

LỜI CẢM ƠN

Chúng em rất hân hạnh được cảm ơn tất cả thầy cô, bạn bè đã nhiệt tình ủng hộ, giúp đỡ trong quá trình làm luận án. Để có thể hoàn thành tốt đề tài này thì cần yêu cầu một sự nỗ lực lớn từ chúng em để đạt được kết quả như vậy trong một thời gian ngắn. Và vai trò của giáo viên hướng dẫn là rất lớn, tiến sĩ Lê Mỹ Hà đã rất nhiệt tình cùng chúng em nghiên cứu để có thể giúp chúng em hoàn thành tốt đề tài này. Trong quá trình làm đề tài nhóm cũng phát sinh nhiều khó khăn, nhiều lúc gặp vấn đề tưởng chừng như không thể giải quyết, chúng em cũng đôi lúc nản chí. Và với kinh nghiệm của một người thầy, thầy luôn theo sát, gợi ý đưa ra hướng xử lý. Thầy không chỉ giúp chúng em trong việc giải quyết vấn đề mà còn động viên và khích lệ tinh thần giúp chúng em có thể hoàn thành tốt nhiệm vụ đề tài. Chúng em xin chân thành cảm ơn thầy tiến sĩ Lê Mỹ Hà.

Để có thể hoàn thiện đề tài thì sự tích lũy kiến thức trong 4 năm học đại học là yếu tố quan trọng vì vậy chúng em xin chân thành cảm ơn các thầy cô trong khoa đã cung cấp các kiến thức hữu ích không chỉ cho việc thực hiện đề tài như điều khiển ổn định hệ thống, xử lý ảnh trong công nghiệp... mà còn là kiến thức về sau này trong công việc sau khi ra trường. Bên cạnh đó chúng em cũng xin cảm ơn các bạn trong nhóm ISLAB – Intelligent Systems Laboratory đã nhiệt tình hỗ trợ, giúp đỡ vật chất và tinh thần, các kiến thức để giúp chúng em vượt qua những lúc khó khăn.

TÓM TẮT

Luận văn “Robot hai bánh tự cân bằng bám theo đối tượng sử dụng kỹ thuật xử lý ảnh” được phát triển từ robot hai bánh tự cân bằng dựa trên lý thuyết mô hình con lắc ngược kết hợp với quá trình xử lý ảnh trên máy tính mini Raspberry. Ý tưởng của sự kết hợp này dựa trên sự nhận thấy thế mạnh của Raspberry Pi và Arduino là một sự bổ sung cho nhau, Raspberry Pi vượt trội về các ứng dụng phần mềm trong khi Arduino xử lý mạnh mẽ cho các dự án thuần về phần cứng.

Robot hai bánh tự cân bằng có khả năng giữ cân bằng ổn định, di chuyển linh hoạt, theo dõi và bám mục tiêu cụ thể là quả bóng tennis và duy trì khoảng cách mong muốn. Thông qua đề tài, nhóm hiểu rõ cách điều khiển hệ thống ổn định bằng bộ điều khiển PID, nguyên lý và cách sử dụng cảm biến góc nghiêng, động cơ step và lập trình xử lý ảnh bằng ngôn ngữ lập trình Python.

Luận văn này sẽ trình bày bao gồm các lý thuyết, thuật toán được áp dụng và cả phần cứng: Arduino Mega 2560, cảm biến góc nghiêng MPU6050, module bluetooth HC05, động cơ bước, raspberry pi 3, picamera, ...

ABSTRACT

The thesis “Two-wheel balancing robot tracking the object using image processing” developed from our team from “Two-wheel balancing robot” based on “Inverted Pendubot” combined with image processing on mini computer raspberry. The idea of this combination based on the strongest face of Arduino and Raspberry, Raspberry Pi excels in software applications while Arduino handles powerful hardware projects.

A two-wheel balancing robot has the ability to maintain a stable balance, move in a flexible manner, track and target a specific tennis ball and maintain the desired distance. Throughout the subject, the team understands how to control the stability of the system using a PID controller, the principles and uses of tilt sensors, step motors and image processing software in the Python programming language.

Arduino Mega 2560, MPU6050 tilt sensor, HC05 bluetooth module, stepper motor, raspberry pi 3, picamera, etc. will be included in this thesis.

MỤC LỤC

LỜI CẢM ƠN	i
TÓM TẮT	ii
ABSTRACT	iii
MỤC LỤC.....	iv
DANH MỤC CÁC TỪ VIẾT TẮT.....	vii
DANH MỤC CÁC BẢNG BIỂU	viii
DANH MỤC CÁC BIỂU ĐỒ VÀ HÌNH ẢNH.....	ix
TÀI LIỆU THAM KHẢO	xii
CHƯƠNG 1: TỔNG QUAN VỀ ĐỒ ÁN ROBOT CÂN BẰNG	1
1.1 ĐẶT VẤN ĐỀ.....	1
1.2 NHIỆM VỤ ĐỒ ÁN	2
1.2.1 Lý do hình thành đề tài	2
1.2.2 Ý nghĩa khoa học thực tiễn	3
1.2.3 Mục tiêu nghiên cứu.....	3
1.2.4 Đối tượng	4
1.2.5 Giới hạn đề tài	4
1.3 CẤU TRÚC ĐỒ ÁN	4
CHƯƠNG 2: CÁC LÝ THUYẾT ÁP DỤNG TRONG ROBOT HAI BÁNH TỰ CÂN BẰNG BẮM THEO ĐỐI TƯỢNG	6
2.1 NGUYÊN LÝ HOẠT ĐỘNG CỦA ROBOT 2 BÁNH TỰ CÂN BẰNG	6
2.2 BỘ ĐIỀU KHIỂN PID	8
2.2.1 Lý thuyết về bộ điều khiển PID	9
2.2.1.1 Khâu tỉ lệ.....	10
2.2.1.2 Khâu tích phân	11
2.2.1.3 Khâu vi phân	12
2.2.2 Áp dụng bộ điều khiển PID trong xe cân bằng.....	14
2.3 BỘ LỌC COMPLEMENTARY	15
2.4 XỬ LÝ ẢNH.....	19

2.4.1 Thư viện OpenCV	20
2.4.2 Không gian màu RGB và HSV	21
2.4.2.1 Không gian màu RGB.....	21
2.4.2.2 Không gian màu HSV	22
2.4.2.3 Phép toán chuyển đổi không gian màu RGB sang không gian màu HSV [12]	23
2.4.3 Phương pháp lấy ngưỡng trong không gian màu HSV	25
2.4.4 Các thuật toán xử lý hình thái học (Morphology) [14]	26
2.4.4.1 Phép co ảnh (Erosion).....	26
2.4.4.2 Phép giãn ảnh (Dilation)	28
2.4.4.3 Bộ lọc Gaussian [15].....	29
CHƯƠNG 3: LỰA CHỌN CÁC THIẾT BỊ PHẦN CỨNG CHO ROBOT CÂN BẰNG BĂM THEO ĐỐI TƯỢNG	33
3.1 SƠ ĐỒ CHUNG KẾT NỐI PHẦN CỨNG	33
3.2 LỰA CHỌN CÁC THIẾT BỊ PHẦN CỨNG.....	34
3.2.1 Board xử lý ảnh	34
3.2.2 Vi điều khiển các thiết bị phần cứng.....	36
3.2.3 Camera Raspberry Pi V2 8MP.....	38
3.2.4 Cảm biến góc nghiêng MPU6050	41
3.2.5 Động cơ bước	47
3.2.6 Mạch điều khiển động cơ bước A4899	51
3.2.7 Module Bluetooth HC05	56
3.2.8 Mạch chuyển đổi tín hiệu USB to TTL CP2102.....	57
3.2.9 Động cơ RC Servo	58
3.2.10 Nguồn cấp cho robot	60
CHƯƠNG 4 XÂY DỰNG VÀ ÁP DỤNG THUẬT TOÁN XỬ LÝ ẢNH VÀO ROBOT	63
4.1 KHÁI QUÁT VỀ NGUYÊN LÝ HOẠT ĐỘNG CỦA ROBOT	63
4.2 CÁC BƯỚC XỬ LÝ ẢNH TRONG RASPBERRY	64

4.3 THUẬT TOÁN XỬ LÝ ẢNH XÁC ĐỊNH ĐỐI TƯỢNG	65
4.4 TÍNH KHOẢNG CÁCH GIỮA ROBOT VÀ ĐỐI TƯỢNG.....	68
CHƯƠNG 5: KẾT QUẢ THỰC NGHIỆM.....	69
5.1 MÔ HÌNH PHẦN CỨNG	69
5.2 XỬ LÝ TÍN HIỆU TỪ CẢM BIẾN MPU6050	70
5.3 KẾT QUẢ CỦA BỘ ĐIỀU KHIỂN PID ĐƯỢC ÁP DỤNG TRONG ROBOT HAI BÁNH TỰ CÂN BẰNG	71
5.4 ỨNG DỤNG XỬ LÝ ẢNH VÀO ROBOT HAI BÁNH TỰ CÂN BẰNG BẮM THEO ĐỐI TƯỢNG.....	74
5.5 PHÉP TÍNH KHOẢNG CÁCH TỪ ROBOT ĐẾN ĐỐI TƯỢNG	76
5.5 THỰC NGHIỆM VỀ KHẢ NĂNG BẮM THEO VẬT CỦA ROBOT	77
CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	80
6.1 KẾT LUẬN	80
6.2 HƯỚNG PHÁT TRIỂN	80

DANH MỤC CÁC TỪ VIẾT TẮT

ACC	:	A ccelerometer
DC	:	D irect C urrent
GY	:	G yroscope
HSV	:	H ue S aturation V alue
I2C	:	I nter- I ntegrated C ircuit
IC	:	I ntegrated C ircuit
IDE	:	I ntegrated D evelopment E nvironment
IMU	:	I nertial M easurement U nit
IOT	:	I nternet O f T hings
MEMS	:	M icro E lectro M echanical S ystems
MP	:	M ega P ixel
MPU	:	M icro P rocessor U nit
OpenCV	:	O pen C omputer V ision
PID	:	P roportional I ntegral D erivative
PWM	:	P ulse W idth M odulation
RGB	:	R ed G reen B lue
SCL	:	S erial C lock
SDA	:	S erial D ata

DANH MỤC CÁC BẢNG BIỂU

Bảng 2.1: Tác động của việc tăng một thông số độc lập	13
Bảng 3.1: Thông số kỹ thuật chính của Raspberry Pi 3 Model B:.....	35
Bảng 3.2: Các thông số kỹ thuật chính	36
Bảng 3.3: Thông số kỹ thuật	39
Bảng 3.4: Thông số kỹ thuật cảm biến MPU6050	42
Bảng 3.5: Thông số cơ bản của động cơ bước	49
Bảng 3.6: Tiêu chuẩn về Bước Góc của động cơ bước:	50
Bảng 3.7: Thiết lập chế độ chia bước:	55
Bảng 3.8: Thông số kỹ thuật module bluetooth HC05	56
Bảng 3.9: Thông số kỹ thuật chính:	59
Bảng 3.10: Thông số kỹ thuật chính	62
Bảng 5.1: Thời gian quay về vị trí cân bằng với từng góc lệch khác nhau.....	73

DANH MỤC CÁC BIỂU ĐỒ VÀ HÌNH ẢNH

Hình 2.1 Nguyên lý di chuyển của robot hai bánh tự cân bằng	6
Hình 2.2: Robot tự cân bằng di chuyển.....	7
Hình 2.3: Robot cân bằng trên mặt phẳng nghiêng.....	8
Hình 2.4: Sơ đồ khối bộ điều khiển PID	9
Hình 2.5: Thể hiện tác động của khâu tỉ lệ trong hệ thống	10
Hình 2.6: Ảnh hưởng của khâu tích phân trong hệ thống	12
Hình 2.7: Ảnh hưởng của khâu vi phân trong hệ thống	13
Hình 2.8: Bộ điều khiển PID dùng trong robot cân bằng	14
Hình 2.9: Sử dụng bộ lọc Complementary để tính góc từ cảm biến MPU6050	16
Hình 2.10: Đo góc bằng cảm biến Accelerometer	16
Hình 2.11: Góc nghiêng được đo từ cảm biến Gyroscope.....	17
Hình 2.12: Ảnh thể hiện giá khả năng đo góc của 2 cảm biến trong miền tần số.....	18
Hình 2.13: Bộ lọc Complementary kết hợp 2 giá trị accelerometer và gyroscope ...	19
Hình 2.14: Không gian màu RGB	21
Hình 2.15: Biểu diễn không gian màu RGB trên hệ trục tọa độ xyz	21
Hình 2.16: Không gian màu HSV	22
Hình 2.17: Chuyển không gian màu RGB sang không gian màu HSV	24
Hình 2.18: Ảnh biểu diễn các giá trị HSV	25
Hình 2.19: Lấy ngưỡng trong không gian màu HSV	26
Hình 2.20: Sử dụng phép co ảnh	27
Hình 2.21: Phần tử cấu trúc (Structure element)	27
Hình 2.22: Sử dụng phép giãn ảnh	28
Hình 2.23: Cấu trúc phần tử	28
Hình 2.24: Gaussian distribution with mean 0 and $\sigma = 1$	29
Hình 2.25: Gaussian distribution with mean (0,0) and $\sigma=1$	30
Hình 2.26: Tính xấp xỉ rời rạc với hàm Gaussian với $\sigma = 1.0$	31
Hình 2.27: Ảnh sau khi sử dụng bộ lọc Gaussian	32

Hình 3.1: Sơ đồ kết nối phần cứng của robot.....	33
Hình 3.2: Board Raspberry Pi 3 Model B	34
Hình 3.3: Board Arduino Mega 2560	36
Hình 3.4: Camera Raspberry Pi V2 8MP	38
Hình 3.5: Kết nối picamera với raspberry pi.....	39
Hình 3.6: Cấu hình raspberry	40
Hình 3.7: Enabled Camera	40
Hình 3.8: Cảm biến góc nghiêng MPU6050	41
Hình 3.9: Sơ đồ chân ra của cảm biến MPU6050.....	42
Hình 3.10: Kết nối cảm biến MPU6050 với Arduino Mega2560.....	43
Hình 3.11: Động cơ bước	47
Hình 3.12: Sơ đồ nguyên lý hoạt động của động cơ bước	49
Hình 3.13: Cấu tạo của động cơ bước.....	51
Hình 3.14: Mạch điều khiển động cơ bước A4988.....	51
Hình 3.15: Sơ đồ kết nối mạch điều khiển động cơ bước	52
Hình 3.16: Các chân điều khiển động cơ bước A4988	53
Hình 3.17: Sơ đồ kết nối động cơ bước với driver A4988.....	54
Hình 3.18: Module Bluetooth HC05	56
Hình 3.19: Module USB to TTL CP2102	57
Hình 3.20: Động cơ RC Servo EMAX ES08.....	58
Hình 3.21: Các dây kết nối của servo	59
Hình 3.22: Khung servo hai trục	60
Hình 3.23: Pin Lipo 3S 2200mAh.....	60
Hình 3.24: Mạch ổn áp DC-DC mini	61
Hình 4.1: Sơ đồ hoạt động của robot	63
Hình 4.2: Các bước cơ bản xử lý hình ảnh trong raspberry pi.....	64
Hình 4.3: Sơ đồ khối biểu diễn quá trình xử lý ảnh.....	65
Hình 4.4: Sơ đồ khối biểu diễn thuật toán xác định đối tượng có trong hình bằng xử lý ảnh	66

Hình 4.5: Tính khoảng cách của robot và đối tượng.....	68
Hình 5.1: Mô hình Robot hai bánh tự cân bằng	69
Hình 5.2: Bộ lọc Complementary trong tính góc nghiêng	70
Hình 5.3: Sơ đồ bộ điều khiển PID được ứng dụng trong robot.....	71
Hình 5.4: Tác động lực lên robot	72
Hình 5.5 Góc điều khiển tốc độ	74
Hình 5.6: Ảnh sau trước và sau khi xử lý	75
Hình 5.7: Chuyển đổi không gian màu và xử lý lấy ngưỡng	75
Hình 5.8: Khoảng cách từ robot đến đối tượng.....	76
Hình 5.9: Tính khoảng cách từ robot đến đối tượng	77
Hình 5.10: Robot xử lý bám đối tượng vòng số 8.....	77
Hình 5.11: Robot bám theo đối tượng theo vòng số 8	78

TÀI LIỆU THAM KHẢO

- [1] K. Čapek, Rossum's Universal Robots, Czech: Science fiction, 1922.
- [2] K. C. G. a. L. Y. Ang, PID control system analysis, design, and technology, IEEE, 2005 .
- [3] "www.pocket-lint.com," Maggie Tillman and Adrian Willings, 29 may 2018. [Online]. Available: <https://www.pocket-lint.com/gadgets/news/134820-real-life-robots-that-will-make-you-think-the-future-is-now>.
- [4] T. Deyle, "www.hizook.com," US Military, 16 June 2010. [Online]. Available: <http://www.hizook.com/blog/2010/06/16/tbot-reconfigurable-140kg-balancing-robot-us-military>.
- [5] P. P. HANNA DOTEVALL, Balancing of an inverted pendulum through adjusting angle and thrust of propellers, Sweden: KTH, 2016.
- [6] wikipedia, "en.wikipedia.org," wikipedia, January 2010. [Online]. Available: https://en.wikipedia.org/wiki/Inverted_pendulum.
- [7] wikipedia, "www.vi.wikipedia.org," wikipedia, 09 March 2018. [Online]. Available: https://vi.wikipedia.org/wiki/B%E1%BB%99_%C4%91i%E1%BB%81u_khi%E1%BB%83n_PID.
- [8] A. Rosenfeld, Picture Processing by Computer, New York: Academic Press, 1969.
- [9] K. Pulli, A. Baksheev, K. Korniyakov and V. Eruhimov, Realtime Computer Vision with OpenCV, 2012.
- [10] "Commission internationale de l'éclairage," Commission internationale de l'éclairage, [Online]. Available: <http://www.cie.co.at/>.
- [11] H. Levkowitz and G. T. Herman, GLHS: A Generalized Lightness, Hue and Saturation Color Model, 1993.
- [12] M. K. Agoston, Computer Graphics and Geometric Modeling: Implementation and Algorithms, London: Agoston, 2005.

- [13] R. C. & W. R. E. Gonzalez, Thresholding. In Digital Image Processing, Pearson Education, 2002.
- [14] P. Soille, Morphological Image Analysis; Principles and Applications, ISBN 3-540-65671-5, 2003.
- [15] Fisher, Perkins, Walker & Wolfart, Spatial Filters - Laplacian of Gaussiam, Retrieved 2010-09-13., 2003.
- [16] T. M. Magazine, Raspberry Pi 3 is out now! Specs, Benchmarks & More, The MagPi Magazine, 2016.
- [17] microchip, ATmega640/1280/1281/2560/2561 - Complete Datasheet, microchip, 2014.

CHƯƠNG 1: TỔNG QUAN VỀ ĐỒ ÁN ROBOT CÂN BẰNG

1.1 ĐẶT VẤN ĐỀ

Thuật ngữ “Robot” lần đầu tiên xuất hiện năm 1922 trong tác phẩm “Rossum’s Universal Robot” của Karel Capek. Theo tiếng Séc thì Robot là người làm tạp dịch. Trong tác phẩm này nhân vật Rossum và con trai ông đã tạo ra những chiếc máy gần giống như con người để hầu hạ con người. [1]

Hơn 20 năm sau, ước mơ viễn tưởng của Karel Capek đã bắt đầu hiện thực. Ngay sau chiến tranh thế giới lần thứ 2, ở Mỹ đã xuất hiện những tay máy chép hình điều khiển từ xa, trong các phòng thí nghiệm phóng xạ. Năm 1959, Devol và Engelber đã chế tạo Robot công nghiệp đầu tiên tại công ty Unimation.

Ngày nay, robotic đã đạt được những thành tựu to lớn trong sản xuất công nghiệp cũng như trong đời sống. Sản xuất robot là ngành công nghiệp trị giá hàng tỉ USD và ngày càng phát triển mạnh, trong các họ robot chúng ta không thể không nhắc tới Mobile Robot với những đặc thù riêng mà các loại robot khác không có.

Mobile robot có thể di chuyển một cách rất linh hoạt, do đó tạo nên không gian hoạt động lớn và cho đến nay nó đã dần khẳng định vai trò quan trọng không thể thiếu trong nhiều lĩnh vực, thu hút được rất nhiều sự đầu tư và nghiên cứu. Mobile robot cũng được chia ra làm nhiều loại: robot dò đường line, robot tránh vật cản, robot tìm đường cho mê cung,...trong số đó robot hai bánh tự cân bằng là một trong những loại robot dễ dàng ứng dụng nhiều trong cuộc sống.

Robot cân bằng là một thành phần có vai trò quan trọng trong ngành Robot học. Cùng với sự phát triển mạnh mẽ của các hệ thống tự động hóa, robot cân bằng ngày một được hoàn thiện và càng cho thấy lợi ích của nó trong công nghiệp và sinh hoạt. Một vấn đề rất được quan tâm khi nghiên cứu về robot cân bằng là làm thế nào để robot có thể tự cân bằng và tự động đi theo một quỹ đạo đã định.

Để điều khiển điều khiển robot theo quỹ đạo, người thiết kế lập trình xác định độ lệch tương đối giữa quỹ đạo của robot và quỹ đạo mong muốn, sau đó so

sánh độ lệch đó thành các mức và điều khiển lái robot quay về quỹ đạo. Phương pháp này có ưu điểm đơn giản nhưng robot chạy không ổn định, lúc rất nhanh, lúc lại chậm, tính ổn định phụ thuộc nhiều vào động cơ và kết cấu cơ khí của robot. Trong nghiên cứu này tập trung vào khắc phục nhược điểm đó bằng cách áp dụng bộ điều khiển PID điều khiển vị trí robot. [2]

Từ đó, nhóm đã nghiên cứu, tìm tòi về cách thức vận hành, hoạt động của robot. Kết hợp với những kiến thức tiếp thu được trong quá trình học tập và sự tìm hiểu một số tài liệu trên mạng, nhóm đã có đủ thông tin để thực hiện báo cáo “Thiết kế robot hai bánh tự cân bằng bám theo vật sử dụng kỹ thuật xử lý ảnh” làm đề tài báo cáo tốt nghiệp.

1.2 NHIỆM VỤ ĐỒ ÁN

1.2.1 Lý do hình thành đề tài

Tính cấp thiết và ứng dụng của cảm biến góc nghiêng vào đời sống, công nghiệp và nghiên cứu khóa học bây giờ rất quan trọng.

Nó được sử dụng rất nhiều trong các thiết bị như mobile, các thiết bị bảo vệ con người trong thực tế như động đất hay trong các vụ tai nạn giao thông. Ngoài ra ứng dụng trong công nghiệp như máy bay, tàu biển, các nhà máy, ...

Song song với đó là sự phát triển của công nghệ xử lý ảnh. Đây là ngành công nghệ không thể thiếu trong tự động hóa và đang phát triển rất mạnh mẽ trong những năm gần đây khi nó được áp dụng cực kỳ rộng rãi trong cả đời sống, xã hội lẫn trong công nghiệp, quân sự,...

Một số ứng dụng có thể kể đến của ngành xử lý ảnh là dây chuyền lắp ráp, phân loại, kiểm tra sản phẩm tự động trong nhà máy, camera quan sát, phát hiện trộm hoặc những vật thể lạ,...

Vì các ứng dụng thực tế như vậy, chúng em đã quyết định sẽ nghiên cứu cảm biến đo góc cùng với kỹ thuật xử lý ảnh và hình thành đề tài “Thiết kế xe hai bánh tự cân bằng bám theo vật ứng dụng kỹ thuật xử lý ảnh”.

1.2.2 Ý nghĩa khoa học thực tiễn

Cùng với phát triển của khoa học kỹ thuật, ngày nay robot có khả năng thay thế con người làm việc trong môi trường độc hại, trong sản xuất hoặc phục vụ con người về nhu cầu giải trí... Nhờ vào sự phát triển liên tục của công nghệ, robot đã và đang được chế tạo để phục vụ cho nhiều mục đích khác nhau.

Robot cân bằng đang được ứng dụng rất nhiều vào đời sống cũng như các lĩnh vực khác: [3]

- Phương tiện di chuyển giải trí, đã được thương mại hóa rộng rãi và nhận được sự hưởng ứng từ cộng đồng.
- Làm phương tiện di chuyển, vận chuyển hàng hóa ở những nơi nhỏ hẹp, trong các tòa nhà, phòng làm việc...
- Được mở rộng và nâng cấp để ứng dụng trong công nghiệp, hàng không vũ trụ, quân sự, ... [4]

1.2.3 Mục tiêu nghiên cứu

Nghiên cứu giải thuật PID.

Nghiên cứu lý thuyết về nguyên lý hoạt động con lắc ngược và áp dụng vào thiết kế mô hình và giải thuật cho robot hai bánh tự cân bằng.

Nghiên cứu, lựa chọn những thiết bị phần cứng phù hợp nhất cho mô hình như cảm biến, vi điều khiển, động cơ, driver, nguồn cung cấp, các module giao tiếp, Trong bài sẽ tập trung vào nghiên cứu cảm biến đo góc MPU6050, arduino 2560, raspberry pi và động cơ bước.

Sử dụng bộ lọc Complementary để lấy tín hiệu từ cảm biến MPU6050.

Ứng dụng bộ điều khiển PID vào điều khiển robot hai bánh tự cân bằng.

Nghiên cứu các giải thuật xử lý ảnh, ứng dụng thư viện OpenCV vào hệ thống xác định đối tượng và khoảng cách.

1.2.4 Đối tượng

Ngày nay đã có rất nhiều nghiên cứu và đề tài robot tự cân bằng. Đối tượng hướng đến trong đề tài của tui em là ứng dụng một cách hiệu quả bộ điều khiển PID vào việc điều khiển xe hai bánh tự cân bằng. Đối tượng tiếp theo là sử dụng công nghệ xử lý ảnh để xác định đối tượng, cụ thể là trái bóng tennis.

1.2.5 Giới hạn đề tài

Vì đây là lần đầu tui em thiết kế robot hai bánh tự cân bằng, nghiên cứu và ứng dụng xử lý ảnh vào thực tế nên có một số hạn chế như sau:

- Thời gian cho phép chỉ có một học kì.
- Thiếu kinh nghiệm và trải nghiệm thực tế.
- Hạn chế về kinh phí nên các thiết bị không thực sự hiệu quả như mong đợi.

Chính vì những hạn chế trên nên robot có thể không cân bằng và bám theo trái banh một cách chính xác và nhanh chóng.

1.3 CẤU TRÚC ĐỒ ÁN

Đồ án “Thiết kế xe hai bánh tự cân bằng sử dụng kỹ thuật xử lý ảnh” bao gồm các chương như sau:

Chương 1: Tổng quan về đồ án robot hai bánh tự cân bằng bám theo đối tượng sử dụng kỹ thuật xử lý ảnh

Chương này cung cấp cho chúng ta những vấn đề thực tế, các lĩnh vực quan tâm, tóm tắt những nghiên cứu trong đời sống có liên quan đến đồ án.

Ngoài ra chương 1 còn nêu tính cấp thiết và lý do hình thành đề tài, ý nghĩa khoa học thực tiễn, mục tiêu nghiên cứu, đối tượng, phạm vi giới hạn và tóm tắt cấu trúc của đồ án.

Chương 2: Các lý thuyết áp dụng trong robot hai bánh tự cân bằng bám theo đối tượng sử dụng kỹ thuật xử lý ảnh

Chương này nêu ra các khái niệm và phương pháp giải quyết vấn đề của đồ án bao gồm lý thuyết và ứng dụng của bộ điều khiển PID, bộ lọc Complementary, các giải thuật xử lý ảnh được sử dụng trong bài và áp dụng các giải thuật đó vào mô hình xe hai bánh tự cân bằng.

Chương 3: Lựa chọn các thiết bị phần cứng cho robot hai bánh tự cân bằng bám theo đối tượng

Chương này liệt kê các thiết bị được sử dụng để xây dựng lên robot, các thông số kỹ thuật cơ bản, cấu tạo, cách sử dụng và lý do lựa chọn. Các thiết bị đó bao gồm Arduino Mega 2560, cảm biến góc nghiêng MPU6050, module bluetooth HC05, động cơ bước, raspberry pi 3, picamera, ...

Chương 4: Xây dựng và áp dụng thuật toán xử lý ảnh vào robot

Chương này nêu rõ cách thức thực hiện và áp dụng giải thuật xử lý ảnh vào trong robot, các giải thuật bao gồm xác định đối tượng có trong hình, định vị xác định được tọa độ của đối tượng.

Chương 5: Kết quả thực nghiệm

Chương này nêu ra các kết quả mà đề tài đã đạt được bằng các số liệu dẫn chứng cụ thể thực tế sau khi nghiên cứu và xây dựng hoàn chỉnh mô hình robot hai bánh tự cân bằng bám theo đối tượng sử dụng kỹ thuật xử lý ảnh.

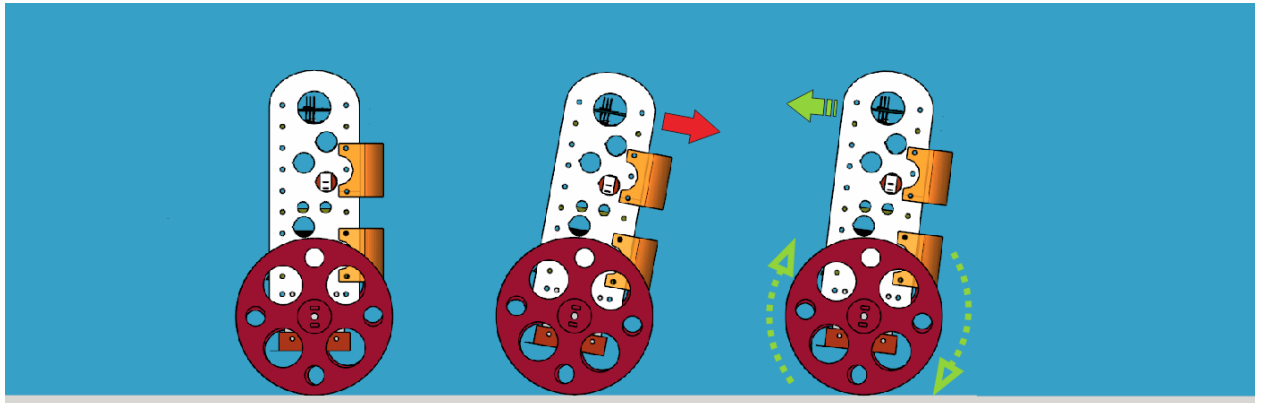
Chương 6: Kết luận và hướng phát triển

Các kết luận và nhận xét về kết quả, ưu điểm và nhược điểm trong cách hoạt động của robot.

Trình bày hướng phát triển của robot trong tương lai: cải thiện độ chính xác, tốc độ xử lý của robot, các ý tưởng xây dựng kết hợp hoàn thiện robot có mục đích gần gũi áp dụng vào đời sống.

CHƯƠNG 2: CÁC LÝ THUYẾT ÁP DỤNG TRONG ROBOT HAI BÁNH TỰ CÂN BẰNG BẮM THEO ĐỐI TƯỢNG

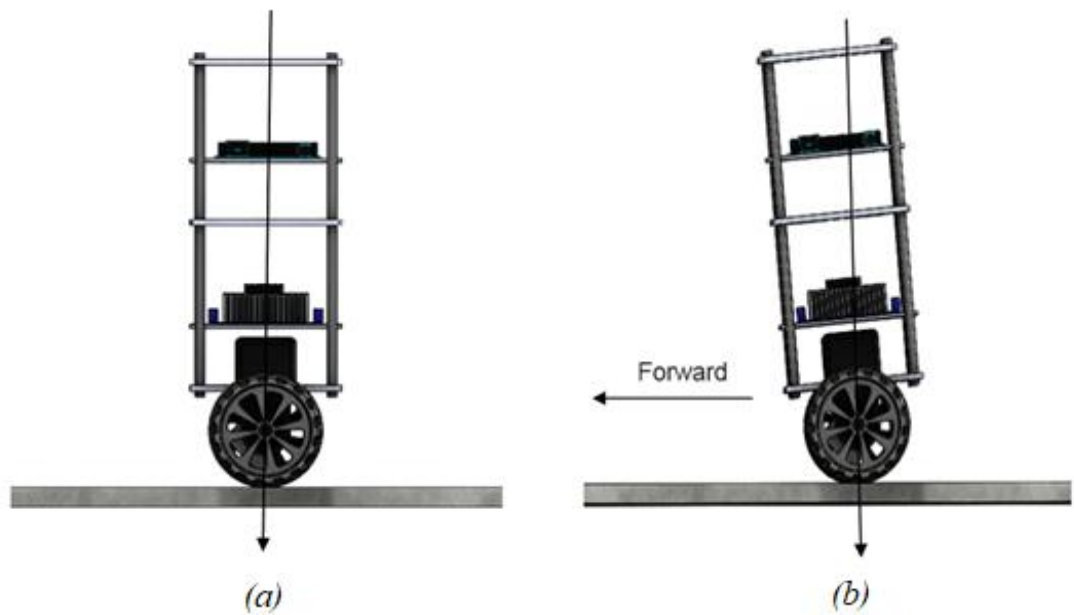
2.1 NGUYÊN LÝ HOẠT ĐỘNG CỦA ROBOT 2 BÁNH TỰ CÂN BẰNG



Hình 2.1 Nguyên lý di chuyển của robot hai bánh tự cân bằng

Một robot có khả năng duy trì sự cân bằng khi di chuyển trên hai bánh xe của nó được gọi là robot hai bánh tự cân bằng. Quá trình cân bằng là một trong những phần quan trọng nhất của robot hai bánh tự cân bằng và cũng thường được gọi là điều khiển ổn định. Hai bánh xe cũng là thành phần chính và nằm bên dưới chân robot để giữ cho robot di chuyển về phía trước hoặc phía sau mà robot không bị đổ. Như hình (H.2.1), khi xe nghiêng về phía nào thì bánh xe phải ngay lập tức điều hướng về phía đó để xe giữ được vị trí cân bằng của mình.

Loại robot này khiến cho chúng ta gặp nhiều khó khăn trong khi thường xuyên gặp phải vấn đề với một điều kiện không ổn định. Lý thuyết được sử dụng để duy trì sự ổn định của robot này dựa trên lý thuyết con lắc ngược [5]. Lý thuyết này đòi hỏi một kiến thức sâu về các thông số trong hệ thống để phát triển một hệ thống đáng tin cậy và có khả năng điều khiển. Lý thuyết con lắc ngược nổi tiếng là một lý thuyết Pole & Cart [6]. Mặc dù nó không có vẻ giống nhau nhưng vẫn có một số nguyên tắc chung giữa robot cân bằng hai bánh và mô hình Pole & Cart, ví dụ trong mô hình hệ thống, Cart tương đương với bánh xe và pole không khác nhiều so với khung gầm của robot.



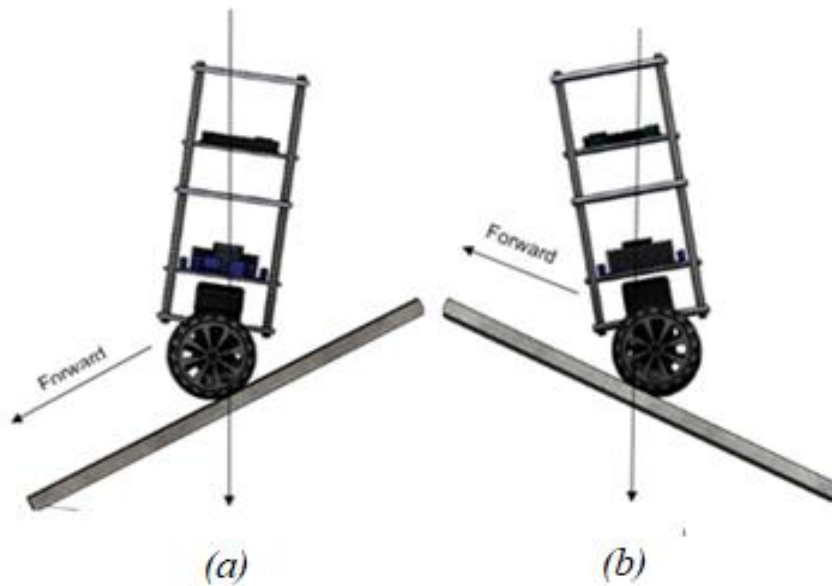
Hình 2.2: Robot tự cân bằng di chuyển

(a) Robot đứng yên;

(b) Robot di chuyển về phía trước

Quy tắc cần thiết để khái quát hóa các đặc điểm của bánh xe bên trái và bên phải và kết hợp chúng với nhau cùng với góc nghiêng của xe. Điều này làm cho việc tính toán dễ dàng vì cả hai bánh xe sẽ hoạt động đồng loạt để duy trì sự ổn định (H. 2.2(a)), hai bánh xe phải đồng bộ với nhau để giữ cân bằng cho robot.

Mục đích của nguyên lý lắc ngược là giữ cho bánh xe bên dưới trung tâm khối lượng của khung gầm robot. Nếu khung nghiêng về phía trước, hai bánh xe sẽ di chuyển về phía trước sao cho góc nghiêng bằng không thì robot sẽ duy trì sự cân bằng. Mặt khác, nếu khung robot nghiêng về phía sau, hai bánh xe sẽ di chuyển về phía sau. Để ngăn không cho tốc độ của robot bị vọt lố, robot cân bằng hai bánh chỉ cần di chuyển theo hướng ngược lại thì tốc độ sẽ giảm dần. Nếu nguyên tắc này không được duy trì, robot sẽ bị ngã. Dựa vào quy tắc này, robot có thể di chuyển về phía trước (H.2.2b). Ta cũng cần phải chú ý đến chiều cao và chiều rộng của robot, nếu robot quá cao, xe sẽ khó có thể di chuyển được ổn định. Trọng tâm của robot cũng cần được quan tâm, ta nên thiết kế robot có trọng tâm ở chính giữa và nằm dưới gầm robot để robot có thể ổn định hơn.



Hình 2.3: Robot cân bằng trên mặt phẳng nghiêng

(a) Robot di chuyển xuống dốc;

(b) Robot di chuyển lên dốc

Khi hệ thống đi xuống dốc (H.2.3a), robot sẽ hướng về hướng ngược lại để giảm tốc và giữ thăng bằng. Ở hướng ngược lại (H.2.3b), khi robot lên dốc, góc nghiêng của robot sẽ lớn hơn góc dốc.

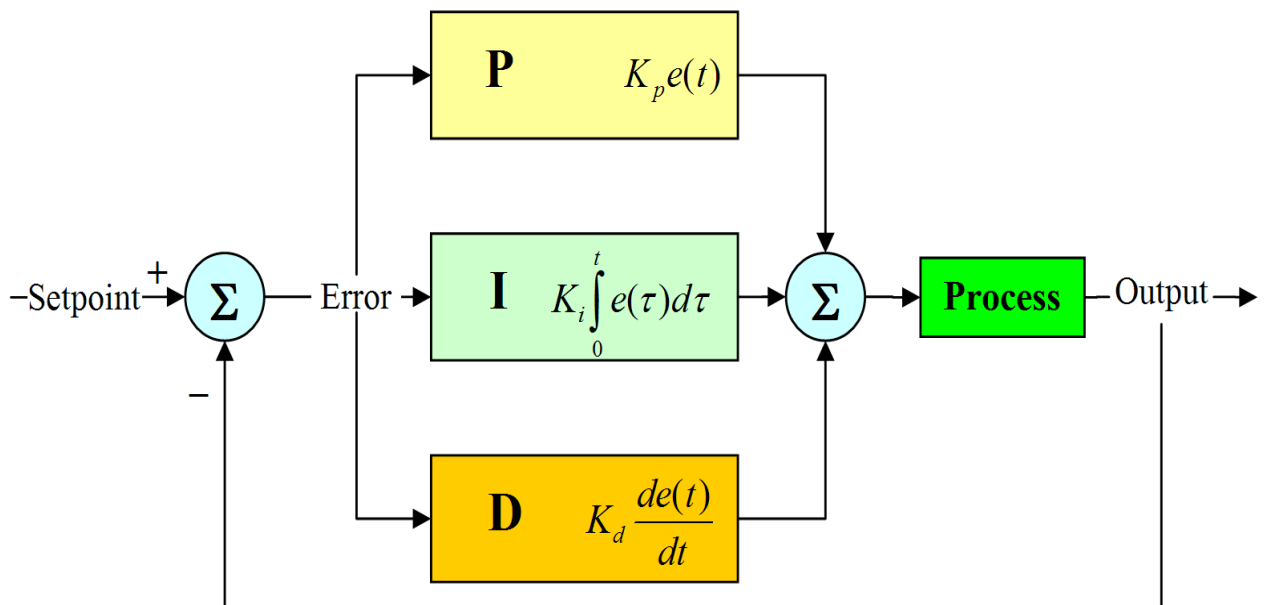
2.2 BỘ ĐIỀU KHIỂN PID

Bộ điều khiển PID là một bộ điều khiển vi tích phân tỉ lệ (PID- Proportional Integral Derivative) có hồi tiếp và được sử dụng rộng rãi trong các hệ thống điều khiển công nghiệp – bộ điều khiển PID là bộ điều khiển được sử dụng nhiều nhất trong các bộ điều khiển có phản hồi. Bộ điều khiển PID sẽ tính toán giá trị "sai số" là hiệu số giữa giá trị đo (thông số biến đổi) và giá trị đặt mong muốn. Bộ điều khiển sẽ thực hiện giảm tối đa sai số bằng cách điều chỉnh giá trị điều khiển đầu vào. Bộ điều khiển PID được áp dụng vào hệ thống mà không cần mô hình toán học của hệ thống đó. Trong trường hợp không có mô hình toán học của hệ thống hoặc hệ thống có mô hình toán quá phức tạp thì bộ điều khiển PID là sẽ bộ điều khiển tốt nhất. Tuy nhiên, để đạt được kết quả như mong muốn, các thông số PID sử dụng

trong tính toán phải điều chỉnh theo tính chất của hệ thống - trong khi kiểu điều khiển là giống nhau, các thông số phải phụ thuộc vào đặc thù của hệ thống. [7]

2.2.1 Lý thuyết về bộ điều khiển PID

Giải thuật tính toán bộ điều khiển PID bao gồm 3 thông số riêng biệt, do đó đôi khi nó còn được gọi là điều khiển ba khâu: các giá trị tỉ lệ, tích phân và vi phân, viết tắt là P, I, và D. Giá trị tỉ lệ xác định tác động của sai số hiện tại, giá trị tích phân xác định tác động của tổng các sai số quá khứ, và giá trị vi phân xác định tác động của tốc độ biến đổi sai số. Tổng chập của ba tác động này dùng để điều chỉnh quá trình thông qua một phần tử điều khiển như vị trí của van điều khiển hay bộ nguồn của phần tử gia nhiệt, cụ thể về robot cân bằng trong bài là 2 bánh xe. Nhờ vậy, những giá trị này có thể làm sáng tỏ về quan hệ thời gian: P phụ thuộc vào sai số hiện tại, I phụ thuộc vào tích lũy các sai số quá khứ, và D dự đoán các sai số tương lai dựa vào tốc độ thay đổi hiện tại, sơ đồ khối của bộ điều khiển như hình sau:



Hình 2.4: Sơ đồ khối bộ điều khiển PID

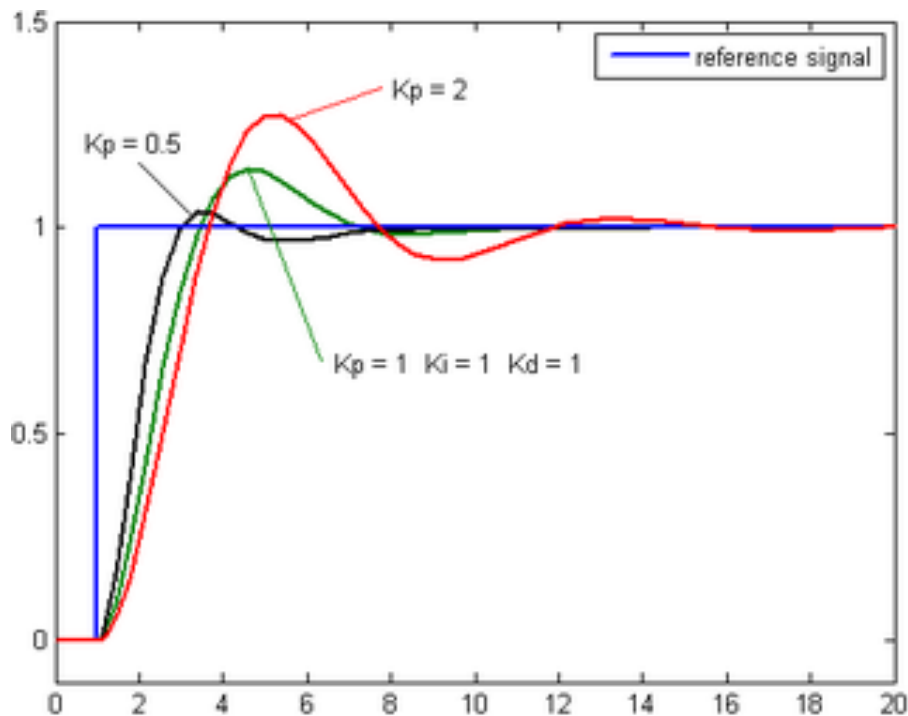
Bằng cách điều chỉnh 3 thông số trong giải thuật của bộ điều khiển PID (H.2.4), bộ điều khiển có thể dùng trong những thiết kế có yêu cầu đặc biệt. Đáp ứng của bộ điều khiển có thể được mô tả dưới dạng độ nhảy sai số của bộ điều khiển, giá trị mà bộ điều khiển vọt lố, điểm đặt và giá trị dao động của hệ thống. Lưu ý là công dụng của giải thuật PID trong điều khiển không đảm bảo tính tối ưu hoặc ổn định cho hệ thống.

Phương trình chung của bộ điều khiển PID:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{d}{dt} e(t) \quad (2.1)$$

2.2.1.1 Khâu tỉ lệ

Khâu tỉ lệ (đôi khi còn được gọi là độ lợi) làm thay đổi giá trị đầu ra, tỉ lệ với giá trị sai số hiện tại. Đáp ứng tỉ lệ có thể được điều chỉnh bằng cách nhân sai số đó với một hằng số K_p , được gọi là độ lợi tỉ lệ.



Hình 2.5: Thể hiện tác động của khâu tỉ lệ trong hệ thống

Khâu tỉ lệ được cho bởi công thức:

$$P_{out} = K_p e(t) \quad (2.2)$$

Trong đó:

- P_{out} : thừa số tỉ lệ đầu ra
- K_p : độ lợi tỉ lệ, thông số điều chỉnh
- e : sai số
- t : thời gian tức thời

Trong bộ điều khiển PID, khâu tỉ lệ ảnh hưởng đến giá trị sai số nhiều nhất, đây là khâu chính giúp giảm sai số đáng kể cho hệ thống (H.2.5).

2.2.1.2 Khâu tích phân

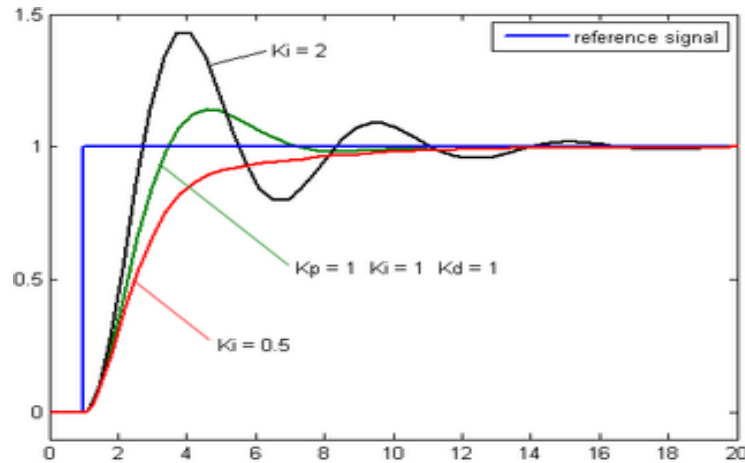
Phân phối của khâu tích phân (đôi khi còn gọi là reset) tỉ lệ thuận với cả biên độ sai số lẫn quãng thời gian xảy ra sai số. Tổng sai số tức thời theo thời gian (tích phân sai số) cho ta tích lũy bù đã được hiệu chỉnh trước đó. Tích lũy sai số sau đó được nhân với độ lợi tích phân và cộng với tín hiệu đầu ra của bộ điều khiển. Biên độ phân phối của khâu tích phân trên tất cả tác động điều chỉnh được xác định bởi độ lợi tích phân K_i .

Thừa số tích phân được cho bởi công thức:

$$I_{out} = K_i \int_0^t e(t) dt \quad (2.3)$$

Trong đó:

- I_{out} : thừa số tích phân của đầu ra
- K_i : độ lợi tích phân, thông số điều chỉnh
- e : sai số
- t : thời gian tức thời



Hình 2.6: Ảnh hưởng của khâu tích phân trong hệ thống

Khâu tích phân (khi cộng thêm khâu tỉ lệ) sẽ tăng tốc chuyển động của quá trình tới điểm đặt và khử số dư sai số ổn định với một tỉ lệ chỉ phụ thuộc vào bộ điều khiển. Tuy nhiên, vì khâu tích phân là đáp ứng của sai số tích lũy trong quá khứ, nó có thể khiến giá trị hiện tại vượt quá giá trị đặt (ngang qua điểm đặt và tạo ra một độ lệch với các hướng khác) (H.2.6).

2.2.1.3 Khâu vi phân

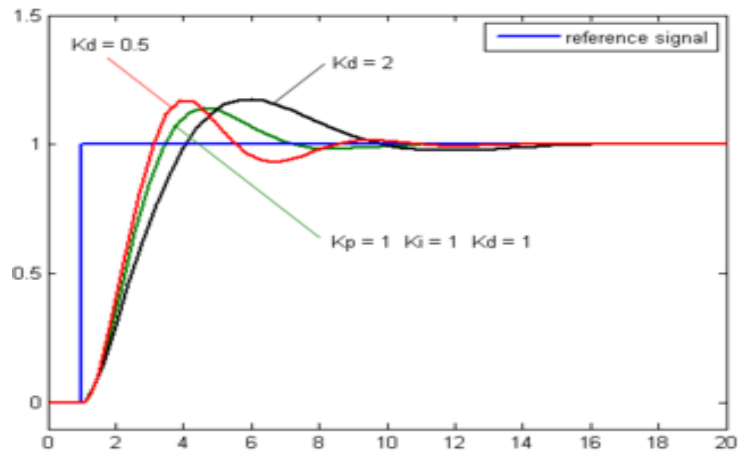
Tốc độ thay đổi của sai số qua trình được tính toán bằng cách xác định độ dốc của sai số theo thời gian (tức là đạo hàm bậc một theo thời gian) và nhân tốc độ này với độ lợi tỉ lệ K_d . Biên độ của phân phối khâu vi phân (đôi khi được gọi là tốc độ) trên tất cả các hành vi điều khiển được giới hạn bởi độ lợi vi phân K_d .

Thừa số vi phân được cho bởi công thức:

$$D_{out} = K_d \frac{d}{dt} e(t) \quad (2.4)$$

Trong đó:

- D_{out} : thừa số vi phân của đầu ra
- K_d : độ lợi vi phân, một thông số điều chỉnh
- e : sai số
- t : thời gian tức thời



Hình 2.7: Ảnh hưởng của khâu vi phân trong hệ thống

Khâu vi phân làm chậm tốc độ thay đổi của đầu ra bộ điều khiển và đặc tính này là đáng chú ý nhất để đạt tới điểm đặt của bộ điều khiển. Từ đó, điều khiển vi phân được sử dụng để làm giảm biên độ vọt lố được tạo ra bởi thành phần tích phân và tăng cường độ ổn định của bộ điều khiển hỗn hợp. Tuy nhiên, phép vi phân của một tín hiệu sẽ khuếch đại nhiễu và do đó khâu này sẽ nhạy hơn đối với nhiễu trong sai số, và có thể khiến quá trình trở nên không ổn định nếu nhiễu và độ lợi vi phân đủ lớn (H.2.7).

2.2.1.4 Tác động độc lập của các khâu trong bộ điều khiển PID

Bảng 2.1: Tác động của việc tăng một thông số độc lập

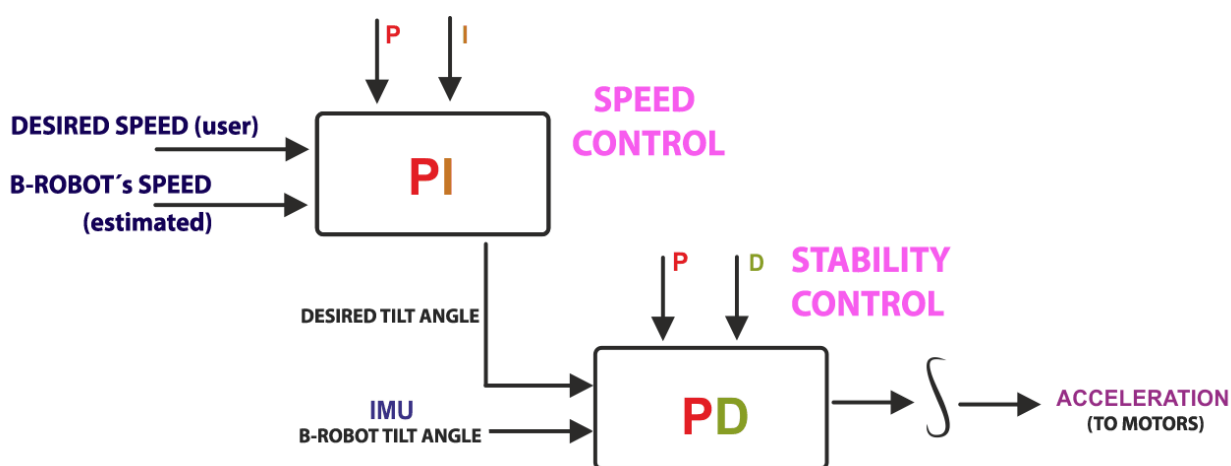
Thông số	Thời gian khởi động	Độ vọt lố	Thời gian xác lập	Sai số ổn định	Độ ổn định
Kp	Giảm	Tăng	Thay đổi nhỏ	Giảm	Giảm cấp
Ki	Giảm	Tăng	Tăng	Giảm đáng kể	Giảm cấp
Kd	Giảm ít	Giảm ít	Giảm ít	Về lý thuyết không tác động	Cải thiện nếu Kd nhỏ

Bảng 2.1 cho chúng ta thấy sự tác động của các thành phần K_p , K_i , K_d vào hệ thống như thế nào, và chúng ta có thể dựa vào sự tác động này để tìm được các thông số K_p , K_i , K_d phù hợp cho hệ thống.

2.2.2 Áp dụng bộ điều khiển PID trong xe cân bằng

Để giữ cho xe cân bằng, chúng ta sử dụng bộ điều khiển PD với mục đích là triệt tiêu độ vọt lố, giúp cho xe đạt được vị trí cân bằng mà không bị quá độ. Giá trị đặt cho bộ điều khiển PD là góc chúng ta cần robot hướng đến, giả sử như để xe cân bằng, góc đặt sẽ bằng 0 độ. Giá trị trả về để so sánh với giá trị đặt là góc đo được từ cảm biến MPU6050 cũng chính là góc nghiêng thực tế của robot, sai số của bộ điều khiển sẽ bằng hiệu của góc đặt và góc đo được, từ đó bộ điều khiển PD sẽ tính toán ra giá trị truyền vào vi điều khiển để điều khiển tốc độ 2 động cơ với mục đích là triệt tiêu sai số, làm sai số dần về 0, khi đó góc đọc từ cảm biến MPU6050 cũng là góc nghiêng của robot sẽ dần về góc đặt là 0 độ, robot sẽ cân bằng.

Hình 2.8 là bộ điều khiển PID sử dụng trong robot cân bằng. Với sự kết hợp của bộ PI và PD để điều khiển cân bằng, điều khiển di chuyển cho robot.



Hình 2.8: Bộ điều khiển PID dùng trong robot cân bằng

Để robot cân bằng di chuyển, chúng ta chỉ cần tăng hoặc giảm góc đặt trong bộ điều khiển PD, lúc đó góc đo được từ cảm biến MPU6050 hay cũng chính là góc nghiêng của robot sẽ hướng theo góc đặt vào bộ điều khiển, lúc đó robot sẽ di chuyển theo chiều nghiêng của góc đặt.

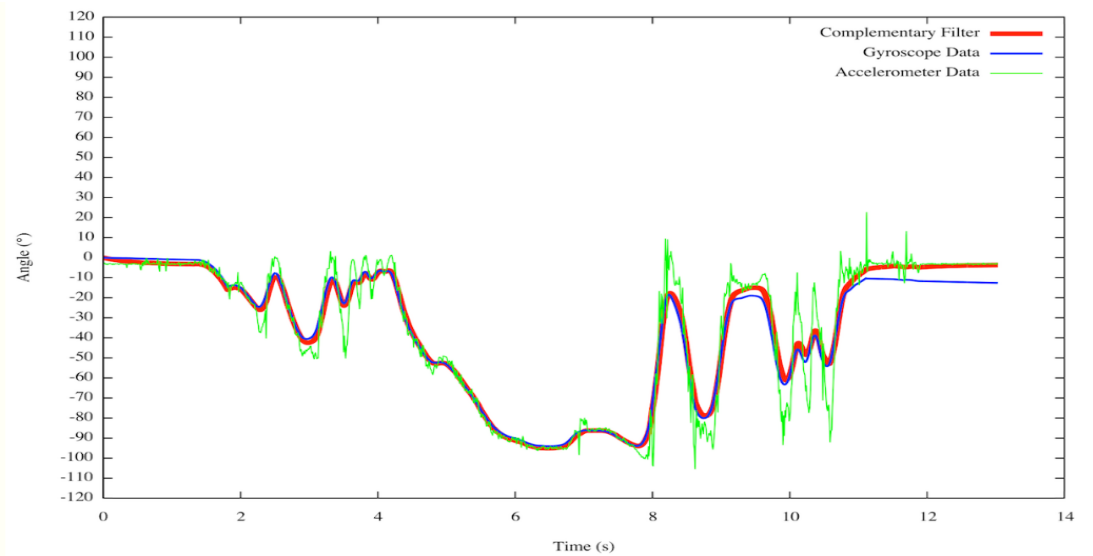
Vấn đề trong việc điều khiển robot là tốc độ di chuyển của robot sẽ tăng liên tục nếu robot chỉ nghiêng 1 góc độ. Chúng ta sẽ cần một thuật toán để ổn định tốc độ di chuyển của robot dựa vào nguyên lý điều khiển góc nghiêng trong bộ điều khiển PD, mà cụ thể là thay đổi góc đặt của bộ điều khiển PD.

Để giải quyết vấn đề này, chúng em sử dụng bộ điều khiển PI, một bộ điều khiển rất thông dụng trong việc điều khiển tốc độ động cơ khi mà bộ điều khiển này giúp hệ thống tác động nhanh và triệt tiêu sai số dư. Giá trị đặt vào bộ điều khiển PI là tốc độ chúng ta mong muốn, giá trị trả về để so sánh với tốc độ đặt là tốc độ đo được từ động cơ. Sai số của bộ điều khiển chính là hiệu của tốc độ đặt và tốc độ đo được từ động cơ. Từ sai số này, bộ điều khiển PI sẽ tính toán ra giá trị góc là góc đặt vào bộ điều khiển ổn định PD để bộ điều khiển PD sẽ tính toán ra tốc độ điều khiển động cơ để bám theo góc đặt vào.

Như vậy dựa vào bộ điều khiển PID, chúng em đã thiết kế được bộ điều khiển PD cho khâu cân bằng và PI cho khâu di chuyển của robot cân bằng

2.3 BỘ LỌC COMPLEMENTARY

Bộ lọc Complementary hay còn gọi là bộ lọc bù, nó có tác dụng lọc nhiễu tín hiệu của một quá trình dựa trên đặc tính của giá trị đó nằm trong miền tần số. Bộ lọc này thường có hai phần là bộ lọc thông thấp và bộ lọc thông cao, bộ lọc thông thấp được áp dụng cho các thiết bị hoạt động trong miền tần số thấp, khi đó bộ lọc chỉ lấy số liệu trong miền tần số thấp, còn bộ lọc thông cao thì ngược lại, thường được dùng cho các thiết bị hoạt động trong vùng tần số cao. Khi ta kết hợp kết hợp hai bộ lọc này lại, ta sẽ lọc được giá trị chính xác mong muốn.

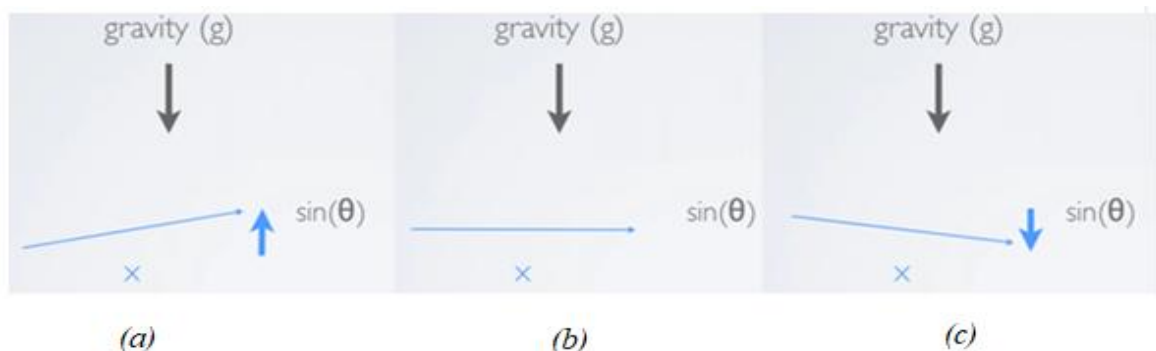


Hình 2.9: Sử dụng bộ lọc Complementary để tính góc từ cảm biến MPU6050

Do đặc tính lọc trong miền tần số, bộ lọc complementary thường xuyên được sử dụng trong việc lấy giá trị góc từ cảm biến MPU6050 (H.2.9).

Cảm biến MPU6050 tính góc từ hai cảm biến chính là cảm biến gia tốc (Accelerometer) và cảm biến vận tốc (Gyroscope) và hai cảm biến này hoạt động chính xác trong hai vùng tần số khác nhau. Cảm biến gia tốc (Accelerometer) hoạt động tốt trong vùng tần số thấp, ngược lại cảm biến vận tốc (Gyroscope) trả giá trị chính xác trong vùng tần số cao.

Cảm biến gia tốc kế (Accelerometer): cảm biến này sẽ đo độ lớn gia tốc góc dựa vào lực hấp dẫn, từ đó tính ra góc nghiêng của vật.



Hình 2.10: Đo góc bằng cảm biến Accelerometer

(a) $x: -/ < 1/g$;

(b) $x: 0g$;

(c) $x: +/ < 1/g$

Độ lớn của gia tốc trọng trường ở mỗi góc nghiêng của cảm biến là khác nhau, được thể hiện như Hình 2.10. Khi cảm biến đặt song song với mặt đất, gia tốc của cảm biến sẽ trả về giá trị là $0g$ (H.2.10b). Còn với mỗi góc nghiêng khác nhau, cảm biến sẽ trả về các giá trị khác nhau có độ lớn là $\pm|<1|g$ (H.2.10a, H2.10c).

Cảm biến con quay hồi chuyển (Gyroscope): đây là cảm biến đo độ lớn vận tốc góc, từ giá trị vận tốc góc này tính toán ra góc nghiêng của robot dựa vào độ lớn và hướng của nó.

Giá trị cảm biến vận tốc trả về là vận tốc góc có hướng (H.2.11). Khi cảm biến đứng yên, vận tốc góc có độ lớn bằng 0 (H.2.11a), khi robot di chuyển, vận tốc góc của cảm biến sẽ thay đổi tùy theo hướng và vận tốc của robot (H.2.11b, c), chúng ta dựa vào độ lớn và hướng của vận tốc góc để tính ra góc nghiêng của robot.

Cảm biến Accelerometer đo góc dựa vào sự thay đổi của trọng lực nên chỉ có thể phản hồi tín hiệu góc chính xác khi vật thể có sự thay đổi chậm về góc, còn các ứng dụng có sự thay đổi góc nhanh như con lắc ngược trong giai đoạn trả về vị trí ban đầu thì cảm biến Accelerometer không thể cho ra giá trị chính xác.



Hình 2.11: Góc nghiêng được đo từ cảm biến Gyroscope

(a) $Gyro = 0$;

(b) $Gyro +$;

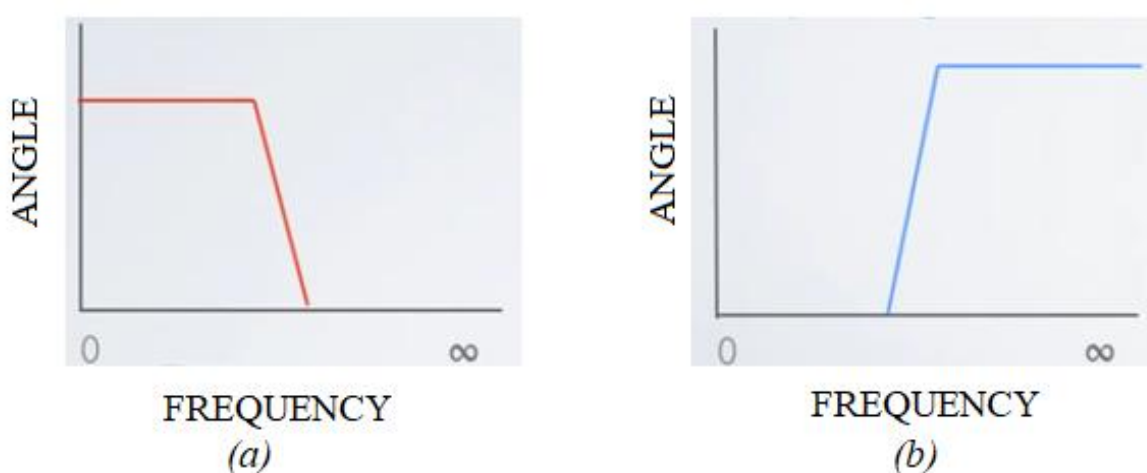
(c) $Gyro -$

Ngược lại, với cảm biến Gyroscope tính góc dựa vào sự thay đổi tốc độ và góc sẽ trả về 0 khi không có sự thay đổi về tốc độ. Chính vì đặc điểm này, cảm biến Gyroscope rất dễ bị trả về sai góc lệnh khi mà trong quá trình quay có giai đoạn robot quay chậm hoặc dừng quay. Nên cảm biến Gyroscope chỉ thích hợp để đo góc cho những trường hợp quá trình có vận tốc góc nhanh – góc thay đổi nhanh.

Kết hợp hai đặc tính trên của 2 cảm biến Accelerometer và Gyroscope, ta sẽ tìm được giá trị góc chính xác hơn.

Cảm biến Accelerometer chỉ trả về góc chính xác trong miền tần số thấp (H.2.12a), ngược lại thì cảm biến Gyroscope cho giá trị góc chính xác trong miền tần số cao (H.2.12b).

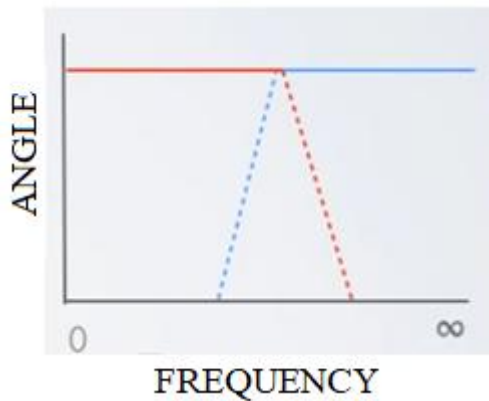
Vậy nên ở đây ta sẽ sử dụng bộ lọc Complementary lọc thông thấp cho cảm biến Accelerometer và lọc thông cao cho cảm biến Gyroscope.



Hình 2.12: Ảnh thể hiện giá khả năng đo góc của 2 cảm biến trong miền tần số

(a) Accelerometer: Low-Pass;

(b) Gyroscope: High-Pass;



Hình 2.13: Bộ lọc Complementary kết hợp 2 giá trị accelerometer và gyroscope

Công thức tổng quát của bộ lọc Complementary như sau (H.2.13):

$$góc = 0.99 * (góc + gyro * dt) + 0.01 * acc \quad (2.4)$$

Nếu tần số là 100Hz, vòng lặp được thực hiện 100 lần trong 1 giây thì thời gian để thực hiện cả lọc thông thấp và lọc thông cao là:

$$t = \frac{a * dt}{1 - a} = \frac{0.99 * 0.01(s)}{1 - 0.99} = 0.99 \text{ giây} \quad (2.5)$$

Ưu điểm của bộ lọc Complementary:

- Không cần mô hình hóa, không có các phương trình trạng thái toán học phức tạp.
- Thuật toán đơn giản, dễ hiểu và áp dụng được cho nhiều ứng dụng thực tế.
- Chỉ xử lý phép toán đơn giản, cho thời gian xử lý nhanh.
- Phù hợp với mọi loại vi điều khiển mini, có tốc độ xử lý thấp.

2.4 XỬ LÝ ẢNH

Xử lý ảnh là một phân ngành trong xử lý số tín hiệu với tín hiệu xử lý là ảnh. Đây là một phân ngành khoa học mới rất phát triển trong những năm gần đây. Xử lý ảnh gồm 4 lĩnh vực chính: xử lý nâng cao chất lượng ảnh, nhận dạng ảnh, nén ảnh và truy vấn ảnh. Sự phát triển của xử lý ảnh đem lại rất nhiều lợi ích cho cuộc sống của con người. [8]

Ngày nay xử lý ảnh đã được áp dụng rất rộng rãi trong đời sống như: photoshop, nén ảnh, nén video, nhận dạng biển số xe, nhận dạng khuôn mặt, nhận dạng chữ viết, xử lý ảnh thiên văn, ảnh y tế,....

2.4.1 Thư viện OpenCV

OpenCV (Open Computer Vision) [9] là một thư viện mã nguồn mở hàng đầu cho xử lý về thị giác máy tính, machine learning, xử lý ảnh. OpenCV được viết bằng C/C++, vì vậy có tốc độ tính toán rất nhanh, có thể sử dụng với các ứng dụng liên quan đến thời gian thực. OpenCV có các interface cho C/C++, Python Java vì vậy hỗ trợ được cho Window, Linux, MacOS lẫn Android, iOS OpenCV có cộng đồng hơn 47 nghìn người dùng và số lượng download vượt quá 6 triệu lần

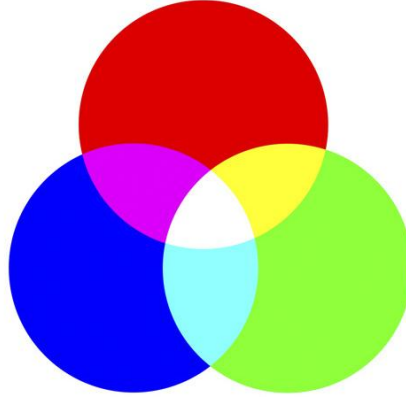
Dự án về OpenCV được khởi động từ những năm 1999, đến năm 2000 nó được giới thiệu trong một hội nghị của IEEE về các vấn đề trong thị giác máy và nhận dạng, tuy nhiên bản OpenCV 1.0 mãi tới tận năm 2006 mới chính thức được công bố và năm 2008 bản 1.1 pre-release) mới được ra đời. Tháng 10 năm 2009, bản OpenCV thế hệ thứ hai ra đời thường gọi là phiên bản 2.x), phiên bản này có giao diện của C++ (khác với phiên bản trước có giao diện của C) và có khá nhiều điểm khác biệt so với phiên bản thứ nhất.

Thư viện OpenCV ban đầu được sự hỗ trợ từ Intel, sau đó được hỗ trợ bởi Willow Garage, một phòng thí nghiệm chuyên nghiên cứu về công nghệ robot. Cho đến nay, OpenCV vẫn là thư viện mở, được phát triển bởi nguồn quỹ không lợi nhuận (none-profit foundation) và được sự hưởng ứng rất lớn của cộng đồng.

OpenCV hỗ trợ đa nền tảng. Nó hỗ trợ cả Windows và Linux, và gần đây hơn là MacOSX. Với giao diện của nó là nền tảng độc lập.

2.4.2 Không gian màu RGB và HSV

2.4.2.1 Không gian màu RGB

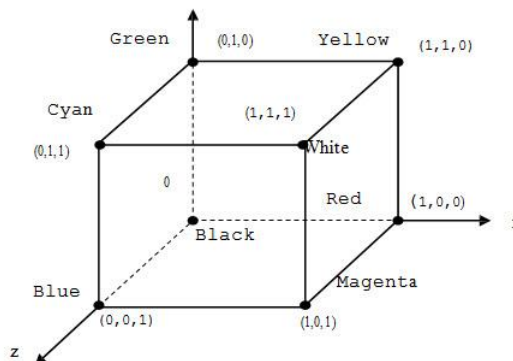


Hình 2.14: Không gian màu RGB

Tổ chức quốc tế về chuẩn hóa màu CIE (Commission Internationale d'Eclairage) [10] đưa ra một số chuẩn để biểu diễn màu. Các hệ này có các chuẩn riêng. Hệ chuẩn màu CIE-RGB dùng 3 màu cơ bản R, G, B và ký hiệu RGBCIE để phân biệt với các chuẩn khác (H.2.14). Như đã nêu trên, một màu là tổ hợp của các màu cơ bản theo một tỷ lệ nào đó. Như vậy, mỗi pixel ảnh màu ký hiệu P_x , được viết: (T trong công thức dưới đây là ký hiệu chuyển vị).

$$P_x = [red, green, blue]^T \quad (2.6)$$

Người ta dùng hệ tọa độ ba màu R-G-B (tương ứng với hệ tọa độ x-y-z) để biểu diễn màu như sau:



Hình 2.15: Biểu diễn không gian màu RGB trên hệ trục tọa độ xyz

Trong cách biểu diễn này (H.2.15) ta có công thức:

$$\text{đỏ} + \text{lục} + \text{lam} = 1 \quad (2.7)$$

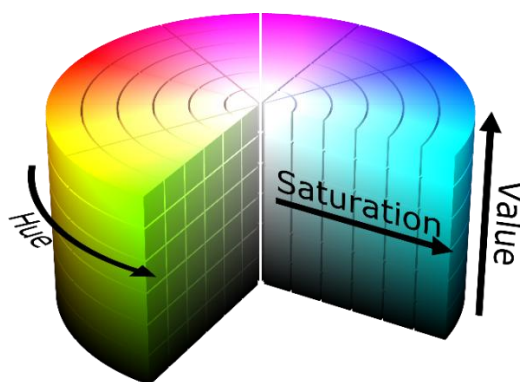
Công thức này gọi là công thức Maxwell. Trong hình trên, tam giác tạo bởi ba đường đứt đoạn gọi là tam giác Maxwell. Màu trắng trong hệ tọa độ này được tính bởi:

$$\text{trắng}_{CIE} = (\text{đỏ}_{CIE} + \text{lục}_{CIE} + \text{lam}_{CIE}) = 1 \quad (2.8)$$

RGB là không gian màu rất phổ biến được dùng trong đồ họa máy tính và nhiều thiết bị kỹ thuật số khác. Ý tưởng chính của không gian màu này là sự kết hợp của 3 màu sắc cơ bản: màu đỏ (R, Red), xanh lục (G, Green) và xanh lơ (B, Blue) để mô tả tất cả các màu sắc khác.

Nếu như một ảnh số được mã hóa bằng 24 bit, nghĩa là 8 bit cho kênh R, 8 bit cho kênh G, 8 bit cho kênh B, thì mỗi kênh màu này sẽ nhận giá trị từ 0-255. Với mỗi giá trị khác nhau của các kênh màu kết hợp với nhau ta sẽ được một màu khác nhau, như vậy ta sẽ có tổng cộng $255 \times 255 \times 255 = 1.66$ triệu màu sắc. Ví dụ: màu đen là sự kết hợp của các kênh màu (R, G, B) với giá trị tương ứng (0, 0, 0) màu trắng có giá trị (255, 255, 255), màu vàng có giá trị (255, 255, 0), màu tím đậm có giá trị (64, 0, 128) ... Nếu ta dùng 16bit để mã hóa một kênh màu (48 bit cho toàn bộ 3 kênh màu) thì dải màu sẽ trải rộng lên tới $3 \times 2^{16} = \dots$ Một con số rất lớn.

2.4.2.2 Không gian màu HSV



Hình 2.16: Không gian màu HSV

Một không gian màu khác cũng thường được sử dụng trong thực tế đó là không gian màu HSV (H.2.16) [11]. Đây là không gian màu rất phổ biến để sử dụng trong chỉnh sửa hình ảnh, phân tích hình ảnh và ứng dụng trong công nghệ vision computer. HSV (Hue, Saturation, Value) còn được gọi là HSB (Hue, Saturation, Brightness) được định nghĩa là một không gian màu được cấu thành theo ba thành phần: Hue là loại màu như màu đỏ, lục lam, xanh lục hoặc vàng trong phạm vi 0-360 độ. Trong phạm vi từ 0-120 độ, Hue thay đổi từ màu đỏ thành màu xanh lá cây. Trong phạm vi 120-240 độ, Hue thay đổi từ màu xanh lá cây thành màu xanh dương và trong phạm vi 240-360 độ, Hue thay đổi từ màu xanh dương thành màu đỏ. Saturation là độ bão hòa là nói đến cường độ của màu sắc cụ thể. Độ bão hòa nằm trong khoảng từ 0 đến 255 (từ 0 đến 1). $S = 0$ là trung tâm của hình nón, là nơi mà Hue có độ bão hòa nhạt nhất. $S = 1$ là bên ngoài của hình nón nơi mà Hue có độ bão hòa đậm nhất. Value đề cập đến độ sáng của màu sắc, có phạm vi từ 0 đến 255 (0 đến 1) tương ứng với đáy đến đỉnh của hình nón.

2.4.2.3 Phép toán chuyển đổi không gian màu RGB sang không gian màu HSV [12]

Khi sử dụng không gian màu RGB trong quá trình xử lý hình ảnh, chúng ta sẽ gặp phải vấn đề về ảnh hưởng của ánh sáng tác động từ môi trường vào vì không gian màu RGB xác định màu sắc dựa vào sự kết hợp của các màu cơ bản, do đó sẽ rất khó để có thể xác định được màu sắc trong môi trường có ánh sáng thay đổi liên tục. Vì việc mô tả được màu sắc đóng vai trò quan trọng trong việc xử lý hình ảnh nên không gian màu HSV thường được ưu tiên sử dụng hơn không gian màu RGB vì không gian màu HSV mô tả các màu sắc gần giống với cách con người cảm nhận về màu sắc.

Sau đây là cách thay đổi không gian màu RGB sang không gian màu HSV

Trong trường hợp hình 8 bit và 16 bit, R, G và B được chuyển đổi sang các điểm số thực và được tỉ lệ trong khoảng từ 0 đến 1.

$$V = \max(R, G, B) \quad (2.9)$$

$$S = \begin{cases} \frac{V - \min(R, G, B)}{V} & \text{nếu } V \neq 0 \\ 0 & \text{nếu } V = 0 \end{cases} \quad (2.10)$$

$$H = \begin{cases} \frac{60(G - B)}{V - \min(R, G, B)} & \text{nếu } V = R \\ 120 + \frac{60(B - R)}{V - \min(R, G, B)} & \text{nếu } V = G \\ 240 + \frac{60(R - G)}{(V - \min(R, G, B))} & \text{nếu } V = B \end{cases} \quad (2.11)$$

Nếu $H < 0$ thì $H = H + 360$.

Đầu ra sẽ là $0 \leq V \leq 1, 0 \leq S \leq 1, 0 \leq H \leq 360$

Kết quả:



(a)



(b)

Hình 2.17: Chuyển không gian màu RGB sang không gian màu HSV

(a) Ảnh trong không gian màu RGB; (b) Ảnh trong không gian màu HSV

Hình 2.17a là hình ảnh thể hiện trong không gian màu RGB, trong không gian màu này, màu sắc được thể hiện rất rõ ràng và sống động, sau khi ra chuyển từ không gian màu RGB sang không gian màu HSV (H.2.17b), hình ảnh được thể hiện theo các khoảng màu sắc và các vùng sáng tối một cách rõ nét, nhờ đó mà việc xử lý hình ảnh sẽ dễ dàng hơn.

2.4.3 Phương pháp lấy ngưỡng trong không gian màu HSV

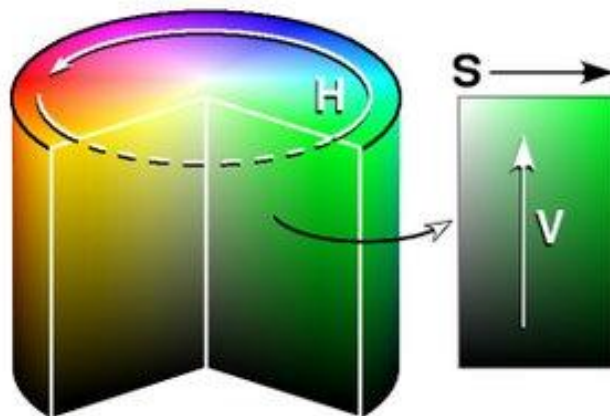
Sau khi chuyển đổi sang không gian màu HSV, hình ảnh sẽ được xử lý lấy ngưỡng để xác định đối tượng cần bám theo. Trong không gian màu HSV, mỗi giá trị H, S, V là một giá trị được xác định riêng biệt và không bị phụ thuộc vào điều kiện môi trường như ánh sáng, nên đây là một không gian màu được sử dụng nhiều nhất để lọc và lấy ngưỡng đối tượng ra khỏi nền [13].

Đầu tiên ta lấy khoảng giá trị H là màu của đối tượng cần tách ra khỏi nền.

Sau đó ta lấy khoảng giá trị S là độ bão hòa của màu của đối tượng.

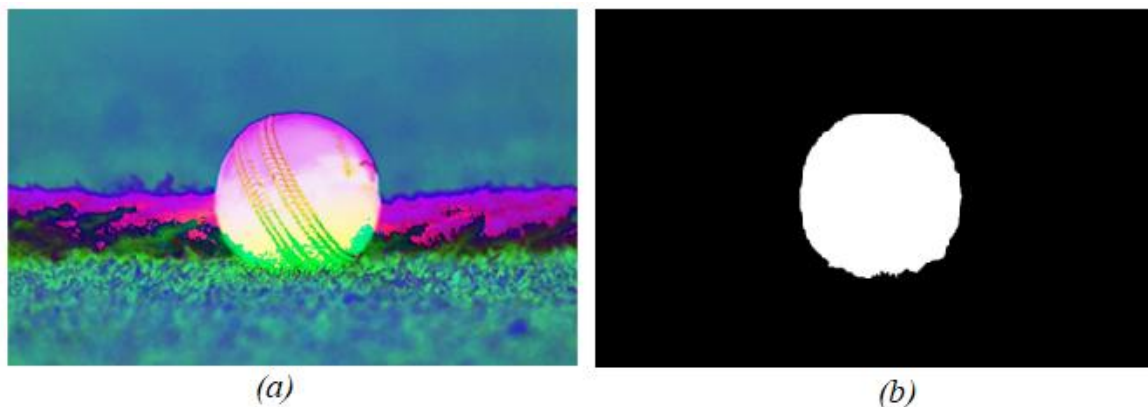
Cuối cùng là giá trị V, là độ sáng của màu đó.

Với phương pháp này, ta có thể lọc được màu của đối tượng mà không lo ảnh hưởng bởi ánh sáng đến từ môi trường bên ngoài tác động vào.



Hình 2.18: Ảnh biểu diễn các giá trị HSV

Hình ảnh trong không gian màu HSV được thể hiện rõ nét trong hình ảnh trên (H.2.18), chọn màu trong khoảng giá trị H và độ sáng tối trong giá trị S và V giúp thị giác máy tính dễ dàng hiểu được đối tượng.



Hình 2.19: Lấy ngưỡng trong không gian màu HSV

(a) Ảnh trong không gian màu HSV; (b) Ảnh sau khi lấy ngưỡng

Hình 2.19a là ảnh trong không gian màu HSV, sau khi chọn các giá trị H, S, V, ta được ảnh sau khi lấy ngưỡng là ảnh nhị phân trắng đen (H.2.19b). Với đối tượng chúng ta tách ra sẽ là màu trắng (giá trị 1), các màu khác là nền có màu đen (giá trị 0).

2.4.4 Các thuật toán xử lý hình thái học (Morphology) [14]

Trong các ứng dụng thị giác máy tính, các phép xử lý hình thái học được sử dụng để nhận dạng đối tượng, nâng cao chất lượng ảnh, phân đoạn ảnh và kiểm tra khuyết điểm trên ảnh, được sử dụng rất nhiều để giảm các lỗi trong quá trình nhận dạng. Đây là một trong những kỹ thuật được áp dụng trong giai đoạn tiền xử lý. Các phép toán thường dùng là phép giãn nở (Dilation) và phép co (Erosion). Từ hai phép toán cơ bản này người ta phát triển thành một số phép toán như phép đóng (Closing) và phép mở (Opening).

Các phép biến đổi hình thái học cần 2 đầu vào, một là hình ảnh ban đầu đã được xử lý thành ảnh nhị phân, hai là phần tử cấu trúc hay còn được gọi là kernel, nó giống như một chiếc mặt nạ sẽ quyết định bản chất của quá trình.

2.4.4.1 Phép co ảnh (Erosion)

Phép co ảnh là phép thu nhỏ hình ảnh lại dựa trên phần tử cấu trúc của nó. Nó là kết quả của phép AND giữa hình ảnh và phần tử cấu trúc, khi mà các pixel

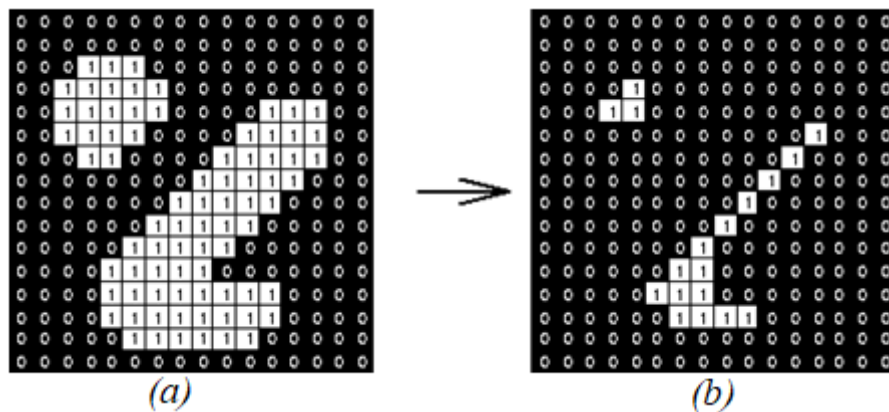
của hình ảnh và phần tử cấu trúc đều bằng 1 thì kết quả là chúng ta sẽ được 1 pixel là 1, còn trường hợp ngược lại là 0. Nhờ đó phép co ảnh làm cho đối tượng trong ảnh bị giảm pixel hay đơn giản là vùng pixel bằng 1 sẽ ít đi. Phép co ảnh thường được dùng để loại bỏ các vùng bị nhiễu, tách các đối tượng nằm đè lên nhau, ...

Phép co ảnh có công thức tổng quát như sau:

$$g[x, y] = AND[W\{f[x, y]\}] = erode(f, W) \quad (2.12)$$

Với:

- $f[x, y]$ là đối tượng
- $W[x, y]$ là phần tử cấu trúc



Hình 2.20: Sử dụng phép co ảnh

(a) Ảnh chưa chỉnh sửa;

(b) Ảnh sau khi sử dụng phép co ảnh

1	1	1
1	1	1
1	1	1

Set of coordinate points =

{ (-1, -1), (0, -1), (1, -1),
 (-1, 0), (0, 0), (1, 0),
 (-1, 1), (0, 1), (1, 1) }

Hình 2.21: Phần tử cấu trúc (Structure element)

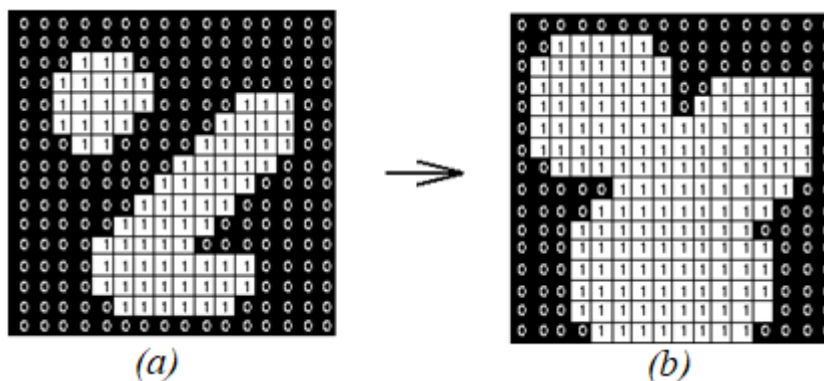
Ta thấy, dựa vào phần tử cấu trúc (H.2.21) và phép co ảnh (2.12), ảnh ban đầu (H2.20a) bị giảm bớt các phần tử màu trắng (đối tượng) và cho được ảnh sau đó là Hình 2.20b.

2.4.4.2 Phép giãn ảnh (Dilation)

Phép giãn ảnh trong xử lý ảnh là phép làm cho đối tượng ban đầu trong ảnh to lên về kích thước. Đó là kết quả của phép OR giữa các pixel ban đầu và phần tử cấu trúc, chỉ cần có 1 pixel bằng 1 giữa đối tượng trong hình và phần tử cấu trúc thì phép toán sẽ ra được pixel 1. Phép giãn ảnh thường được sử dụng sau phép co ảnh, để đối tượng sau khi bị co lại sẽ được hiển thị một cách rõ ràng.

Phép giãn ảnh được thể hiện qua công thức như sau:

$$g[x, y] = OR[W\{f[x, y]\}] = dilate(f, W) \quad (2.13)$$



(a) Ảnh chưa chỉnh sửa;

(b) Ảnh sau khi sử dụng phép giãn ảnh

Hình 2.22: Sử dụng phép giãn ảnh

1	1	1
1	1	1
1	1	1

Set of coordinate points =

{ (-1, -1), (0, -1), (1, -1),
(-1, 0), (0, 0), (1, 0),
(-1, 1), (0, 1), (1, 1) }

Hình 2.23: Cấu trúc phần tử

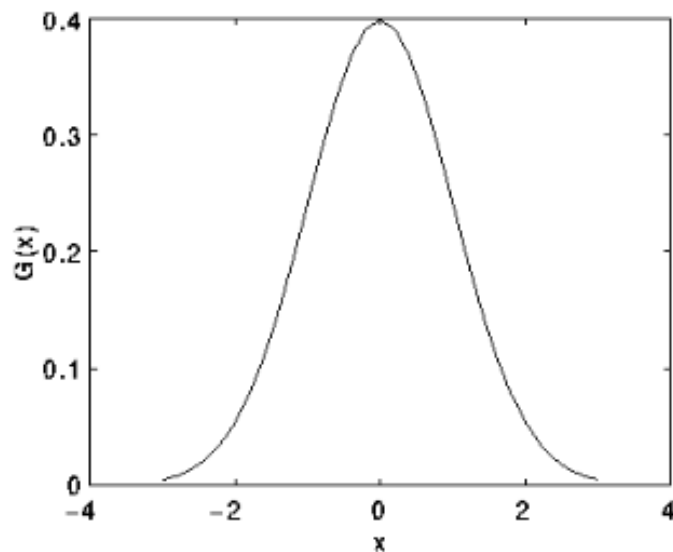
Hiệu quả của phép giãn ảnh được thể hiện trên các hình trên. Đối tượng trên hình ảnh ban đầu (H.2.22a) được giãn ra như Hình 2.22b bằng phép OR hình ban đầu với phân tử cấu trúc (H.2.23).

2.4.4.3 Bộ lọc Gaussian [15]

Gaussian blurring là cách làm mờ một ảnh bằng hàm Gaussian. Phương pháp này được ứng dụng rộng rãi và hiệu quả trong các phần mềm xử lý đồ họa. Nó cũng là công cụ phổ biến để thực hiện quá trình tiền xử lý hình ảnh dùng để làm dữ liệu đầu vào tốt cho các phân tích cao cấp hơn trong Computer Vision, hoặc cho các giải thuật được thực hiện trong một tỉ lệ khác của hình đã được cho. Như vậy, ta có thể nói Gaussian blur là một loại bộ lọc làm mờ ảnh, sử dụng lý thuyết hàm Gaussian (cũng được biết đến như là dạng phân tán chuẩn (Normal Distribution) trong thống kê) để tính toán việc chuyển đổi (Transformation) mỗi điểm ảnh của hình, giúp làm giảm nhiễu và mức độ chi tiết (không mong muốn) của hình ảnh. Đây là phương trình hàm Gaussian (Gaussian Distribution) trong không gian một chiều.

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (2.14)$$

Trong đó σ phương sai chuẩn của phân tán Gaussian.



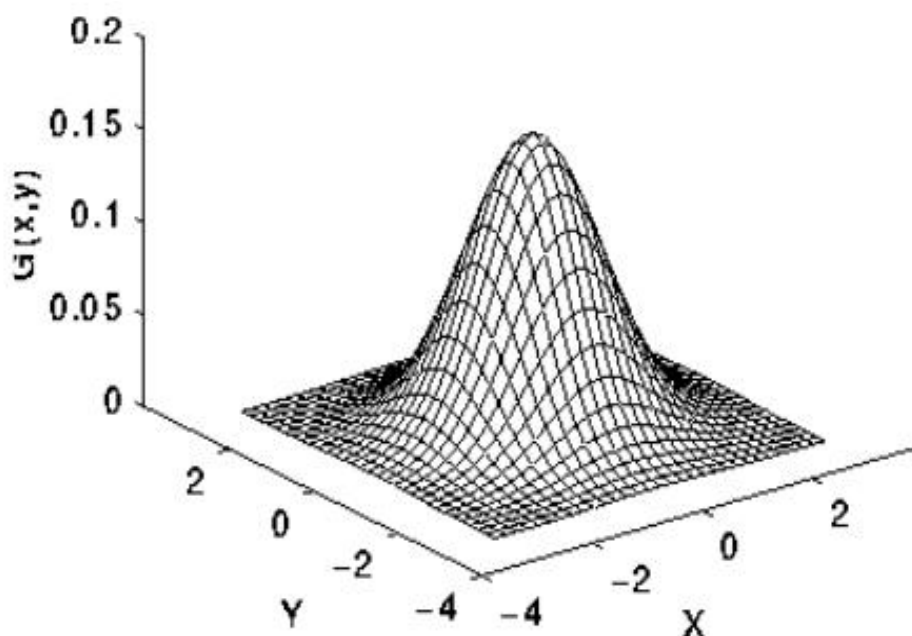
Hình 2.24: Gaussian distribution with mean 0 and $\sigma = 1$

Hàm Gaussian trong không gian một chiều được thể hiện như Hình 2.24 với điểm trung tâm là 0 và phương sai là 1.

Hàm Gaussian trong không gian hai chiều được thể hiện như sau:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.15)$$

Với x, y là tọa độ theo hai trục đứng và ngang. Trong không gian hai chiều, công thức này sản sinh ra những đường viền là những đường tròn đồng tâm, tuân theo logic phân tán Gaussian từ điểm trung tâm (H.2.25). Giá trị từ hệ thống phân tán này sẽ được sử dụng để xây dựng một ma trận tích chập dùng tính toán phép tích chập với ảnh gốc. Phần tử ở trung tâm có giá trị lớn nhất, thể hiện đặc tính của đối tượng, các phần tử phân tán ra có giá trị nhỏ dần và đồng đều nhau, giúp cho ảnh bị mờ nhưng không mất đi nét đặc trưng của nó.



Hình 2.25: Gaussian distribution with mean (0,0) and $\sigma=1$

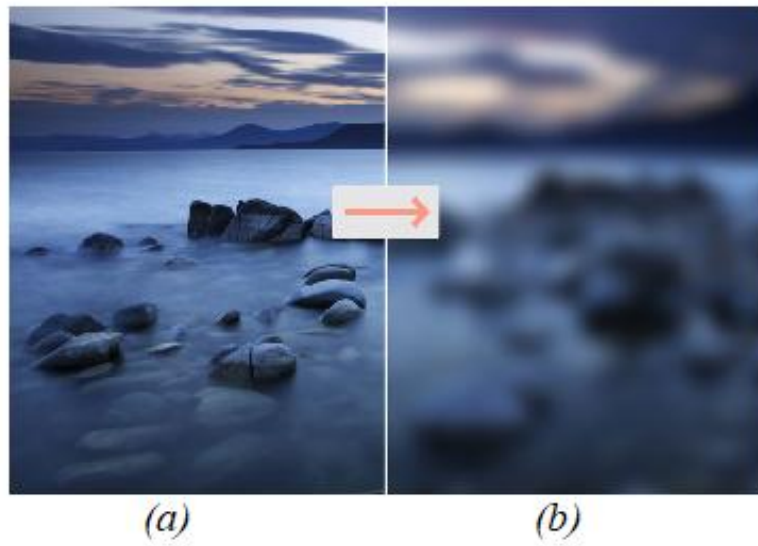
	1	4	7	4	1
	4	16	26	16	4
$\frac{1}{273}$	7	26	41	26	7
	4	16	26	16	4
	1	4	7	4	1

Hình 2.26: Tính xấp xỉ rời rạc với hàm Gaussian với $\sigma = 1.0$

Cấu trúc phần tử được xây dựng từ hàm Gaussian hai chiều được thể hiện như Hình 2.26.

Giá trị mới sau khi tính tích nhân chập với phần tử cấu trúc (kernel) đại diện cho hàm Gaussian có thể coi là trung bình lượng giá của các điểm ảnh xung quanh nó. Ta thấy rằng giá trị lượng giá của phần tử trung tâm kernel tương ứng với điểm ảnh đang xét là lớn nhất, giá trị này sẽ nhỏ hơn đối với các phần tử tương ứng với các điểm ảnh kế cận một cách đối xứng và tỉ lệ thuận với khoảng cách của phần tử này với trung tâm. Tính chất này giúp giữ lại đường viền và đường biên cũng như làm mờ một cách đồng bộ hơn so với các phương pháp khác. Trong lý thuyết, hàm Gaussian tại mỗi điểm trên hình là khác 0. Điều này có nghĩa là Gaussian kernel nên có kích thước bằng với hình ảnh và giá trị tại mỗi phần luôn khác 0. Tuy nhiên trong thực hành, do việc tính toán dựa trên xấp xỉ rời rạc (Discrete Approximation) cho nên giá trị của các phần tử trên bề mặt Gaussian ở khoảng cách lớn hơn 3σ so với trung tâm gần như không đáng kể (tiệm cận 0). Do vậy các Gaussian distribution ngoài bán kính này sẽ bị bỏ qua, đó cũng là lí do mà thông thường Gaussian kernel có kích thước giới hạn 3, 5, 7.. (Cái này tùy thuộc vào giá trị phương sai chuẩn mà bạn chọn). Khoảng cách giữa hai điểm gần nhau trong Gaussian kernel là σ . Thông thường chọn σ dựa vào kích thước ma trận mặt nạ Gaussian, ksize là kích thước ma trận.

$$\sigma = 0,3 \times [(ksize - 1) \times 0,5 - 1] + 0,8 \quad (2.16)$$



Hình 2.27: Ảnh sau khi sử dụng bộ lọc Gaussian

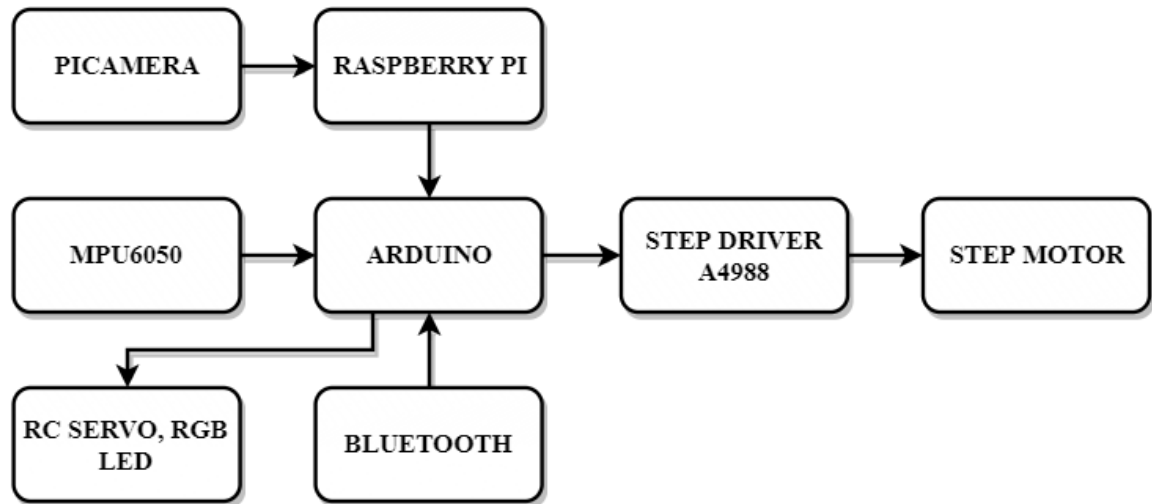
(a) Ảnh chưa chỉnh sửa;

(b) Ảnh sau khi sử dụng bộ lọc Gaussian

Hình ảnh ban đầu (H.2.27a) ta thấy khá rõ nét và không bị mờ, nhưng sau khi sử dụng bộ lọc Gaussian, ảnh sau đó (H.2.27b) ta thấy bị mờ đi thấy rõ. Nhận xét về ảnh sau khi bị làm mờ thì ta thấy ảnh mờ một cách toàn cục, các phần tử trong ảnh vẫn giữ được bản chất của nó. Lý do chúng ta thường hay làm mờ ảnh trước khi xử lý là để loại bỏ đi các pixel “gai” trên đối tượng, làm đối tượng “mượt” hơn và quá trình xử lý ảnh sẽ chính xác, dễ dàng hơn.

CHƯƠNG 3: LỰA CHỌN CÁC THIẾT BỊ PHẦN CỨNG CHO ROBOT CÂN BẰNG BẢM THEO ĐỐI TƯỢNG

3.1 SƠ ĐỒ CHUNG KẾT NỐI PHẦN CỨNG



Hình 3.1: Sơ đồ kết nối phần cứng của robot

Hình 3.1 thể hiện tổng quan về sơ đồ phần cứng của robot hai bánh tự cân bằng bảm theo đối tượng, picamera làm nhiệm vụ thu thập dữ liệu và gửi các hình ảnh đó dưới dạng ảnh màu 3 lớp trong không gian màu RGB về cho raspberry pi xử lý, raspberry pi sẽ xử lý các hình ảnh đó qua các bước như chuyển đổi không gian màu, tiền xử lý, sử dụng các thuật toán hình thái học để tách đối tượng có trong hình, xác định vị trí đối tượng và gửi thông tin vị trí đối tượng cho arduino.

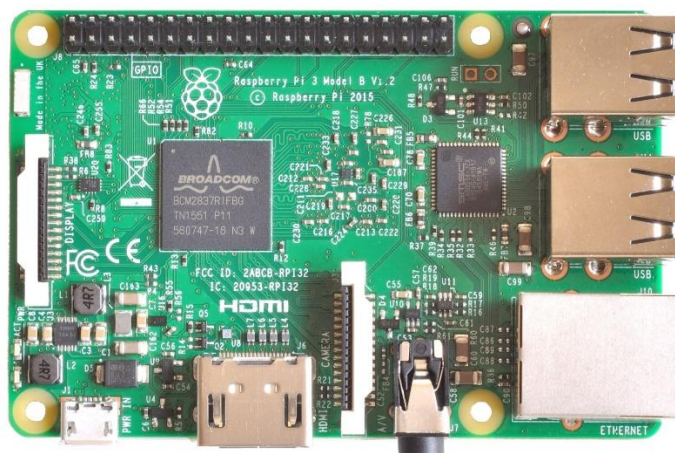
Arduino sẽ lấy tín hiệu từ cảm biến MPU6050 để xử lý và đưa tín hiệu sau khi xử lý qua step driver để điều khiển động cơ bước, với mục đích giữ cho robot cân bằng.

Tín hiệu từ raspberry và bluetooth arduino sẽ sử dụng để điều khiển rc servo, led và điều khiển robot di chuyển.

RC Servo có nhiệm vụ điều khiển vị trí của camera để camera có thể bảm theo đối tượng khi đối tượng di chuyển.

3.2 LỰA CHỌN CÁC THIẾT BỊ PHẦN CỨNG

3.2.1 Board xử lý ảnh



Hình 3.2: Board Raspberry Pi 3 Model B

Raspberry Pi (H.3.2) [16] là chiếc máy tính kích thước nhỏ được tích hợp nhiều phần cứng mạnh mẽ đủ khả năng chạy hệ điều hành và cài đặt được nhiều ứng dụng trên nó, ngoài việc sử dụng như một máy tính bình thường chạy hệ điều hành Linux hoặc Windows 10 IoT, máy còn có khả năng xuất tín hiệu ra 40 chân GPIO giúp bạn có thể giao tiếp và điều khiển vô số các board mạch và ngoại vi bên ngoài để thực hiện vô số các ứng dụng khác nhau. Với giá chỉ vài chục USD, Raspberry hiện đang là mini computer nổi bật nhất hiện nay. Ban đầu, tổ chức Raspberry Pi Foundation phát triển dự án Raspberry với mục tiêu chính là giảng dạy máy tính cho trẻ em và tạo ra một công cụ giá rẻ (chỉ vài chục USD) để sinh viên nghiên cứu học tập. Tuy nhiên, sau khi xuất hiện, Raspberry Pi được cộng đồng đánh giá cao về tính ứng dụng với phần cứng được hỗ trợ tốt, Pi đã nhanh chóng phát triển một cách rộng rãi. Pi phù hợp cho những ứng dụng cần khả năng xử lý mạnh mẽ, đa nhiệm hoặc giải trí và đặc biệt cần chi phí thấp. Hiện nay đã có hàng ngàn ứng dụng đa dạng được cài đặt trên Raspberry Pi.

Tổ chức Raspberry Pi Foundation được thành lập năm 2009, và cùng với sự giúp đỡ của 3 hãng cung cấp linh kiện điện tử lớn là EGOMAN, QSIDA,

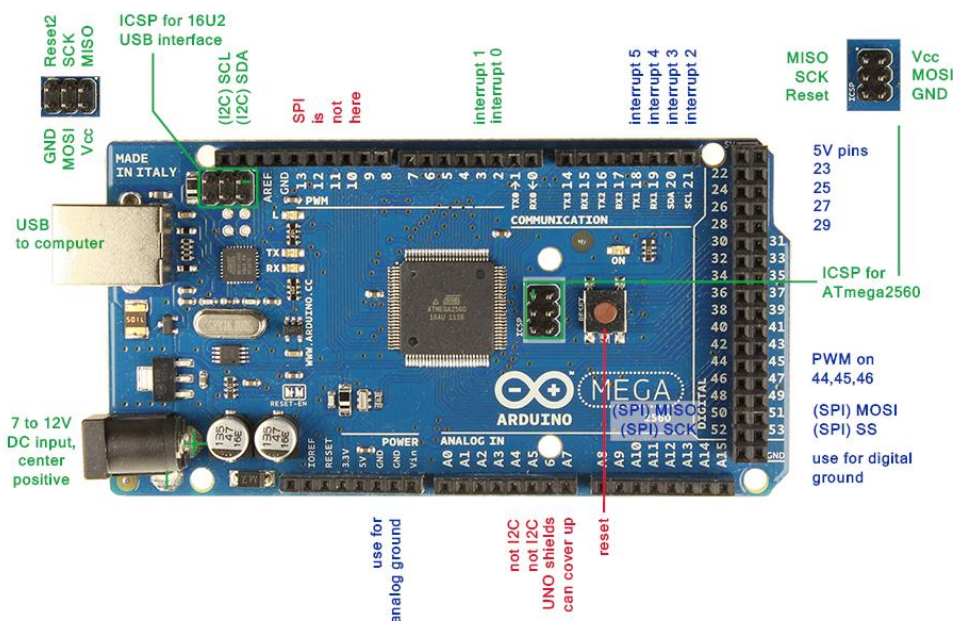
SONY phát triển lên Raspberry Pi. Trên thế giới, bạn có thể tìm mua tại những nhà phân phối của hãng như EGOMAN, ELEMENT14 hoặc RS Component.

Phiên bản Raspberry Pi đầu tiên được phát hành tháng 2 năm 2012, và tới nay đã có nhiều phiên bản khác nhau, với sự nâng cấp của phần cứng, cũng như hướng tới những mục tiêu khác nhau. Phiên bản theo thứ tự ra mắt là : Pi A → Pi A+ → Pi 1 B → Pi 1B+ → Pi 2B → Pi Zero → Pi 3B

Bảng 3.1: Thông số kỹ thuật chính của Raspberry Pi 3 Model B:

Raspberry	Pi 3 Model B
Vi xử lý	Broadcom BCM2837, ARMv8 (64bit) quad-core
GPU	Broadcom VideoCore IV, OpenGL ES 2.0, OpenVG 1080p60 , 400 MHz
Tốc độ xử lý	1.2 Ghz
Power Ratings	800mA, 5V
RAM (chia sẻ với GPU)	1GB LPDDR2 (900Mhz)
Bộ nhớ	Micro SD
GPIO	40
Kết nối	4xUSB 2.0 10/100mb Ethernet wifi 802.11 n Bluetooth 4.1 CSI, DSI
Video & audio	1080p HDMI, stereo audio 3.5mm jack
Kích thước	85x56mm

3.2.2 Vi điều khiển các thiết bị phần cứng



Hình 3.3: Board Arduino Mega 2560

Board mạch Arduino Mega 2560 (H.3.3) sử dụng chip ATmega2560 thuộc dòng vi xử lý 8bit megaAVR của Atmel [17]. Cho phép lập trình các ứng dụng điều khiển phức tạp do được trang bị cấu hình mạnh với các loại bộ nhớ ROM, RAM và Flash, các ngõ vào ra digital I/O trong đó có nhiều ngõ có khả năng xuất tín hiệu PWM, các ngõ đọc tín hiệu analog và các chuẩn giao tiếp đa dạng như UART, SPI, TWI(I2C).

Bảng 3.2: Các thông số kỹ thuật chính

Vi điều khiển	ATmega2560
Điện áp hoạt động	5V
Tốc độ xung	16MHz
Điện áp cấp	7V-12V
Điện áp cấp giới hạn	6V – 20V
Số chân digital	54 (15 PWM outputs)

Số chân analog	16
Dòng cấp trên mỗi chân I/O	20mA
Dòng cấp cho chân 3.3V	50mA
Bộ nhớ flash	256KB
SRAM	8KB
EEPROM	4KB

- Nguồn: Cấp nguồn cho board Arduino: có 1 jack để cắm nguồn từ 7-12V cho board, ngoài ra còn 1 chân Vin cũng có thể dùng để cấp nguồn cho Arduino. Nguồn từ Arduino ra: Arduino hỗ trợ 1 chân 5V và 1 chân 3.3V ra để sử dụng.

- Chân cấp xung PWM: có 15 chân Digital cấp xung PWM (chân 2-13, chân 44, 45, 46) trong đề tài dùng chân 44 và 45 cho động cơ phải và trái

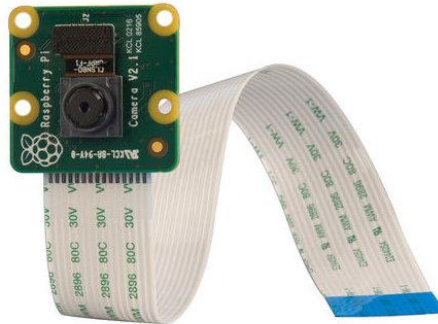
- Giao tiếp I2C: 2 chân Digital giao tiếp I2C là 20 (SDA) và 21 (SCL).

- Giao tiếp Serial: hỗ trợ đến 4 cổng Serial 0, 1, 2, 3 (Thứ tự các chân là 0, 1, 19, 18, 17, 16, 15, 14). Ở đây ta dùng Serial 0 để giao tiếp với module Bluetooth HC05.

- Ngắt: có 6 chân ngắt ngoài 0, 1, 2, 3, 4, 5 (Thứ tự chân là 2, 3, 21, 20, 19, 18).

- Chip Atmega2560 có 5 timer 16bit là timer 0, timer 1, timer 3, timer 4, timer 5, ở đây chúng em sử dụng 2 timer là timer 3 và 5 để tạo xung điều khiển động cơ, timer 1 để điều xung PWM điều khiển 2 servo, timer 0 để chạy xung vòng loop, timer 1 để ngắt MPU6050.

3.2.3 Camera Raspberry Pi V2 8MP



Hình 3.4: Camera Raspberry Pi V2 8MP

Camera Raspberry Pi V2 8MP (H.3.4) là phiên bản Camera Module dành cho Raspberry Pi mới nhất sử dụng cảm biến ảnh IMX219 8-megapixel từ Sony thay cho cảm biến cũ là OV5647. Với cảm biến IMX219 8-megapixel từ Sony, Camera Module cho Raspberry Pi đã có được sự nâng cấp vượt trội về cả chất lượng hình ảnh, video cũng như độ bền.

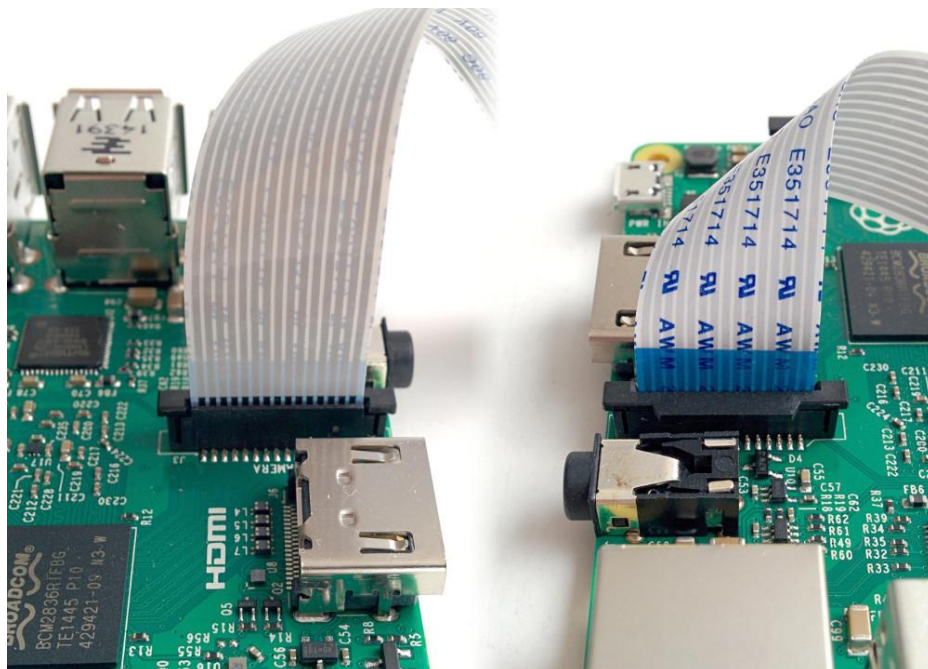
Camera Raspberry Pi V2 8MP có thể sử dụng với Raspberry Pi để chụp hình, quay phim với chất lượng HD 1080p30, 720p60 hoặc VGA90, cách sử dụng cũng như lập trình với Camera Module trên Raspberry Pi cũng rất dễ dàng, chỉ cần cắm vào cổng Camera trên Raspberry Pi và Config 1 chút là có thể dùng được.

Camera Raspberry Pi V2 8MP có thể sử dụng tương thích với tất cả các dòng Raspberry Pi 1, 2 và 3 từ trước đến nay, có thể điều khiển thông qua MMAL và V4L APIs, có rất nhiều bộ thư viện được cộng đồng Raspberry Pi phát triển trên Python giúp cho việc tìm hiểu và sử dụng trở nên dễ dàng hơn rất nhiều.

Bảng 3.3: Thông số kỹ thuật

Cảm biến IMX219	Sony
Số điểm ảnh	8MP
Lens	Fixed focus
Độ phân giải của camera	3280 x 2464 pixel stills
Độ phân giải video	HD 1080p30, 720p60 và 640 x 480p90
Kích thước	25 x 23 x 9 mm
Cân nặng	~3g
Cổng kết nối	Ribbon
Giao tiếp	CSI

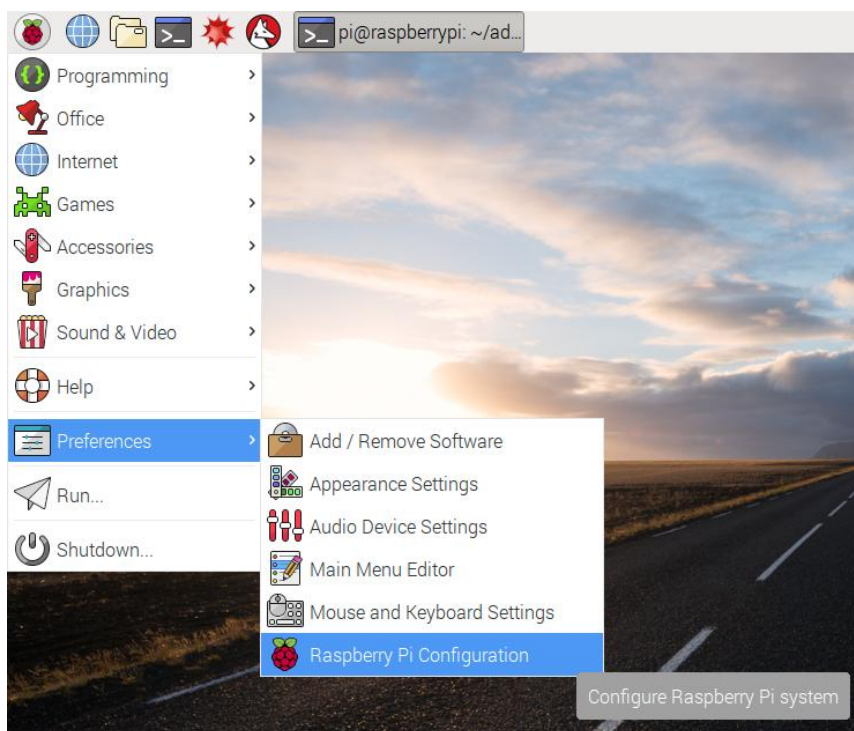
Kết nối picamera với raspberry pi



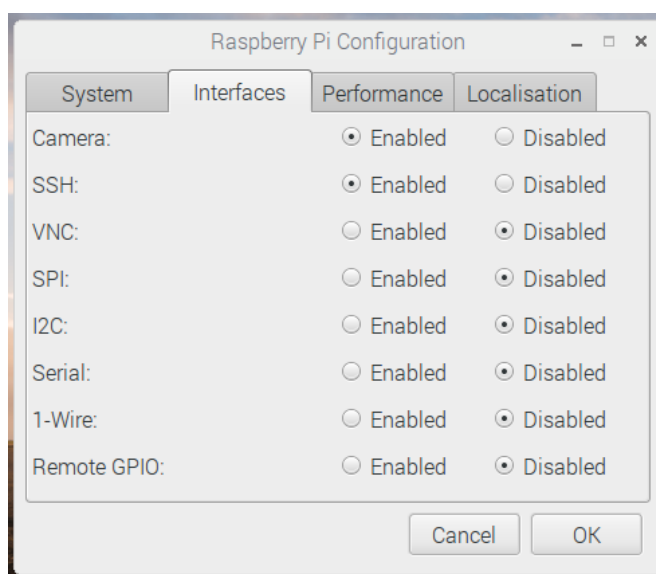
Hình 3.5: Kết nối picamera với raspberry pi

Ta dùng cáp FFC/FPC 15 pin ngược chiều để kết nối picamera với raspberry pi qua khe cắm cáp camera được thiết kế sẵn trên board raspberry pi như Hình 3.5.

Sau khi kết nối picamera vào raspberry pi, bước tiếp theo là chúng ta phải thiết lập kết nối camera phần mềm trong raspberry pi.



Hình 3.6: Cấu hình raspberry



Hình 3.7: Enabled Camera

Ta nháy chuột vào biểu tượng trái dâu và chọn thanh Preferences như, ta vào cấu hình raspberry pi bằng cách chọn Raspberry Pi Configuration như Hình 3.6.

Ta chọn mở (Enabled) các chế độ Camera và SSH trong cửa sổ thiết lập thông số raspberry pi để có thể sử dụng picamera như Hình 3.7.

Nếu mà vẫn chưa thấy camera hoạt động, chúng ta phải enable camera lại và khởi động lại raspberry pi.

3.2.4 Cảm biến góc nghiêng MPU6050

Cảm biến MPU6050 (H.3.8) là một trong những loại cảm biến IMU (Inertial Measurement Unit) – thiết bị kết hợp từ hai bộ cảm biến Accelerometer (cảm biến gia tốc) và cảm biến Gyroscope (cảm biến con quay hồi chuyển). Ngày nay, IMU là một trong những cảm biến được sử dụng phổ biến trong các thiết bị điện tử. Chúng có thể được tìm thấy trong các smartphone, điều khiển cầm tay, các thiết bị điện tử đeo trên người... Cảm biến IMU giúp chúng ta xác định được trạng thái của một vật được gắn liền với cảm biến theo không gian ba chiều.

MPU6050 là cảm biến 6 bậc tự do (6 DOF) hay cảm biến IMU 6 trục mà đầu ra cảm biến cho ta 6 giá trị. Ba giá trị từ cảm biến gia tốc, ba giá trị từ cảm biến vận tốc. MPU6050 được chế tạo dựa trên MEMS (Micro Electro Mechanical Systems) – công nghệ vi cơ điện tử. Cả hai bộ cảm biến gia tốc góc và vận tốc góc được tích hợp trong một con chip sử dụng phương thức I2C để giao tiếp với các thiết bị khác.

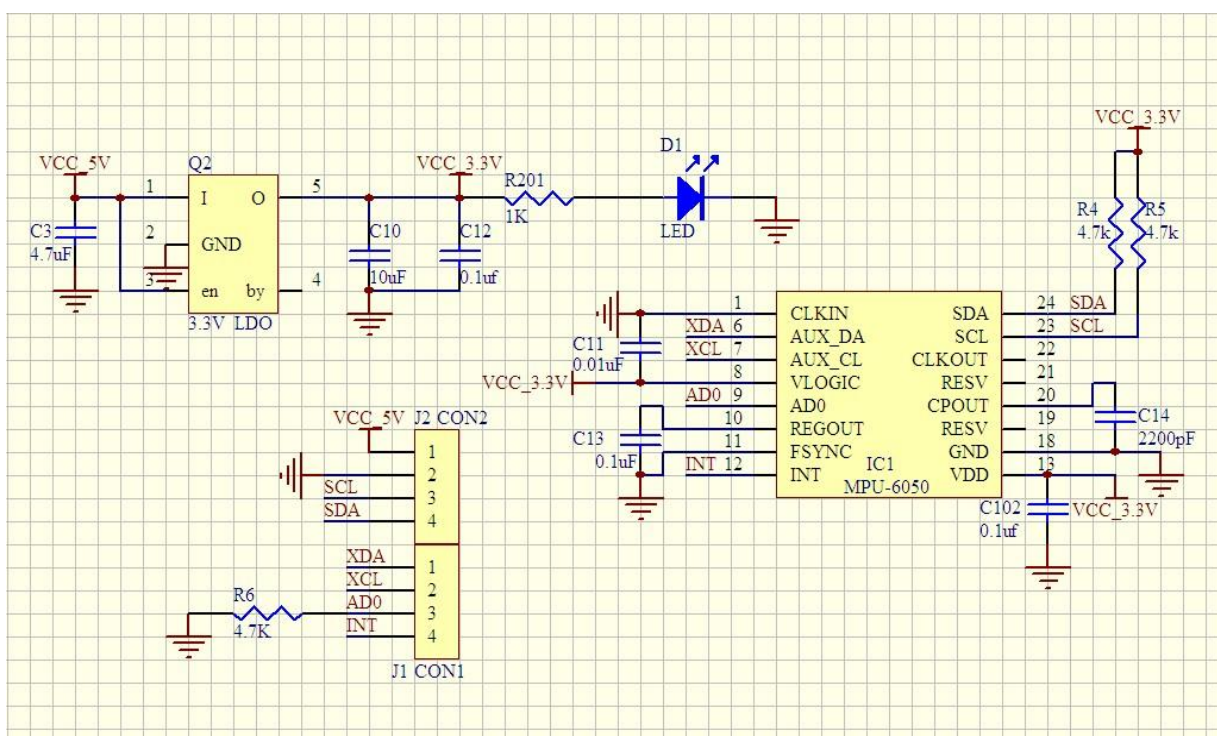


Hình 3.8: Cảm biến góc nghiêng MPU6050

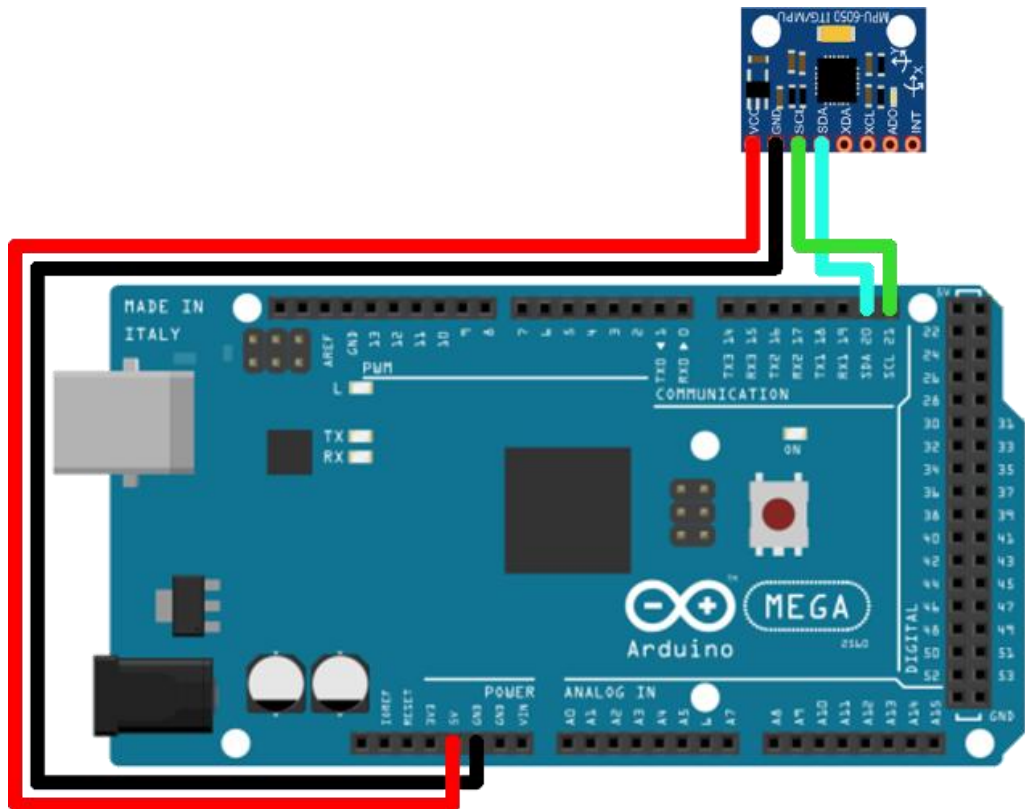
Trong đề tài này, cảm biến MPU6050 được sử dụng để đọc giá trị gia tốc và vận tốc, từ đó tính ra giá trị góc nghiêng của robot so với mặt đất để đưa vào đầu vào của bộ điều khiển PID.

Bảng 3.4: Thông số kỹ thuật cảm biến MPU6050

Cảm biến	MPU6050
Điện áp sử dụng	3 ~ 5 VDC
Điện áp giao tiếp	3 ~ 5 VDC
Chuẩn giao tiếp	I2C
Giá trị Gyroscope trong khoảng	+/- 250 500 1000 2000 degree/sec
Giá trị Accelerometer trong khoảng	+/- 2g, +/- 4g, +/- 8g, +/- 16g



Hình 3.9: Sơ đồ chân ra của cảm biến MPU6050



Hình 3.10: Kết nối cảm biến MPU6050 với Arduino Mega2560

Ta kết nối cảm biến MPU6050 như Hình 3.10, cấp nguồn cho cảm biến MPU6050 qua 2 chân VCC và GND, SCL và SDA của cảm biến nối vào 2 chân SCL và SDA của board arduino.

Các chân SCL, SDA lần lượt ra 2 chân cấp xung và chân dữ liệu theo chuẩn I2C. Chân XDA và XDL dùng để giao tiếp với những cảm biến khác, chân INT là chân ngắt.

Trong bài chỉ sử dụng địa chỉ mặc định I2C để lấy dữ liệu và không sử dụng ngắt ngoài hay các cảm biến mở rộng khác nên các chân XDA, XCL, INT, AD0 sẽ không được sử dụng.

Trong giao tiếp I2C MPU6050 với arduino, arduino là master và MPU6050 ở chế độ slave, tốc độ truyền cao nhất là 400 Khz.

Để lập trình giao tiếp I2C, chúng ta sử dụng thư viện Wire.h trong Arduino cung cấp.

Cấu hình thanh ghi cảm biến MPU6050:

Thanh ghi WHO_AM_I:

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
75	117	-	WHO_AM_I[6:1]						-

Thanh ghi này được dùng để xác minh danh tính của thiết bị. Nội dung trong thanh ghi này chiếm 6 bit trên thanh ghi địa chỉ của MPU6050. Bit thấp nhất được xác định bởi giá trị của chân AD0. Nếu chân AD0 được nối với 5V, thanh ghi địa chỉ sẽ là 0x69.

Thanh ghi Power Management: thanh ghi quản lý năng lượng

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
6B	107	DEVICE_RESET	SLEEP	CYCLE	-	TEMP_DIS	CLKSEL[2:0]		

Thanh ghi này cho phép người dùng cấu hình nguồn và xung clock. Chế độ clock mặc định là 8MHz với dao động của thanh anh trong.

- DEVICE_RESET: Khi set lên 1, bit này sẽ reset tất cả các thanh ghi nội về giá trị mặc định. Bit sẽ tự động xóa về 0 một lần khi quá trình reset hoàn tất.
- SLEEP: khi set bit này lên 1, MPU sẽ vào chế độ sleep.
- CYCLE: khi bit này được set lên 1 và SLEEP bị vô hiệu hóa, cảm biến MPU6050 sẽ hoạt động 1 chu kỳ giữa chế độ sleep và waking để lấy giá trị mẫu của cảm biến hoạt động dưới tần số của bit LP_WAKE_CTRL.
- TEMP_DIS: khi bit này hoạt động, cảm biến đo nhiệt độ bị vô hiệu hóa.
- CLKSEL: 3 bit chưa có giá trị được ghi, xác định nguồn xung của thiết bị.

Thanh ghi Configuration

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1A	26	-	-	EXT_SYNC_SET[2:0]			DLPF_CFG[2:0]		

Thanh ghi này cấu hình đồng bộ hóa cho các cảm biến mở rộng bên ngoài và cấu hình bộ lọc thông thấp số (Digital Low Pass Filter – DLPF) cho cảm biến gyroscope và accelerometer.

DLPF_CFG	Accelerometer (F _s = 1kHz)		Gyroscope		
	Bandwidth (Hz)	Delay (ms)	Bandwidth (Hz)	Delay (ms)	Fs (kHz)
0	260	0	256	0.98	8
1	184	2.0	188	1.9	1
2	94	3.0	98	2.8	1
3	44	4.9	42	4.8	1
4	21	8.5	20	8.3	1
5	10	13.8	10	13.4	1
6	5	19.0	5	18.6	1
7	RESERVED		RESERVED		8

Thanh ghi Gyroscope Configuration:

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1B	27	XG_ST	YG_ST	ZG_ST	FS_SEL[1:0]		-	-	-

Thanh ghi này được dùng để kích hoạt gyroscope self-test và cấu hình phạm vi scale của gyroscope.

FS_SEL	Full Scale Range
0	± 250 °/s
1	± 500 °/s
2	± 1000 °/s
3	± 2000 °/s

Gyroscope self-test cho phép người dùng kiểm tra các phần cơ khí và điện của con quay hồi chuyển trong cảm biến. Việc tự kiểm tra cho mỗi trục con quay có thể được kích hoạt bằng các bit XG_ST, YG_ST và ZG_ST của thanh ghi.

Thanh ghi Accelerometer Configuration

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1C	28	XA_ST	YA_ST	ZA_ST	AFS_SEL[1:0]		-		

Thanh ghi này được dùng để kích hoạt chế độ self-test của accelerometer và cấu hình phạm vi đo gia tốc của cảm biến accelerometer.

AFS_SEL	Full Scale Range
0	$\pm 2g$
1	$\pm 4g$
2	$\pm 8g$
3	$\pm 16g$

Việc tự kiểm tra cho mỗi trục gia tốc có thể được kích hoạt bằng cách kiểm soát các bit XA_ST, YA_ST và ZA_ST của thanh ghi này. Tự kiểm tra cho mỗi trục có thể được thực hiện độc lập hoặc tất cả cùng một lúc.

Thanh ghi Sample Rate Divider

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
19	25	SMPLRT_DIV[7:0]							

Thanh ghi này chỉ định dải phân chia tốc độ đầu ra con quay hồi chuyển để tạo ra tốc độ lấy mẫu cho MPU6050. Tốc độ lấy mẫu được tạo ra bằng cách chia tỷ lệ đầu ra con quay hồi chuyển của SMPLRT_DIV: Tốc độ lấy mẫu = Tốc độ đầu ra con quay hồi chuyển / (1 + SMPLRT_DIV).

Thanh ghi Interrupt Enable:

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
38	56		-		FIFO_OFLOW_EN	I2C_MST_INT_EN	-	-	DATA_RDY_EN

Thanh ghi này cho phép tạo ngắt từ các nguồn ngắt.

FIFO_OFLOW_EN: bit này cho phép tràn bộ đệm FIFO để tạo ra ngắt.

I2C_MST_INT_EN: bit này cho phép tạo ngắt từ master I2C.

DATA_RDY_EN: bit này cho phép ngắt Data Ready, ngắt này sẽ xuất hiện sau khi các thanh ghi được ghi xong.

Thanh ghi Accelerometer Measurements:

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
3B	59	ACCEL_XOUT[15:8]							
3C	60	ACCEL_XOUT[7:0]							
3D	61	ACCEL_YOUT[15:8]							
3E	62	ACCEL_YOUT[7:0]							
3F	63	ACCEL_ZOUT[15:8]							
40	64	ACCEL_ZOUT[7:0]							

Thanh ghi này chứa các giá trị mới nhất của cảm biến gia tốc góc. Ta thấy các giá trị đó được chứa vào các thanh ghi như bảng trên.

Thanh ghi Gyroscope Measurements

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
43	67	GYRO_XOUT[15:8]							
44	68	GYRO_XOUT[7:0]							
45	69	GYRO_YOUT[15:8]							
46	70	GYRO_YOUT[7:0]							
47	71	GYRO_ZOUT[15:8]							
48	72	GYRO_ZOUT[7:0]							

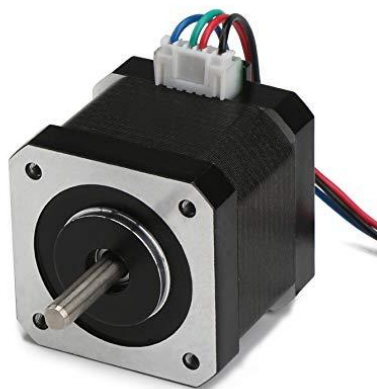
Thanh ghi này chứa các giá trị gần nhất của cảm biến gyroscope.

Tính giá trị góc từ cảm biến gia tốc và vận tốc góc:

Góc từ gia tốc góc: $\text{góc} = \arctan(\text{accX} / \text{accZ})$ (rad)

Góc từ vận tốc góc: ta lấy giá trị gyro chia cho giá trị 65.5 LSB/ deg/ s.

3.2.5 Động cơ bước



Hình 3.11: Động cơ bước

Động cơ bước (H.3.11) thực chất là một động cơ đồng bộ dùng để biến đổi các tín hiệu điều khiển dưới dạng các xung điện rời rạc kế tiếp nhau thành các chuyển động góc quay hoặc các chuyển động của roto và có khả năng cố định roto vào những vị trí cần thiết. Động cơ bước làm việc được là nhờ có bộ chuyển mạch điện tử đưa các tín hiệu điều khiển vào stato theo một thứ tự và một tần số nhất định. Tổng số góc quay của roto tương ứng với số lần chuyển mạch, cũng như chiều quay và tốc độ quay của roto, phụ thuộc vào thứ tự chuyển đổi và tần số chuyển đổi. Khi một xung điện áp đặt vào cuộn dây stato (phần ứng) của động cơ bước thì roto (phần cảm) của động cơ sẽ quay đi một góc nhất định, góc ấy là một bước quay của động cơ. Khi các xung điện áp đặt vào các cuộn dây phần ứng thay đổi liên tục thì roto sẽ quay liên tục. (Nhưng thực chất chuyển động đó vẫn là theo các bước rời rạc).

Về cách hoạt động của động cơ bước là động cơ bước không quay theo cơ chế thông thường, chúng quay theo từng bước nên có độ chính xác rất cao về mặt điều khiển học. Chúng làm việc nhờ các bộ chuyển mạch điện tử đưa các tín hiệu điều khiển vào stato theo thứ tự và một tần số nhất định. Tổng số góc quay của rôto tương ứng với số lần chuyển mạch, cũng như chiều quay và tốc độ quay của rôto phụ thuộc vào thứ tự chuyển đổi và tần số chuyển đổi.

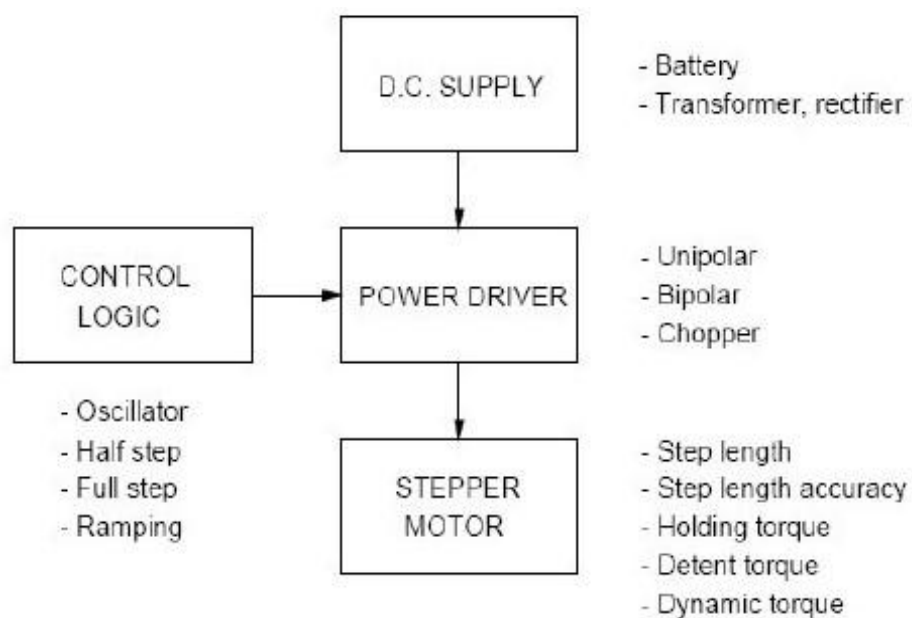
Động cơ bước có 5 đặc tính cơ bản sau:

- Brushless (không chổi than): STEP là loại động cơ không chổi than.
- Load Independent (độc lập với tải): động cơ bước quay với tốc độ ổn định trong tầm moment của động cơ.
- Open loop positioning (điều khiển vị trí vòng hở): thông thường chúng ta có thể đếm xung kích ở động cơ để xác định vị trí mà không cần phải có cảm biến hồi tiếp vị trí, nhưng đôi khi trong những ứng dụng đòi hỏi tính chính xác cao STEP thường được sử dụng kết hợp với các cảm biến vị trí như: encoder, biến trở ...
- Holding Torque (moment giữ): STEP có thể giữ được trục quay của nó, so với động cơ DC không có hộp số thì moment giữ của STEP lớn hơn rất nhiều.

- Excellent Response (Đáp ứng tốt): STEP đáp ứng tốt khi khởi động, dừng lại và đảo chiều quay một cách dễ dàng.

Bảng 3.5: Thông số cơ bản của động cơ bước

Động cơ	Nema 17 42SHDC3025-24B
Điện áp hoạt động	3.96 VDC
Dòng hoạt động	1.68 A
Tốc độ quay	1000 rpm
Momen xoắn	4.0 kg.cm
Góc bước	1.8 độ
Nhiệt độ cao nhất	80 độ C
Độ chính xác vị trí	5%
Kích thước	42 x 42 x 40 mm



Hình 3.12: Sơ đồ nguyên lý hoạt động của động cơ bước

Giải thích sơ đồ nguyên lý Hình 3.12:

D.C.SUPPLY: Có nhiệm vụ cung cấp nguồn một chiều cho hệ thống. Nguồn một chiều này có thể lấy từ pin nếu động cơ có công suất nhỏ. Với các động cơ có công suất lớn có thể dùng nguồn điện được chỉnh lưu từ nguồn xoay chiều.

CONTROL LOGIC: Đây là khối điều khiển logic. Có nhiệm vụ tạo ra tín hiệu điều khiển động cơ. Khối logic này có thể là một nguồn xung, hoặc có thể là một hệ thống mạch điện tử. Nó tạo ra các xung điều khiển. Động cơ bước có thể điều khiển theo cả bước hoặc theo nửa bước.

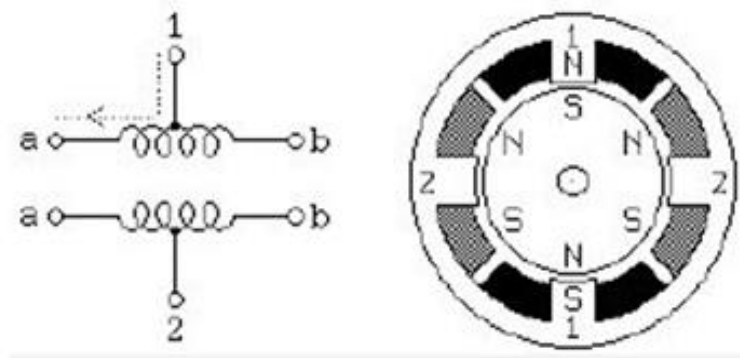
POWER DRIVER: Có nhiệm vụ cấp nguồn điện đã được điều chỉnh để đưa vào động cơ. Nó lấy điện từ nguồn cung cấp và xung điều khiển từ khối điều khiển để tạo ra dòng điện cấp cho động cơ hoạt động.

STEPPER MOTOR: Động cơ bước. Các thông số của động cơ gồm có: Bước góc, sai số bước góc, mômen kéo, mômen hãm, mômen làm việc. Đối với hệ điều khiển động cơ bước, ta thấy đó là một hệ thống khá đơn giản vì không hề có phản tử phản hồi. Điều này có được vì động cơ bước trong quá trình hoạt động không gây ra sai số tích lũy, sai số của động cơ do sai số trong khi chế tạo. Việc sử dụng động cơ bước tuy đem lại độ chính xác chưa cao nhưng ngày càng được sử dụng phổ biến. Vì công suất và độ chính xác của bước góc đang ngày càng được cải thiện.

Bảng 3.6: Tiêu chuẩn về Bước Góc của động cơ bước:

Step angle	Steps per revolution
0.9°	400
1.8°	200
3.6°	100
3.75°	96
7.5°	48
15.0°	24

Bảng 3.6 là một số tiêu chuẩn về góc bước của các động cơ bước phổ biến, mà phổ biến nhất là 1.8 độ.



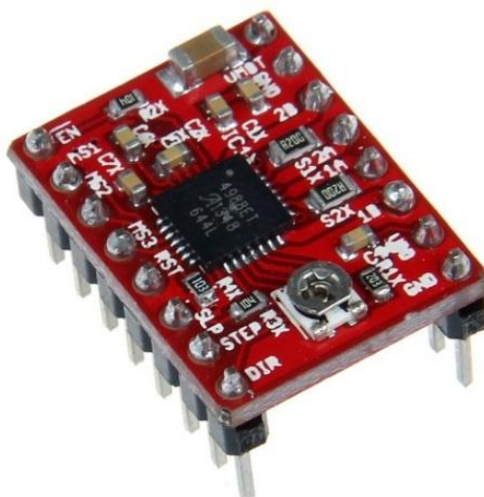
Hình 3.13: Cấu tạo của động cơ bước

Động cơ bước có cấu tạo như Hình 3.13, có các cuộn dây gần giống như là một động cơ đồng bộ có 2 cuộn dây.

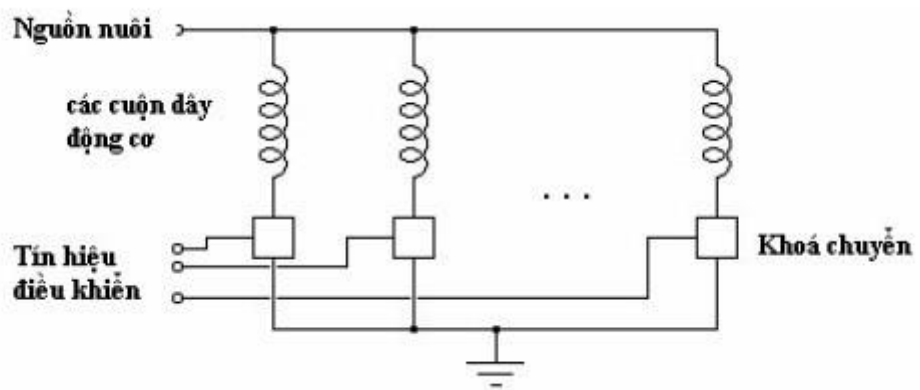
3.2.6 Mạch điều khiển động cơ bước A4899

Mạch điều khiển động cơ bước (H3.14) bao gồm một số chức năng sau đây:

- Tạo các xung với những tần số khác nhau.
- Chuyển đổi các phần cho phù hợp với thứ tự kích từ.
- Làm giảm các dao động cơ học.



Hình 3.14: Mạch điều khiển động cơ bước A4988



Hình 3.15: Sơ đồ kết nối mạch điều khiển động cơ bước

Hình 3.15 là sơ đồ chung của các mạch điều khiển động cơ bước. Đầu vào của mạch điều khiển là các xung. Thành phần của mạch là các bán dẫn, vi mạch. Kích thích các phần của động cơ bước theo thứ tự 1-2-3-4 do các transistor công suất T1 đến T4 thực hiện. Với việc thay đổi vị trí bộ chuyển mạch, động cơ có thể quay theo chiều kim đồng hồ hoặc ngược lại. Dưới đây là sơ đồ kết nối mạch điều khiển động cơ bước.

Điện áp được cấp qua các khoá chuyển để nuôi các cuộn dây, tạo ra từ trường làm quay rotor. Các khoá ở đây không cụ thể, có thể là bất cứ thiết bị đóng cắt nào điều khiển được như rơle, transistor công suất... Tín hiệu điều khiển có thể được đưa ra từ bộ điều khiển như vi mạch chuyên dụng, máy tính

Động cơ bước có nhiều cách điều khiển. Có thể điều khiển các dây trực tiếp qua 4 cổng qua MCU thông qua driver đệm công suất. cách này hơi phức tạp một chút, cần phải hiểu rõ bên trong động cơ và thường chỉ điều khiển được full bước

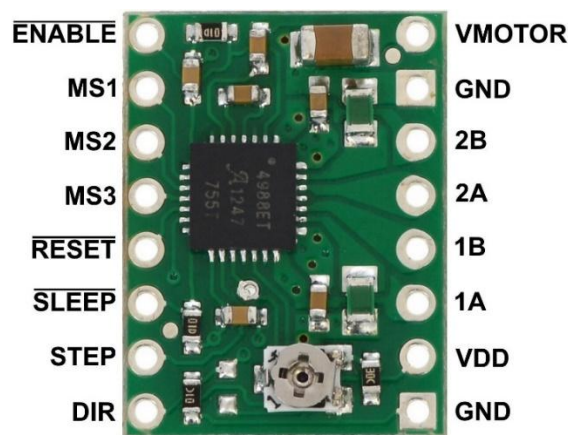
Cách thông dụng nhất là dùng các IC chuyên dụng điều khiển động cơ bước. Các IC hay gặp nhất là TB6560, TB6600, L297, A4988, DRV8825, MA860H... Việc lựa chọn dùng loại driver nào phụ thuộc vào loại động cơ và công suất động cơ định điều khiển. Ví dụ như A4988, DRV8825 dùng để điều khiển các loại động cơ nhỏ có công suất bé ví dụ như trong máy photo hoặc máy in 3d, TB6560 hoặc TB6600 thì lại dùng để điều khiển các loại động cơ lớn hơn một chút ví dụ như

trong các loại máy cnc mini. Các loại động cơ to thì người ta hay dùng MA860H hoặc các driver khủng hơn nữa.

Nhìn chung cách giao tiếp với các module Driver này tương đối giống nhau. Chúng đều có 3 port cơ bản là DIR (để điều khiển hướng quay động cơ), EN (để điều khiển bật tắt động cơ), CLK (xuất xung để dịch chuyển từng step). Một số loại module như tb6560 tb6600 hoặc MA860H thì mỗi port đều có 2 pin. Ví dụ như EN+ EN- CW+ CW- CLK+ CLK-. Vì thế người dùng có thể tùy chọn điều khiển theo mức 0 hoặc mức 1. Mọi người thường kéo 3 chân xuống thấp và điều khiển bằng 3 chân còn lại.

Điều quan trọng nhất là các module này là chúng có thể điều khiển được vi bước như 1/16 step, 1/8 step, 1/2 step và full step. nếu full step thì bạn cần 200 step để quay hết 1 vòng đối với loại 1.8 độ 1 step. Còn nếu dùng chế độ vi bước 1/16 step thì một xung động cơ chỉ dịch chuyển 1,8/16 độ. Tức là bạn phải cần $200 \times 16 = 3200$ xung để quay hết một vòng. Điều này làm tăng độ phân giải và tăng độ chính xác cho step.

Cách kết nối: với động cơ 4 dây thì chúng ta tìm các dây xem dây nào là kênh A- A+ B- B+ và nối vào driver. với động cơ 5 dây 6 dây hoặc 8 dây thì thường phải cắt bỏ hết những dây chung đi và chỉ sử dụng 4 dây ở 2 đầu cuộn dây để điều khiển.



Hình 3.16: Các chân điều khiển động cơ bước A4988

Hình 3.16 thể hiện các chân của mạch điều khiển động cơ bước A4988, sau đây là chức năng chính của các chân driver A4988:

ENABLE: kích hoạt động cơ, tích cực mức thấp.

MS1, MS2, MS3: 3 chân này dùng để chọn chế độ chia bước cho động cơ.

DIR: chân điều hướng động cơ – Direction.

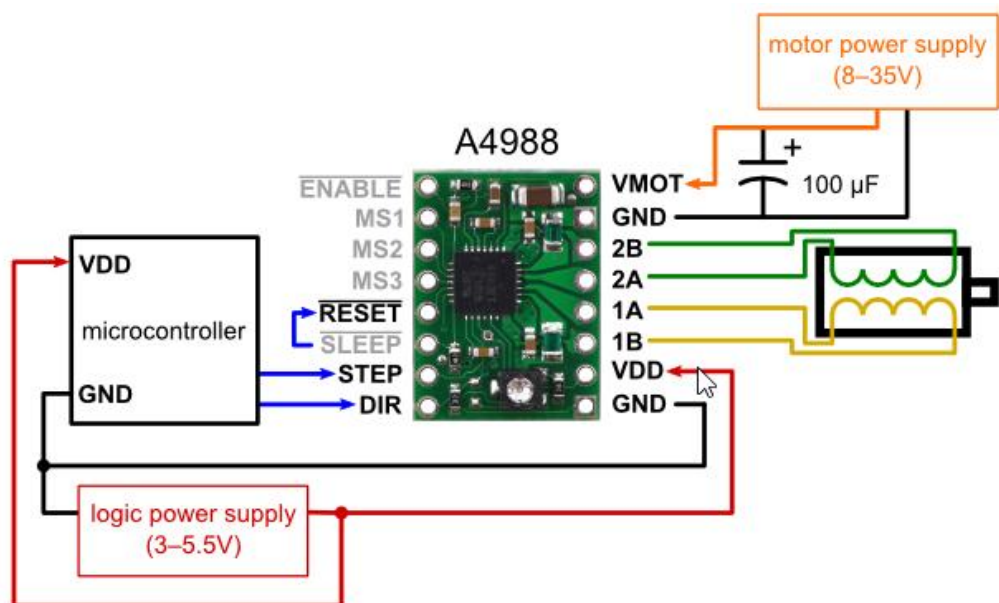
STEP: cấp xung cho chân này để động cơ quay.

VDD, GND: đây là hai chân cấp nguồn cho động cơ.

1A, 1B, 2A, 2B: là 4 chân nối với 4 dây của động cơ, là 2 cuộn dây của động cơ.

Tính năng:

- Điều khiển đơn giản
- Điều khiển được động cơ hoạt động với điện áp lên tới 35V dòng lên tới 2A
- Có 5 chế độ: full bước, 1/2 bước, 1/4 bước, 1/8 bước, 1/16 bước
- Điều chỉnh dòng ra bằng triết áp (bế xú) nằm bên trên Current Limit = $V_{REF} \times 2.5$
- Tự động Shutdown khi quá nóng...



Hình 3.17: Sơ đồ kết nối động cơ bước với driver A4988

Hình 3.17 là sơ đồ kết nối động cơ bước với mạch điều khiển A4988 gồm 4 dây của 2 cuộn dây của động cơ step nối vào các chân 1A, 1B, 2A, 2B, cấp nguồn cho động cơ qua hai chân VMOT và GND, cấp nguồn cho driver qua hai chân VDD và GND, điều khiển hướng và tốc độ động cơ qua hai chân DIR và STEP. Và có một lưu ý khi điều khiển động cơ bước là ta thường phải chỉnh biến trở để điều chỉnh dòng vào động cơ, để động cơ có thể chạy khỏe và ổn định nhất.

Lựa chọn chế độ full hay 1/2, 1/4, 1/8 hoặc 1/16 sẽ được thông qua 3 pin MS1 MS2 MS3. Người ta thường nối thẳng 3 pin này với công tắc bit 3p để dễ thiết lập từ trên phần cứng. Lưu ý là nếu thả nổi 3 pin này tức là mode full step. Chế độ này giúp ta chia nhỏ bước ra để tăng độ chính xác khi điều khiển động cơ step có chia bước lớn. Chế độ này chỉ xuất hiện ở các driver điều khiển hiện đại như ngày nay, nó gần giống như điều xung PWM của động cơ DC.

Bảng 3.7: Thiết lập chế độ chia bước:

MS1	MS2	MS3	Microstep Resolution
Low	Low	Low	Full step
High	Low	Low	Half step
Low	High	Low	Quarter step
High	High	Low	Eighth step
High	High	High	Sixteenth step

- Bật tắt động cơ thì thông qua pin ENABLE, mức LOW là bật module, mức HIGH là tắt module.
- Điều khiển chiều quay của động cơ thông qua pin DIR.
- Điều khiển bước của động cơ thông qua pin STEP, mỗi xung là tương ứng với 1 bước (hoặc vi bước).
- Hai chân Sleep với Reset nối với nhau.

3.2.7 Module Bluetooth HC05



Hình 3.18: Module Bluetooth HC05

Module bluetooth HC05 (H.3.18) dùng để thiết lập kết nối Serial giữa 2 thiết bị bằng sóng bluetooth.

Bảng 3.8: Thông số kỹ thuật module bluetooth HC05

Module	Bluetooth HC05
Điện áp hoạt động	3.3 ~ 5 VDC
Dòng hoạt động	Khi Pairing 30mA, sau khi pairing hoạt động truyền nhận 8mA
Baudrate UART	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
Support profile	Bluetooth serial port (master và slave)
Bluetooth protocol	Bluetooth specification v2.0 + EDR
Tần số hoạt động	2.4 GHz ISM band
Độ nhạy	-84 dBm tại 0.1% BER
Kích thước	15.2 x 35.7 x 5.6 mm

Các thông số thiết lập mặc định:

Thiết lập UART mặc định: Baurate 38400 hoặc 9600, N, 8, 1.

Pairing code mặc định: 1234 hoặc 0000.

Trong bài, module bluetooth HC05 được sử dụng để điều khiển robot bằng smart phone.

Module bluetooth sẽ truyền tín hiệu từ điện thoại đến board Arduino qua cổng Serial, arduino xử lý tín hiệu đó và điều khiển di chuyển của robot.

3.2.8 Mạch chuyển đổi tín hiệu USB to TTL CP2102

USB to TTL UART CP2102 (H.3.19) là module chuyển đổi tín hiệu USB sang tín hiệu Serial tuân tự theo chuẩn TTL.

Mạch sử dụng chip CP2102 là chip USB to UART của Silabs. CP2102 có kích thước nhỏ gọn và yêu cầu rất ít thành phần bên ngoài để hoạt động được ngay. CP2102 không sử dụng thạch anh ngoài như các chip PL2303 nên không sợ bị hư thạch anh.



Hình 3.19: Module USB to TTL CP2102

Module có sẵn ngõ ra điện áp 3.3V với dòng tải tối đa 100mA.

Trên mạch có 5 cổng đầu ra: 3.3V RST 5V Tx Rx Gnd. Trong đó chân RST được sử dụng để ngắt kết nối USB của module với máy tính.

Chân RST có thể được chỉnh sửa để trở thành chân RESET dùng để nạp chương trình cho Arduino Pro Mini mà không cần phải nhấn nút RESET.

Tốc độ tối đa 115200 bps.

Module USB to Serial CP 2102 có sẵn LED

LED nguồn sáng khi gắn vào máy tính và LED báo hiệu Tx/Rx, LED này sẽ sáng khi module nhận/gửi dữ liệu.

Driver tương thích với nhiều hệ điều hành. Đã test và hoạt động tốt trên Windows, Mac OS, Linux. Riêng với Gnu/Linux, driver đã được biên dịch sẵn trong kernel, bạn không cần cài thêm gì cả.

Trong bài, module USB to TTL CP2102 được sử dụng để giao tiếp Raspberry với Arduino.

3.2.9 Động cơ RC Servo

Động cơ Servo RC Emax ES08MA (H.3.20) là sản phẩm mới. Có hộp số bằng kim loại cùng kích thước hợp lý tạo nên 1 Servo RC rất tốt cho các ứng dụng yêu cầu độ chính xác cao. Ứng dụng để điều khiển Robot, Máy bay và các thiết bị tự động khác.



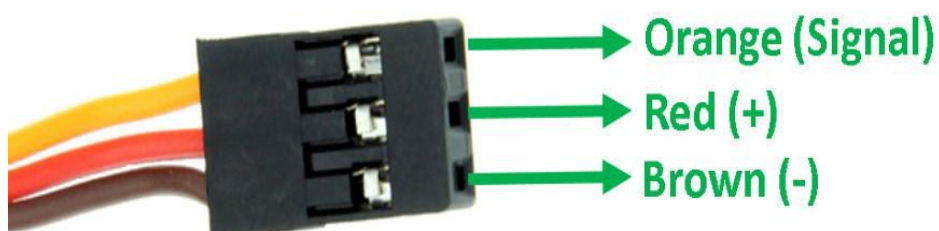
Hình 3.20: Động cơ RC Servo EMAX ES08

Servo trong bài được sử dụng để điều hướng camera, giúp camera bám theo đối tượng.

Bảng 3.9: Thông số kỹ thuật chính:

Động cơ	RC Servo EMAX ES08
Kích thước	32 x 11.5 x 24 mm
Trọng lượng	12 gram
Điện áp hoạt động	4.8V ~ 6.0V
Dòng điện hoạt động	200 A/60 độ
Tốc độ	0.12 giây/60 độ khi 4.8V, 0.10 giây/60 độ khi 6V
Momen xoắn	1.5/1.8 kg.cm (4.8V ~6V)
Vật liệu	Nhong đồng
Chiều dài Servo	25 cm
Tiêu chuẩn cắm	Fubata/JR phổ

Động cơ RC Servo trên thị trường thường có ba dây, bao gồm 2 dây nguồn VCC màu đỏ, GND màu nâu và 1 dây tín hiệu màu vàng (H.3.21).



Hình 3.21: Các dây kết nối của servo



Hình 3.22: Khung servo hai trục

Để điều hướng camera, ta thiết kế khung servo (H.3.22) có hai trục và kết hợp với camera.

Với kiểu thiết kế như thế này sẽ giúp robot có được tầm nhìn rộng, bao quát, khả năng mở rộng và phát triển thêm sau này.

Để điều khiển servo, ta sử dụng thư viện Servo.h trong Arduino.

3.2.10 Nguồn cấp cho robot



Hình 3.23: Pin Lipo 3S 2200mAh

Trong robot chúng em sử dụng nguồn là pin Lipo 3S 2200mAh (H.3.23).

Pin có dung lượng là 11.1V, áp sau khi sạc đầy có thể lên đến 12.6V và dòng tiêu thụ trong 1 giờ tối đa là 220mA.

Ưu điểm: Có dung lượng lớn trên cùng kích thước so với các chuẩn pin khác. Có thể sản xuất, chế tạo dưới nhiều hình dạng, kích thước đa dạng khác nhau. Pin nhẹ và mỏng hơn Pin Li-Ion. Dòng xả mạnh đáp ứng tốt cho mọi thiết bị (Smartphone, máy bay và xe mô hình,...). Thường được sử dụng rộng rãi trong các thiết bị như điện thoại di động, các mạch watch dog timer cảnh báo, các mạch chuyển tiếp duy trì điện khi mất điện trong các thiết quan trọng,

Nhược điểm: Giá còn khá đắt so với các loại pin sạc khác. Tuổi thọ khoảng 1000 vòng sạc xả (Cứ khoảng 300 vòng sạc xả khi đó pin suy giảm dung lượng đi 20% so với lúc mới) và tuổi thọ này bị ảnh hưởng khá nhiều bởi cách xả và sạc Pin. Phải sử dụng cẩn thận, khi không dùng ta phải để dung lượng pin khoảng 80% - 85% dung lượng lớn nhất để giữ cho pin không bị phồng, hư hại, có thể gây ra cháy nổ nếu không sử dụng đúng cách.

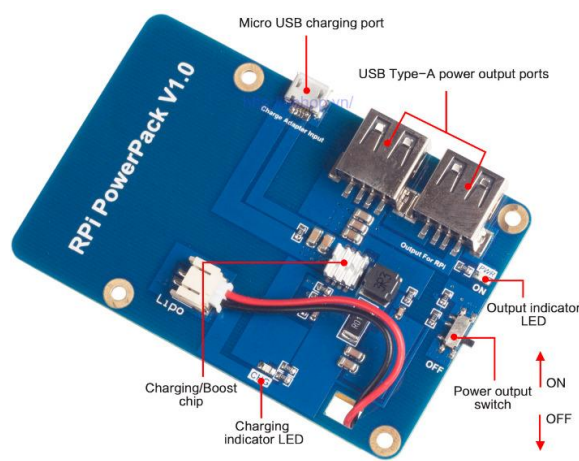
Để nguồn cấp cho board arduino được ổn định, chúng em sử dụng mạch ổn áp DC-DC mini.

Ưu điểm của mạch này là tạo ra áp ổn định cho board, dòng ra lên đến 3A và mạch ổn áp này rất nhỏ gọn, phù hợp với các ứng dụng robot nhỏ.



Hình 3.24: Mạch ổn áp DC-DC mini

Ngoài ra, robot của nhóm còn sử dụng raspberry nên phải có nguồn riêng cho raspberry pi động một cách ổn định nhất.



Hình 3.25: Mạch Raspberry Pi Lithium Battery Power Pack 3800mAh

Mạch Raspberry Pi Lithium Battery Power Pack 3800mAh được sử dụng để cấp nguồn bằng pin cho Raspberry Pi trong những trường hợp không thể dùng được nguồn AC cố định, mạch được thiết kế gồm một cổng sạc chuẩn Micro USB và hai cổng USB Output cho Raspberry Pi và màn hình hoặc các phụ kiện khác sử dụng nguồn USB.

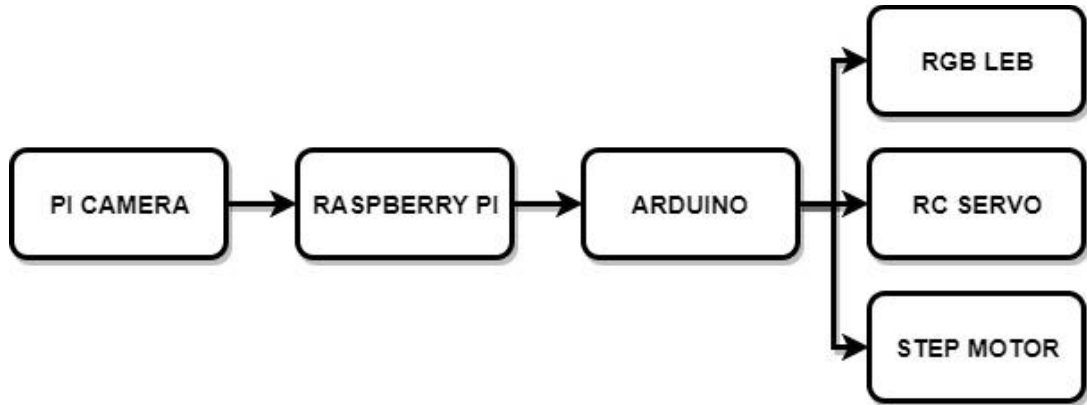
Mạch Raspberry Pi Lithium Battery Power Pack 3800mAh sử dụng pin Lithium dung lượng 3800mAh cho thời gian sử dụng lâu (tối đa lên đến 10h), dòng xả lớn và độ bền cao.

Bảng 3.10: Thông số kỹ thuật chính

Thiết bị	Raspberry Pi Lithium Battery Power Pack 3800mAh
Cổng sạc	Micro USB 5VDC
Dung lượng pin	3800mAh
Dòng xả tối đa	1.8A
Điện áp ra không tải	$5.1V \pm 0.1V$
Dòng/Áp sạc mặc định	1.0A/5.0V
Áp chứa tối đa khi sạc đầy	4.18 – 4.2V

CHƯƠNG 4 XÂY DỰNG VÀ ÁP DỤNG THUẬT TOÁN XỬ LÝ ẢNH VÀO ROBOT

4.1 KHÁI QUÁT VỀ NGUYÊN LÝ HOẠT ĐỘNG CỦA ROBOT



Hình 4.1: Sơ đồ hoạt động của robot

Theo sơ đồ trên (H.4.1), Picamera có nhiệm vụ thu thập hình ảnh về để raspberry pi xử lý. Ảnh của picamera có dạng ảnh màu 3 lớp RGB theo dạng số hóa.

Raspberry có nhiệm vụ xử lý ảnh được lấy từ picamera, từ đó xác định được đối tượng có trong hình và vị trí của đối tượng cần theo dõi, cuối cùng là tính toán ra tọa độ của đối tượng và truyền dữ liệu đó vào arduino.

Arduino có nhiệm vụ điều khiển các cơ cấu chấp hành có trong robot như servo và step motor. Các động cơ servo giúp cho camera di chuyển theo vật, còn 2 bánh step motor giúp xe giữ cân bằng và bám theo đối tượng.

RGB led báo trạng thái của robot.

4.2 CÁC BƯỚC XỬ LÝ ẢNH TRONG RASPBERRY



Hình 4.2: Các bước cơ bản xử lý hình ảnh trong raspberry pi

Các bước để xử lý ảnh (H.4.2) là: Picamera sẽ thu hình ảnh về dưới dạng ảnh màu 3 lớp RGB. Chất lượng hình ảnh phụ thuộc vào chất lượng camera, độ phân giải và số hình ảnh đọc trên 1 giây (frame per second). Ở đây chúng em sử dụng camera có độ phân giải 8MP, hình ảnh xuất ra có độ lớn 320x320 để gia tăng tốc độ xử lý cho raspberry, với framerate là 40.

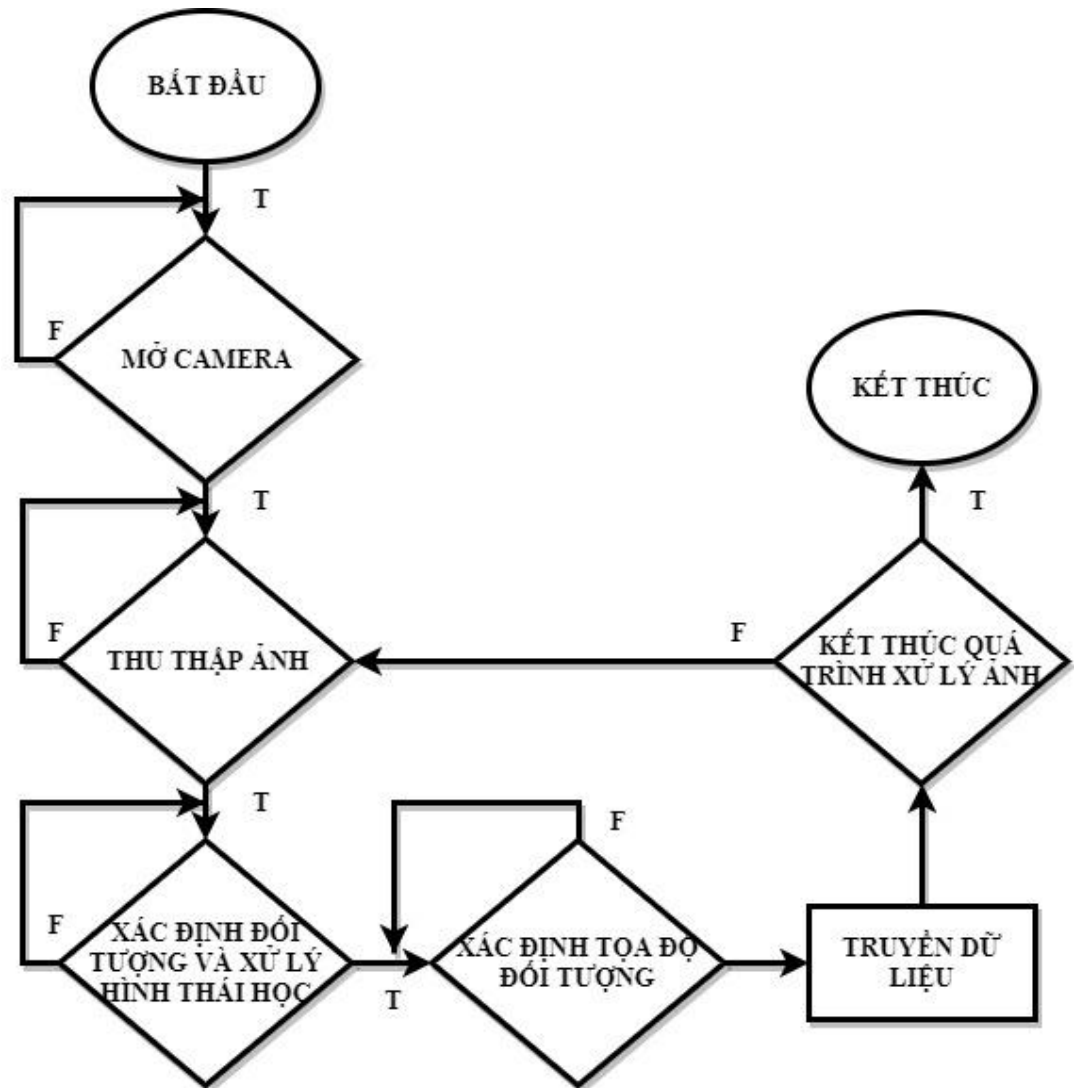
Tiền xử lý: hình ảnh sau khi thu thập từ picamera sẽ được đưa vào khâu tiền xử lý để gia tăng chất lượng hình ảnh bằng bộ lọc Gaussian, làm mờ hình ảnh, loại bỏ một số nhiễu từ môi trường. Khâu này rất quan trọng để gia tăng khả năng xử lý ảnh.

Xử lý hình ảnh: trong khâu xử lý hình ảnh, đối tượng sẽ được chuyển đổi không gian màu, lấy ngưỡng để xác định đối tượng cần tìm trong hình, các thuật toán về hình thái học sẽ được sử dụng để lọc nhiễu và tăng độ chính xác trong nhận dạng đối tượng.

Xác định tọa độ dựa vào góc của servo và độ cao của robot.

Raspberry pi truyền thông tin tọa độ đến Arduino thông qua cổng USB, ta sẽ chuyển đổi tín hiệu từ USB sang tín hiệu serial TTL UART để đọc bằng Arduino, Arduino sẽ xử lý tín hiệu tọa độ đó điều khiển servo và robot bám theo đối tượng.

4.3 THUẬT TOÁN XỬ LÝ ẢNH XÁC ĐỊNH ĐỐI TƯỢNG

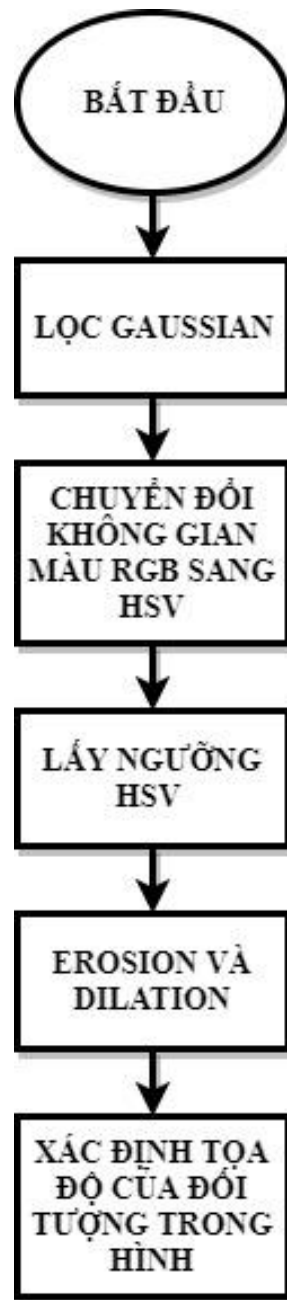


Hình 4.3: Sơ đồ khối biểu diễn quá trình xử lý ảnh

Giải thích sơ đồ khối (H.4.3):

Bắt đầu ta sẽ mở camera hoạt động, camera sẽ thu thập hình ảnh và hình ảnh đó sẽ qua các bước xử lý hình thái học để xác định được đối tượng có trong hình. Khi ta đã xác định được đối tượng có trong hình rồi, ta sẽ xác định tọa độ của đối tượng đó và truyền tọa độ đó về board arduino để arduino xử lý.

Để kết thúc quá trình xử lý ảnh, nhóm có lập trình một nút bấm để thoát khỏi vòng lặp xử lý ảnh.



Hình 4.4: Sơ đồ khối biểu diễn thuật toán xác định đối tượng có trong hình bằng xử lý ảnh

Dựa vào Hình 4.4, bắt đầu là chúng ta lấy hình ảnh thu được từ camera đi vào lọc Gaussian để làm mịn hình ảnh trước khi đi vào xử lý chính. Hàm sử dụng ở đây là `cv2.GaussianBlur(img,(5,5),0)`

- `img`: đầu vào là hình ảnh từ camera.
- `(5,5)` là cấu trúc phần tử (kernel) có kích thước `5x5`

Ta đưa ảnh sau khi lọc chuyển đổi từ không gian màu RGB sang không gian màu HSV bằng hàm `cv2.cvtColor(blurred, cv2.COLOR_BGR2HSV)`

- `Blurred`: là hình ảnh sau khi lọc Gaussian
- `cv2.COLOR_BGR2HSV`: là chuyển từ không gian màu RGB sang HSV

Sau khi chuyển không gian màu, ta lấy ngưỡng trong không gian màu HSV theo màu của đối tượng cần tìm có trong hình. OpenCV cung cấp cho ta hàm `cv2.inRange(hsv, greenLower, greenUpper)`

- `hsv`: là hình ảnh trong không gian màu HSV
- `greenLower`: ngưỡng dưới của màu của đối tượng
- `greenUpper`: ngưỡng trên của màu của đối tượng

Ta sử dụng các phép biến đổi hình thái học để xử lý các tín hiệu nhiễu đến từ môi trường có trong hình ảnh. Đó là các phép Erosion và Dilation, lần lượt là các hàm `cv2.erode(mask, None, iterations=2)`, `cv2.dilate(mask, None, iterations=2)`. Trong đó:

- `Mask`: ảnh sau khi lấy ngưỡng
- `None`: cấu trúc phần tử mặc định
- `Iterations`: độ lớn của cấu trúc phần tử (`2x2`)

Sau khi đã xử lý được đối tượng và nền, lúc này hình ảnh sẽ ở dạng ảnh nhị phân. Đối tượng sẽ có màu trắng và nền là màu đen. Chúng ta sử dụng hàm `cv2.findContours(thresh, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)` để xác định xem có đối tượng trong hình không.

- `Thresh`: ảnh nhị phân sau khi đã tách đối tượng ra khỏi nền.
- `cv2.RETR_TREE`: chế độ đi tìm đường viền.
- `cv2.CHAIN_APPROX_SIMPLE`: phương pháp xấp xỉ đường viền.

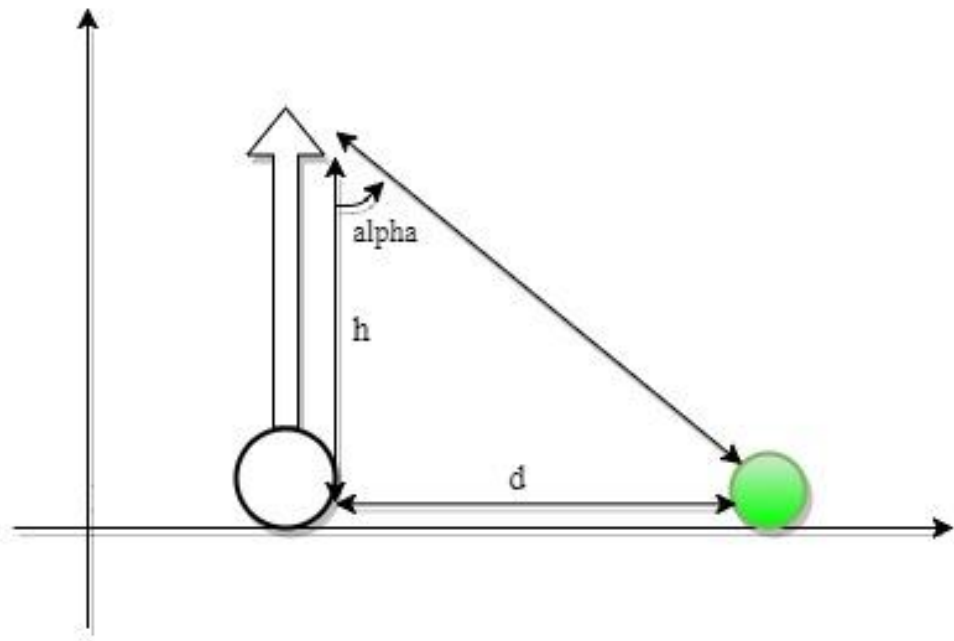
Nếu trong hình có đối tượng, giá trị trả về của hàm findContours sẽ lớn hơn 0. Ta tìm các đặc tính của đối tượng bằng hàm $M = cv2.moments(img)$. Điểm trung tâm của đối tượng có trong hình được tính bằng công thức sau:

$$\{\bar{x}, \bar{y}\} = \left\{ \frac{M10}{M00}, \frac{M01}{M00} \right\} \quad (4.1)$$

Sau khi xác định được tọa độ, ta gửi tọa độ này về cho arduino để arduino điều khiển các cơ cấu chấp hành.

4.4 TÍNH KHOẢNG CÁCH GIỮA ROBOT VÀ ĐỐI TƯỢNG

Ta dựa vào góc của servo để tính toán ra khoảng cách giữ robot và đối tượng.



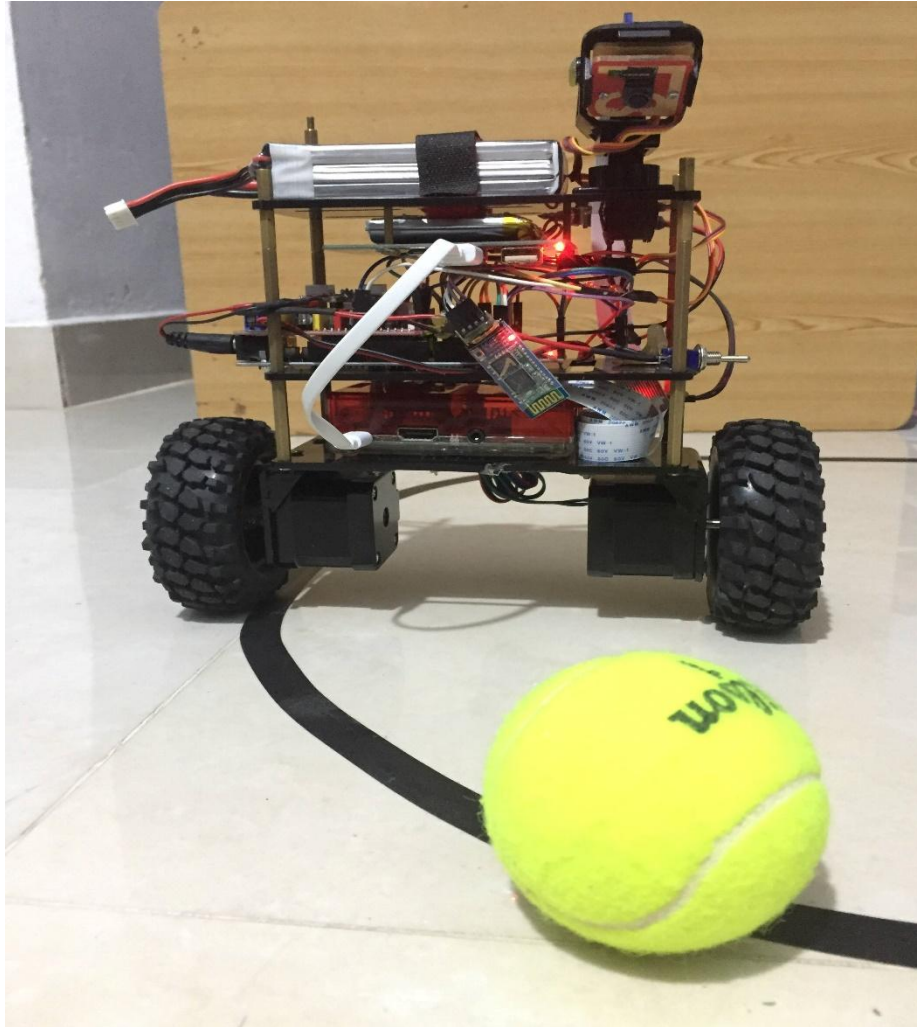
Hình 4.5: Tính khoảng cách của robot và đối tượng

Dựa vào Hình 4.5, ta thấy góc alpha là góc của servo so với phương thẳng đứng, h là chiều cao của robot và d là khoảng cách giữa đối tượng và robot, ta có công thức tính khoảng cách d là:

$$d = h \times \tan \alpha \quad (4.2)$$

CHƯƠNG 5: KẾT QUẢ THỰC NGHIỆM

5.1 MÔ HÌNH PHẦN CỨNG

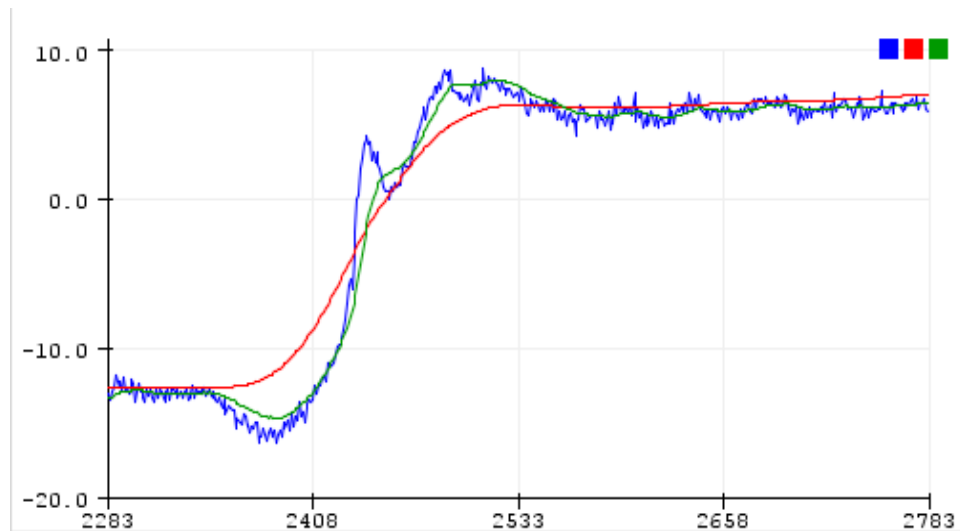


Hình 5.1: Mô hình Robot hai bánh tự cân bằng

Hình 5.1 là mô hình robot hai bánh tự cân bằng mà nhóm đã thực hiện được. Như trên hình, các phần cứng mà nhóm đã sử dụng bao gồm Arduino Mega 2560, Raspberry Pi Model 3, module bluetooth HC05, cảm biến góc MPU6050, động cơ bước, picamera, nguồn gồm pin lipo cho động cơ bước và pin lithium cho raspberry. Với thiết kế xây tầng đứng như trên mô hình, ta có thể dễ dàng điều chỉnh trọng tâm của robot và giúp robot di chuyển nhanh hơn, linh hoạt hơn do mô hình không quá cao

5.2 XỬ LÝ TÍN HIỆU TỪ CẢM BIẾN MPU6050

Ở đây nhóm sử dụng bộ lọc Complementary để kết hợp hai giá trị gia tốc và vận tốc góc để tính được tín hiệu góc nghiêng thực tế.



Hình 5.2: Bộ lọc Complementary trong tính góc nghiêng

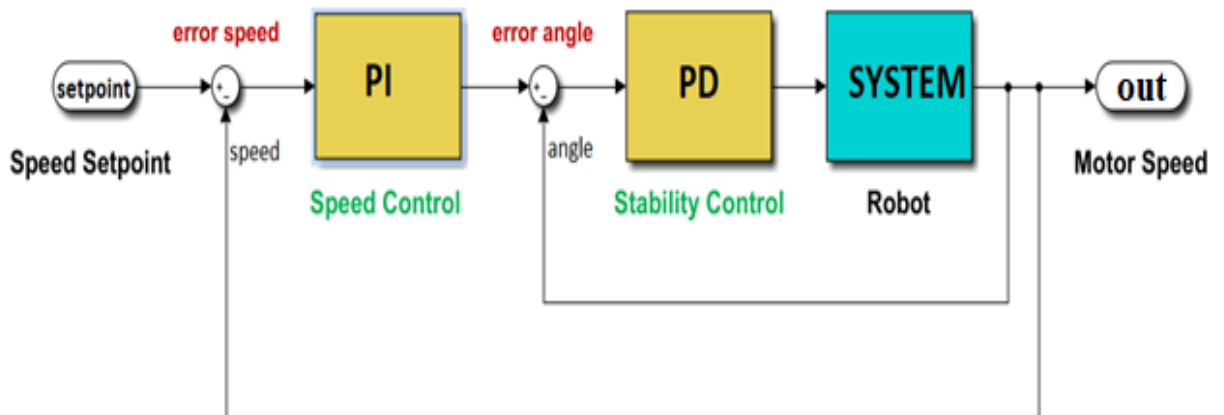
- Đường màu xanh nước biển: cảm biến gia tốc trọng trường.
- Đường màu xanh lá cây: cảm biến vận tốc góc.
- Đường màu đỏ: góc từ bộ lọc complementary

Nhìn vào biểu đồ Hình 5.2, ta thấy tín hiệu góc của cảm biến gia tốc rất không ổn định do giá trị gia tốc trả về không nhanh, những khúc góc thay đổi nhanh thì cảm biến gia tốc không bắt kịp gia tốc chính xác của robot tại thời điểm đó.

Còn tín hiệu góc của cảm biến vận tốc trả về góc nhanh và theo thời gian, nhưng tại thời điểm góc thay đổi chậm, vận tốc góc không lớn thì cảm biến vận tốc bị sai số trượt do không đo được góc lệch lúc đó, nên ta thấy trên Hình 5.2, giá trị góc của cảm biến vận tốc góc có những lúc thấp hơn so với giá trị góc của gia tốc góc.

Sau khi sử dụng bộ lọc complementary để kết hợp giá trị góc của 2 loại cảm biến gia tốc và vận tốc, kết quả thu được là tốt, góc ra chính xác và không bị nhiễu nhiều.

5.3 KẾT QUẢ CỦA BỘ ĐIỀU KHIỂN PID ĐƯỢC ÁP DỤNG TRONG ROBOT HAI BÁNH TỰ CÂN BẰNG



Hình 5.3: Sơ đồ bộ điều khiển PID được ứng dụng trong robot

Các bộ điều khiển được sử dụng trong robot cân bằng (H.5.3):

Bộ điều khiển cân bằng PD

Thông số tìm được:

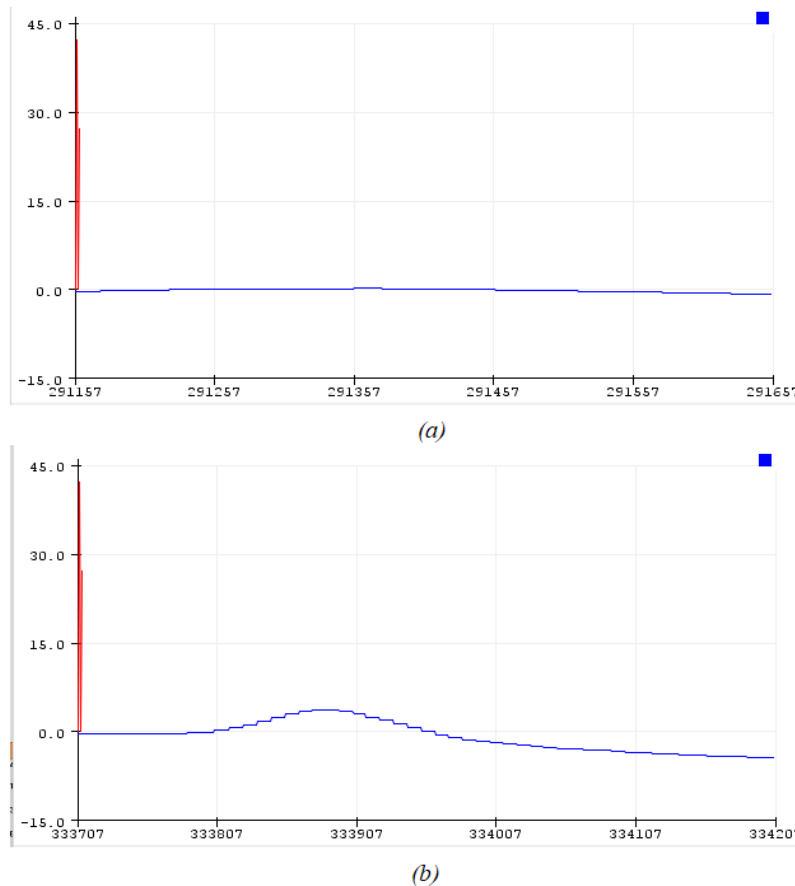
$$K_p = 0.32$$

$$K_d = 0.15$$

Để tìm được các thông số này, đầu tiên ta đặt các thông số K_p và K_d bằng 0. Sau đó ta tăng K_p dần lên cho đến khi robot lắc quanh điểm cân bằng thì dừng lại. Lúc này sai số hệ thống sẽ dần về 0 là điểm cân bằng nhưng vẫn còn dao động. Ta tăng K_d dần lên cho đến khi robot tự cân bằng một cách ổn định nhất thì dừng lại. Lưu ý là khi tăng K_d quá cao, thời gian xác lập sẽ tăng theo và robot sẽ vào điểm cân bằng lâu hơn.

Kết quả: Bộ điều khiển PD sẽ điều khiển động cơ bước sao cho góc nghiêng từ MPU6050 trả về dần về góc đặt tại setpoint.

Sau khi áp dụng bộ điều khiển PID vào robot, robot đã có thể cân bằng tại vị trí setpoint của nó và di chuyển mà không hề bị đổ.



Hình 5.4: Tác động lực lên robot

(a) Góc trả về của robot tại vị trí cân bằng

(b) Góc trả về của robot khi robot bị tác động lực

Đường màu xanh trong Hình 5.4 chính là giá trị góc nghiêng của robot do cảm biến MPU6050 trả về.

Hình 5.4a là góc của robot khi đang đứng yên tại vị trí cân bằng trên mặt phẳng ngang so với mặt đất. Nhận xét thấy robot có dao động qua lại tại vị trí cân bằng nhưng rất ổn định với góc lệch không lớn so với góc đặt.

Hình 5.4b là góc nghiêng của robot khi robot bị tác động một lực, ta thấy góc nghiêng của robot trả về giá trị rất đều và mượt, bộ điều khiển PID lập tức điều

kiểm động cơ bước để đưa góc lệch về vị trí cân bằng bằng cách di chuyển robot về hướng ngược lại.

Bảng 5.1: Thời gian quay về vị trí cân bằng với từng góc lệch khác nhau

Góc lệch (độ)	Thời gian xác lập (s)
0	0
4	0.5
10	0.9
15	1.51

Theo bảng 5.1, ta thấy với góc lệch càng lớn, thời gian để robot quay về vị trí cân bằng càng lâu, với mô hình robot thực tế của nhóm đã thực hiện, robot có thể nghiêng tối đa với góc lệch là 40 độ so với phương thẳng đứng.

Với góc nghiêng quá lớn trên 40 độ, nhóm có thiết lập một bộ điều khiển với cách thông số PID cao hơn, để robot về trạng thái cân bằng một cách nhanh nhất bằng cách giảm thông số K_p và tăng thông số K_d lên. Lý do giảm thông số K_p là vì khâu P cao sẽ làm robot di chuyển quá nhanh, tăng độ vọt lố. Khi giảm K_p thì sai số sẽ giảm, nên ta tăng K_p để khâu tích phân bù lại sai số cho khâu tỉ lệ, đồng thời giúp tăng thời gian khởi động, xe nhanh chóng đạt được vị trí cân bằng.

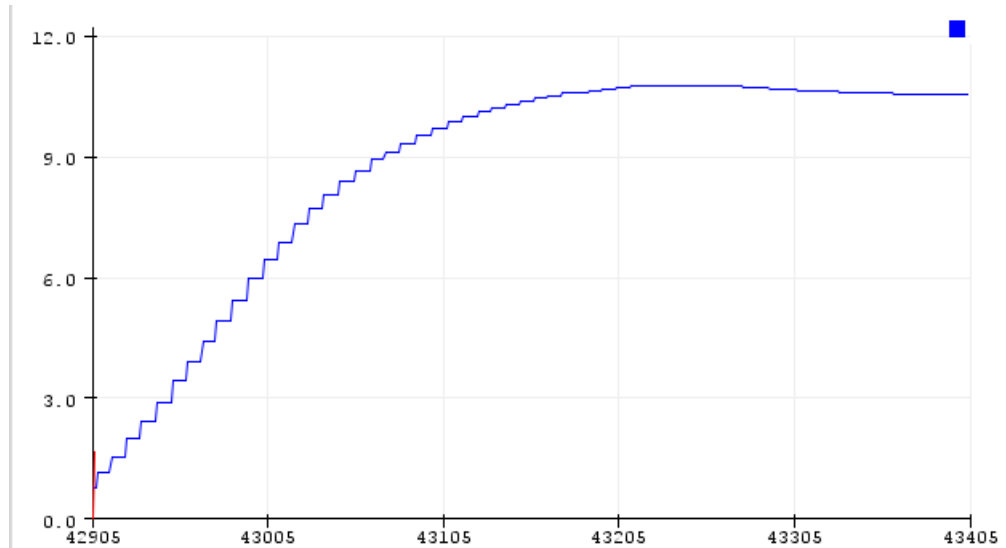
Bộ điều khiển tốc độ PI

Thông số tìm được:

$$K_p = 0.08$$

$$K_i = 0.1$$

Phương pháp mà chúng em áp dụng để tìm được thông số PI này là đầu tiên chúng em phải đặt giá trị K_p và K_i bằng 0. Sau đó truyền thông số tốc độ cho bộ điều khiển vào tăng nhẹ K_p lên cho đến khi xe di chuyển và đạt được tốc độ mong muốn, cuối cùng là tăng nhẹ K_i để xe đạt được tốc độ đặt nhanh hơn, thông số này rất quan trọng để xe có thể đạt được tốc độ một cách nhanh nhất.



Hình 5.5 Góc điều khiển tốc độ

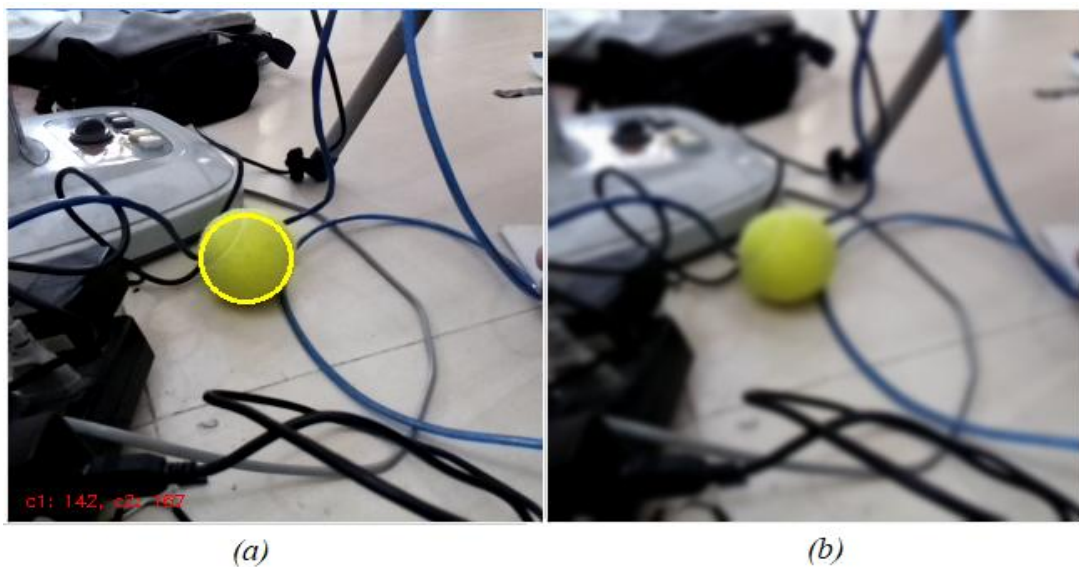
Dựa vào Hình 5.5, ta có thể thấy góc tăng lên từ 0 đến 12 độ một cách nhanh chóng và ổn định nhờ vào bộ điều khiển PI.

Tùy vào tốc độ đặt vào mà góc đặt vào sẽ khác nhau.

Theo thực tế thì mô hình Robot hai bánh tự cân bằng có thể giữ thăng bằng tối đa với góc nghiêng là 40 độ so với phương thẳng đứng, góc nghiêng để đạt tốc độ tối đa là 15 độ, những con số này sẽ thay đổi khi mô hình thay đổi về chiều cao và chiều rộng của robot.

5.4 ỨNG DỤNG XỬ LÝ ẢNH VÀO ROBOT HAI BÁNH TỰ CÂN BẰNG BẢM THEO ĐỐI TƯỢNG

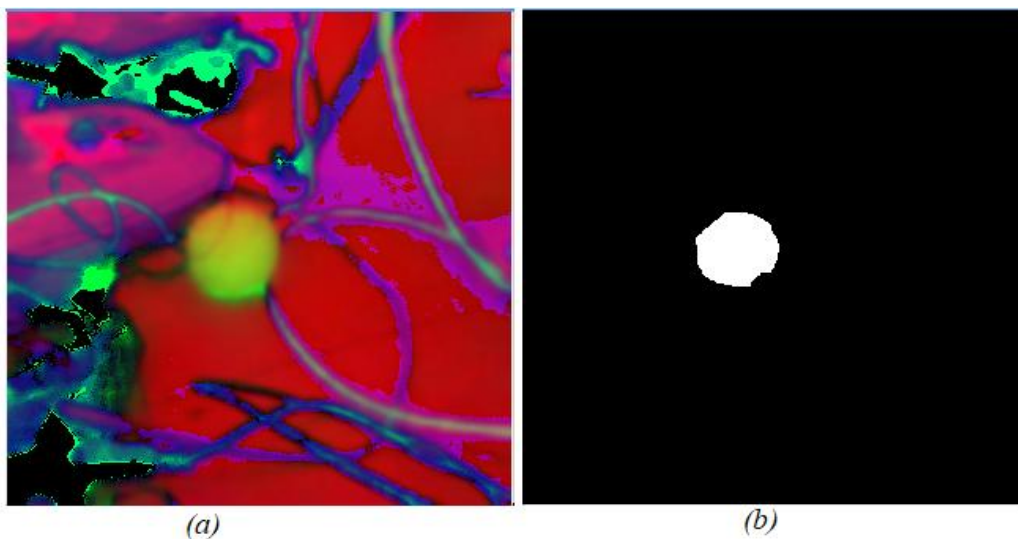
Phần xử lý ảnh bao gồm thu thập hình ảnh từ camera, tiền xử lý hình ảnh bằng bộ lọc Gaussian, chuyển đổi không gian màu từ RGB sang HSV, lấy ngưỡng HSV để lọc màu đối tượng và nền, sử dụng các phép hình thái học để loại bỏ nhiễu, xác định đối tượng.



Hình 5.6: Ảnh sau trước và sau khi xử lý

(a) Ảnh ban đầu lấy từ camera; (b) Ảnh đã được làm mờ bằng lọc Gaussian

Hình 5.6 là kết quả của bộ lọc Gaussian, ta thấy hình ảnh được làm mờ (H.5.6b) một cách toàn cục, vẫn giữ được bối cảnh như hình ảnh gốc (H.5.6a). Bước tiền xử lý này giúp chúng ta làm “mượt” ảnh, loại bỏ được các phần tử “gai” trong hình ảnh gốc, giúp phần xử lý ảnh dễ dàng hơn.



Hình 5.7: Chuyển đổi không gian màu và xử lý lấy ngưỡng

(a) Ảnh trong không gian màu HSV; (b) Lấy ngưỡng màu của đối tượng

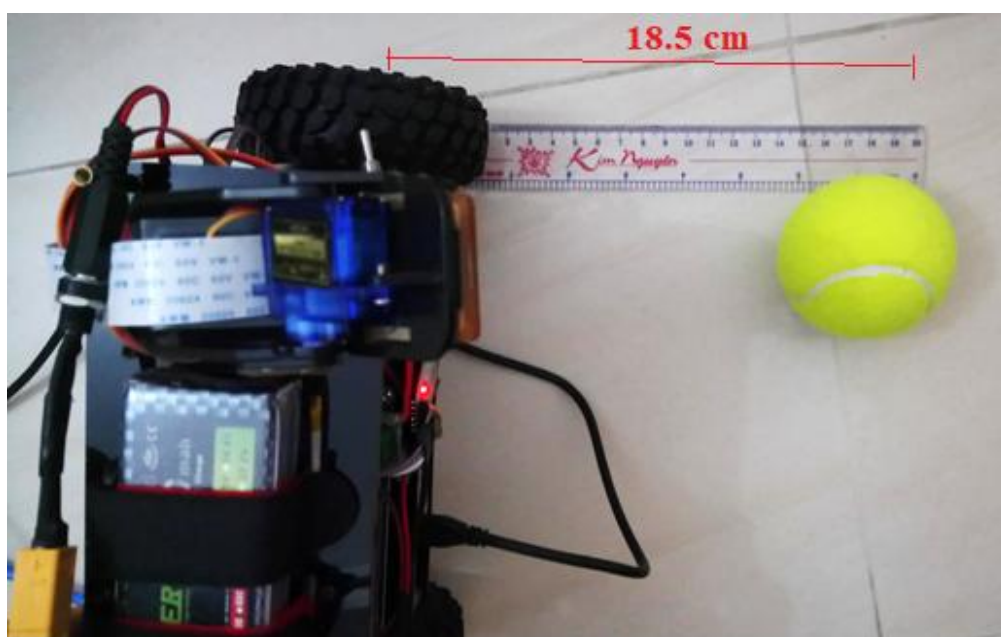
Kết quả của thuật toán chuyển không gian màu RGB sang không gian màu HSV được thể hiện trong Hình 5.7a. Ta có thể thấy rõ ràng rằng hình ảnh trong không gian màu RGB (H.5.6a) không thể hiện rõ nét các khoảng màu được như hình trong không gian màu HSV(H.5.7a), đây là lý do mà không gian màu HSV được sử dụng phổ biến trong các giải thuật xử lý ảnh.

Sau khi chuyển đổi sang không gian màu HSV, ta sẽ lấy ngưỡng màu của đối tượng, cụ thể trong Hình 5.7 ở đây là trái banh tennis. Thông số lấy ngưỡng của nhóm như sau

- H: 29 – 64
- S: 86 – 255
- V: 10 – 255

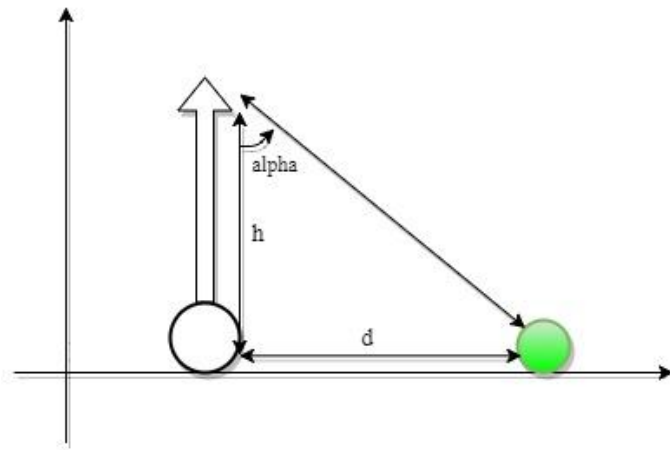
Kết quả sau khi lấy ngưỡng được thể hiện trên Hình 5.7b

5.5 PHÉP TÍNH KHOẢNG CÁCH TỪ ROBOT ĐẾN ĐỐI TƯỢNG



Hình 5.8: Khoảng cách từ robot đến đối tượng

Hình 5.8 là khoảng cách thực tế của robot so với đối tượng là 18.5 cm.

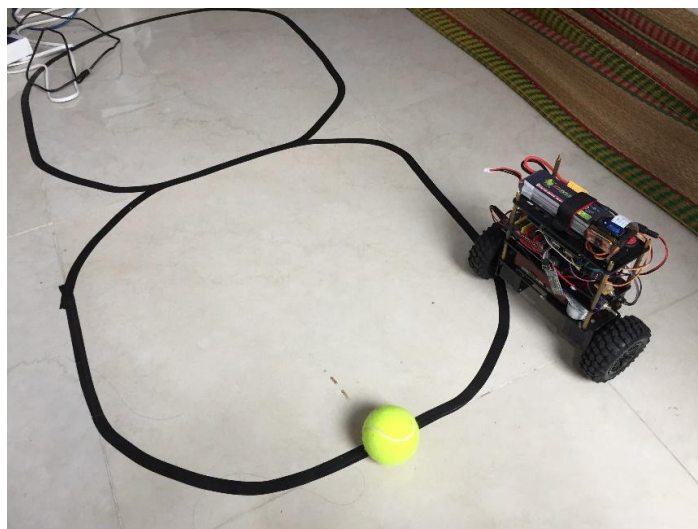


Hình 5.9: Tính khoảng cách từ robot đến đối tượng

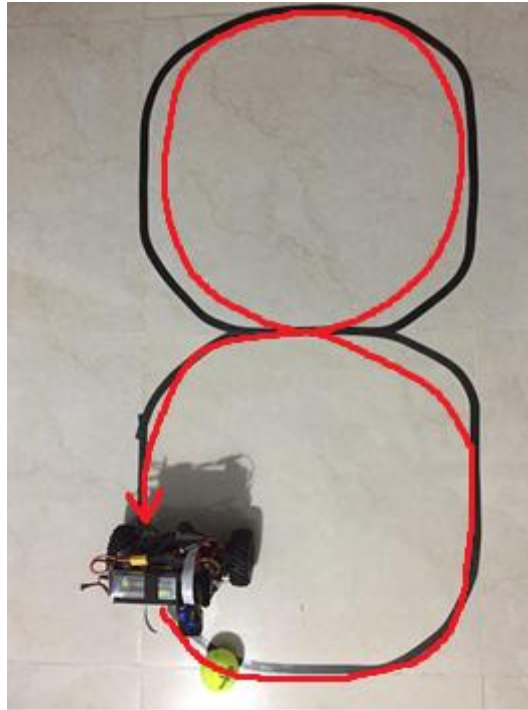
Giá trị servo trả về góc α là 40° và chiều cao h của robot là 22 cm, vậy khoảng cách d ta tính được là 18.46 cm. So với khoảng cách thực tế là 18.5 cm, ta thấy servo trả về giá trị khá chính xác.

Ta có thể thấy được độ chính xác về phép tính khoảng cách của robot so với đối tượng trong phương pháp này là tương đối cao nhờ vào phép tính đơn giản, độ chính xác của phần cứng như servo và cách xây dựng mô hình robot. Chính vì vậy robot có thể duy trì khoảng cách với đối tượng một cách đơn giản và chính xác.

5.5 THỰC NGHIỆM VỀ KHẢ NĂNG BẮM THEO VẬT CỦA ROBOT



Hình 5.10: Robot xử lý bám đối tượng vòng số 8



Hình 5.11: Robot bám theo đối tượng theo vòng số 8

Nhóm đã kiểm tra khả năng hoạt động bám theo đối tượng của robot bằng cách cho đối tượng đi theo vòng số 8 và cho robot bám theo. Dựa vào khả năng giữ cân bằng của robot khi đứng yên, khi di chuyển, khả năng hoạt động của động cơ bước, khả năng bắt đối tượng của RC servo, khả năng giữ khoảng cách của robot để nhận xét tổng quan về ưu và nhược điểm của mô hình này.

- Robot giữ cân bằng tốt khi đứng yên và khi di chuyển.
- Động cơ RC hướng theo đối tượng khá chính xác, giúp camera có thể lia theo đối tượng nhanh bám theo thời gian thực.
- Theo hình 5.11, dựa vào đường màu đỏ là đường robot đã di chuyển, ta thấy robot có sự linh hoạt trong những khúc cua, khả năng bẻ lái khá tốt và bám sát đối tượng nhờ vào động cơ bước phản hồi tốt.
- Robot duy trì được khoảng cách so với trái banh, khi ở xa đối tượng, robot sẽ tiến lại gần, khi đối tượng lại gần, robot sẽ tự động lùi lại để có thể giữ khoảng cách chính xác với đối tượng.

Nhược điểm:

- Động cơ RC Servo bị trượt mất đối tượng khi đối tượng di chuyển quá nhanh, vượt ra khỏi khung hình.
- Ảnh hưởng xấu từ môi trường như ánh sáng, màu sắc sẽ gây nhiễu cho quá trình xử lý ảnh, khiến robot không thể xác định chính xác đối tượng.
- Quỹ đạo di chuyển của đối tượng di chuyển nhanh sẽ làm robot bị mất dấu quỹ đạo di chuyển cũ và sẽ đi theo quỹ đạo mới.
- Không thể bám theo đối tượng sau khi đã mất dấu đối tượng ra khỏi khung hình.

CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 KẾT LUẬN

Sau 4 tháng nghiên cứu, chúng em đã hoàn thành đề án “Thiết kế robot hai bánh tự cân bằng sử dụng kỹ thuật xử lý ảnh” với các thành quả đã đạt được:

- Robot có thiết kế nhỏ gọn giúp khả năng tự cân bằng, di chuyển một cách hiệu quả.
- Robot có khả năng bám theo và giữ khoảng cách với đối tượng tốt trong điều kiện cho phép.

Bên cạnh đó thì robot vẫn còn tồn tại một số nhược điểm sau:

- Kỹ thuật xử lý ảnh đơn thuần không thực sự tối ưu khi có những vật cùng màu xung quanh đối tượng và các ảnh hưởng khác từ môi trường gây nhiễu màu.
- Robot không thể bám theo đối tượng nếu đối tượng di chuyển quá nhanh hay bị mất dấu đối tượng khi đang bám theo.

GIẢI PHÁP

- Ta có thể sử dụng các phương pháp xử lý ảnh nâng cao hơn là trí tuệ nhân tạo, học thuật máy tính, ...
- Tối ưu hóa lập trình sẽ giúp robot hoạt động trơn tru và xử lý nhanh hơn.
- Thiết lập một thuật toán vẽ bản đồ tìm đối tượng dựa trên đường đi trước đó của đối tượng sẽ giúp một phần nào khả năng bám theo của robot.

6.2 HƯỚNG PHÁT TRIỂN

Dựa trên các nghiên cứu về robot cân bằng, học thuật xử lý ảnh thì hướng phát triển robot tự cân bằng của chúng em là:

- Nhận dạng đối tượng bằng phương pháp trí tuệ nhân tạo.
- Thêm các chức năng tự động phân loại, sắp xếp đồ vật.