẽ được nối với Rx của thiết bị kia, hình sao mô tả cách kết nối giữa vi điều khiển Atmega644p với máy tính.

Hình 2‑11. Sơ đồ kết nối với máy tính



* Mạch chuyển đổi thông dụng nhất hiện nay là mạch sử dụng IC PL2303. Đây là IC tương đối ổn định và hiện nay được sử dụng rất rộng rãi. Dưới đây là mạch giao tiếp[[[2]](#footnote-2)] sử dụng PL2303:



Hình 2‑12. Mạch chuyển đổi USB sang COM

* Thiết lập UART cho vi điều khiển:
* Baud rate 115200.
* Stop bit 1.
* None parity.

Giá trị các thanh ghi điều khiển cho uart

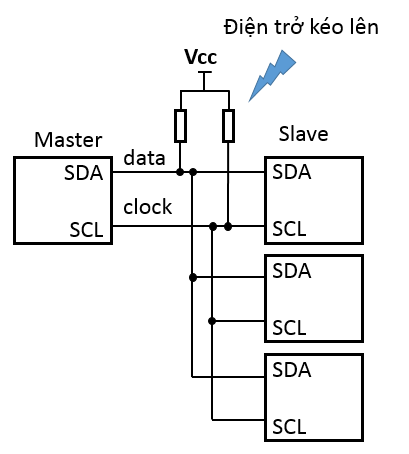
|  |
| --- |
| UBRR1 = 10;  UCSR1A = 0;  UCSR1B = \_BV(TXEN1) | \_BV(RXEN1) | \_BV(RXCIE1);  UCSR1C = \_BV(UCSZ11) | \_BV(UCSZ10); |

#### I2C

Chuẩn giao tiếp 2 dây I2C, I2C (I2C) là từ viết tắt của cụm từ Inter-Integrated Circuit là giao tiếp nối tiếp nhiều master, nhiều slave được phát minh bởi Philips Semiconductor[7], với mục đích thực hiện truyền nhận dữ liệu nối tiếp giống như SPI và chỉ áp dụng khoảng cách giao tiếp gần giống như UART. Tốc độ truyền nhận của hầu hết thiết bị dùng I2C là từ 100kHz đến 400Khz, đặc biệt (như atmega328) tốc độ có thể thiết lập cho những trường hợp đặc biệt lên đến 1Mhz (fast-mode plus), 3.4Mhz (high-speed mode) hay tối đa 5Mhz (ultral-fast mode)[8].

Đặc tả phần cứng của I2C: mỗi bus I2C sử dụng 2 chân tín hiệu là SDA và SCL. SDA dùng để thực hiện truyền dữ liệu và SCL là tín hiệu xung clock điều khiển. Tín hiệu clock điều khiển được điều khiển bới chip master.

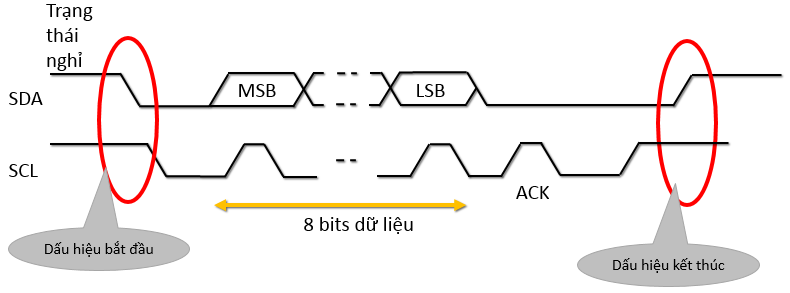
Tất cả các thiết bị dùng I2C đều có cực hở, nghĩa là sẽ không có mức điện áp cao ở các chân SDL và SCL mà chỉ kéo xuống mức 0v mà thôi. Vì vậy để I2C hoạt động được cần thiết phải có một điện trở kéo lên Vcc ở giữa master và salve. Điện trở này thường dùng là 4k7 Ohm với khoảng cách dưới 3m, khi dùng nhiều slave hay khoảng cách xa hơn cần phải sử dụng điện trở có giá trị nhỏ hơn.



Hình 2‑13. Kết nối cho các thiết bị sử dụng I2C

Hình trên mô phỏng kết nối cho các thiết bị sử dụng I2C, như đã trình bày cần có các điện trở nối với nguồn Vcc cho các chân SDA và SCL. Hoạt động của giao tiếp I2C cơ bản như sau: Khi ở trạng thái bình thường không hoạt động cả 2 chân SDL và SCL điều ở trạng thái nghỉ (IDLE) mức cao và khi master muốn thực hiện việc truyền nhận thì master sẽ kéo chân SDA xuống thấp trong khi chân SCL vẫn ở mức cao, đây là dấu hiệu bắt đầu việc truyền nhận. Sau khi có dấu hiệu bắt đầu thì trên chân SDA sẽ là 8 bits dữ liệu và bit ACK/NACK. Tám bits dữ liệu này có thể phát đi từ master hay slave tùy thuộc vào master muốn nhận hay muốn truyền dữ liệu.

Tuy nhiên, đối với lần thực hiện việc truyền nhận đầu tiên (dấu hiệu bắt đầu từ trạng thái nghỉ) master cần truyền 8 bit dữ liệu trong đó có 7 bit địa chỉ và một bit W/R (chỉ ra rằng master muốn ghi hay muốn đọc dữ liệu vào slaver) và 1 bit ACK/NACK (gửi từ slaver) thông báo việc nhận dữ liệu có thành công hay không. Như ta thấy trên đường truyền SDA sẽ có 9 bits dữ liệu. Sau khi xác nhận được slaver thì việc truyền nhận được thực hiện dữ liệu tương tự như truyền địa chỉ. Việc truyền nhận kết thúc khi master thực hiện một điều kiện kết thúc, điều kiện kết thúc là master sẽ kéo chân SDA lên cao trong khi SCL đang ở mức cao. Hình sau mô tả cho hoạt động truyền nhận dữ liệu I2C.



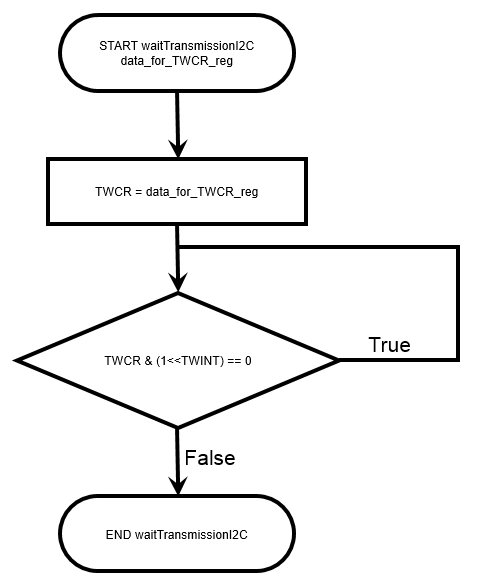
Hình 2‑14. Hoạt động truyền nhận dữ liệu I2C

Thiết lập I2C cho chip atmega644p: Các thiết lập thanh ghi hoạt động cho chip như sau:

|  |
| --- |
| TWSR = 0; // no prescaler => prescaler = 1  TWBR = ((F\_CPU / 400000) - 16) / 2; // set the I2C clock rate to 400kHz  TWCR = 1 << TWEN; |

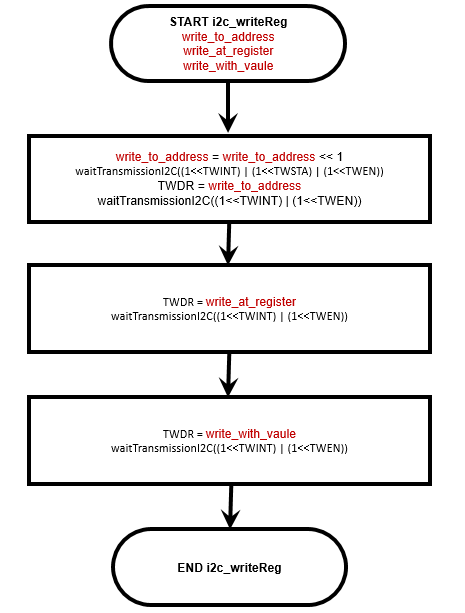
Sơ đồ khối cho giao tiếp I2C master (Atmega644p) slaver (mpu6050).

* Hàm **waitTransmissionI2C**, chờ giá trị cần thiết lập và bit báo ngắt có giá 1, sẽ được dùng cho các hàm **i2c\_writeReg** và **i2c\_read\_reg\_to\_buf**.



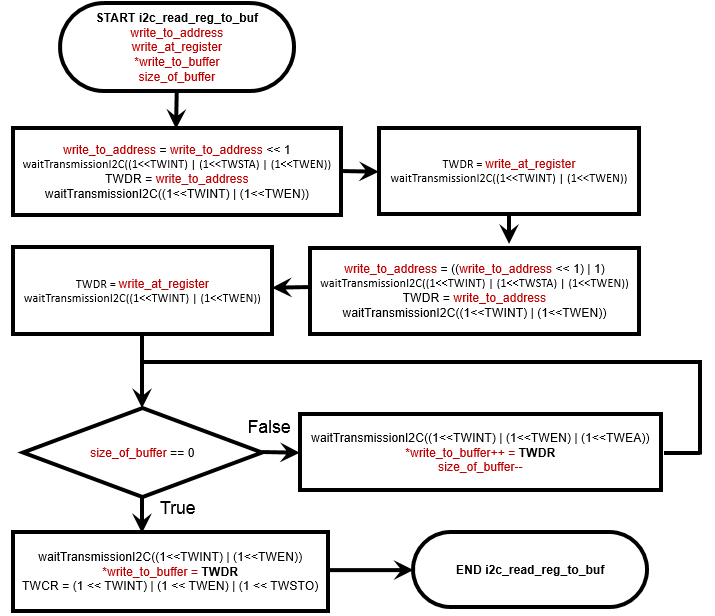
Hình 2‑15. Quy trình truyền - nhận dữ liệu I2C

* Hàm **i2c\_writeReg**. Ghi dữ liệu vào mpu6050 với địa chỉ mong muốn.



Hình 2‑16. Quy trình ghi dữ liệu I2C

* Hàm **i2c\_read\_reg\_to\_buf** đọc giá trị trả về từ salve và gán giá trị vào con trỏ để trả về.



Hình 2‑17Quy trình đọc mảng dữ liệu I2C

#### PWM

PWM là từ viết tắt của cụm từ Pulse Width Modulation (phương pháp điều chế độ rộng xung). Đây là phương pháp điều khiển các thiết bị khác bằng cách thay đổi độ rộng của xung cao.

Một số định nghĩa cần biết khi sử dụng PWM:

* Period: Là chu kỳ của xung.
* Pulse Width: là độ rộng của xung cao.
* Duty: là tỉ lệ phần trăm của Pulse Width với Period.

Thiết bị nhận được tín hiệu PWM sẽ dựa vào Duty khác nhau để thực hiện các lệnh khác nhau. Đối với trong ứng dụng điều khiển tốc độ thì Duty tỉ lệ thuận với tốc độ quay của thiết bị, ví dụ như Duty 70% sẽ cho ra tốc độ 70% so với tốc độ tối đa. Áp dụng PWM để điều khiển RC servo cần phải biết thông số kỹ thuật của servo để việc điều khiển được chính xác.

Vi điều khiển Atmega644p cung cấp tối đa 6 kênh PWM nên có thể dễ dàng để điều khiển ba động cơ servo cho ba trục gimbal. Giá trị thiết lập cho các thanh ghi của chip AVR.

|  |
| --- |
| OCR1A = TCNT1;  OCR1B = TCNT1;  TIFR1 |= \_BV(OCF1A) | \_BV(OCF1B);// clear OCA and OCB interrupt  TIMSK1 |= \_BV(OCIE1A) | \_BV(OCIE1B);// enable OCA and OCB interrupt |

### Tối ưu hiệu năng hệ thống nhúng

Trong một hệ thống nhúng thì hiệu năng có ý nghĩa như là sự thành công, một hệ thống nhúng sẽ phải hoạt động với nguồn tài nguyên hạn chế và thời gian hoạt động dài. Khác với các chương trình trên máy tính, chương trình trên hệ thống nhúng cần đáp ứng các yếu tố về bộ nhớ RAM, ROM, EPPROM, tốc độ xung nhịp cấp vào và các yếu tố này có giới hạn. Vì vậy việc lập trình cần chú ý đến hiệu quả sử dụng bộ nhớ, tránh tình trạng rõ rĩ bộ nhớ (memory leak). Đối với tốc độ xử lý, một nguyên tắc cần theo là trong một lệnh C/C++ sinh ra càng ích mã máy càng tốt.

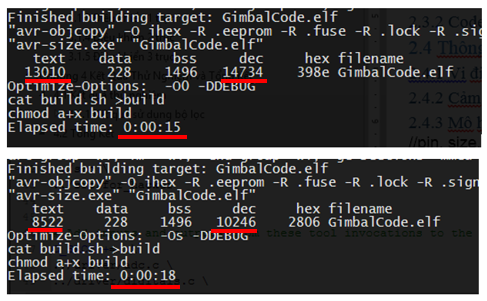
Một yếu tố về hiệu măng là thời gian mà hệ thống hoạt động. Một chương trình nhúng cần phải hoạt động ổn định trong một thời gian dài. Để tối ưu được hiệu năng, nhóm thực hiện hai cách sau đó là dựa vào trình biên dịch và dựa vào chính các thành viên trong nhóm.

Trình biên dịch mà nhóm sử dụng là avr-gcc, đây là trình biên dịch C được viết cho AVR. Bộ biên dịch này miễn phí nằm ở dự án AVR libs (không nằm trong dự án GNU[2]) bao gồm các công cụ hỗ trợ như avr-binutils, avr-gcc và avr-libc là trái tim của công cụ lập trình cho avr.

Trong các tùy chọn của avr-gcc có tùy chọn tối ưu lúc biên dịch, bao gồm các cờ -O0, -O1, -O2, -O3 và –Os[3, Ch. 2.2] tương ứng với các mức tối ưu khác nhau.Nội dung của các cờ như sau:

* -O0 không sử dụng tối ưu lúc biên dịch
* -O1 trình biên dịch cố gắng giảm bớt kích thước mã nguồn và thời gian thực thi nhưng không thực hiện bất kỳ tối ưu nào phải mất nhiều thời gian biên dịch.
* -O2 tối ưu nhiều hơn –O1, thực hiện tất cả các tối ưu được hỗ trợ mà không liên hệ đến space-speed, tùy chọn này làm tăng thời gian biên dịch.
* -O3 mở tất cả các tối ưu của –O2 đồng thời mở thêm các tối ưu đặc biệt.
* -Os tối ưu làm giảm kích thước code được sinh ra.

Dưới đây là kết quả tối ưu và thời gian biên dịch với với 2 tùy chọn (qua một số lần chạy thử kết quả dưới đây là kết quả trung bình):



Hình 2‑18. Kết quả tối ưu việc biên dịch

Ta thấy rõ thời gian biên dịch của tùy chọn không bật cờ tối ưu sẽ nhanh hơn thời gian biên dịch bằng cờ -Os, đổi lại là kích thước text lại cao hơn so với tùy chọn –Os. Tổng kết lại ta có kết quả như sau.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Tùy chọn –O0 | | Tùy chọn –Os | |
| Thời gian biên dịch | 15s | | 18s | |
| Bộ nhớ Flash | Text: 13010 | Tổng: 13238 bytes | Text: 8522 | Tổng: 8750 bytes |
| Data: 228 | Data: 228 |
| Bộ nhớ RAM | BSS: 1496 | Tổng: 1724 bytes | BSS: 1496 | Tổng: 1724 bytes |
| Data: 228 | Data: 228 |

Nội dung của vùng nhớ text, data, bss, hex như sau[4]:

* text: chứa mã và dữ liệu chỉ đọc (biến constant) trong ứng dụng (bytes).
* data: dữ liệu đọ và ghi trong chương trình (các biến bình thường) (bytes).
* bss: chỉ ra dữ liệu khởi tạo ban đầu là zero.
* dec: tổng số byte của text + data + bss.
* hex: dạng hệ 16 của dec.

## PHƯƠNG PHÁP NGHIÊN CỨU

### Phương pháp nghiên cứu trực tiếp

Đặc điểm của cách giải quyết vấn đề này là đều xác định trực tiếp được lời giải qua một thủ tục tính toán (công thức, hệ thức, định luật…) hoặc qua các bước căn bản để có được lời giải. Đối với phương pháp này, việc giải quyết vấn đề trên máy tính chỉ là thao tác lập trình hay là sự chuyển đổi lời giải từ ngôn ngữ  
bên ngoài sang các ngôn ngữ được sử dụng trong máy tính. Tìm hiểu về phương pháp này chính là tìm hiểu về kỹ thuật lập trình trên máy tính.

Các nguyên lý được áp dụng trong phương pháp này được kể ra như sau:

* Nguyên lý 1: Chuyển đổi dữ liệu bài toán thành dữ liệu của chương trình, có nghĩa là “Dữ liệu của bài toán sẽ được biểu diễn lại dưới dạng các biến của chương trình thông qua các quy tắc xác định của ngôn ngữ lập trình cụ thể”
* Nguyên lý 2: Chuyển đổi quá trình tính toán của bài toán thành các cấu trúc của chương trình, có nghĩa là “Mọi quá trình tính toán đều có thể mô tả và thực hiện dựa trên ba cấu trúc cơ bản: Cấu trúc tuần tự, cấu trúc rẽ nhánh và cấu trúc lặp”.
* Nguyên lý 3: Phân chia bài toán ban đầu thành những bài toán nhỏ hơn, có nghĩa là “Mọi vấn đề-bài toán đều có thể giải quyết bằng cách phân chia thành những vấn đề-bài toán nhỏ hơn”.

### Phương pháp nghiên cứu gián tiếp

Đặc điểm của phương pháp gián tiếp là được sử dụng khi chưa tìm ra lời giải chính xác của vấn đề. Chính vì thế, phương pháp gián tiếp chỉ đưa ra các giải pháp mang đặc trưng của máy tính.

Các phương pháp nghiên cứu gián tiếp đã sử dụng trong đề tài đó là:

* Phương pháp thử - sai:

Các nguyên lý áp dụng trong phương pháp này bao gồm:

* *Nguyên lý vét cạn toàn bộ*: Đây là phương pháp đơn giản nhất, liệt kê tất cả các trường hợp có thể xảy ra.
* *Nguyên lý mắt lưới*: Vận dụng nguyên lý này để sàn lọc ra những trường hợp nào phù hợp với các điều kiện đưa ra, giống như lưới bắt cá chỉ bắt được những con các có kích thước lớn hơn kích thước của mắt lưới.
* *Nguyên lý giảm độ phức tạp của thử và sai*: Thu hẹp trường hợp trước và trong khi duyệt, đồng thời đơn giản hóa tối đa điều kiện chấp nhận một trường hợp.
* *Nguyên lý thu gọn không gian tìm kiếm*: Loại bỏ những trường hợp hoặc nhóm trường hợp chắc chắn không dẫn đến lời giải.

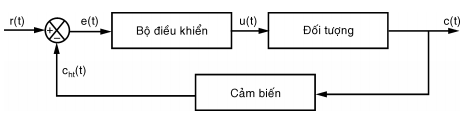
## CÁC KHÁI NIỆM LIÊN QUAN

Đây là một đề tài nghiên cứu thuộc lĩnh vực điều khiển tự động, vì thế trước hết ta cần tìm hiểu về điều khiển và các nguyên tắc điều khiển.

### Khái niệm điều khiển

Điều khiển là quá trình thu thập thông tin, xử lý thông tin và tác động lên hệ thống để đáp ứng của hệ thống “gần” với mục đích định trước. Điều khiển tự động là quá trình điều khiển không cần sự tác động của con người.

### Các thành phần cơ bản của hệ thống điều khiển



Hình 2‑19. Các thành phần cơ bản của hệ thống điều khiển

Chú thích các ký hiệu viết tắt:

* r(t): (reference input) tín hiệu vào, tín hiệu chuẩn
* c(t): (controlled output) tín hiệu ra
* cht(t): tín hiệu hồi tiếp
* e(t): (error) sai số
* u(t): tín hiệu điều khiển

Hình. Sơ đồ khối hệ thống điều khiển

### Các bài toán cơ bản trong lĩnh vực điều khiển tự động

Phân tích hệ thống: cho hệ thống tự động đã biết đã biết cấu trúc và thông số. bài toán đặt ra là trên cơ sở những thông tin đã biết tìm đáp ứng của hệ thống và đánh giá chất lượng hệ thống. Bài toán này luôn giải được.

Thiết kế hệ thống: biết cấu trúc và thông số của đối tượng điều khiển. Bài toán đặt ra là thiết kế bộ điều khiển để được hệ thống thỏa mãn các yêu cầu về chất lượng. Bài toán này nói chung là giải được.

Nhận dạng hệ thống: chưa biết cấu trúc và thông số của hệ thống. Vấn đề lớn nhất cần giải quyết là là tìm ra cấu trúc và các thông số này. Bài toán này không phải lúc nào cũng giải được.

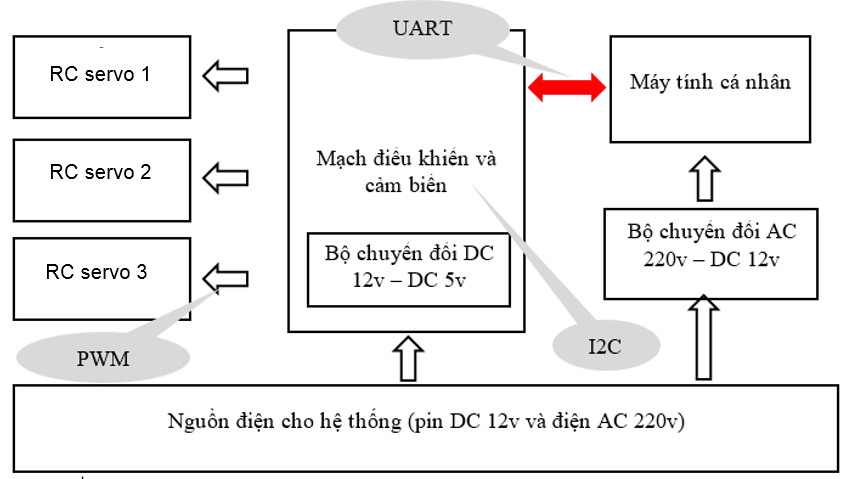
## GIẢ THUYẾT NGHIÊN CỨU

Đọc tín hiệu trả về từ bộ ADC của cảm biến. Lọc nhiễu tín hiệu bằng bộ lọc số. Kết hợp giá trị của con quay hồi chuyển và gia tốc kế để tính ra độ nghiêng lệch.

Bằng các giải thuật điều khiển tự động, hệ không gian tọa độ Oxyz, tính toán góc quay cần thiết cho các động cơ RC Servo để khắc phục độ nghiêng lệch, giúp cân bằng ổn định cho hệ thống máy ảnh.

# PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## MÔ HÌNH TỔNG QUÁT CỦA HỆ THỐNG



Hình 3‑1. Mô hình tổng quát của hệ thống

**Chi tiết của mô hình:**

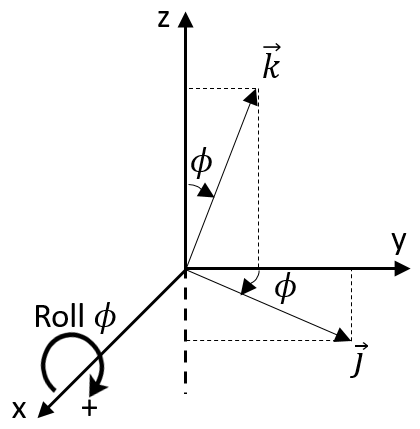
* Khối trung tâm là mạch điều khiển và cảm biến. Mạch điều khiển nhóm sử dụng vi điều khiển ATmega644p với cảm biến gia tốc và con quay hồi chuyển mpu6050, vi điều khiển và cảm biến sẽ kết nối với nhau thông qua chuẩn giao tiếp hai dây I2C, vi điều khiển đóng vai trò là master và cảm biến là slave. Mạch điều khiển trung tâm được mua ở ngoài thị trường nhằm giảm bớt công việc thiết kế mạch, đồng thời tăng tính ổn định của hệ thống lên.
* Khối trung tâm có nhiệm vụ đọc giá trị trả về của cảm biến, sau đó lọc đi dữ liệu lỗi, nhiễu và áp dụng giải thuật điều khiển PID để cân bằng cho hệ thống.
* Tín hiệu điều khiển được xuất là là tín hiệu PWM như đã trình bày ở trên, tín hiệu hiệu này sẽ điều khiển góc quay cho 3 động cơ RC servor.
* Khối trung tâm còn có nhiệm vụ giao tiếp với máy tính thông qua chuẩn giao tiếp UART, việc giao tiếp với máy tính sẽ dễ dàng cho việc kiểm lỗi và giám sát hệ thống.
* Ngoài ra trên khối trung tâm còn có mạch nguồn chuyển đổi điện áp 12v từ pin sang điện áp 5v với dòng điện cung cấp tối đa là 3A.
* Máy tính cá nhân sẽ giao tiếp với mạch trung tâm, nhận dữ liệu trả về và gửi dữ liệu điều khiển.
* Ba động cơ RC sẽ là cơ cấu chấp hành, sẽ hoạt động theo lệnh điều khiển của mạch trung tâm.
* Khối nguồn bao gồm pin lipo 12v, 2200mAh và điện gia đình 220v (cung cấp cho máy tính). Nhiệm vụ của khối này cực kỳ quan trọng, giúp tạo nguồn điện cung cấp cho toàn bộ hệ thống.

## MÔ HÌNH TOÁN HỌC CỦA HỆ THỐNG

### Ma trận xoay

Góc roll () được tạo ra khi ta xoay hệ trục tọa độ theo trục Ox (chiều dương như hình 3-2). Khi đó ta có ma trận xoay theo trục Ox:

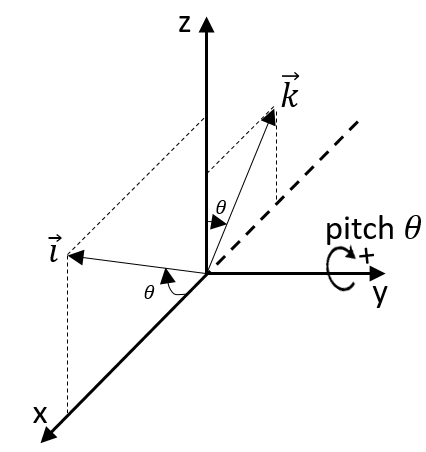




Hình 3‑2. Xoay theo góc roll

Góc pitch () được tạo ra khi ta xoay hệ trục tọa độ theo trục Oy (chiều dương như hình 3-3). Khi đó ta có ma trận xoay theo trục Oy:

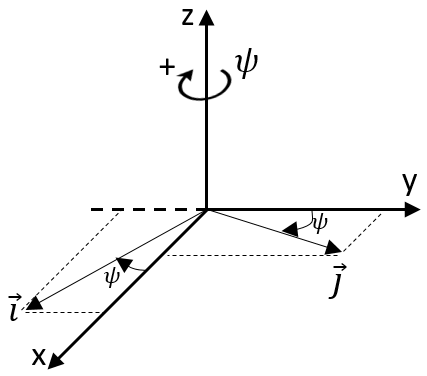




Hình 3‑3. Xoay theo góc pitch

Góc yaw () được tạo ra khi ta xoay hệ trục tọa độ theo trục Oz (chiều dương như hình 3-4). Khi đó ta có ma trận xoay theo trục Oz:





Hình 3‑4. Xoay theo góc yaw

Xét ma trận , ma trận này mô tả được sự thay đổi của tọa độ x và y trong khi tọa độ z không đổi, suy luận tương tự cho các ma trận ta thấy được các ma trận này tương tự như các ma trận xoay hai chiều ở từng mặt phẳng Oxy, Oxz và Oyz.

Muốn có được ma trận xoay ba chiều, ta thực hiện phép nhân các ma trận  lại với nhau:



Trong đó: 

### Động cơ và sức kéo

Động cơ được sử dụng là servo KST DS215MG: hoạt động ở mức điện áp từ 4.8v đến 7.4v, điện áp càng cao sẽ cho được tốc độ và lực kéo càng lớn.

Điển hình ở điện áp 6v, servo chỉ cần 60 ms để quay được góc 60 độ.

## NGUYÊN LÝ ĐIỀU KHIỂN GIMBAL

### Nguyên lý điều khiển

Điều khiển hệ thống cân bằng máy ảnh – gimbal là quá trình làm triệt tiêu các dao động ảnh hưởng đến máy ảnh, làm cho máy ảnh có chất lượng hình ảnh tốt nhất.

Phương pháp điều khiển cơ bản là đọc giá trị nghiêng lệch từ cảm biến, điểu khiển cơ cấu chấp hành để đưa máy ảnh trở lại vị trí cân bằng và loại bỏ tất cả các dao động khiển máy ảnh bị rung lắc.

Hệ thống điều khiển phải có đáp ứng nhanh, ổn định và có tính chất của một hệ thống thời gian thực.

### Các trạng thái cơ bản

Gimbal có rất nhiều trạng thái cấu hình tùy thuộc vào yêu cầu của người dùng. Qua quá trình nghiên cứu, nhóm nhận thấy có hai kiểu thường được sử dụng nhất:

* Cố định, bám mục tiêu: hệ thống gimbal sẽ giữ cho máy ảnh luôn nhìn về một hướng cố định. Được ứng dụng trong các máy bay fly-cam, để ghi hình các vật thể đặt biệt hoặc các biểu diễn của ca sĩ…
* Cố định một cách tương đối: hệ thống gimbal sẽ giúp máy ảnh loại bỏ các rung động, đồng thời cũng xoay theo hướng xoay của cả hệ thống với tốc độ thấp, để đáp ứng được quá trình lấy nét, điều chỉnh tiêu cự của máy ảnh. Mode hoạt động này thường được sử dụng khi người sử dụng muốn có các cảnh quay rộng.

## HIỆN THỰC HÓA SẢN PHẨM

### Tài nguyên sử dụng

Để hoàn thành nghiên cứu này, nhóm đồng thời sử dụng các nguồn tài nguyên có sẵng và miễn phí, đồng thời sử dụng tối ưu kinh phí hỗ trợ từ trường. Các tài nguyên bao gồm:

* Mạch vi điều khiển Atmega644p bao gồm cảm biến gia tốc và con quay hồi chuyển mpu6050.
* USB to UART: thiết bị tạo cổng COM ảo để giao tiếp với thiết bị thông qua RS232.
* Động cơ điều khiển, sử dụng ba động cơ digital RC servo, với góc điều khiển nhỏ.
* Mica để làm mô hình.
* Trình biên dịch Atmel Studio.
* Phần mền thiết kế Inventor.
* Quản lý qua Mercurial cung cấp bởi bitbucket (miễn phí với với dự án nhỏ hơn 5 thành viên).

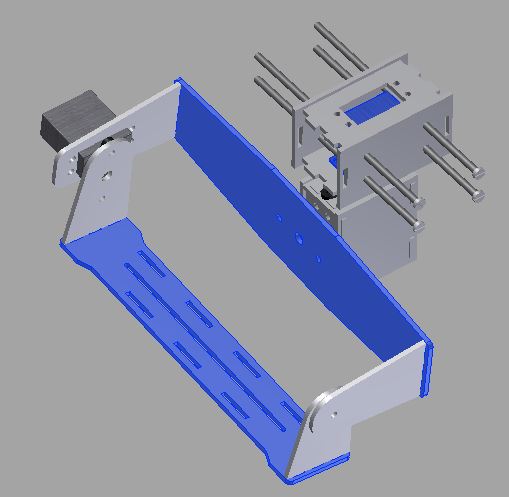
### Mô hình phần cứng

Mô hình phần cứng bao gồm mạch điều khiển và khung của sản phẩm.

Đối với mạch phần cứng, nhóm sử dụng mạch điều khiển Atmega644p có tích hợp cảm biến MPU6050.

Mạch được bố trí trên hệ thống sao cho cảm biến được ổn định nhất. Qua quá trình nghiên cứu và thực nghiệm, nhóm nhận thấy được cảm biến cần được đặt ở vị trí gần tay cầm của người dùng, điều này làm giảm đi rõ rệt sự rung lắc cảm biến khi quay servo.

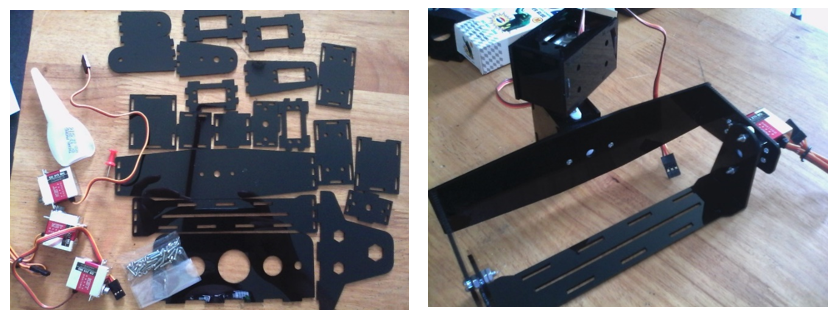
Khung của sản phẩm được thiết kế qua thông phần mền Inventor để sản phẩm có độ chính xác và ổn định trong quá trình hoạt động.



Hình 3‑5. Bản vẽ trên máy tính

Sau khi có bản thiết kế sẽ tiến hành cắt, vật liệu sử dụng là mica đen.

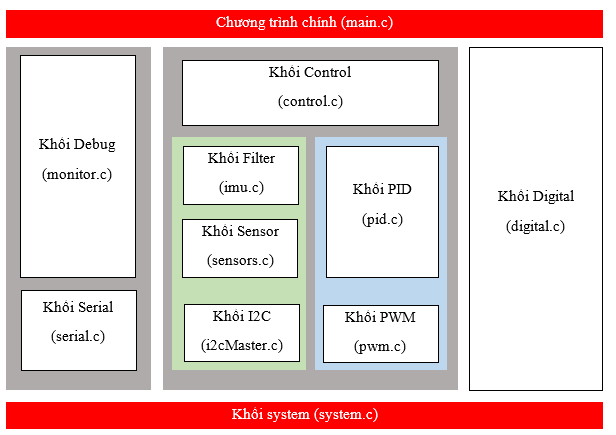
Đây là mô hình hoàn chỉnh của thiết kế, mô hình đáp ứng được yêu cầu về hoạt động cơ học, chịu được va chạm trong quá trình thử nghiệm.



Hình 3‑6. Các chi tiết sau khi cắt

### Mô hình phần mềm

Để thuận tiện cho quá trình phát triển mã nguồn cũng như kiểm lỗi và thu thập kết quả điều khiển, nhóm đã thiết kế mô hình phần mềm như sau:



Hình 3‑7. Mô hình giải thuật điều khiển

Chi tiết từng khối sẽ được trình bày ngay sau đây. Việc trình bày chỉ mô tả hoạt động và vai trò của từng khối trong mã nguồn, bao gồm liệt kê các hàm chính yếu trong khối (mô tả chi tiết tính năng của hàm sẽ được trình bày ở phần phụ lục).

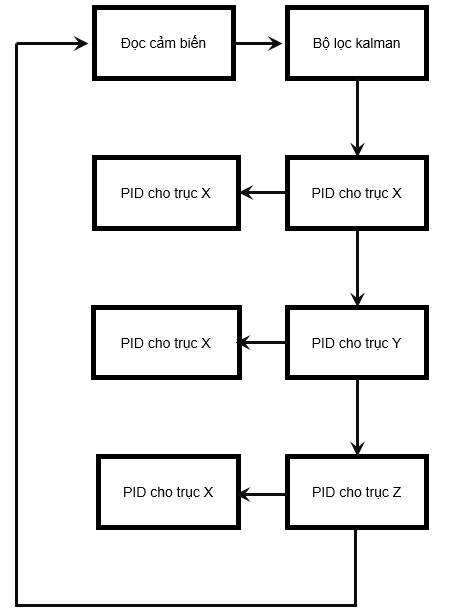
* Lớp thấp nhất của hệ thống nằm ở khối system, khối system được hiện thực qua file mã nguồn system.c. Khối system sẽ khai báo các thanh ghi dùng để khởi tạo hệ thống, bao gồm khai báo các port sử dụng trên vi điều khiển, khai báo timer (hàm setup()) và các hàm lấy thời gian của hệ thống ( gồm hàm millis(), micros(), ticks()) khởi tạo các mode cho giao tiếp uart, khởi tạo pwm và khởi tạo i2c để giao tiếp với cảm biến.

Khối system còn cung cấp các macro và define dùng trong cho hệ thống (thông qua include file system.h), ví dụ như MIN, MAX, ABS, FILTER, F\_CPU, DEBUG, ROLL, PITH, YALL...

* Phần cảm biến bao gồm các khối I2C, Sensor và Filter, thứ tự từ thấp đến cao như trong hình vẽ (số mấy), nhiệm vụ của phần cảm biến là trả về góc lệch của đối tượng, góc lệch này cần phải chính xác, ổn định, không bị nhiễu bởi các rung lắc. Để hiện thực nên phần này, nhóm chia thành ba khối nhỏ để phân chia công việc rõ ràng, cụ thể:
* Khối I2C (file i2cMaster.c) sẽ hiện thực các hàm I2C để giao tiếp với cảm biến (ví dụ như hàm i2c\_writeReg(), I2c\_getSixRawADC(), i2c\_readReg16()...) các hàm này đọc dữ liệu sẽ được trả về từ cảm biến.
* Việc đọc các hàm này được thực hiện qua khối Sensor (sensors.c), dữ liệu đọc được là dữ liệu thô, chưa qua xử lý và cần được lọc bởi khối Filter.
* Khối Filter hiện thực qua file imu.c, ở đây sẽ dùng hàm imuCalculate() để lọc đi dữ liệu nhiễu, việc lọc dữ liệu này đã được trình bày ở phần 2.1.2
* Khối digial là khối phụ, dùng để điều khiển đèn led thông báo, việc thông báo bằng đền led sẽ có lợi cho việc giám sát khi cho chạy hệ thống thực tế, nó là tín hiệu báo lỗi cực kỳ hữu ích.
* Một phần quan trọng của hệ thống nữa là phần Debug, phần này có nhiệm vụ rất nhiều trong quá trình phát triển mã cho hệ thống, để module Debug có thể hoạt động cần thiết lập tùy chọn ở compiler setting là –D DEBUG. Trong phần Debug có các khối nhỏ bao gồm: Khối Serial và khối Debug.
* Khối Serial hiện thực ở file serial.c, trong file này nhóm hiện thực nên các hàm cơ bản như serialWriteChar() serialReadChar(). Các hàm cơ bản này sẽ ghi hay đọc dữ liệu vào queue, chuẩn giao tiếp uart cho tốc độ truyền nhận không cao vì vậy cần một queue để dữ liệu được buffering trước khi truyền, tránh trường hợp hệ thống bị lỗi khi truyền dữ liệu quá nhiều.
* Khối serial còn override hàm printf() trong C để việc xuất dữ liệu thêm dễ dàng.
* Ở file monitor.c là khối Debug, khối này sẽ nhận dữ liệu điều khiển từ máy tính và sau đó thực hiện các hàm đã được định nghĩa trước. Ví dụ một số hàm cơ bản như ver, help, wr… Lệnh help sẽ xuất ra một số thông tin hướng dẫn.
* Thuộc tính của các lệnh điều khiển:
  + - * Độ dài lệnh bao gồm cả đối số: CMD\_SIZE 32
      * Đối số là kiễu dữ liệu float
      * Số lượng tối đa của đối số: MAX\_PARMS 8
      * Độ dài một lệnh tối đa: PARMS\_SIZE 8
* File main.c là file chương trình chính chứa hàm main của hệ thống, file này và bao gồm file main.h sẽ chữa các biến extern để sử dụng cho toàn bộ các file khác trong chương trình.

### Điều khiển 3 trục

Tổng quan việc điều khiển 3 trục quay cho gimbal được thực hiện theo các bước như sơ đồ sau:

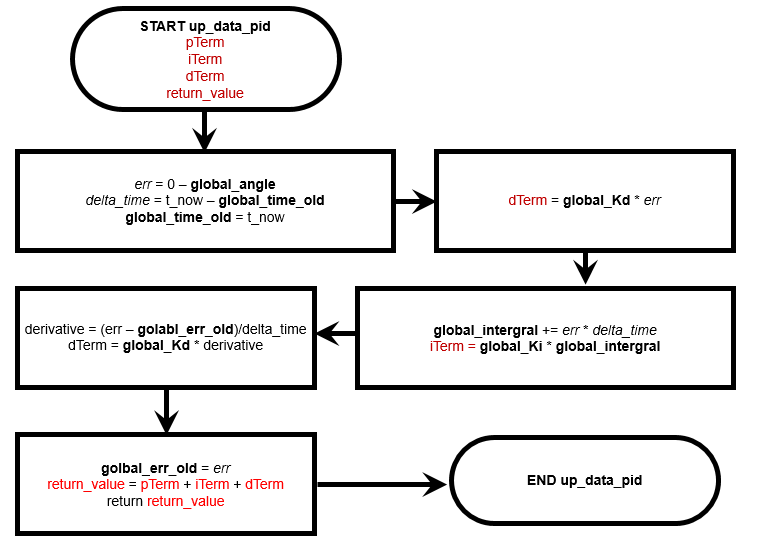


Chi tiết của từng thành phần trong sơ đồ và giải thuật như sau:

#### Đọc cảm biến và bộ lọc

Việc đọc cảm biến sẽ cho ta được kết quả góc nghiêng của gimbal, cảm biến mpu650 trả về giá trị số của ADC đọc được từ 3 trục

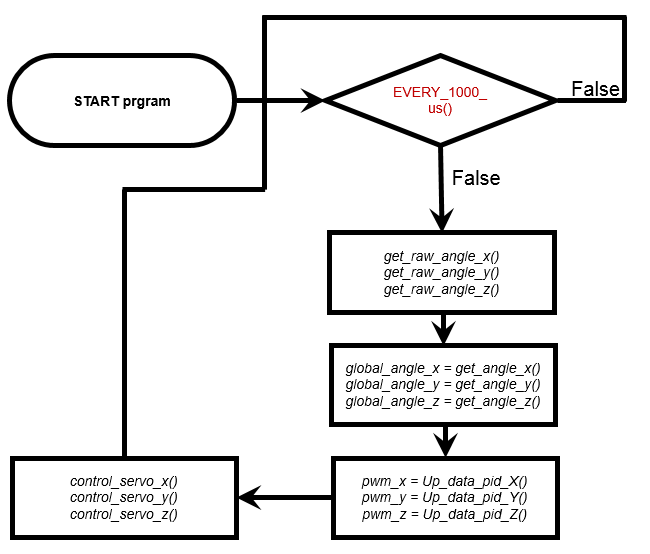
#### Sơ đồ giải thuật PID



Các biến cục bộ global\_err\_old lưu lại giá trị góc quay lỗi cho lần tính PID tiếp theo, global\_time\_old lưu giá trị của thời gian mỗi lần gọi hàm up\_data\_pid. Các biến global\_Kd, global\_Ki và global\_Kp là các biến hệ số của bộ lọc, dùng để cân chỉnh cho hệ thống. Biến global\_intergral dùng để lưu thành phần intergral dùng lại cho mỗi lần gọi hàm up\_data\_pid.

Giá trị trả về của hàm up\_data\_pid được dùng trực tiếp vào việc điều khiển cho servo, có 3 bộ PID tương ứng với 3 trục đáp ứng cho 3 servo. Sơ đồ giải thuật PID của 3 trục là giống nhau.

#### Sơ đồ giải thuật cho toàn bộ hệ thống

Như sơ đồ trên hoạt động của hệ thống sẽ là cập nhật giá trị của cảm biến, cho giá trị cả cảm biến qua bộ lọc và có được góc lệch so với vị trí cân bằng, từ góc lệch này bộ điều khiển PID của mỗi trực sẽ tính ra được giá trị cần điều khiển cho servo.

# KẾT QUẢ THỬ NGHIỆM VÀ ĐÁNH GIÁ

## QUY TRÌNH THỬ NGHIỆM

### Các thành phần của hệ thống thử nghiệm

* Bộ khung giá đỡ.
* Ba servo digital KST DS521MG.
* Board mạch với chip Atmega644 kết nối cảm biến MPU6050.
* USB to USART: sử dụng để Debug.
* Mạch nạp chip: USB ASP[[[3]](#footnote-3)].

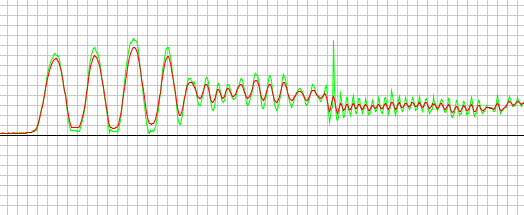
### Tiến hành thử nghiệm

Kết nối servo vào board điều khiển, nạp chương trình vào chip.

Cung cấp nguồn điện cho servo và tiến hành kiểm tra hoạt động.

## CÁC KẾT QUẢ THỬ NGHIỆM

### Kết quả sử dụng bộ lọc



Hình 4‑1. Kết quả sử dụng bộ lọc Kalman

Chú thích: Đường màu xanh lá cây là tín hiệu chưa qua bộ lọc, đường màu đỏ là tín hiệu sau khi đã đi qua bộ lọc.

Dựa vào biểu đồ cho thấy được hiệu quả của bộ lọc Kalman, đã lọc được tương đối các nhiễu của hệ thống, biến những sự rung lắc mạnh trở nên mượt hơn.

### Tính ổn định của hệ thống

Qua các kết quả thử nghiệm, hệ thống đã đạt được một độ ổn định tương đối. có thể ứng dụng vào việc cân bằng máy ảnh để chụp ảnh.

# KẾT LUẬN VÀ KIẾN NGHỊ

## KẾT LUẬN

### Các kết quả đạt được

Qua thời gian nghiên cứu và thực hiện đề tài nghiên cứu này, chúng tôi đã đạt được những kết quả như sau:

* Chế tạo được một hệ thống cân bằng máy ảnh hoạt động tương đối ổn định.
* Tìm hiểu được vi điều khiển mới trong họ vi điều khiển AVR là atmega644p.
* Tìm hiểu được cảm biến nghiêng mpu6050, cách hoạt động của cảm biến nghiêng và con quay hồi chuyển cũng như cách giao tiếp với vi điều khiển qua chuẩn giao tiếp I2C.
* Giao tiếp với máy tính thông qua UART, là hình thức giao tiếp phổ biến được dùng rất nhiều trong các hệ thống nhúng.
* Tìm hiểu và hiện thực bộ lọc Kalman trên vi điều khiển.
* Tìm hiểu và hiện thực bộ điều khiển PID.
* Ứng dụng bộ điều khiển PID và giá trị góc nghiêng sau khi qua bộ lọc Kalman để điều khiển ba trục của gimbal. Kết quả cuối cùng, hệ thống hoạt động, camera được giữ cân bằng khi bị tác dụng bởi lực bên ngoài làm cho hệ thống bị nghiêng.
* Ngoài ra hai thành viên đã chia sẽ được công việc với nhau, sắp xếp thời gian để hoàn thành công việc của mình.

### Chưa đạt được

Tuy đã đạt được rất nhiều kết quả về mặt kiến thức, tuy nhiên kỹ năng làm việc chưa thật sự tốt, không tập trung vào các vấn đề chính, quan trọng.

Trong 3 trục của gimbal có trục Oz (yaw) bị “trôi” theo thời gian. Giá trị góc lệch Yaw quay quanh trục Oz có hiện tượng tăng hoặc giảm đi một giá trị tương đối nhỏ (khoảng 0.5 độ/giây). Điều này xuất phát từ nguyên nhân cảm biến MPU6050 có nhược điểm bị “drift” theo thời gian. Vấn đề này có thể được khắc phục bằng cách sử dụng một cảm biến khác với giá thành cao hơn.

## KIẾN NGHỊ

Đề tài nghiên cứu – chế tạo hệ thống cân bằng máy ảnh là một đề tài mang tính mới mẽ, dễ dàng tạo được cảm hứng cho các sinh viên khoa Kỹ thuật máy tính nói riêng và sinh viên thuộc nhóm ngành điều khiển tự động nói chung.

Trong tương lai có thể phát triển hệ thống ổn định hơn, bằng cách:

* Sử dụng động cơ brushless (một loại động cơ không chổi quét, điều khiển chính xác qua các bước quay khác nhau) để cho hiệu quả ổn định tốt hơn. Thực tế các hệ thống gimbal thương mại phần lớn sử dụng loại động cơ này hơn là sử dụng động cơ RC Servo vì RC Servo được dùng nhiều trong nghiên cứu.
* Thiết kế hệ thống khung chắc chắn hơn để có thể ứng dụng ngay vào trong thực tế.
* Tinh chỉnh bộ lọc cho cảm biến và thay cảm biến khác để tránh hiện tượng bị “trôi” như đã trình bày ở trên.
* Ứng dụng hệ thống này cũng như những kiến thức học được vào khóa luận tốt nghiệp.

# DANH MỤC TÀI LIỆU THAM KHẢO

[1] “Ghost Gimbals | Spaulding International Cinema.” [Online]. Available: http://ghostgimbals.com/collections/ghost-gimbals. [Accessed: 07-Nov-2015].

[2] “AVR C Runtime Library - Summary [Savannah].” [Online]. Available: http://savannah.nongnu.org/projects/avr-libc/. [Accessed: 07-Nov-2015].

[3] Atmel Corporation, “Tips and Tricks to Optimize Your C Code for 8-bit AVR Microcontrollers,” p. 18, 2011.

[4] “FlashRamSize - \*\* Code Red Support Site \*\*.” [Online]. Available: http://www.support.code-red-tech.com/CodeRedWiki/FlashRamSize. [Accessed: 07-Nov-2015].

[5] ATmel, “ATmega 6444/V,” pp. 1–376, 2012.

[6] *EIA standard RS-232-C: Interface between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange*. Washington: Electronic Industries Association. Engineering Dept, 1969.

[7] “I2C - Wikipedia, the free encyclopedia.” [Online]. Available: https://en.wikipedia.org/wiki/I%C2%B2C. [Accessed: 21-Nov-2015].

[8] “I2C - learn.sparkfun.com.” [Online]. Available: https://learn.sparkfun.com/tutorials/i2c. [Accessed: 21-Nov-2015].

1. [] <http://ghostgimbals.com/> [↑](#footnote-ref-1)
2. [] Được thiết kế theo sơ đồ nguyên lý của câu lạc bộ CEEC [↑](#footnote-ref-2)
3. [] www.fischl.de/**usbasp**/ [↑](#footnote-ref-3)