

**ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CƠ KHÍ**



LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC

**XE HAI BÁNH TỰ CÂN BẰNG
DI CHUYỂN
TRÊN ĐỊA HÌNH PHẲNG**

MÃ NGÀNH: 128

**SVTH : MAI TUẤN ĐẠT
CBHD : KS. VÕ TƯỜNG QUÂN**

**CHƯƠNG TRÌNH ĐÀO TẠO KỸ SƯ CHẤT LƯỢNG CAO
KHÓA 2: 2000 – 2005
TP. HỒ CHÍ MINH, 07/2005**

**ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CƠ KHÍ**



LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC

XE HAI BÁNH TỰ CÂN BẰNG DI CHUYỂN TRÊN ĐỊA HÌNH PHẪNG

MÃ NGÀNH:128

**SVTH :MAI TUẤN ĐẠT
MSSV :P0000016
CBHD :KS. VÕ TƯỜNG QUÂN**

**CHƯƠNG TRÌNH ĐÀO TẠO KỸ SƯ CHẤT LƯỢNG CAO
KHÓA 2: 2000 – 2005
TP. HỒ CHÍ MINH, 07/2005**

Lời cảm ơn

Tôi không thể theo đuổi và hoàn thành đề tài của luận văn trong vòng 16 tuần nếu không có sự giúp đỡ của những người thân và người bạn xung quanh. Do vậy, với sự trân trọng và cảm kích, tôi xin gửi lời cảm ơn đến ông bà và cha mẹ, những người thân trong gia đình hết lòng chăm sóc, an ủi khi gặp trở ngại và động viên tôi trong thời gian thực hiện luận văn, xin cảm ơn TS. Nguyễn Văn Giáp và giáo viên trực tiếp hướng dẫn luận văn, thầy Võ Tường Quân đã cho phép tôi theo đuổi đề tài và cho những lời khuyên xác đáng, kịp thời những lúc gặp khó khăn khi thực hiện trong suốt quá trình làm luận văn tốt nghiệp đại học. Ngoài ra, tôi cũng xin chân thành cảm ơn anh Quân và anh Kiên ở công ty máy tính Bách Khoa đã hỗ trợ một phần kinh phí và thiết bị để thực hiện đề tài; cảm ơn người anh – Th.S Trần Công Bình, giảng viên bộ môn Thiết bị Điện – nhiệt tình giúp đỡ về mặt lý thuyết để hoàn thành phần điện động cơ công suất cao, một phần khá hóc búa của đề tài. Ngoài ra, cũng xin cảm ơn Thy và Tâm, hai người bạn thân thiết nhất đã giúp tôi hoàn thành bản thuyết minh mà chúng ta đang có trên tay.

Cuối cùng em xin cảm ơn tất cả quý Thầy Cô tham gia giảng dạy chương trình Kỹ sư chất lượng cao Việt Pháp khóa 2000-2005, và Khoa Cơ Khí, bộ môn Cơ Điện tử, Trường Đại Học Bách Khoa TP.HCM đã trang bị cho em những kiến thức cơ sở cũng như đã giúp đỡ tôi trong thời gian làm Luận văn tốt nghiệp.

Tp. Hồ Chí Minh, ngày 03 tháng 07 năm 2005

Mai Tuấn Đạt

MỤC LỤC

Lời cảm ơn	
Mục lục	i
Tóm tắt đề tài	iv
Abstract	v
CHƯƠNG 1 TỔNG QUAN	1
1.1 Lời nói đầu	1
1.2 Thế nào là xe hai bánh tự cân bằng (two wheels self balancing)	2
1.3 Tại sao phải thiết kế xe hai bánh tự cân bằng	3
1.4 Ưu nhược điểm của xe hai bánh tự cân bằng	4
1.4.1 Ưu điểm của xe scooter tự cân bằng trên hai bánh	4
1.4.2 Nhược điểm của xe	4
1.5 Khả năng ứng dụng	5
1.6 Tình hình nghiên cứu trong và ngoài nước	5
1.6.1 Một số dạng xe hai bánh tự cân bằng dùng trên robot	5
1.6.2 Một số dạng scooter hai bánh tự cân bằng	9
1.7 Nhu cầu thực tế	14
CHƯƠNG 2 NHIỆM VỤ LUẬN VĂN	15
2.1 Mục tiêu đề tài	15
2.2 Phương pháp nghiên cứu	15
CHƯƠNG 3 LÝ THUYẾT TIẾP CẬN	17
3.1 Phương pháp tính động lực học	17
3.2 Thuật toán điều khiển - Kỹ thuật điều khiển hiện đại	24
3.3 Các phương pháp xử lý tín hiệu từ cảm biến	29
3.3.1 Lọc bổ phụ thông tần (complementary filter)	29
3.3.2 Lọc thích nghi - Bộ lọc Kalman	32
3.3.3 So sánh các bộ lọc với bộ lọc Kalman	40
3.4 Mô hình lý thuyết động cơ DC	43
CHƯƠNG 4 TÍNH TOÁN MÔ PHỎNG	45
4.1 Các thông số trong mô hình mô phỏng được xây dựng	45

4.2 Mô phỏng MatLAB	46
4.2.1 <i>Giới thiệu về phần mềm MatLAB, công cụ Simulink</i>	46
4.2.2 <i>Kết quả tính bằng MatLAB</i>	46
4.3 Mô phỏng VisualNastran và Simulink	48
4.3.1 <i>Giới thiệu về phần mềm VisualNastran</i>	48
4.3.2 <i>Cách thực hiện mô phỏng bằng vN Desktop 4D</i>	49
4.3.4 <i>Kết quả mô phỏng</i>	50
CHƯƠNG 5 THỰC HIỆN	54
5.1 Thiết kế cơ khí	54
5.1.1 <i>Tóm tắt thiết kế</i>	54
5.1.2 <i>Tính toán sức bền</i>	54
5.2 Mạch điện tử	59
5.2.1 <i>Nguồn điện</i>	60
5.2.2 <i>Mạch công suất điều khiển động cơ</i>	61
5.2.2.1 <i>Bộ đệm (MOSFET driver)</i>	61
5.2.2.2 <i>MOSFET công suất – mắc bổ phụ</i>	63
5.2.2.3 <i>Mạch Snubber</i>	66
5.2.2.4 <i>MOSFET thắng</i>	66
5.2.3 <i>Cảm biến</i>	66
5.2.3.1 <i>Thiết bị đo góc gyro Murata ENC-03</i>	67
5.2.3.2 <i>ADXL202A</i>	68
5.2.3.3 <i>Cảm biến đo vị trí- encoder</i>	73
5.2.3.4 <i>Cảm biến đo dòng hồi tiếp (Điện trở shunt)</i>	75
5.2.4 <i>Bộ xử lý trung tâm - vi điều khiển PIC 18F452</i>	76
5.2.4.1 <i>Các khả năng của vi điều khiển Microchip PIC 18F452:</i>	76
5.2.4.2 <i>Mạch điều khiển trung tâm</i>	79
5.2.5 <i>Bảng điều khiển và hiển thị</i>	80
5.2.6 <i>Động cơ</i>	80
5.2.7 <i>Hình chụp các mạch điện tử</i>	85
5.3 Giải thuật - Lưu đồ chương trình	88
5.3.1 <i>Chương trình chính</i>	88
5.3.2 <i>Chương trình ngắt</i>	89
5.3.3 <i>Cập nhật encoder</i>	91
5.3.4 <i>Điều khiển động cơ</i>	92
5.4 Kết quả	94
CHƯƠNG 6 CÁCH VẬN HÀNH	95
6.1 <i>Cách sử dụng</i>	95
6.2 <i>Bảo dưỡng</i>	97
CHƯƠNG 7 KẾT LUẬN	98

7.1 Những kết quả đạt được	98
7.2 Những kết quả chưa đạt được	98
7.3 Những vấn đề chưa giải quyết	99
7.4 Hướng phát triển	99
TÀI LIỆU THAM KHẢO	100
PHỤ LỤC	102
1. GIỚI THIỆU PHẦN MỀM VISUALNASTRAN	102
2. LỌC THÍCH NGHI – BỘ LỌC KALMAN	105
3. GYRO MURATA ENC-03	118
4. CẢM BIẾN GIA TỐC ACCELEROMETER ADXL202	122
5. CHUẨN TRỰC CÁC CẢM BIẾN ĐO GÓC	128
6. TÍNH NĂNG VI ĐIỀU KHIỂN PIC 18FXX2	131

TÓM TẮT ĐỀ TÀI

Đề tài này có thể xem là một câu nổi kinh nghiệm từ mô hình thăng bằng con lắc ngược đến việc nghiên cứu và chế tạo các loại robot hai chân và robot người (humanoid robot) trong tương lai. Mục tiêu của đề tài là thiết kế và chế tạo một xe hai bánh tự cân bằng, dựa trên lý thuyết cân bằng con lắc ngược. Không giống như các xe scooter hay xe 2 bánh thông thường có hai bánh xe nằm trước sau, xe scooter trong đề tài có hai bánh nằm song song với nhau, giúp nó trở nên cực kỳ gọn gàng để di chuyển bằng những bánh xe trong những khoảng chật hẹp mà thường chỉ có thể đi bộ.

Đề tài này được quan tâm từ việc tính toán các thông số đầu vào và ra, dựa trên đó để xây dựng các mô phỏng, đến việc thiết kế mô hình, thực hiện phần điện tử và điều khiển, viết các chương trình điều khiển với mục đích cuối cùng là tạo ra một mô hình xe di chuyển cân bằng trên hai bánh xe đồng trục được lắp trên hai động cơ dựa theo các định luật cơ học Newton và cơ học vật rắn: điều khiển để luôn duy trì bề mặt chân đế (hai bánh xe) ở vị trí ngay dưới trọng tâm của xe khi đứng yên, và tạo một sai số nhỏ về góc nghiêng của thân xe với nền khi muốn xe chuyển động.

Sự cân bằng được tính toán và mô phỏng bằng 2 phần mềm MatLAB-SIMULINK và Visual Nastran, để chứng minh rằng hoàn toàn có khả năng để điều khiển một mô hình xe tự cân bằng chỉ nhờ một hệ thống điều khiển hoạt động của động cơ điện gắn trên mỗi bánh xe.

Mô hình bao gồm một thân mang hai động cơ DC được tích hợp trong mỗi bánh xe đạp điện 400 mm phổ biến trong thời gian gần đây tại Việt Nam, bo mạch sử dụng bộ điều khiển trung tâm PIC18Fxxx của hãng Microchip để điều khiển những mạch khuếch đại công suất, lái công suất (MOSFET driver) cho những động cơ, điều khiển những cảm biến cần thiết để đo các giá trị góc và quãng đường đi. Các tín hiệu đo góc từ hai cảm biến accelerometer và gyro được thông qua một bộ lọc Kalman được lập trình trên vi điều khiển PIC để có các thông số đo góc chính xác. Bảng điện kiểm soát và hiển thị chức năng hoạt động của xe. Bình điện được lắp dưới sàn xe bằng nhôm để cung cấp toàn bộ năng lượng cho xe hoạt động.

ABSTRACT

This project can be an useful experiment to the research and manufacture in balancing robot and humanoid robot in future. The main purposes of my project are designing and manufacturing a self-balancing scooter, based on the theory of the balancing inverted pendulum. It is unlike the popular scooter or bicycle, which have two wheels being in a same surface (the wheel's axes are parallel). Its parallel wheels configuration make it compact enough to be maneuvered through most pedestrian spaces that accommodate wheelchairs.

Calculating parameters of the model to construct the simulation, designing the model, making electronic boards and controller, and programming the microcontroller are the missions in the project, to reach the main goal of building a scooter that could balance in its two coaxial wheels driven by two intergrated motors. The method analysing the auto-balancing scooter's dynamic is roughly based on Newton's laws and mechanics of solid. To keep the scooter remains balanced when scooter don't move, it must drive the wheels staying under the scooter's gravity, and making a small error in tilt angle (angle of the chassis with respect to the ground) when the scooter moves.

The balance of scooter is also calculated and simulated by MatLAB-SIMULINK and Visual Nastran, to show that it is clearly possible to control such a system using an electric motor mounted on each of the two wheels.

The self-balancing scooter is structured of a chassis carrying two wheels coupled a DC motors for each. The wheel which is used in my final project is a wheel of electric bicycle (400 mm of diameter), lately popular in Viet Nam. PIC18Fxxx, a micro-controller of Microchip's family is used to implement as the main controller of scooter's system, manages the works of the electric power amplifiers, MOSFET driver for the motors and of the necessary sensors to measure the vehicle's states. To have the exact information of angle received from the noisy accelerometer and piezo-electric gyro, a discrete Kalman filter is implemented in PIC microcontroller. A control board is used to display the state of sensors, operation of scooter and to control the speed and steering. Batteries are bolted under the chassis of scooter, supply electric energies for scooter's operation.

Chương 1

TỔNG QUAN

1.1 Lời nói đầu

Bài luận văn xuất phát từ ý tưởng đã được thương mại hóa của công ty Segway: kết hợp ý tưởng về cách giữ thăng bằng của con người trên đôi chân và độ cơ động trong di chuyển của các loại xe di chuyển bằng bánh. Thông qua bài nghiên cứu, ta có thể phần nào nắm bắt những ý tưởng giữ thăng bằng cho các loại humanoid robot (robot dạng người), cách phối hợp và xử lý tín hiệu tốt nhất từ cảm biến. Tuy vậy, giá thành của sản phẩm Segway không rẻ (khoảng 5000USD/xe) do chi phí rất cao từ các cảm biến đã được tích hợp và xử lý với độ chính xác và tin cậy cao (khoảng 900USD/bộ). Do vậy, chúng ta sẽ tìm cách kết hợp các cảm biến riêng lẻ với giá thành thấp (4 - 40USD/cảm biến) và xử lý tín hiệu cảm biến của chúng để có được các tín hiệu tinh khiết và chính xác như mong muốn với giá thành không cao.

Mô hình là một chiếc xe có hai bánh được đặt dọc trục với nhau (khác với xe đạp là trục của hai bánh xe song song). Trên mô hình sử dụng các cảm biến để đo góc nghiêng của thân xe, vận tốc quay (lật) của sàn xe quanh trục bánh và vận tốc di chuyển của xe so với mặt đất. Nhờ các cảm biến này, xe sẽ có thể tự giữ thăng bằng và di chuyển. Với cấu trúc này, trọng tâm của mô hình phải luôn nằm trong vùng đỡ của bánh xe (supporting area) để có thể thăng bằng khi di chuyển ở mọi bề mặt từ đơn giản đến phức tạp.

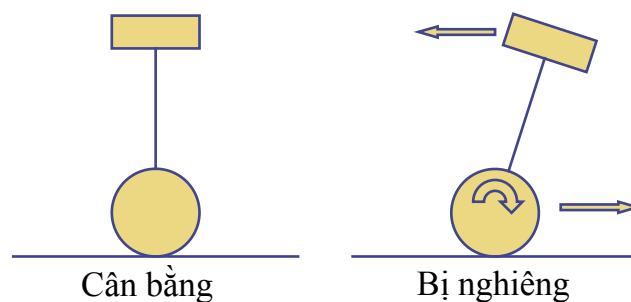
Trong hệ thống các cảm biến, để loại trừ các tín hiệu nhiễu từ hệ thống và nhiễu từ tín hiệu đo, sai số của ngõ ra, đồng thời có thể ước lượng chính xác giá trị đo trong tương lai của cảm biến cũng như kết hợp các tín hiệu, bộ lọc Kalman được nghiên cứu và sử dụng nhằm cho một kết quả tối ưu về tình trạng của xe gồm góc nghiêng, vận tốc quay của xe từ mô hình và các cảm biến thành phần. Nói cách khác, hệ thống xử lý tín hiệu và lọc Kalman là công cụ để biến các cảm biến đơn giản, giá rẻ thành tập hợp cảm biến có giá trị trong hệ thống. Từ các tín hiệu đo, thông qua một số đại lượng đặc trưng của mô hình (khối lượng, chiều dài, chiều cao vật, đường kính bánh...) ta sẽ tính được momen quán tính nghiêng (lật của mô hình), từ đó đưa ra các giá trị điều khiển phù hợp cho các bánh xe để giữ cho mô hình luôn đứng vững hoặc di chuyển với một vận tốc ổn định.

Toàn bộ mô hình được điều khiển bằng một vi điều khiển PIC 18F452. Đây là thế hệ tương đối cao cấp của họ PIC có thể xử lý và thực thi chương trình ở tốc độ cao (đạt đến 10MIPs) trong việc tính toán các giá trị cảm biến và đưa ra bộ truyền động (động cơ điện). Bộ vi điều khiển đóng vai trò thứ nhất trong đề tài như một bộ lọc Kalman với tín hiệu vào từ thiết bị inclinometer và gyro. Với các dữ liệu về góc đã xử

lý và tín hiệu hồi tiếp về vị trí do encoder đưa về (incremental encoder), vai trò thứ hai của vi điều khiển trong đề tài sẽ tính toán và đưa ra tín hiệu điều khiển bộ truyền động, đến bánh xe để giữ thăng bằng/di chuyển, đi thẳng, quay, quẹo.

Đây là một phương tiện vận chuyển mới tại các thành phố trong tương lai với nhiều ưu điểm: gọn, nhẹ, ít chiếm diện tích đường phố, dễ mang vác, tháo lắp và vận chuyển, nhiên liệu sạch, dễ điều khiển cho người lớn và trẻ em, đi được trên một số địa hình phức tạp.

1.2 Thế nào là xe hai bánh tự cân bằng (two wheels self balancing)

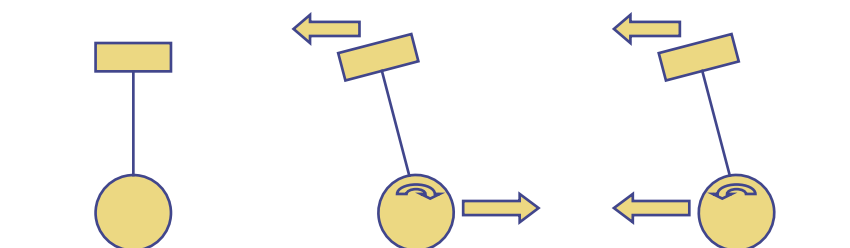


Hình 1.1 Mô tả nguyên lý giữ thăng bằng

Đối với các xe ba hay bốn bánh, việc thăng bằng và ổn định của chúng là nhờ trọng tâm của chúng nằm trong bề mặt chân đế do các bánh xe tạo ra. Đối với các xe 2 bánh có cấu trúc như xe đạp, việc thăng bằng khi không di chuyển là hoàn toàn không thể, vì việc thăng bằng của xe dựa trên tính chất con quay hồi chuyển ở hai bánh xe khi đang quay. Còn đối với xe hai bánh tự cân bằng, là loại xe chỉ có hai bánh với trục của hai bánh xe trùng nhau, để cho xe cân bằng, trọng tâm của xe (bao gồm cả người sử dụng chúng) cần được giữ nằm ngay giữa các bánh xe. Điều này giống như ta giữ một cây gậy dựng thẳng đứng cân bằng trong lòng bàn tay.

Thực ra, trọng tâm của toàn bộ scooter không được biết nằm ở vị trí nào, cũng không có cách nào tìm ra nó, và có thể không có khả năng di chuyển bánh xe đủ nhanh để giữ nó luôn ở dưới toàn bộ trọng tâm.

Về mặt kỹ thuật, góc giữa sàn scooter và chiều trọng lực có thể biết được. Do vậy, thay vì tìm cách xác định trọng tâm nằm giữa các bánh xe, tay lái cần được giữ thẳng đứng, vuông góc với sàn xe (góc cân bằng khi ấy là zero).



Hình 1.2 Mô tả cách bắt đầu di chuyển

Nếu tay lái được đẩy hơi nghiêng tới trước, scooter sẽ chạy tới trước và khi nó được đẩy nghiêng ra sau, scooter sẽ chạy lùi. Đây là một phân tích lý tính. Hầu hết mọi người đều có thể kiểm soát tay lái trong vòng vài giây để giữ lấy nó.

Để dừng lại, chỉ cần kéo trọng tâm xe nghiêng ngược hướng đang di chuyển thì tốc độ xe giảm xuống. Do tốc độ cảm nhận và phản ứng thăng bằng của mỗi người là khác nhau, nên xe scooter hai bánh tự cân bằng chỉ được thiết kế cho một người sử dụng.

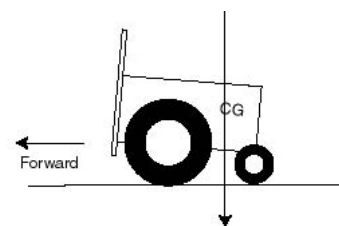
1.3 Tại sao phải thiết kế xe hai bánh tự cân bằng

Những mobile robot xây dựng hầu hết robot là những robot di chuyển bằng ba bánh xe, với hai bánh lái được lắp ráp đồng trục, và một bánh đuôi nhỏ. Có nhiều kiểu khác nhau, nhưng đây là kiểu thông dụng nhất. Còn đối với các xe 4 bánh, thường một đầu xe có hai bánh truyền động và đầu xe còn lại được gắn một hoặc hai bánh lái.

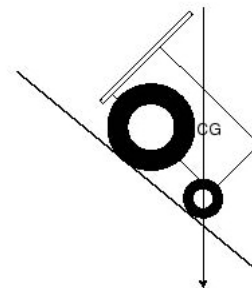
Việc thiết kế ba hay bốn bánh làm cho xe/mobile robot được thăng bằng ổn định nhờ trọng lượng của nó được chia cho hai bánh lái chính và bánh đuôi, hay bất kỳ cái gì khác để đỡ trọng lượng của xe. Nếu trọng lượng được đặt nhiều vào bánh lái thì xe/robot sẽ không ổn định dễ bị ngã, còn nếu đặt nhiều vào bánh đuôi thì hai bánh chính sẽ mất khả năng bám. Nhiều thiết kế xe/robot có thể di chuyển tốt trên địa hình phẳng, nhưng không thể di chuyển lên xuống trên địa hình lồi lõm (mặt phẳng nghiêng). Khi di chuyển lên đồi, trọng lượng xe/robot dồn vào đuôi xe làm bánh lái mất khả năng bám và trượt ngã, đối với những bậc thang, thậm chí nó dừng hoạt động và chỉ quay tròn bánh xe.

Khi di chuyển xuống đồi, sự việc còn tệ hơn, trọng tâm thay đổi về phía trước và thậm chí làm xe/robot bị lật úp khi di chuyển trên bậc thang. Hầu hết những xe/robot này có thể leo lên những dốc ít hơn là khi chúng di chuyển xuống, bị lật úp khi độ dốc chỉ 15° hay 20° . Việc bố trí bốn bánh xe, giống như xe hơi đồ chơi hay các loại xe bốn bánh hiện đang sử dụng trong giao thông không gặp vấn đề nhưng điều này sẽ làm các mobile robot không gọn gàng và thiết kế bộ phận lái (cua quẹo) gặp một chút phiền toái để có thể xác định chính xác quãng đường đã đi ^[16].

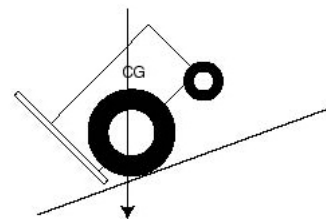
Ngược lại, các xe dạng hai bánh đồng trục lại thăng bằng rất linh động khi di chuyển trên địa hình phức tạp, mặc dù bản thân là một hệ thống không ổn định. Khi nó leo sườn dốc, nó tự động nghiêng ra trước và giữ cho trọng lượng dồn về hai bánh lái



3-wheel robot center of gravity (CG) on level surface. Weight distributed between main drive wheels and castor tail wheel.



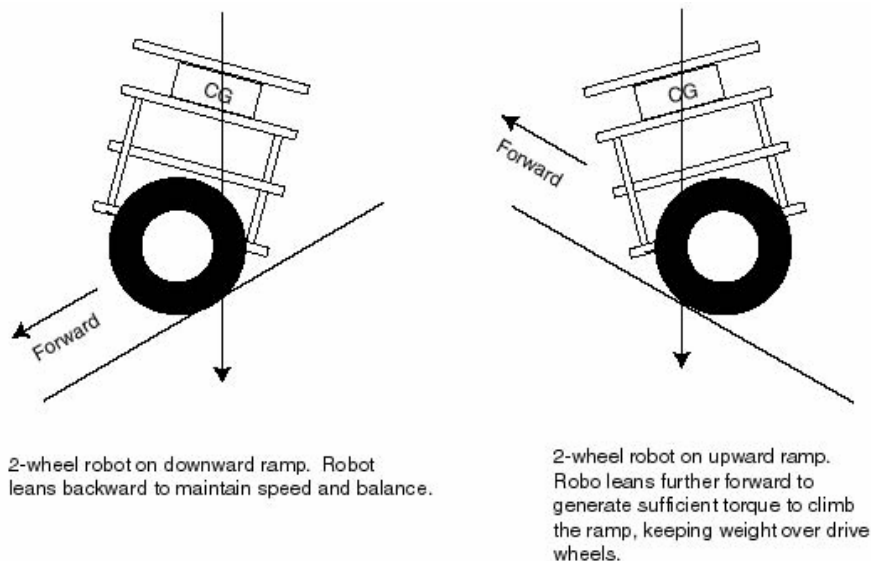
3-wheel robot on upward ramp. Center of gravity now situated over rear wheel, causing main wheels to lose traction.



3-wheel robot on downward ramp. Center of gravity now shifts ahead of main drive wheel axle, causing robot to tip over forward.

Hình 1.3 Trạng thái xe ba bánh khi di chuyển trên địa hình bằng phẳng, dốc ^[16]

chính. Tương tự vậy, khi bước xuống dốc, nó nghiêng ra sau và giữ trọng tâm rơi vào các bánh lái. Chính vì vậy, không bao giờ có hiện tượng trọng tâm của xe rơi ra ngoài vùng đỡ của các bánh xe để có thể gây ra sự lật úp.



Hình 1.4 Trạng thái xe hai bánh đồng trục khi di chuyển trên địa hình bằng phẳng, dốc^[16]

Đối với những địa hình lồi lõm và những ứng dụng thực tế, sự thăng bằng của xe hai bánh có thể sẽ mang lại nhiều ý nghĩa thực tiễn trong giới hạn ổn định hơn là đối với xe ba bánh truyền thống.

1.4 Ưu nhược điểm của xe hai bánh tự cân bằng

1.4.1 Ưu điểm của xe scooter tự cân bằng trên hai bánh

- Không ô nhiễm, sử dụng bình điện, và có thể sạc điện.
- Sử dụng không gian hiệu quả, đa năng (sử dụng trong nhà và ngoài phố).
- Dễ dàng lái xuống đường, dừng lại và trò chuyện với bạn bè. Scooter tự cân bằng này khác hẳn với các loại xe đạp hay xe đẩy bình thường, vì chúng dễ kéo đẩy và không gây khó khăn khi dừng lại.
- Khá dễ để lái vòng quanh trong văn phòng, chạy ngang qua cửa ra vào do tốc độ thấp. Ngoài ra, nó còn có thể xuống các bậc thềm/ bậc thang thấp.
- Chiếm ít diện tích (chỉ hơn một con người) nên nó không gây tắc nghẽn giao thông như các loại xe bốn bánh. Như một phương tiện vận chuyển trên vỉa hè, nó cho phép di chuyển trong nơi đông đúc, và hoàn toàn có thể đi trên lòng đường.
- Giá thành thấp hơn so với xe hơi.
- Cuốn hút người sử dụng cũng như mọi người xung quanh vì hình dáng kỳ lạ của nó, phá vỡ các hình ảnh thường thấy về các phương tiện giao thông của con người.

1.4.2 Nhược điểm của xe

- Không thể thư giãn và khá mệt khi lái do phải đứng trong khi điều khiển. Vì đứng trên mặt sàn rung (do động cơ gây ra) và cứng làm chân mỏi. Do luôn giữ tư thế thẳng đứng để trọng lượng cơ thể đặt ở trọng tâm và đôi lúc gặp những đoạn đường xấu khiến cơ thể người điều khiển mệt mỏi.
- Không thể làm các việc khác khi đứng trên scooter này, chẳng hạn vừa đi vừa nghe điện thoại, hoặc vừa uống nước.
- Scooter không đủ nhanh để đi đường trường và không đủ an toàn để lên xuống lề đường.
- Không thể vận chuyển hai người trên cùng một xe. Việc này không thành vấn đề khi xe tự cân bằng đóng vai trò một platform của mobile robot, vì khối lượng tải là tĩnh.
- Không thể leo bậc thang có chiều cao quá $\frac{1}{2}$ bán kính bánh xe.

1.5 Khả năng ứng dụng

Xây dựng được một phương tiện vận chuyển mới trong khu vực chật hẹp có thể di chuyển ngay trong các chung cư tòa nhà cao tầng, dùng trợ giúp di chuyển cho người già, và trẻ em vận chuyển.

Làm phương tiện vận chuyển hàng hoá đến những nơi đã được lập trình sẵn ở trong các tòa nhà, phòng làm việc, những không gian chật hẹp, khó xoay trở.

Thậm chí kết hợp trên các humanoid robot, nếu được kết hợp với các robot camera, robot dò đường, robot lái mặt đường thì hiệu quả các công dụng cụ thể cực kỳ linh hoạt. Tuy vậy, cần phải tiến hành giải quyết thêm về phần xuống cầu thang (không thể leo lên các bậc thang cao).

1.6 Tình hình nghiên cứu trong và ngoài nước

Hiện nay chưa có thông tin cụ thể nào về việc chế tạo xe hai bánh tự cân bằng dùng trên robot cũng như xe hai bánh tự cân bằng ở Việt Nam. Nhưng trên thế giới, ở một vài nước, các kỹ thuật viên và một số sinh viên đã nghiên cứu và cho ra đời các dạng xe hai bánh như thế. Dưới đây là một số thông tin về chúng.

1.6.1 Một số dạng xe hai bánh tự cân bằng dùng trên robot

1.6.1.1 *nBot*^[16]

nBot do David P. Anderson sáng chế. nBot được lấy ý tưởng để cân bằng như sau: các bánh xe sẽ phải chạy xe theo hướng mà phần trên robot sắp ngã. Nếu bánh xe có thể được lái theo cách



Hình 1.5 nBot

đứng vững theo trọng tâm robot, robot sẽ vẫn được giữ cân bằng. Trong thực tế, điều này đòi hỏi hai cảm biến thông tin phản hồi: cảm biến góc nghiêng để đo góc nghiêng của robot với trọng lực, và encoder trên bánh xe để đo vị trí cơ bản của robot. Bốn thông số ngõ vào để xác định hoạt động và vị trí của xe con lắc ngược cân bằng là:

- 1) góc nghiêng.
- 2) đạo hàm của góc nghiêng, vận tốc góc.
- 3) vị trí bánh xe.
- 4) đạo hàm vị trí bánh, vận tốc bánh xe.

Bốn giá trị đo lường được cộng lại và phản hồi tới điện áp động cơ, tương ứng với momen quay, cân bằng, và bộ phận lái robot.

1.6.1.2 Balance bot I^[28]

Balance-bot I (do Sanghyuk, Hàn Quốc thực hiện) là một robot hai bánh tự cân bằng bằng cách kiểm soát thông tin phản hồi. Hệ thống cao 50cm. Khung chính được làm bằng nhôm. Nó có hai trục bánh xe nối với hộp giảm tốc và động cơ DC cho sự phát động. Tổng cộng có ba bộ vi xử lý Atmel được sử dụng. Vi điều khiển chính (master) thi hành những nguyên lý kiểm soát và thuật toán ước lượng. Một vi điều khiển khác kiểm soát tất cả cảm biến analog. Vi điều khiển thứ ba điều khiển động cơ DC.

Linear quadratic regulator (LQR) được thiết kế và thực thi mạch điều khiển. Nó có bốn giá trị khác nhau – góc nghiêng, vận tốc góc nghiêng, góc quay bánh xe, và vận tốc góc quay, sau đó nó tạo lệnh cho động cơ DC để điều chỉnh tốc độ bánh xe.



Hình 1.6 Balance-bot

1.6.1.3 Balancing robot (Bbot^[26])

Vào năm 2003, Jack Wu và Jim Bai là những sinh viên trường Đại học Carnegie Mellon dưới sự trợ giúp của GS. Chris Atkeson đã thực hiện đề tài robot hai bánh tự cân bằng như luận văn tốt nghiệp. Robot này có thể xác định vị trí hướng của nó đối với môi trường và lái động cơ theo hướng này.

Để đo góc nghiêng của robot, các sinh viên này đã sử dụng hệ thống đo lường góc 2DOF được tích hợp sẵn của hãng Rotomotion. Hệ thống này gồm gia tốc kế



Hình 1.7 Balancing robot

ADXL202 và mạch con quay hồi chuyển. Vi mạch điều khiển dùng trên robot này là BasicX 24, có nhiều tính năng khác nhau. Nó được dùng như bộ điều khiển động cơ, COM1 được nối với Pocket PC và COM3 thì nối với bộ điều khiển servo Mini SSC 12. Nó còn được sử dụng như CPU chính cho việc điều khiển thăng bằng cho robot.

1.6.1.4 JOE^[18]

Phòng thí nghiệm điện tử công nghiệp của Viện Công nghệ Federal, Lausanne, Thụy Sĩ, đã tạo ra cuộc cách mạng đầu tiên khi xây dựng mô hình xe hai bánh. Robot JOE cao 65cm, nặng 12kg, tốc độ tối đa khoảng 1,5m/s, có khả năng leo dốc nghiêng đến 30°. Nguồn điện cấp là nguồn pin 32V khả năng 1,8Ah.

Hình dạng của nó gồm hai bánh xe trục, mỗi bánh gắn với một động cơ DC, chiếc xe này có thể chuyển động xoay theo hình U. Hệ thống điều khiển được lắp từ hai bộ điều khiển state-space tách rời nhau, kiểm soát động cơ để giữ cân bằng cho hệ thống. Những thông tin về trạng thái của JOE được cung cấp bởi hai encoder quang và vận tốc của con quay hồi chuyển.

JOE được điều khiển bởi một bộ điều khiển từ xa R/C thường được sử dụng để điều khiển các máy bay mô hình. Bộ điều khiển trung tâm và xử lý tín hiệu là một board xử lý tín hiệu số (DSP) được phát triển bởi chính nhóm và của viện Federal, có khả năng xử lý dấu chấm động (SHARC floating point), FPGA XILINX, 12 bộ biến đổi A/D 12bit và 4 bộ biến đổi D/A 10bit.

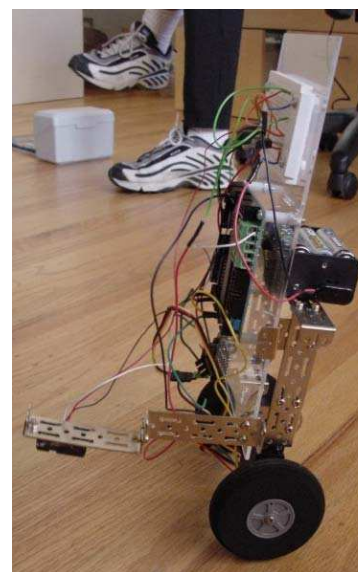


Hình 1.8 Hình chụp JOE

1.6.1.5 Equibot^[27]

Equibot là robot cân bằng do Dan Piponi thực hiện. Cơ bản nó dựa vào vi điều khiển ATmega32 RISC.

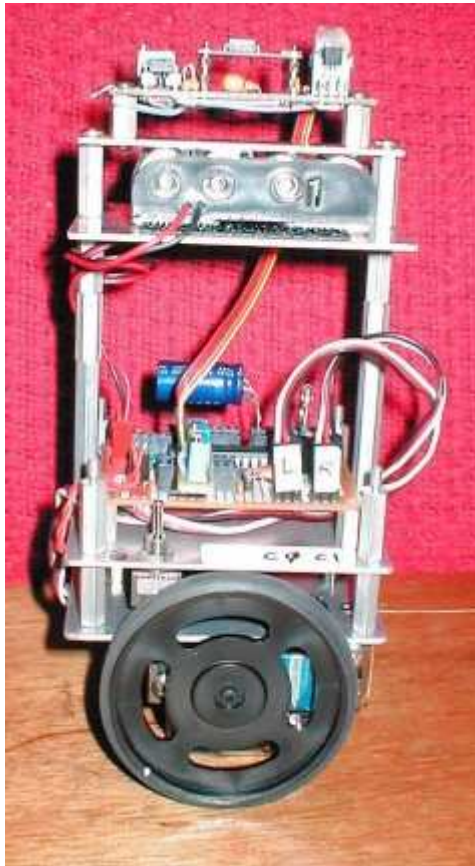
Cả hai servo Hitec HS-311 chuẩn được sửa đổi cho xoay vòng 360° và nguồn điện vào được nối trực tiếp với các động cơ để PWM kiểm soát chúng. Một trong hai servo được gắn với bộ điều khiển tứ cực LQR, đó là phần phức tạp nhất trong cấu trúc robot, bánh còn lại bắt chước tốc độ của bánh thứ nhất. Equibot chỉ có một loại cảm biến hồng ngoại Sharp thay cho cảm biến về góc. Nó được đặt thấp để đo khoảng cách với sàn. Ngõ ra từ thiết bị được dùng để xác định hướng robot di chuyển.



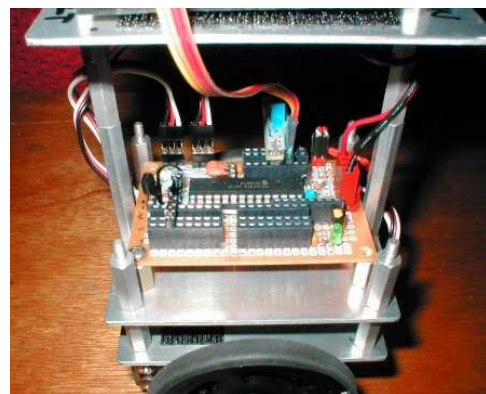
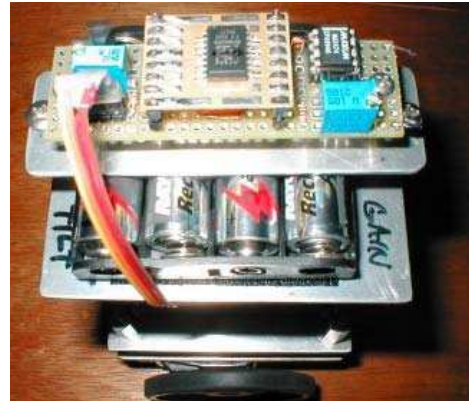
Hình 1.9 Equibot

1.6.1.6 BaliBot^[29]

Balibot, một robot hai bánh tự cân bằng, là một trong các mẫu đầu tiên về robot hai bánh có trọng tâm phía trên các bánh xe. Không có hệ thống điều khiển hoạt động, robot sẽ bị ngã. Khi robot có nhận biết hướng mà nó sắp ngã, các bánh xe sẽ di chuyển về phía ngã và thẳng góc với chính nó.



Hình 1.10 Balibot



Hình 1.11 Các tầng mạch, gồm nguồn, vi điều khiển và cảm biến

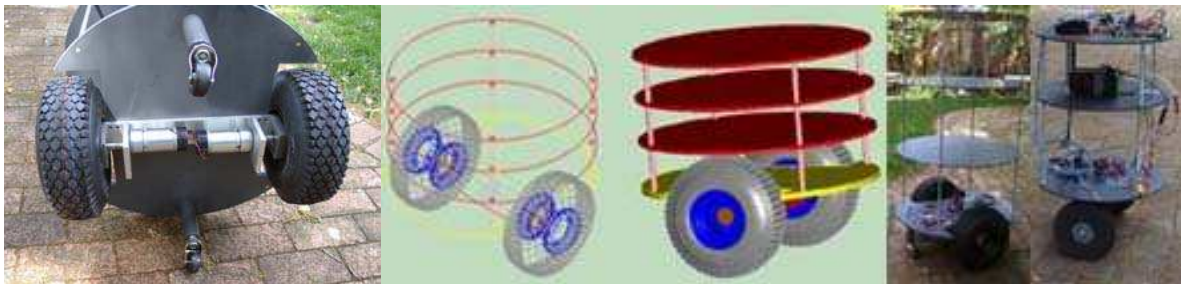
Cảm biến góc nghiêng để đo góc nghiêng của robot, gia tốc kế Motorola MMA2260 được sử dụng, thiết bị có cấu trúc MEMS.

PIC16F876 của hãng Microchip[©] được chọn làm trung tâm điều khiển cho robot. PIC tích hợp một bộ biến đổi A/D nhiều kênh để đo cảm biến góc nghiêng và các ngõ I/O để kiểm soát hai servo được mô tả cho sự quay vòng tiếp theo. Điện được cung cấp bằng bốn cục pin AA và được ổn áp dropout. Nguồn điện 6V không qua ổn áp được phân phối đến động cơ servo qua tụ điện 3300 μ F qua bù năng lượng cho vi mạch điều khiển khi công suất ngõ ra từ các servo được hoạt động. Mạch điện tử được xây dựng trên bảng project board Radio Shack RS 276-150 và lắp ráp phía trên các motor servo, trên khung bằng nhôm. Nguồn điện được đặt gần đỉnh và hoạt động như trọng lượng

của con lắc ngược. Một phiên bản khác của BaliBot sử dụng các cảm biến hồng ngoại để đo khoảng cách thay vì dùng các cảm biến đo góc.

1.6.1.7 Bender^[21]

Robot cân bằng Bender là đề án do TedLarson, San Francisco thực hiện. Mục tiêu hiện tại của ông là xây dựng robot tự cân bằng trên mặt sàn, và từ đó dùng làm nền cơ bản (platform) để xây dựng robot tự hành dùng bánh xe.



Hình 1.12 Hình chụp robot Bender

1.6.1.8 Loại Robot phục vụ con người, kiểu rolling phục vụ con người của hãng TOYOTA



Hình 1.13 Loại robot, kiểu Rolling của TOYOTA

Đây là một trong những loại robot có công dụng phục vụ cho con người do hãng TOYOTA thiết kế. Nó cao 100cm và nặng 35kg. Mẫu robot này có khả năng di chuyển nhanh mà không chiếm một không gian lớn, đồng thời đôi tay của nó có thể làm nhiều công việc khác nhau, chủ yếu được dùng làm trợ lý trong công nghiệp.

1.6.2 Một số dạng scooter hai bánh tự cân bằng

1.6.2.1 Segway^[33]

Không giống như một chiếc xe hơi, Segway chỉ có hai bánh – trông nó như một chiếc xe đẩy bằng tay thông thường – nó còn kiểm soát hoạt động ở tư thế thẳng đứng.

Để di chuyển đến trước hay lùi ra sau, người lái đứng trên Segway chỉ việc hơi nghiêng về phía trước hay phía sau. Để quẹo trái hay phải, người lái quay tay lái qua phải hướng ra trước hay ra sau.

Hoạt động cân bằng ở Segway là một điều thú vị nhất, đó là chiếc chìa khóa của quá trình hoạt động. Xem xét về mô hình Karmen về thăng bằng của cơ thể người để hiểu hệ thống làm việc như thế nào. Nếu ta đứng và nghiêng người về phía trước, không còn thăng bằng, bạn sẽ ngã về trước. Bộ não biết rằng bạn không còn thăng bằng nữa, bởi vì chất dịch trong tai trong dao động, nên nó truyền tín hiệu ra lệnh cho chân bạn đặt lên phía trước và bạn lấy lại thăng bằng. Nếu bạn giữ mình trong trạng thái nghiêng về trước, bộ não điều khiển chân bạn đặt lên trước và giữ bạn đứng thẳng. Thay vì ngã, bạn bước đến trước.



Hình 1.14 Segway

Segway tạo ra khá giống như vậy, ngoại trừ nó có bánh xe thay vì đôi chân, động cơ thay cho bắp cơ, tập hợp các vi mạch xử lý thay cho một bộ não và một dãy các cảm biến nghiêng thay cho hệ thống cân bằng tai trong. Như bộ não của bạn, Segway nhận biết khi ta hướng về trước. Để duy trì cân bằng, nó quay bánh xe đến trước chỉ với tốc độ vừa phải (chính xác), nên ta di chuyển đến trước.

Sự phân chia rõ ràng (con quay hồi chuyển chính yếu – trạng thái cân bằng). Khối Segway được lắp đặt nhiều hơn hai bánh xe. Thiết bị lái tận dụng cả công nghệ drive-by-wire và thiết bị cơ khí có hệ thống. Trong khi việc thiết kế bốn bánh đưa đến vận động dễ dàng và tốc độ cao hơn một tí, người lái có thể chọn lựa giữa việc sử dụng bốn bánh hay chỉ hai bánh xe.

Điều cơ bản nhất, Segway là sự kết hợp của một dãy các cảm biến, một hệ thống kiểm soát và một hệ thống động cơ.

Hệ thống cảm biến chủ yếu là sự kết hợp các con quay hồi chuyển (gyroscope). Một con quay hồi chuyển cơ học cơ bản là một bánh xe quay tròn bên trong cơ cấu vững chắc. Mục đích sự quay tròn nhằm kháng lại sự thay đổi trục quay của nó, bởi vì lực tác động di chuyển dọc theo cơ cấu. Nếu ta đẩy một điểm trên bánh xe quay, ví dụ, điểm này di chuyển quanh bánh trước trong khi nó vẫn còn giữ lực tác động. Khi một điểm lực giữ di chuyển, nó kết thúc lực tác dụng đối diện với điểm cuối của bánh xe – không còn cân bằng lực.

Bởi vì nó kháng đối với lực bên ngoài, bánh xe quay hồi chuyển sẽ duy trì vị trí của nó trong không gian (liên hệ với mặt đất) thậm chí nếu bạn nghiêng nó đi. Nhưng hệ thống con quay hồi chuyển sẽ di chuyển tự do trong không gian. Bằng việc đo lường vị trí của bánh xe quay liên hệ với cơ cấu, cảm biến chính xác có thể cho ta biết độ dốc của vật (nó nghiêng bao nhiêu so với vị trí thẳng đứng) cũng như tốc độ dốc (nó nghiêng nhanh như thế nào).

Một con quay hồi chuyển thông thường sẽ cồng kềnh và khó bảo dưỡng xe, nên Segway tiếp thu hiệu quả này với hình thức khác của cơ khí. Segway vận dụng một cảm biến tốc độ nghiêng bán dẫn đặc biệt được tạo từ silic. Loại con quay hồi chuyển này quy định sự quay vòng của vật thể sử dụng hiệu ứng Coriolis trên một lớp rất nhỏ.

Segway HT có năm cảm biến hồi chuyển, mặc dù nó chỉ cần ba cảm biến để phát hiện ra mức đẩy ra trước và ra sau cũng như nghiêng bên trái hay bên phải. Các cảm biến còn lại làm cho phương tiện chắc chắn hơn. Thêm vào đó, Segway có hai cảm biến nghiêng chứa đầy dung dịch điện phân. Giống như tai trong, hệ thống nhận biết vị trí nghiêng có liên hệ với mặt đất trong trạng thái nghiêng của bề mặt chất dịch.

Tất cả thông tin về trạng thái nghiêng truyền đến “bộ não” của xe, hai bảng mạch điều khiển điện tử bao gồm một bộ vi mạch xử lý. Segway có tổng cộng 10 bảng mạch vi xử lý, với năng lực gấp ba lần năng lực PC điển hình. Thông thường cả hai bảng mạch làm việc chung với nhau nhưng nếu một bảng bị hư, bảng còn lại nhận tất cả các chức năng để hệ thống báo tín hiệu cho người lái biết sự trục trặc để khởi động lại.

Segway đòi hỏi năng lực làm việc cao của bộ não vì nó cần điều chỉnh cực kỳ chính xác để giữ không bị ngã. Trong những máy thông thường, bảng mạch điều khiển kiểm tra vị trí cảm biến khoảng 100 lần/giây. Mạch vi xử lý điều hành phần mềm tương thích để phát tín hiệu tất cả các thông tin ổn định và điều chỉnh tốc độ cho nhiều động cơ điện phù hợp. Động cơ điện được nạp năng lượng từ một cặp pin (Ni-MH) có thể sạc lại, làm quay độc lập mỗi bánh xe với tốc độ khác nhau.

Khi xe nghiêng về trước, động cơ làm cả hai bánh xe quay về trước và giữ về trạng thái nghiêng. Khi xe nghiêng ra sau, động cơ làm cả hai bánh xe quay ra sau. Khi người lái điều khiển tay lái quẹo trái hay phải, động cơ làm một trong hai bánh xe quay nhanh hơn bánh xe kia hay hai bánh xe quay ngược chiều để xe xoay quanh.

Nó chỉ đi khoảng 12 dặm/giờ (20km/giờ), và nó cần nạp điện khoảng 6 giờ để dự trữ dùng đủ cho một chuyến đi 15 dặm (24km).

Segway là sự lựa chọn cao trong thành phố. Vì các xe hơi đắt tiền và nếu có lượng lớn xe hơi chạy trên đường phố sẽ gây nên ùn tắc giao thông, và thiếu chỗ đậu xe. Tất cả những điều ấy, xe hơi không là phương tiện tối ưu nhất trong khu dân cư đông đúc.

Segway không thể đưa con người đi đến nơi muốn đến với tốc độ cao nhất, nhưng Segway có thể đi bằng sự di chuyển chậm, nối đuôi nhau. Một khi chúng đến nơi, người lái có thể mang Segway vào bên trong mà không phải lo lắng gì về chỗ đậu xe. Và cũng không cần dừng ở những trạm xăng dầu, mà chỉ cần nạp điện cho xe tại nhà.

Segway cũng là chiếc máy tốt dùng để đi trong các kho hàng, nơi có nhiều hành lang. Người ta còn nhận thấy sự hữu dụng khi đi quanh trong các khu dân cư, sân bay hay công viên. Thật sự không có giới hạn không gian trong việc sử dụng xe. Segway giúp bạn đi nhanh hơn mà không mất nhiều năng lượng.

- Tốc độ cao nhất: 12,5 dặm/giờ (20 km/giờ). Gấp ba lần tốc độ đi bộ bình thường.
- Trọng lượng không tải: 80 lbs (36 kg).
- Chiều rộng: không gian bao phủ trên mặt đất của Segway là 19 – 25 inch (48 – 63,5 cm). Segway có chiều rộng gần bằng kích thước của một người trung bình, nên nó không mất nhiều diện tích trên đường. Bàn đạp dài 8 inch (20 cm).
- Tải trọng: một người nặng 250 pound (110kg) với hàng hóa nặng 75 pound (34kg).
- Phạm vi: đi khoảng 17 dặm (28 km) với một bình sạc đơn. Trên mô hình tính toán, người thiết kế ước tính xe đi trong phạm vi 11 dặm (17 km) với một bình sạc đơn.
- Giao diện hiển thị xe hoạt động: Segway có màn hình LCD nhỏ cho người lái biết năng lượng pin còn bao nhiêu và hoạt động của xe như thế nào, còn tốt không. Màn hình trình bày như bề mặt hoạt hình, biểu diễn trạng thái chung của phương tiện.

1.6.2.2 *Balancing scooter*^[17]

Trevor Blackwell chế tạo ra xe scooter dựa theo Segway của hãng Mỹ. Xe scooter tự cân bằng này được xây dựng từ những bộ phận giống động cơ xe lăn và từ các cục pin xe RC. Những bộ phận và module để chế tạo có giá thành thấp hơn phân nửa Segway. Nó không cần phần mềm thực thi cao hay phức tạp. Phiên bản đầu tiên được viết trong Python và sử dụng port số để truyền thông tin đến con quay hồi chuyển và mạch điều khiển động cơ.

Xe được sử dụng vi điều khiển 8-bit từ Atmel, chạy trên code C với một số điểm trôi. Nó gửi những lệnh kiểm soát tốc độ ra port serial khoảng 9600 baud trong ASCII đối với bộ phận lái động cơ, có giá 10USD do Digikey tạo. Một con quay hồi chuyển ceramic và gia tốc kế hai trục để điều chỉnh hướng chính xác, cùng hoạt động với vi mạch điều khiển Atmel, với giá 149USD do Rotomotion tạo ra.



Hình 1.15 Xe 2 bánh tự cân bằng của Trevor Blackwell

1.6.2.3 *HTV*^[18]

Nhóm sinh viên ngành kỹ thuật HTV của trường đại học Camosun gồm các thành viên Brian Beckwith, Eric Desjardins, Chris Howard, Joel Murphy, Matt Uganecz, Jack Woolley đến từ các bang khác nhau Victoria, British Columbia của Canada. Tháng 3/2004, họ đã cho ra đời sản phẩm scooter HTV như một đề án tốt nghiệp đại học của họ.



Hình 1.16 Xe tự cân bằng HTV và nhóm thực hiện

Nhóm HTV đã sử dụng ADXR150EB từ thiết bị analog đo vận tốc góc nghiêng. Đó là một gyro tuyệt vời, có các tính năng, như: loại bỏ độ rung cao, tỉ số cao $\pm 150^\circ/\text{s}$, độ nhạy cao $12\text{mV}/\text{deg}/\text{s}$, được cài đặt sẵn tín hiệu điều kiện. MMA2260D từ Motorola, một gia tốc kế có độ nhạy cao ($1200\text{mV}/\text{g}$), và cũng được cài đặt sẵn tín hiệu điều kiện, dùng đo góc nghiêng tĩnh. Bộ điều khiển sử dụng Logic mờ (Fuzzy Logic) trong việc điều khiển cân bằng và di chuyển của xe.

1.6.2.4 Spider^[20]

Francisco Lobo cho ra đời Spider vào cuối tháng 2/2004, trông giống là scooter hơn là robot, tuy nhiên nó có ứng dụng trong cả hai lĩnh vực. Nó có thể giữ cân bằng hầu như ở mọi tình huống, di chuyển, lượn vòng quanh. Scooter được điều khiển bằng hai động cơ của hãng NPC và gia tốc kế hai trục bằng thiết bị analog, chứa hai thành phần chính: Gyro kỹ thuật cảm biến silicon và BasicX (vi điều khiển). Khung xe được chế tạo từ khung nhôm và sợi carbon. Bộ lái MOSFET động cơ lái là module từ Roboteq được dùng trên robot chiến đấu. Nguồn điện là loại dùng trong mô hình RC (NiMh 3000mAh).



Hình 1.17 Spider

1.7 Nhu cầu thực tế

Hiện tại, trong điều kiện đường xá giao thông ngày càng chật hẹp, không khí ngày càng ô nhiễm, việc nghiên cứu và chế tạo một mô hình xe điện gọn nhẹ, dễ xoay

xở, không sử dụng nhiên liệu đốt trong là một nhu cầu thực sự. Bên cạnh đó, thiết kế một platform cho mobile robot cũng là một đề tài cần thiết trong lĩnh vực tự động hóa ngày nay, nhằm trợ giúp cho trẻ em, người già, vận chuyển hàng hóa, giám sát ... trong cuộc sống hàng ngày vốn có nhiều nhu cầu trong việc đi lại và vận chuyển tại các thành phố lớn.

Về khía cạnh khoa học và công nghệ, mô hình xe hai bánh tự cân bằng thực sự là một bước đệm quan trọng để có kinh nghiệm trong việc tính toán, mô hình và chế tạo các robot hai chân (biped-robot, humanoid robot), là đỉnh cao về khoa học và công nghệ mà các trường đại học trên toàn thế giới mong muốn vươn tới. Ngoài ra, mô hình cũng sẽ là sự bổ sung cần thiết về các giải pháp công nghệ di chuyển của các mobile robot 3 bánh, 4 bánh cũng như mobile robot có chân, làm phong phú những lựa chọn giải pháp để chuyển động trong không gian cho các robot.

Về yếu tố tâm lý con người, mô hình xe hai bánh tự cân bằng thực sự là một dấu chấm hỏi lớn cho những người từng thấy hay dùng nó: tại sao có thể di chuyển và thăng bằng được? Điều này cuốn hút nhu cầu được sử dụng một chiếc xe hai bánh tự cân bằng. Và đó chính là lý do của sự thành công lớn trên thế giới của mô hình xe Segway trong năm 2003.

Với những lý do khách quan như đã nêu, đề tài có lẽ có một nhu cầu nhất định trong tình hình hiện nay của Việt Nam cũng như toàn thế giới.

Chương 2

NHIỆM VỤ LUẬN VĂN

2.1 Mục tiêu đề tài

Mục tiêu của đề tài là xây dựng phương tiện xe hai bánh cân bằng di chuyển trên địa hình phẳng, dựa trên nền tảng lý thuyết mô hình con lắc ngược. Khả năng di chuyển cân bằng trên hai bánh làm phương tiện di chuyển hiệu quả và linh động hơn, dễ dàng xoay trở trong điều kiện không gian chật hẹp. Trong khuôn khổ 16 tuần thực hiện luận văn tốt nghiệp đại học, những mục tiêu của đề tài được đề ra như sau:

- Tìm hiểu về các loại scooter, nguyên lý cơ bản về cân bằng.
- Tính toán các tham số động lực học, hàm trạng thái (space-state) của mô hình.
- Xây dựng mô phỏng trên MSc Nastran và Matlab 7-Simulink.
- Thiết kế bản vẽ, chế tạo mô hình theo kích thước thực, có thể vận chuyển một người trưởng thành.
- Thiết kế mạch điều khiển trung tâm, làm nhiệm vụ xử lý tín hiệu đo và đưa ra các quyết định điều khiển.
- Thiết kế mạch điện tử kết hợp các cảm biến thực hiện chức năng đo góc (phần cứng).
- Thiết kế mạch lái các MOSFET công suất cho hai động cơ (MOSFET driver) có khả năng hoạt động ở tần số từ 7-15KHz.
- Giải thuật cho vi điều khiển kết hợp và bù trừ các cảm biến để có được giá trị đo góc chính xác.
- Xây dựng thuật toán điều khiển cho động cơ, giữ thăng bằng và ngăn ngừa quá tải của các bánh xe.
- Lập trình điều khiển.

2.2 Phương pháp nghiên cứu

- Xây dựng mô hình lý thuyết gồm có:
 - Tiếp cận từ mô hình tương đương – mô hình con lắc ngược đến mô hình thật của đề tài.
 - Mô phỏng mô hình bằng VN Nastran và MatLAB: scooter tự cân bằng trên hai bánh.

- Tiếp cận mô hình thực, gồm có:
 - Thiết kế khung sườn cơ khí của mô hình.
 - Công suất điện và điện tử (điều khiển bánh xe).
 - Mạch cảm biến (góc, vị trí, vận tốc góc và vận tốc dài).
 - Calibre cảm biến.
 - Bộ điều khiển trung tâm.
 - Lập trình vi điều khiển.

Chương 3

LÝ THUYẾT TIẾP CẬN

3.1 Phương pháp tính động lực học

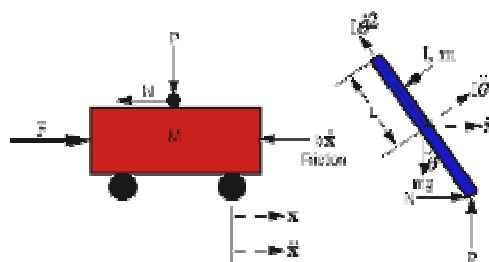
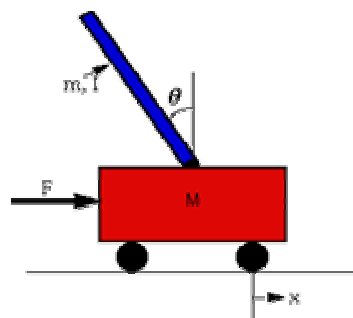
Có nhiều phương pháp dùng để tính động lực học, chẳng hạn: phương pháp Newton, phương pháp Lagrange, phương pháp theo năng lượng... Nhưng trong đề tài này, phương pháp Newton được sử dụng với các ưu điểm của nó. Thứ nhất, nó sử dụng các phương pháp tính cơ học thông thường. Thứ hai, các công thức và hệ phương trình trong quá trình tính không quá phức tạp. Thứ ba, kết quả tính động lực học của mô hình con lắc ngược được phổ biến hiện nay ở các tài liệu tham khảo được sử dụng để kiểm tra sự sai sót trong quá trình tính toán động lực học của mô hình xe hai bánh tự cân bằng.

Bên cạnh các ưu điểm này, nó vẫn có nhược điểm là phải tuyến tính hóa tính toán tại vị trí góc $\alpha = 0^\circ$. Tuy nhiên việc này không trầm trọng trong mô hình của đề tài, vì mô hình chỉ hoạt động xung quanh vị trí 0° ($\pm 10^\circ$).

3.1.1 Nền tảng nghiên cứu từ con lắc ngược

– Ta xem xét mô hình toán học của con lắc ngược với các tham số như sau:

- M Khối lượng xe (kg).
- m Khối lượng con lắc (kg).
- b Ma sát của xe (N).
- L Chiều dài ½ con lắc (m).
- I Momen quán tính của con lắc (Nm).
- F Lực tác động vào xe (N).
- X Vị trí của xe (m).
- θ Góc của con lắc so với phương thẳng đứng (rad).



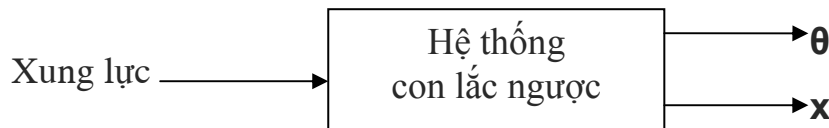
Hình 3.1 Phân tích lực trên xe và trên con lắc [34]

Mô hình nghiên cứu gồm hai phần:

- Một con lắc gắn bởi một khớp bản lề với xe.
- Tác động lực F vào xe.

Mô hình bao gồm hai ngõ vào:

- Dịch chuyển của xe (x).
- Góc lệch của con lắc (θ).



Hình 3.2 Mô hình và sơ đồ khối tín hiệu ngõ vào và ra của con lắc ngược

Phân tích lực cho xe, ta có:

Tổng lực tác dụng lên xe theo phương ngang:

$$M \ddot{x} + b \dot{x} + N = F \quad [3-1]$$

Tổng lực tác dụng lên con lắc theo phương ngang:

$$N = m \ddot{x} + mL \ddot{\theta} \cos \theta - mL \dot{\theta}^2 \sin \theta \quad [3-2]$$

Từ [3-1] suy ra:

$$(M + m) \ddot{x} + b \dot{x} + mL \ddot{\theta} \cos \theta - mL \dot{\theta}^2 \sin \theta = F \quad [3-3]$$

Tổng lực tác dụng ở phương vuông góc với con lắc ta có:

$$P \sin \theta + N \cos \theta + mg \sin \theta = mL \ddot{\theta} + m \ddot{x} \cos \theta \quad [3-4]$$

Tổng momen tại khối tâm con lắc ta có:

$$-PL \sin \theta - NL \cos \theta = I \ddot{\theta} \quad [3-5]$$

Kết hợp hai phương trình trên:

$$(I + mL^2) \ddot{\theta} - mgL \sin \theta = -mL \ddot{x} \cos \theta \quad [3-6]$$

Từ [3-3] và [3-6] xấp xỉ tuyến tính hóa tại 0° :

$$(M + m) \ddot{x} + b \dot{x} + mL \ddot{\theta} = F \quad [3-7]$$

$$(I + mL^2) \ddot{\theta} - mgL \theta = -mL \ddot{x} \quad [3-8]$$

Hàm trạng thái (space-state):

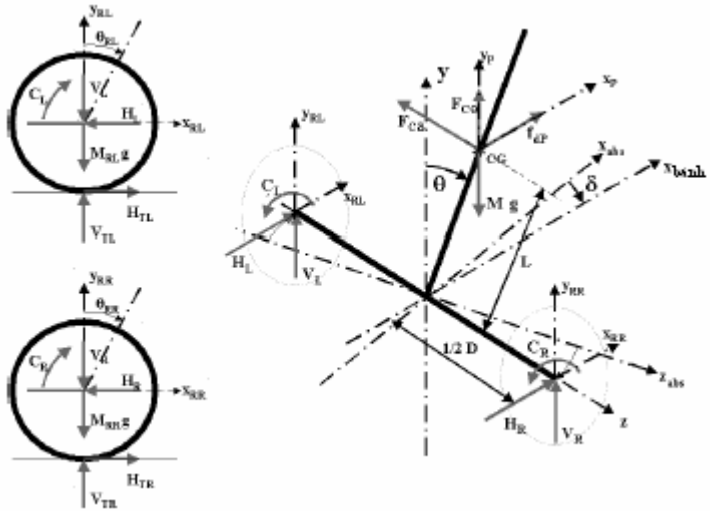
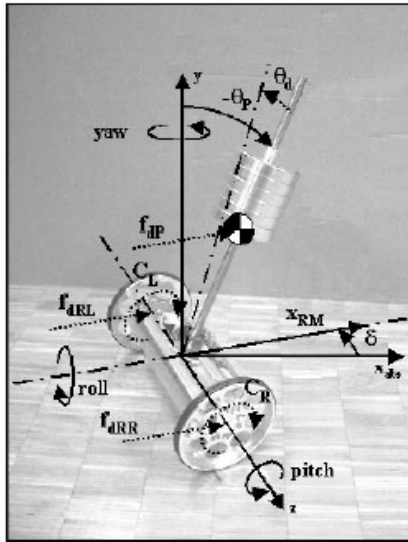
$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I+mL^2)b}{I(M+m)+MmL^2} & \frac{-m^2gL^2}{I(M+m)+MmL^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{mLb}{I(M+m)+MmL^2} & \frac{mgL(M+m)}{I(M+m)+MmL^2} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{I+mL^2}{I(M+m)+MmL^2} \\ 0 \\ \frac{-mL}{I(M+m)+MmL^2} \end{bmatrix} F \quad [3-9]$$

$$\begin{bmatrix} x \\ \theta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} F \quad [3-10]$$

Nếu xem $b \ll 1$, $I \ll 1$, ta được dạng hàm trạng thái đơn giản hơn

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-mg}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{g(M+m)}{ML} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ \frac{-1}{ML} \end{bmatrix} F \quad [3-10']$$

3.1.2 Động lực học mô hình xe tự cân bằng



Hình 3.3 Biểu diễn lực và momen của mô hình ^[17]

- Đối với bánh trái (cũng như bánh phải):

$$\ddot{x}_{RL} M_{RL} = H_{TL} - H_L \quad [3-11]$$

$$\ddot{y}_{RL} M_{RL} = V_{TL} - M_{RL}g - V_L \quad [3-12]$$

$$\ddot{\theta}_{RL} J_{RL} = C_L - H_{TL}R \quad [3-13]$$

Ma sát rất nhỏ do đó bỏ qua H_{TL}

- Đối với thân:

$$\ddot{x}_{thân} M_{thân} = H_R + H_L \quad [3-14]$$

$$\ddot{y}_{thân} M_{thân} = V_L + V_R - M_{thân}g + F_{C\theta} \quad [3-15]$$

$$J_{thân} \ddot{\theta} = (V_L + V_R)L \sin \theta - (H_L + H_R)L \cos \theta - (C_L + C_R) \quad [3-16]$$

$$J_{thânquay} \ddot{\delta} = (H_L - H_R) \frac{D}{2} \quad [3-17]$$

Trong đó:

$J_{thân}$: là momen quán tính của người (Nm^2).

$J_{bánh}, J_{RL}, J_{RR}$: momen quán tính bánh xe (Nm^2).

$M_{thân}, M_{bánh}$: khối lượng của người và bánh (kg).

R : bán kính bánh xe (m).

D : khoảng cách giữa hai bánh xe (m).

L : khoảng cách từ trục z của hai bánh xe đến trọng tâm người (m).

θ : góc lật (rad).

δ : góc quay (rad).

Tổng quát, ta được phương trình:

$$J_{thân} \ddot{\theta} = (V_L + V_R)L \cdot \sin(\theta) - (H_L + H_R)L \cdot \cos(\theta) - (C_L + C_R) \quad [3-18]$$

$$\ddot{x}_{thân} \cdot M_{thân} = (H_L + H_R) \quad [3-19]$$

$$\ddot{y}_{thân} \cdot M_{thân} = (V_L + V_R) - M_{thân} \cdot g + F_{C\theta} = (V_L + V_R) - M_{thân} \cdot g + \frac{(C_L + C_R)}{L} \cdot \sin(\theta) \quad [3-20]$$

- Thế [3-19] và [3-20] vào [3-18]:

$$J_{thân} \ddot{\theta} = (\ddot{y}_{thân} \cdot M_{thân} + M_{thân} \cdot g - \frac{(C_L + C_R)}{L})L \cdot \sin^2(\theta) - \ddot{x}_{thân} \cdot M_{thân}L \cdot \cos(\theta) - (C_L + C_R) \quad [3-21]$$

$$J_{thân} \ddot{\theta} = M_{thân} \cdot L \left[\ddot{y}_{thân} \cdot \sin(\theta) - \ddot{x}_{thân} \cdot \cos(\theta) \right] + M_{thân} \cdot gL \cdot \sin(\theta) - (1 + \sin(\theta^2))(C_L + C_R) \quad [3-22]$$

Xét với mỗi bánh trái và phải:

$$\begin{cases} \ddot{x}_{RL} M_{RL} = H_{TL} - H_L \\ \ddot{x}_{RR} M_{RR} = H_{TR} - H_R \end{cases} \quad [3-23]$$

$$M_{RL} = M_{RR} = M_{bánh} \cdot (\ddot{x}_{RL} + \ddot{x}_{RR}) = 2 \ddot{x}_{RM} \quad [3-24]$$

$$M_{bánh} \cdot (\ddot{x}_{RL} + \ddot{x}_{RR}) = -(H_L + H_R) + (H_{TL} + H_{TR}) \quad [3-25]$$

$$\text{Mà } \ddot{\theta}_{RL} \cdot J_{RL} = C_L - H_{TL} \cdot R \quad [3-26]$$

$$M_{bánh} \cdot (\ddot{x}_{RL} + \ddot{x}_{RR}) = -\ddot{x}_{thân} \cdot M_{thân} + \frac{(C_L + C_R) - (\ddot{\theta}_{RL} \cdot J_{RL} + \ddot{\theta}_{RR} \cdot J_{RR})}{R} \quad [3-27]$$

$$C_L = \frac{1}{2} C_\theta + \frac{1}{2i} C_\delta \quad [3-28]$$

$$C_R = \frac{1}{2} C_\theta - \frac{1}{2i} C_\delta \quad [3-29]$$

$$C_L + C_R = C_\theta \quad [3-30]$$

$\theta_{bánh}$ được đo so với trục tọa độ Galilée tĩnh:

$$J_{thân} \ddot{\theta} = M_{thân} \cdot L \left[\ddot{y}_{thân} \cdot \sin(\theta) - \ddot{x}_{thân} \cdot \cos(\theta) \right] + M_{thân} \cdot gL \cdot \sin(\theta) - (1 + \sin^2(\theta))(C_L + C_R) \quad [3-31]$$

$$2M_{bánh} \cdot \ddot{x}_{RM} = -\ddot{x}_{thân} \cdot M_{thân} + \frac{(C_L + C_R) - (\ddot{\theta}_{RL} \cdot J_{RL} + \ddot{\theta}_{RR} \cdot J_{RR})}{R} \quad [3-32]$$

$$2M_{bánh} \cdot \ddot{x}_{RM} = -\ddot{x}_{thân} \cdot M_{thân} + \frac{C_\theta}{R} - 2 \frac{J_{bánh} \cdot \ddot{\theta}_{bánh}}{R} \quad [3-33]$$

C_θ là giá trị trung bình của C_L và C_R

$$\begin{cases} y_{thân} = -L \cdot (1 - \cos \theta) \\ x_{thân} = x_{bánhTB} + L \cdot \sin \theta \end{cases} \quad [3-34]$$

$$\begin{cases} \dot{y}_{thân} = -L(\sin \theta) \cdot \dot{\theta} \\ \dot{x}_{thân} = \dot{x}_{bánhTB} + L \cdot (\cos \theta) \cdot \dot{\theta} \end{cases} \quad [3-35]$$

$$\begin{cases} \ddot{y}_{thân} = -L(\sin \theta) \cdot \ddot{\theta} - L(\cos \theta) \dot{\theta}^2 \\ \ddot{x}_{thân} = \ddot{x}_{bánhTB} + L \cdot (\cos \theta) \cdot \ddot{\theta} - L \cdot (\sin \theta) \cdot \dot{\theta}^2 \end{cases} \quad [3-36]$$

• Do đó:

$$\ddot{y}_{thân} \sin \theta - \ddot{x}_{thân} \cos \theta = -L \ddot{\theta} - \ddot{x}_{bánhTB} \cos \theta \quad [3-37]$$

- Hệ phương trình trở thành:

$$J_{thân} \ddot{\theta} = -M_{thân} L^2 \ddot{\theta} + gL \cdot M_{thân} \sin \theta - (1 + \sin \theta) C_{\theta} - \ddot{x}_{bánhTB} L M_{thân} \cos \theta \quad [3-38]$$

$$2M_{bánh} \ddot{x}_{bánhTB} = M_{thân} \cdot L(\sin \theta) \dot{\theta}^2 - M_{thân} L(\cos \theta) \ddot{\theta} - 2 \frac{J_{bánh} \ddot{\theta}}{R} + \frac{C_{\theta}}{R} - M_{thân} \ddot{x}_{bánhTB} \quad [3-39]$$

- Momen quán tính của thân xem như là một đoạn thẳng hình trụ chiều dài $2L$, bánh xe xem như là một đĩa tròn xoay:

$$J_{thân} = \frac{1}{3} M_{thân} L^2 \quad [3-40]$$

$$J_{bánh} = \frac{1}{2} M_{bánh} R^2 \quad [3-41]$$

$$\begin{cases} \frac{1}{3} M_{thân} L^2 \ddot{\theta} = -M_{thân} L^2 \ddot{\theta} + gL \cdot M_{thân} \sin \theta - (1 + \sin \theta) C_{\theta} - \ddot{x}_{bánhTB} L M_{thân} \cos \theta \\ (2M_{bánh} + M_{thân}) \ddot{x}_{bánhTB} = M_{thân} \cdot L(\sin \theta) \dot{\theta}^2 - M_{thân} L(\cos \theta) \ddot{\theta} - 2 \frac{\frac{1}{2} M_{bánh} R^2 \ddot{\theta}}{R} + \frac{C_{\theta}}{R} \end{cases} \quad [3-42]$$

$$\begin{cases} \frac{4}{3} M_{thân} L^2 \ddot{\theta} = gL \cdot M_{thân} \sin \theta - (1 + \sin \theta) C_{\theta} - \ddot{x}_{bánhTB} L M_{thân} \cos \theta \\ (2M_{bánh} + M_{thân}) \ddot{x}_{bánhTB} = M_{thân} \cdot L(\sin \theta) \dot{\theta}^2 - M_{thân} L(\cos \theta) \ddot{\theta} - M_{bánh} R \ddot{\theta} + \frac{C_{\theta}}{R} \end{cases} \quad [3-43]$$

- Xấp xỉ các giá trị \sin , $(1 + \sin^2)$, \cos , ta được:

$$\begin{cases} \frac{4}{3} M_{thân} L \ddot{\theta} - gM_{thân} \theta + \frac{C_{\theta}}{L} + M_{thân} \ddot{x}_{bánhTB} = 0 \\ (2M_{bánh} + M_{thân}) \ddot{x}_{bánhTB} + (M_{thân} L + M_{bánh} R) \ddot{\theta} = \frac{C_{\theta}}{R} \end{cases} \quad [3-44]$$

$$\begin{cases} \frac{4}{3} L \ddot{\theta} - g\theta + \frac{C_{\theta}}{M_{thân} L} + \ddot{x}_{bánhTB} = 0 \\ M_{thân} \ddot{x}_{bánhTB} + \frac{M_{thân}}{(2M_{bánh} + M_{thân})} (M_{thân} L + M_{bánh} R) \ddot{\theta} = \frac{M_{thân}}{(2M_{bánh} + M_{thân})} \frac{C_{\theta}}{R} \end{cases} \quad [3-45]$$

- Giải hệ phương trình, ta được:

$$\begin{cases} \ddot{x}_{bánhTB} = -\frac{4}{3}L\ddot{\theta} + g\theta - \frac{C_\theta}{M_{thân}L} \\ \left(\frac{4}{3}M_{thân}L - \frac{M_{thân}}{(2M_{bánh} + M_{thân})}(M_{thân}L + M_{bánh}R) \right) \ddot{\theta} - gM_{thân}\theta = -\left[\frac{M_{thân}}{(2M_{bánh} + M_{thân})} \frac{C_\theta}{R} + \frac{C_\theta}{L} \right] \end{cases} \quad [3-46]$$

$$\begin{cases} \ddot{x}_{bánhTB} = -\frac{4}{3}L\ddot{\theta} + g\theta - \frac{C_\theta}{M_{thân}L} \\ X \cdot \ddot{\theta} - gM_{thân}\theta = -\left[\frac{M_{thân}}{(2M_{bánh} + M_{thân})} \frac{C_\theta}{R} + \frac{C_\theta}{L} \right] \end{cases} \quad [3-47]$$

$$\text{Đặt: } X = \left(\frac{4}{3}M_{thân}L - \frac{M_{thân}}{(2M_{bánh} + M_{thân})}(M_{thân}L + M_{bánh}R) \right) \quad [3-48]$$

$$Y = \frac{M_{thân}}{(2M_{bánh} + M_{thân})} \frac{1}{R} + \frac{1}{L} \quad [3-49]$$

$$\ddot{\theta} = \frac{gM_{thân}}{X} \theta - \frac{Y \cdot C_\theta}{X} \quad [3-50]$$

$$\ddot{x}_{bánhTB} = \left(-\frac{4}{3}L \frac{gM_{thân}}{X} + g \right) \theta + \left(\frac{4LY}{3X} - \frac{1}{M_{thân}L} \right) C_\theta \quad [3-51]$$

- Từ đó, ta có hệ phương trình biến trạng thái:

$$\ddot{\theta} = \frac{gM_{thân}}{X} \theta - \frac{Y \cdot C_\theta}{X} \quad [3-52]$$

$$\ddot{x}_{bánhTB} = \left(-\frac{4}{3}L \frac{gM_{thân}}{X} + g \right) \theta + \left(\frac{4LY}{3X} - \frac{1}{M_{thân}L} \right) C_\theta \quad [3-53]$$

$$\begin{bmatrix} \dot{x}_{bánh} \\ \ddot{x}_{bánh} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & g \left(1 - \frac{4}{3}L \frac{M_{thân}}{X} \right) & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & + \frac{gM_{thân}}{X} & 0 \end{bmatrix} \begin{bmatrix} x_{bánh} \\ \dot{x}_{bánh} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \left(\frac{4LY}{3X} - \frac{1}{M_{thân}L} \right) \\ 0 \\ -\frac{Y}{X} \end{bmatrix} C_\theta \quad [3-54]$$

$$\begin{bmatrix} x_{bánh} \\ \theta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_{bánh} \\ \dot{x}_{bánh} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} C_\theta \quad [3-55]$$

3.2 Thuật toán điều khiển - Kỹ thuật điều khiển hiện đại (Modern control design) - Bộ điều khiển số LQR



Hình 3.4 Mô hình truyền của hàm trạng thái

3.2.1 Thiết kế cổ điển và hiện đại

Hệ thống điều khiển được thiết kế nhằm tự thực thi mệnh lệnh như thế nào và tới đâu của hệ thống. Hệ thống điều khiển rất cần thiết vì ta không muốn cách hoạt động hiện tại của quy trình, và bằng cách làm thay đổi hoạt động của quy trình phù hợp với nhu cầu của ta. Thiết kế hệ thống là thay đổi các tham số (parameter) hệ thống kiểm soát để phù hợp với mục tiêu ổn định đặc trưng, và bền vững. Tham số thiết kế có thể là một hằng số không biết (bất kỳ) trong hàm truyền của bộ điều khiển (transfer function), hoặc ma trận trạng thái đặc trưng (space-state).

Phương pháp thiết kế cổ điển bao gồm cách thay đổi hàm truyền bộ điều khiển đến khi đạt được chu trình đóng (closed-loop). Những dấu hiệu nhận biết (indicator) cổ điển của khả năng vòng điều khiển khép kín ổn định là đáp ứng tần số, hay vị trí cực của hệ thống điều khiển. Đối với hệ thống điều khiển bậc cao, bằng cách thay đổi một số ít hằng số trong hàm truyền của bộ điều khiển, có thể làm thay đổi phần lớn các vị trí cực điều khiển, dù không phải biến đổi là tất cả. Đây là giới hạn chính của phương pháp thiết kế cổ điển.^[11]

Nguyên tắc kiểm tra hệ thống ổn định bằng hàm trạng thái : tìm nghiệm λ của phương trình $|\lambda I - A| = 0$, với A là một thành phần của hàm trạng thái, nếu một trong những eigenvalue của A là $\lambda > 0$, thì vòng kín của hệ điều khiển không ổn định.^[14]

Khả năng điều khiển của hệ thống có thể được xác lập khi hệ thống có được bất kỳ trạng thái ban đầu nào (any initial state), $x(t_0)$, đến trạng thái cuối bất kỳ nào (any final state), $x(t_f)$, trong thời gian xác định (finite time), $(t_f - t_0)$, mỗi một giá trị của vector ngõ vào, $u(t)$, $t_0 \leq t \leq t_f$. Điều này cực kỳ quan trọng, vì có thể thay đổi hệ thống không điều khiển (uncontrollable) được từ trạng thái ban đầu đến trạng thái cuối cùng, hay lấy một lượng thời gian không xác định làm thay đổi hệ thống không điều khiển được, bằng cách dùng vector ngõ vào, $u(t)$. Khả năng điều khiển được của hệ thống dễ dàng được kiểm tra nếu tách riêng ra các phương trình trạng thái của hệ thống. Mỗi phương trình trạng thái vô hướng tách riêng đều tương đương với hệ thống phụ (sub-system).

3.2.2 Định lý về khả năng điều khiển

Một hệ thống tuyến tính, bất biến theo thời gian được mô tả bằng phương trình trạng thái ma trận, $\dot{x}^{(1)}(t) = Ax(t) + Bu(t)$ điều khiển được nếu và chỉ nếu ma trận kiểm tra khả năng điều khiển.

$P = [B; AB; A^2B; A^3B; \dots; A^{n-1}B]$ có hạng là n , là bậc (rank) của hệ thống.^[13]

Hạng của ma trận, P , được xác định như là một định thức khác không lớn nhất hình thành ma trận, P . Nếu P là ma trận vuông, định thức lớn nhất hình thành P là $|P|$. Nếu P không phải là ma trận vuông, thì định thức lớn nhất P được hình thành bằng cách cho tất cả hàng bằng với số cột hoặc là tất cả cột bằng với số hàng của P . Từ đó có hạng của ma trận. Chú ý rằng hệ thống của ngõ vào bậc n với r , ma trận kiểm tra khả năng điều khiển, P , là ma trận $(n \times nr)$. Định thức khác không của P có thể là thứ nguyên của n . Do đó, hạng của P có thể là nhỏ hơn hoặc bằng n .

Một bằng chứng hiển nhiên của định lý kiểm tra khả năng điều khiển đại số có thể thấy trong định lý Friedland (phần đại số ma trận). Một dạng tương tự của định lý có thể áp dụng cho hệ thống thay đổi theo thời gian. Từ đó có thể tiến hành kiểm tra khả năng điều khiển thay đổi theo thời gian (time-varying).

$$P(t) = [B(t); A(t)B(t); A^2(t)B(t); A^3(t)B(t); \dots; A^{n-1}(t)B(t)] \quad [3-56]$$

Và kiểm tra hạng của $P(t)$ đối với tất cả thời điểm, $t \geq t_0$, cho hệ thống tuyến tính biến đổi theo thời gian. Nếu tại bất kỳ t nào, hạng của $P(t)$ nhỏ hơn n , hệ thống không điều khiển được. Tuy nhiên, việc dùng ma trận kiểm tra khả năng điều khiển biến đổi thời gian của phương trình [3-56] cần chú ý đến hệ số trạng thái biến đổi nhanh theo thời gian, vì việc kiểm tra có thể thực thi tại các bước thời gian riêng biệt, và trong vài khoảng thời gian (nhỏ hơn bước thời gian), hệ thống vẫn có khả năng điều khiển trong đó.

$$|P| \geq \text{rank}(P) \rightarrow \text{điều khiển được (controllable)}$$

Việc kiểm tra khả năng điều khiển đại số gồm nhận biết hạng của P , kiểm tra xem liệu nó có bằng n không, hình thành định thức có thể của thứ nguyên n của ma trận P bằng cách loại bỏ vài cột (nếu $m > 1$), và kiểm tra giá trị định thức có khác không. Phần mềm MatLAB cung cấp lệnh `rank(P)` giúp cho việc nhận biết hạng của ma trận P trở nên dễ dàng. Hơn nữa, công cụ Control System Toolbox của MatLAB (CST) cho phép hình thành trực tiếp ma trận kiểm tra khả năng điều khiển P , bằng cách dùng lệnh `ctrb` như sau:

$$>> P = \text{ctrb}(A, B) \quad <\text{enter}>$$

hay

$$>> P = \text{ctrb}(\text{sys}) \quad <\text{enter}>$$

trong đó, A và B là ma trận hệ số trạng thái của hệ thống mà hàm truyền (LTI) của nó là sys .

Nguyên nhân khiến hệ thống không điều khiển được có thể là do tính chính xác, chẳng hạn cách dùng các giá trị trạng thái thừa (nghĩa là các giá trị trạng thái nhiều hơn bậc của hệ thống). Giá trị trạng thái thừa không ảnh hưởng trực tiếp bởi ngõ vào hệ thống, sẽ không tạo ra không gian trạng thái để điều khiển, cho dù hệ thống có thể điều khiển được về mặt vật lý. Nguyên nhân đôi khi làm mất khả năng điều khiển được là có quá nhiều cực đối xứng trong mô hình toán học của hệ thống.

3.2.3 Thiết kế gain cực hồi tiếp biến trạng thái đầy đủ

Hệ thống điều khiển chu trình đóng được thiết kế bởi bộ điều khiển để đặt các cực tại vị trí mong muốn sẽ làm thay đổi đặc tính của hệ điều khiển. Phương pháp thiết kế cổ điển dùng hàm chuyển giao bộ điều khiển với một vài tham số thiết kế không đủ đặt vào tất cả các cực của chu trình đóng tại vị trí mong muốn. Phương pháp hàm trạng thái dùng thông tin phản hồi trạng thái đầy đủ cung cấp đủ tham số thiết kế bộ điều khiển để di chuyển các cực chu trình đóng độc lập nhau^[13]. Thông tin phản hồi trạng thái đầy đủ được phát ra từ vector ngõ vào của bộ điều khiển, $u(t)$, theo định luật điều khiển như sau:

$$u(t) = K [x_d(t) - x(t) - K_d x_d(t) - K_n x_n(t)] \quad [3-57]$$

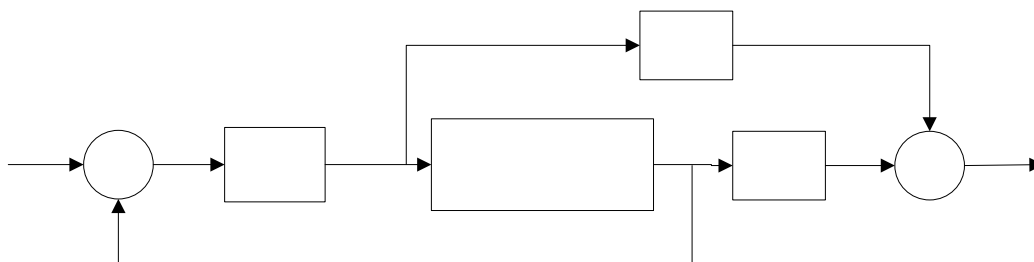
Trong đó $x(t)$ là vector trạng thái của hệ thống, $x_d(t)$ là vector trạng thái mong muốn, $x_n(t)$ là vector trạng thái nhiễu và K , K_d và K_n là ma trận gain của bộ điều khiển. Vector trạng thái mong muốn, $x_d(t)$, và vector trạng thái nhiễu, $x_n(t)$, được tạo ra từ tiến trình ngoài, và hoạt động như những ngõ vào hệ thống điều khiển. Nhiệm vụ của bộ điều khiển là thu được vector trạng thái mong muốn ở tình trạng ổn định (steady), trong khi đó vẫn phản ứng chống lại ảnh hưởng của độ nhiễu. Vector ngõ vào, $u(t)$, được tạo bởi phương trình [3-57] ứng dụng cho hệ thống được mô tả bởi phương trình ngõ ra và trạng thái sau:

$$\dot{x}^{(1)}(t) = Ax(t) + Bu(t) + Fx_n(t) \quad [3-58]$$

$$y(t) = Cx(t) + Du(t) + Ex_n(t) \quad [3-59]$$

trong đó, F và E là ma trận hệ số nhiễu trong phương trình ngõ ra và trạng thái. Thiết kế hệ thống điều khiển dùng thông tin phản hồi trạng thái đầy đủ yêu cầu hệ điều khiển được mô tả bằng phương trình [3-57] phải dễ điều khiển, nếu không thì ngõ vào kiểm soát được tạo ra từ phương trình [3-57] sẽ không ảnh hưởng đến tất cả các giá trị trạng thái khác nhau của hệ thống. Hơn nữa, phương trình [3-57] đòi hỏi tất cả giá trị trạng thái khác nhau của hệ thống phải được đo, và có khả năng phản hồi đến bộ điều khiển.

Ma trận độ lợi (gain) của bộ điều khiển, K , K_d , và K_n là các tham số thiết kế của hệ thống điều khiển được mô tả qua phương trình [3-58] và [3-59]. Chú ý rằng bậc của hệ thống chu trình đóng thông tin phản hồi trạng thái đầy đủ là giống nhau như hệ thống. Biểu đồ của hệ thống điều khiển hồi tiếp trạng thái đầy đủ không nhiễu trình bày ở hình 3.5.



Hình 3.5 Biểu đồ của hệ thống điều khiển hồi tiếp các biến trạng thái và không nhiễu

Một hệ thống kiểm soát mà vector trạng thái mong muốn là 0 được gọi là hệ thống điều chỉnh (regulator). Vì vậy, định luật điều khiển sẽ là:

$$u(t) = -Kx(t) \quad [3-60]$$

Thay vào phương trình [3-60] được phương trình [3-61] và [3-62], từ đó có phương trình ngõ ra và trạng thái chu trình đóng của hệ thống điều chỉnh như sau:

$$\dot{x}^{(1)}(t) = (A - BK)x(t) \quad [3-61]$$

$$y(t) = (C - DK)x(t) \quad [3-62]$$

Phương trình [3-61] và [3-62] cho thấy hệ thống điều chỉnh là hệ thống đồng nhất (homogeneous), được mô tả bởi ma trận hệ số trạng thái chu trình đóng $A_{CL} = A - BK$, $B_{CL} = 0$, $C_{CL} = C - DK$, và $D_{CL} = 0$. Các cực chu trình đóng là eigenvalue của A_{CL} . Do đó, bằng cách chọn ma trận gain của bộ điều khiển, K , các cực chu trình đóng được đặt tại vị trí mong muốn. Đối với hệ thống của ngõ vào bậc n với r , cỡ của K là $(r \times n)$. Từ đó có tổng của tham số thiết kế vô hướng r , n . Đối với hệ thống nhiều ngõ vào (nghĩa là $r > 1$), số lượng tham số thiết kế có nhiều cách chọn vị trí cực n .

3.2.4 Thiết kế hệ thống ổn định gán cực cho ngõ vào đơn

Nếu hệ thống trong biểu mẫu cùng bộ điều khiển, ma trận độ lợi điều chỉnh là:

$$K = (\alpha - a)P'P^{-1} \quad [3-63]$$

Phương trình [3-63] được gọi là công thức không gian đặt cực Ackermann, trong đó P và P' là ma trận vuông $(n \times n)$ [14].

Cho phép thiết kế hệ thống điều chỉnh thông tin phản hồi trạng thái đầy đủ cho mô hình scooter hai bánh. Từ phương trình biểu diễn trạng thái không gian tuyến tính của hệ thống, ma trận hệ số trạng thái của nó như sau:

$$\begin{bmatrix} \dot{x}_{bánh} \\ \ddot{x}_{bánh} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \left(g - \frac{4}{3}L \frac{gM_{thân}}{X}\right) & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{gM_{thân}}{X} & 0 \end{bmatrix} \begin{bmatrix} x_{bánh} \\ \dot{x}_{bánh} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \left(\frac{4LY}{3X} - \frac{1}{M_{thân}L}\right) \\ 0 \\ -\frac{Y}{X} \end{bmatrix} C_{\theta} \quad [3-64]$$

$$\begin{bmatrix} x_{bánh} \\ \dot{x}_{bánh} \\ \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_{bánh} \\ \dot{x}_{bánh} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} C_{\theta} \quad [3-65]$$

với

$$X = \left(\frac{4}{3}M_{thân}L - \frac{M_{thân}}{(2M_{bánh} + M_{thân})} (M_{thân}L + M_{bánh}R) \right) \quad [3-66]$$

$$Y = \frac{M_{thân}}{(2M_{bánh} + M_{thân})} \frac{1}{R} + \frac{1}{L} \quad [3-67]$$

Ngõ vào đơn, $u(t)$, là năng lượng cung cấp theo chiều ngang đối với phương tiện và hai ngõ ra là vị trí góc của con lắc, $\theta(t)$, và vị trí ngang của phương tiện, $x(t)$. Bốn vector trạng thái của hệ thống thứ tự là $x(t) = [\theta(t); x(t); \dot{\theta}(t); \dot{x}(t)]^T$. Cho phép áp dụng giá trị số của tham số hệ thống như sau: $m_{bánh} = 7\text{kg}$; $M_{thân} = 60\text{kg}$; $R = 0,2\text{m}$; $L = 1\text{m}$; và $g = 9,8\text{m/s}^2$. Ma trận A và B được cho như sau:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -15.9620 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 19.3290 & 0 \end{bmatrix}; B = \begin{bmatrix} 0 \\ 0.2037 \\ 0 \\ -0.1653 \end{bmatrix} \quad [3-68]$$

Quyết định có điều khiển hệ thống không. Điều này thực hiện bằng cách tìm ma trận kiểm tra khả năng điều khiển, P, sử dụng lệnh MATLAB (CST) *ctrb* như sau:

$$\begin{aligned} &>> P = \text{ctrb}(A,B) <enter> \\ P &= \begin{bmatrix} 0 & 0.2037 & 0 & 2.5935 \\ 0.2037 & 0 & 2.5935 & 0 \\ 0 & -0.1653 & 0 & -3.1942 \\ -0.1653 & 0 & -3.1942 & 0 \end{bmatrix} \end{aligned} \quad [3-69]$$

Sau đó, việc quyết định ma trận kiểm tra khả năng điều khiển được tính toán như sau:

$$\begin{aligned} &>> \text{det}(P) <enter> \\ \text{ans} &= 0.1701 \end{aligned}$$

Vì $|P| \neq 0$, nó đưa đến hệ thống điều khiển được. Tuy nhiên, độ lớn (cường độ) $|P|$ phụ thuộc khung cỡ (thang tỉ lệ) của ma trận P, và nó không cho biết P cách xa bao nhiêu từ vị trí điều khiển được và như thế hệ thống được điều khiển mạnh bao nhiêu. Cách tốt hơn kiểm tra giá trị đo lường khả năng điều khiển là số điều kiện, đạt được khi dùng hàm MatLAB *cond* như sau:

$$\begin{aligned} &>> \text{cond}(P) <enter> \\ \text{ans} &= 80.251 \end{aligned}$$

Vì về độ lớn (cường độ) của số điều kiện P nhỏ, hệ thống được điều khiển cao. Như vậy, những kết quả bậc của cực được trông đợi chính xác. (Nếu số điều kiện P lớn về độ lớn (cường độ), nó sẽ chỉ báo một hệ thống điều khiển yếu và sự đảo ngược P để lấy ma trận độ lợi thông tin phản hồi sẽ không chính xác). Các cực của hệ thống được tính bằng cách tìm eigenvalue của ma trận A dùng lệnh MATLAB *damp* như sau:

$$>> \text{damp}(A) <enter>$$

Eigenvalue	Damping	Freq. (rad/sec)
4.3965	-1.0000	4.39
0	-1.0000	0
0	-1.0000	0
- 4.3965	1.0000	4.39

Hệ thống không ổn định vì một cặp cực tại $s=0$. Nhiệm vụ của bộ hồi tiếp điều chỉnh là làm ổn định hệ thống.

Các bước tính toán được chương trình hóa trong hàm MatLAB (CST) được gọi là *acker* để tính ma trận độ lợi ổn định cho những hệ thống ngõ vào đơn giản khi dùng công thức Ackermann's. Lệnh *acker* được dùng như sau:

```
>>K = acker(A,B,V) <enter>
```

Với $V = [-0.7-j; -0.7+j; -8.2+3*j; -8.2-3*j]$

Dưới đây là kết quả tính bằng MatLAB (không giới hạn độ lớn ngẫu lực):

– Chọn cực $V =$

```
-0.7000 - 1.0000i
-0.7000 + 1.0000i
-8.2000 + 3.0000i
-8.2000 - 3.0000i
```

sẽ thu được kết quả bộ ổn định cực K có giá trị như sau:

```
K = [-87.4505 -100.9797 -834.0498 -232.1682]
```

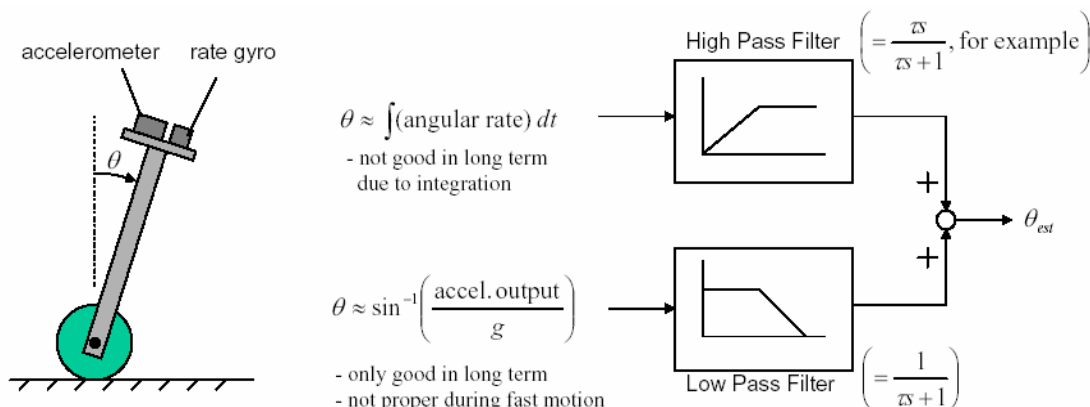
3.3 Các phương pháp xử lý tín hiệu từ cảm biến

Như ở chương 1 đã trình bày, giá trị ngõ ra được quan tâm hàng đầu của xe scooter hai bánh chính là góc giữa tay lái thẳng đứng hoặc sàn xe với chiều trọng lực. Nhiều loại cảm biến có thể dùng để đo góc như encoder, resolver, inclinometer, Nhưng trong mô hình của đề tài, chỉ có hai loại cảm biến để xác định góc giữa sàn xe với trọng lực theo phương pháp không tiếp xúc với sàn là cảm biến đo vận tốc góc và cảm biến gia tốc ứng dụng để đo góc tĩnh (cảm biến đo khoảng cách tới mặt đất bằng hồng ngoại chỉ dùng được với sàn đất phẳng, nằm ngang và lằng). Để làm tròn nhiều và kết hợp tín hiệu từ hai cảm biến accelerometer và gyro, người ta thường dùng các bộ lọc trung bình, lọc bổ phụ thông tần complementary, lọc thích nghi – bộ lọc Kalman và các dạng lọc khác.

3.3.1 Lọc bổ phụ thông tần (complementary filter) ^[16]

Bộ lọc bổ phụ được sử dụng trong trường hợp khi có hai nguồn giá trị khác nhau trong việc ước lượng một giá trị, và đặc tính nhiễu của hai giá trị chẳng hạn một

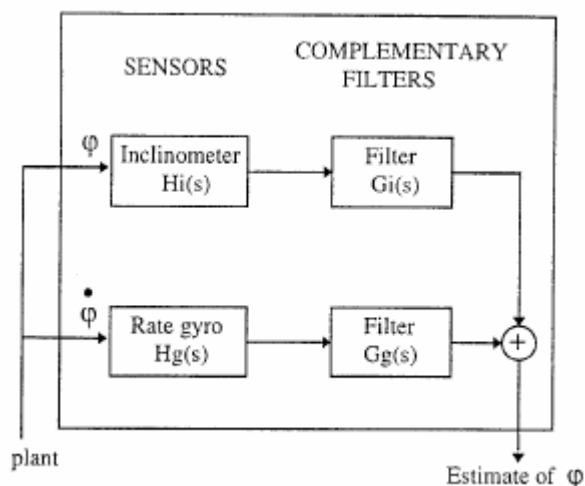
nguồn mang lại thông tin đúng trong vùng tần số thấp trong khi nguồn khác chỉ đúng trong vùng tần số cao. Bộ lọc bổ phụ là kết hợp những ngõ ra của độ nghiêng và vận tốc gyro nhằm thu được khả năng ước lượng tốt nhất của sự định hướng, để đền bù cho sự trôi giá trị gyro (drift) và cho đáp ứng chậm của cảm biến đo nghiêng.



Hình 3.6 Sơ đồ nguyên lý hoạt động của bộ lọc bổ phụ trên mô hình con lắc ngược

Ước lượng góc được thu nhận là tổng của tín hiệu từ hai nhánh giá trị được biểu diễn ở hình 3.6. Góc nghiêng của cảm biến đo nghiêng được đưa vào bộ lọc $G_i(s)$ phân bổ giá trị góc nghiêng để ước lượng chỉ trong vùng tần số thấp. Vận tốc góc của gyro cung cấp vào bộ lọc $G_g(s)$. Bộ lọc này được thiết kế để ngõ ra phân bổ đến ước lượng chỉ trong vùng tần số cao. Yêu cầu cơ bản của bộ lọc có thể được tính như sau:

1. Toàn bộ hệ thống có sự khuếch đại là hằng số và phase tổn thất nhỏ nhất ngay vùng tần số cắt của cảm biến nghiêng.
2. Để giữ độ nhảy offset của gyro là nhỏ nhất, cảm biến nghiêng nên được dùng trong vùng băng thông tần số rộng nhất có thể.
3. Số lượng tham số thiết kế nên nhỏ để dễ cho sự điều chỉnh và áp dụng bộ lọc trong thực tế của các vi điều khiển.



Hình 3.7 Hệ thống ước lượng sử dụng bộ lọc bổ phụ thông tần. Tổng ngõ ra của hai nguồn giá trị thu được khả năng ước lượng của sự định hướng

Để sự ước lượng trình bày động học cũng như góc thật của hướng, bộ lọc chuyển giao hàm $G_i(s)$ và $G_g(s)$ phải được chọn như sau:

$$H_i(s)G_i(s) + sH_g(s)G_g(s) = 1, \forall s \quad [3-70]$$

Trong đó $H_i(s)$ và $H_g(s)$ biểu thị hàm chuyển giao của độ nghiêng và gyro, mà không chú ý đến tất cả không tuyến tính. Hai cách chọn lựa có thể thực hiện bây giờ. Một cách tiếp cận để cho cảm biến lý tưởng, nghĩa là:

$$H_i(s) = H_g(s) = 1 \quad [3-71]$$

Sự tiếp cận thứ hai để lấy động học được biết của góc nghiêng trong cách tính:

$$H_i(s) = \frac{1}{0.53s + 1} \quad \text{và} \quad H_g(s) = 1 \quad [3-72]$$

Thiết kế bộ lọc cho cảm biến lý tưởng:

Nếu cảm biến lý tưởng thu được từ tiêu chuẩn của bộ lọc $G_i(s)$ và $G_g(s)$ do kết hợp phương trình [3-70] và [3-71].

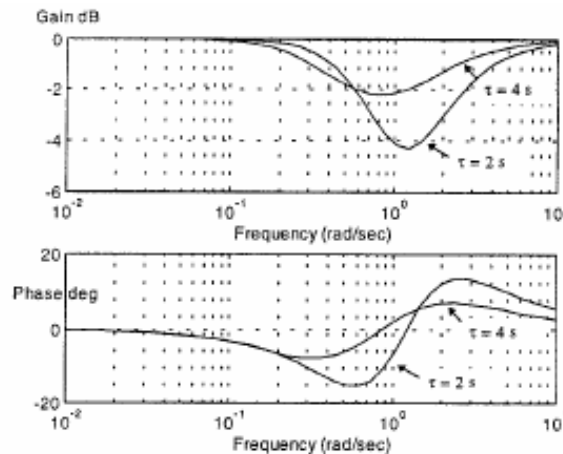
$$G_i(s) + G_g(s) = 1, \forall s \quad [3-73]$$

Để giữ số lượng của bộ lọc thiết kế tham số nhỏ thì chọn bộ lọc bậc hai với cực đôi.

$$G_i(s) = \frac{2\tau + 1}{(\tau + 1)^2} \quad [3-74]$$

$$G_i(s) = \frac{\tau^2 s}{(\tau + 1)^2} \quad [3-75]$$

Bộ lọc $G_i(s)$ phương trình [3-74], lọc tín hiệu vào từ góc nghiêng là bộ lọc thấp bậc nhất trong dãy bộ lọc mắc nối tiếp, nơi mà nhánh của gyro là bộ lọc cao bậc hai, với độ lợi góc là -40dB. $2\tau s$ trong tử số trong [3-74] có thể chuyển thành tử số trong [3-75], do xuất hiện 2τ . Đây là kết quả cho bộ lọc hữu ích, nghĩa là bộ lọc thấp bậc hai cho nhánh độ nghiêng và bộ lọc cao bậc nhất cho nhánh gyro. Bộ lọc cao bậc hai cung cấp sự tiếp hợp tín hiệu vận tốc gyro.



Hình 3.8 Giản đồ Bode của hệ thống ước lượng đối với hai giá trị hằng số lọc thời gian τ

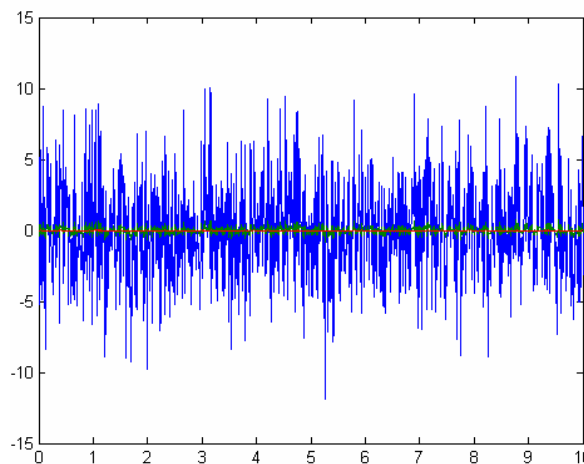
cho cảm biến lý tưởng của bộ lọc bổ phụ

3.3.2 Lọc thích nghi - Bộ lọc Kalman

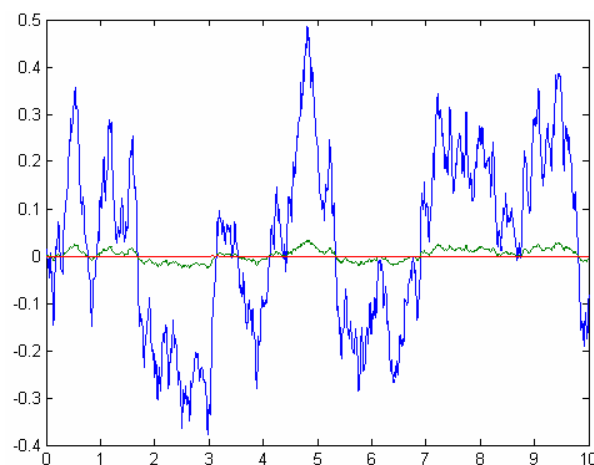
3.3.2.1 Lý thuyết về bộ lọc Kalman

Được đề xuất từ năm 1960 bởi giáo sư Kalman để thu thập và kết hợp linh động các thông tin từ cảm biến thành phần. Một khi phương trình định hướng và mẫu thống kê nhiều trên mỗi cảm biến được biết và xác định, bộ lọc Kalman sẽ cho ước lượng giá trị tối ưu (chính xác do đã được loại sai số, nhiễu) như là đang sử dụng một tín hiệu ‘tinh khiết’ và có độ phân bố không đổi. Trong hệ thống này, tín hiệu cảm biến vào bộ lọc gồm hai tín hiệu: từ cảm biến góc (inclinometer) và cảm biến vận tốc góc (gyro). Tín hiệu ngõ ra của bộ lọc là tín hiệu của inclinometer và gyro đã được loại nhiễu nhờ hai nguồn tín hiệu hỗ trợ và xử lý lẫn nhau trong bộ lọc, thông qua quan hệ (vận tốc góc = đạo hàm/vi phân của giá trị góc).^[15]

Một ví dụ mô phỏng về bộ lọc Kalman cho một máy bay chiến đấu bằng MatLAB^[13]:

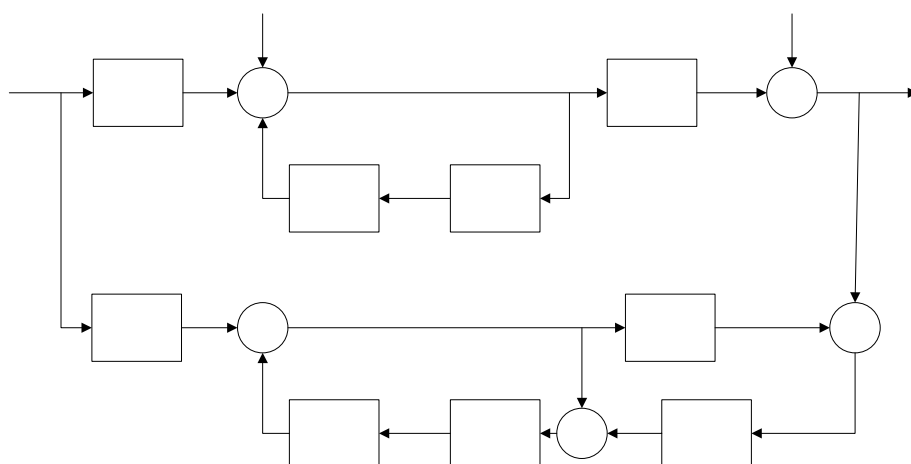


Hình 3.9 Tín hiệu thu nhận chưa được lọc



Hình 3.10 Tín hiệu đã qua bộ lọc Kalman

Bộ lọc Kalman đơn giản là thuật toán xử lý dữ liệu hồi quy tối ưu. Có nhiều cách xác định tối ưu, phụ thuộc tiêu chuẩn lựa chọn trình thông số đánh giá. Nó cho thấy rằng bộ lọc Kalman tối ưu đối với chi tiết cụ thể trong bất kỳ tiêu chuẩn có nghĩa nào. Một khía cạnh của sự tối ưu này là bộ lọc Kalman hợp nhất tất cả thông tin được cung cấp tới nó. Nó xử lý tất cả giá trị sẵn có, ngoại trừ độ sai số, ước lượng giá trị hiện thời của những giá trị quan tâm, với cách sử dụng hiểu biết động học thiết bị giá trị và hệ thống, mô tả số liệu thống kê của hệ thống nhiều, gồm nhiều ồn, nhiều đo và sự không chắc chắn trong mô hình động học, và những thông tin bất kỳ về điều kiện ban đầu của giá trị quan tâm.



Hình 3.11 Sơ đồ nguyên lý tín hiệu ngõ ra của bộ lọc Kalman

3.3.2.2 Quy trình ước lượng

Kalman filter định vị vấn đề chung nhằm ước lượng giá trị $x \in \mathbb{R}^n$ của tiến trình kiểm soát thời gian gián đoạn biểu diễn bằng phương trình tuyến stochastic khác nhau:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}, \quad [3-76]$$

với giá trị $z \in \mathfrak{R}^m$:

$$z_k = Hx_k + v_k. \quad [3-77]$$

Giá trị thay đổi ngẫu nhiên w_k và v_k trình bày cách xử lý và đo độ nhiễu (theo thứ tự). Chúng tách biệt nhau, đó là nhiễu trắng, và với sự sắp xếp bản chất thông thường:

$$p(w) \sim N(0, Q), \quad [3-78]$$

$$p(v) \sim N(0, R) . \quad [3-79]$$

Trong thực tiễn, ma trận tương quan tiến trình nhiễu Q và tương quan đo độ nhiễu R có lẽ thay đổi với mỗi bước thời gian hay giá trị, tuy nhiên giả thuyết rằng ở đây chúng là hằng số.

Ma trận A $n \times n$ trong phương trình [3-76] liên hệ với giá trị ở thời điểm trước $k-1$ đến giá trị ở thời điểm hiện tại k , trong sự thiếu hụt không chỉ ở hàm lái dạng mà còn do nhiều tiến trình. Chú ý rằng trong thực tế A có lẽ thay đổi tại mỗi thời điểm, nhưng ở đây, ta xem nó là hằng số. Ma trận B $n \times 1$ liên quan hoạt động kiểm soát tùy ý $u \in \mathcal{R}^1$ đối với x . Ma trận H $m \times n$ trong phương trình [3-77] liên quan với giá trị đo z_k . Trong thực tế, H có thể thay đổi tại mỗi thời điểm hay giá trị, nhưng ta xem nó là hằng số.

3.3.2.3 Bản chất tính toán học của bộ lọc

Theo định nghĩa $\hat{x}_k^- \in \mathcal{R}^n$ (siêu âm) ước lượng trạng thái priori tại bước k đưa ra tiến trình priori tại bước k , và $\hat{x}_k^- \in \mathcal{R}^n$ là ước lượng trạng thái posteriori tại bước k với giá trị z_k . Từ đó có thể xác định sai số ước lượng priori và posteriori là:

$$\begin{aligned} e_k^- &\equiv x_k - \hat{x}_k^-, \\ e_k &\equiv x_k - \hat{x}_k. \end{aligned}$$

[3-80]

Tương quan sai số ước lượng priori là:

$$P_k^- = E[e_k^- e_k^{-T}],$$

[3-81]

và tương quan sai số ước lượng posteriori là:

$$P_k = E[e_k e_k^T].$$

[3-82]

Bắt nguồn từ việc lấy đạo hàm phương trình của bộ lọc Kalman, với mục tiêu tìm một phương trình ước lượng trạng thái posteriori \hat{x}_k như sự liên kết tuyến tính ước lượng priori \hat{x}_k^- và sự khác biệt đánh giá giữa giá trị thực z_k và giá trị chuẩn tắc $H\hat{x}_k$ được trình bày ở công thức [3-83].

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-)$$

[3-83]

Độ lệch $(z_k - H\hat{x}_k^-)$ được gọi là đại lượng cách tân hay giá trị thặng dư. Giá trị thặng dư phản ánh sự khác nhau giữa giá trị chuẩn tắc $H\hat{x}_k^-$ và giá trị thực z_k . Giá trị thặng dư A của 0 nghĩa là hai thành phần này hoàn toàn nhất quán nhau.

Ma trận K $n \times m$ trong công thức [3-83] được chọn là độ lợi hay hệ số pha trộn tối thiểu hóa tương quan sai số posteriori [3-82]. Sự tối thiểu hóa có thể được thực hiện bằng cách thay thế đầu tiên [3-83] vào giá trị xác định cho e_k , thay thế vào [3-82], trình bày giá trị trông cậy chỉ thị, lấy đạo hàm vết của kết quả đối với K , thiết lập kết quả bằng 0, và sau đó giải quyết K . Một dạng kết quả K tối thiểu hóa [3-82] được trình bày như sau:

$$\begin{aligned} K_k &= P_k^- H^T (H P_k^- H^T + R)^{-1} \\ &= \frac{P_k^- H^T}{H P_k^- H^T + R} \end{aligned}$$

[3-84]

Xem [3-84], thấy rằng tương quan sai số giá trị R tiến tới 0, độ lợi K đánh giá giá trị thặng dư cao hơn. Đặc biệt,

$$\lim_{R_k \rightarrow 0} K_k = H^{-1} \quad [3-85]$$

Hơn nữa, tương quan sai số ước lượng priori P_k tiến tới 0, độ lợi K đánh giá thấp hơn. Đặc biệt,

$$\lim_{P_k \rightarrow 0} K_k = 0. \quad [3-86]$$

Cách nghĩ khác về phụ cấp bằng K thống kê sai lệch đo lường R tiến tới 0, đo lường thực tế z_k đáng tin cậy hơn, trong khi dự đoán đo lường $H\hat{x}_k$ ít tin cậy hơn. Hơn nữa, thống kê sai lệch ước lượng priori P_k tiến tới 0 cách đo thực tế z_k ít tin cậy hơn, trong khi dự đoán cách đo lường $H\hat{x}_k$ đáng tin cậy hơn.

3.3.2.4 Bản chất xác suất của bộ lọc

Sự điều chỉnh cho [3-83] đã xác định rõ bản chất ước lượng priori \hat{x}_k – với điều kiện tất cả các giá trị z_k đều có nghĩa (luật phân bố Bayes). Nó chỉ rõ rằng bộ lọc Kalman duy trì hai khoảng thời gian đầu tiên của sự phân bố trạng thái.

$$\begin{aligned} E[x_k] &= \hat{x}_k \\ E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T] &= P_k. \end{aligned} \quad [3-87]$$

Ước lượng trạng thái posteriori [3-83] phản ánh trung bình (khoảng phân bố thời gian đầu tiên) của sự phân bố trạng thái – thông thường nó được sắp xếp nếu các điều kiện của [3-78] và [3-79] xảy ra cùng lúc. Tương quan sai số ước lượng posteriori [3-82] phản ánh giá trị của sự phân bố trạng thái (khoảng phân bố thời gian thứ hai). Theo cách khác,

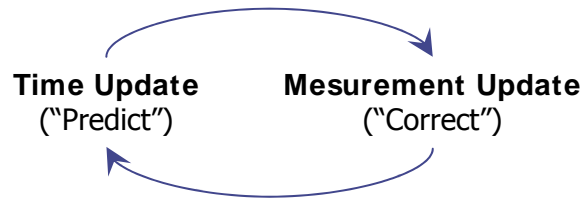
$$\begin{aligned} p(x_k | z_k) &= N(E[x_k], E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]) \\ &= N(\hat{x}_k, P_k). \end{aligned} \quad [3-88]$$

3.3.2.5 Thuật toán bộ lọc Kalman gián đoạn

Bộ lọc Kalman ước lượng tiến trình bằng việc sử dụng hình thức kiểm soát phản hồi: bộ lọc ước lượng trạng thái tiến trình tại vài thời điểm và sau đó thu sự phản hồi trong hình thức của giá trị (độ nhiễu). Chẳng hạn, phương trình của bộ lọc Kalman chia thành hai nhóm: phương trình cập nhật thời gian và phương trình cập nhật giá trị. Phương trình cập nhật thời gian chịu trách nhiệm dự đoán trước (thời gian) giá trị hiện tại và tương quan sai số ước lượng để đạt ước lượng priori cho thời điểm kế tiếp. Phương trình cập nhật giá trị chịu trách nhiệm đối với sự phản hồi – nghĩa là kết hợp giá trị mới vào ước lượng priori nhằm đạt sự cải tiến ước lượng posteriori. ^[14]

Phương trình cập nhật thời gian có thể xem là phương trình chuẩn tắc, trong khi phương trình cập nhật giá trị có thể xem là phương trình chính xác. Thật vậy, thuật

toán ước lượng cuối cùng giống như thuật toán chính xác – chuẩn tắc nhằm giải quyết vấn đề số liệu được trình bày ở hình 3.12.



Hình 3.12 Chu kỳ bộ lọc Kalman gián đoạn

Phương trình đặc biệt cho cập nhật thời gian và giá trị được biểu diễn ở bảng 3-1 và 3-2.

Bảng 3-1: Phương trình cập nhật thời gian bộ lọc Kalman gián đoạn:

$$\hat{x}_k^- = A \hat{x}_{k-1} + B u_{k-1} \quad [3-89]$$

$$P_k^- = A P_{k-1} A^T + Q \quad [3-90]$$

Chú ý rằng phương trình cập nhật thời gian ở bảng 3-1 dự đoán giá trị và tương quan ước lượng trước từ thời điểm $k-1$ đến thời điểm k . A và B theo [3-76], trong khi Q theo [3-78].

Bảng 3-2: Phương trình cập nhật giá trị đo bộ lọc Kalman gián đoạn:

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \quad [3-91]$$

$$\hat{x}_k = \hat{x}_k^- + K_k \left(z_k - H \hat{x}_k^- \right) \quad [3-92]$$

$$P_k = (I - K_k H) P_k^- \quad [3-93]$$

Bài toán đầu tiên trong suốt quá trình cập nhật giá trị được tính toán Kalman gain, K_k . Chú ý rằng phương trình đưa ra ở đây [3-91] giống với [3-84]. Bước kế tiếp là đánh giá thực tiễn trình đo z_k , và sau đó phát ra ước lượng trạng thái posteriori bằng cách kết hợp giá trị như trong [3-92]. Rút gọn [3-92] sẽ được [3-83]. Bước cuối cùng đạt được thông qua ước lượng tương quan sai số posteriori.

Sau mỗi cặp cập nhật thời gian và giá trị, tiến trình được lập lại như trước ước lượng posteriori đã từng dự đoán hay tiên đoán ước lượng priori mới. Trạng thái đệ quy tự nhiên là một trong nhiều điểm đặc trưng thu hút của bộ lọc Kalman – thực tiễn nó tạo sự thi hành khả thi hơn sự thi hành bộ lọc Wiener tạo ra hoạt động của tất cả dữ liệu một cách trực tiếp cho mỗi ước lượng. Bộ lọc Kalman thay thế điều kiện đệ quy ước lượng hiện tại cho giá trị đã qua.

3.3.2.6 Giải thuật bộ lọc Kalman áp dụng cho cảm biến gia tốc và gyro trên PIC C18 Compiler

Cảm biến góc được thiết kế bằng cách kết hợp một cảm biến gia tốc 1 trục và một gyro vận tốc 1 trục. Hai cảm biến này được nối nhau thông qua một bộ lọc Kalman 2 trạng thái, với trạng thái là góc và trạng thái còn lại là giá trị cơ sở gyro (gyro bias).

Gyro_bias được điều chỉnh tự động bởi bộ lọc. Kalman filter là một vấn đề thực sự phức tạp, mặc dù đã được tối ưu nhiều lần đoạn code C.

Khai báo các biến cố định

$$P_{init} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad [3-94]$$

```
float          angle;
float          q_bias;
float          rate;
```

R tượng trưng cho giá trị nhiễu covariance. Trong trường hợp này, nó là ma trận 1x1 được mong đợi 0.03 rad $\approx 18^\circ$ từ gia tốc kế.

$$R_angle = 0.03;$$

Q là ma trận 2x2 tượng trưng cho tiến trình nhiễu covariance. Trong trường hợp này, nó chỉ mức độ tin cậy của gia tốc kế quan hệ với gyro.

$$Q = \begin{bmatrix} Q_angle & 0 \\ 0 & Q_gyro \end{bmatrix} = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.003 \end{bmatrix} \quad [3-95]$$

State_update()

State_update được gọi mỗi dt với giá trị cơ sở gyro bởi người dùng module. Nó cập nhật góc hiện thời và vận tốc ước lượng.

Giá trị gyro_m được chia thành đúng đơn vị thực, nhưng không cần bỏ gyro_bias độ nghiêng. Bộ lọc theo dõi độ nghiêng.

$$\text{Vector giá trị: } X = \begin{bmatrix} angle \\ gyro_bias \end{bmatrix} \quad [3-96]$$

Nó chạy trên sự ước lượng giá trị qua hàm giá trị:

$$\dot{X} = \begin{bmatrix} \dot{angle} \\ \dot{gyro_bias} \end{bmatrix} = \begin{bmatrix} gyro - gyro_bias \\ 0 \end{bmatrix} \quad [3-97]$$

Và cập nhật ma trận covariance qua hàm:

$$\dot{P} = AP + PA' + Q \quad [3-98]$$

$$\dot{P} = A \cdot P + P \cdot A' + Q \text{ (lý thuyết là } P = A \cdot P \cdot A^T + Q \text{)} \quad [3-99]$$

A là Jacobian của \dot{X} với giá trị mong đợi:

$$A = \begin{bmatrix} \frac{d(\dot{angle})}{d(angle)} & \frac{d(\dot{angle})}{d(gyro_bias)} \\ \frac{d(\dot{gyro_bias})}{d(angle)} & \frac{d(\dot{gyro_bias})}{d(gyro_bias)} \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix} \quad [3-100]$$

Vì CPU nhỏ có sẵn trên vi điều khiển, nên tối ưu code C chỉ để tính giới hạn rõ ràng không bằng 0, cũng như khai triển toán ma trận qua vài bước có thể. Cách tính P này làm nó khó đọc hơn, debug và kéo dài hơn thuật toán chính xác của bộ lọc Kalman, nhưng cho phép ít thời gian thực hiện với CPU.

$$\begin{aligned} \dot{P} &= \begin{bmatrix} Q_angle - P[0][1] - P[1][0] & -P[1][1] \\ -P[1][1] & Q_gyro \end{bmatrix} = \\ &= \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} P[0][0] & P[0][1] \\ P[1][0] & P[1][1] \end{bmatrix} + \begin{bmatrix} P[0][0] & P[0][1] \\ P[1][0] & P[1][1] \end{bmatrix} \begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix} + \begin{bmatrix} Q_angle & 0 \\ 0 & Q_gyro \end{bmatrix} \\ &= \begin{bmatrix} -P[1][0] & -P[1][1] \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} -P[0][1] & 0 \\ -P[1][1] & 0 \end{bmatrix} + \begin{bmatrix} Q_angle & 0 \\ 0 & Q_gyro \end{bmatrix} \end{aligned} \quad [3-101]$$

Lưu giữ giá trị ước lượng chưa bias của gyro:

$$rate = q = q_m - q_bias \quad [3-102]$$

Cập nhật ước lượng góc:

$$angle = angle + \dot{angle} \cdot dt = angle + rate \cdot dt \quad [3-103]$$

Cập nhật ma trận covariance:

$$\begin{aligned} P &= P + \dot{P} \cdot dt \\ &= \begin{bmatrix} P[0][0] & P[0][1] \\ P[1][0] & P[1][1] \end{bmatrix} + \begin{bmatrix} Q_angle - P[0][1] - P[1][0] & -P[1][1] \\ -P[1][1] & Q_gyro \end{bmatrix} dt \end{aligned} \quad [3-104]$$

☞ Kalman_update()

Kalman_update được gọi bởi người dùng module khi giá trị gia tốc kế có sẵn. Giá trị angle_m không cần chia thành đơn vị thực tế, nhưng phải được chuẩn mức 0 và có độ chia như nhau.

Procedure này không cần phải gọi mỗi bước thời gian, nhưng có thể nếu dữ liệu gia tốc có sẵn tại vận tốc bằng giá trị vận tốc gyro.

Ma trận C là một ma trận 1x2 (giá trị x trạng thái), đó là ma trận Jacobian của giá trị đo lường với giá trị mong đợi. Trong trường hợp này, C là:

$$C = \begin{bmatrix} \frac{d(angle_m)}{d(angle)} & \frac{d(angle_m)}{d(gyro_bias)} \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad (C \text{ chính là } H) \quad [3-105]$$

Vì giá trị góc đáp ứng trực tiếp với góc ước lượng và giá trị góc không quan hệ với giá trị gyro_bias $\rightarrow C_0$ cho thấy giá trị trạng thái quan hệ trực tiếp với trạng thái ước lượng như thế nào, C_1 cho thấy giá trị trạng thái không quan hệ với giá trị cơ sở gyro ước lượng.

error là giá trị khác nhau trong giá trị đo lường và giá trị ước lượng. Trong trường hợp này, nó khác nhau giữa hai gia tốc kế đo góc và góc ước lượng.

$$angle_error = angle_m - angle \quad [3-106]$$

Tính sai số ước lượng. Từ bộ lọc Kalman:

$$E = CPC^T + R = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} P[0][0] & P[0][1] \\ P[1][0] & P[1][1] \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + R = P[0][0] + R \quad [3-107]$$

$$PCt_0 = C[0]*P[0][0] \quad [3-108]$$

$$PCt_1 = C[0]*P[1][0] \quad [3-109]$$

Ước tính bộ lọc Kalman đạt được. Từ lý thuyết bộ lọc Kalman:

$$\begin{bmatrix} K0 \\ K1 \end{bmatrix} = \begin{bmatrix} PCt_0/E \\ PCt_1/E \end{bmatrix} \quad [3-110]$$

$$P = (I - KH)P = P - KHP = P - KCP \quad [3-111]$$

Ta có phép nhân điểm trôi (floating point):

$$CP = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} P[0][0] & P[0][1] \\ P[1][0] & P[1][1] \end{bmatrix} = \begin{bmatrix} P[0][0] & P[0][1] \end{bmatrix} = \begin{bmatrix} t_0 & t_1 \end{bmatrix} \quad [3-112]$$

$$P = \begin{bmatrix} P[0][0] & P[0][1] \\ P[1][0] & P[1][1] \end{bmatrix} - \begin{bmatrix} K0 \\ K1 \end{bmatrix} \begin{bmatrix} t_0 & t_1 \end{bmatrix} = P - \begin{bmatrix} K0 \cdot t_0 & K0 \cdot t_1 \\ K1 \cdot t_0 & K1 \cdot t_1 \end{bmatrix} \quad [3-113]$$

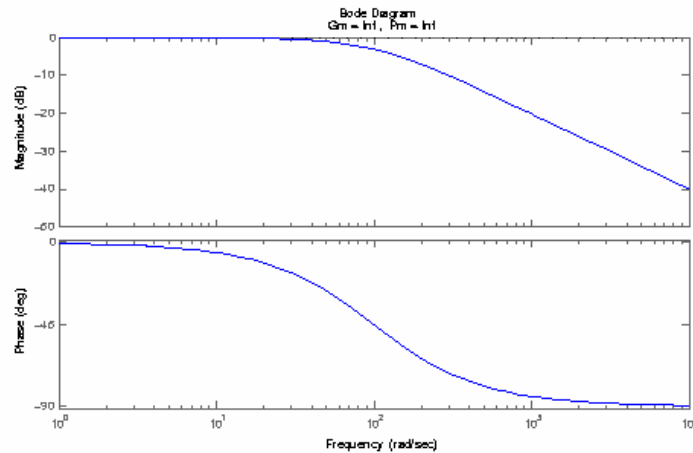
Cập nhật giá trị ước lượng. Lần nữa, từ Kalman:

$$X = \begin{bmatrix} angle \\ gyro_bias \end{bmatrix} = X + K(Z_{measure} - HX_{estimate}) = X + K \begin{bmatrix} angle_err \\ gyro_err \end{bmatrix} \quad [3-114]$$

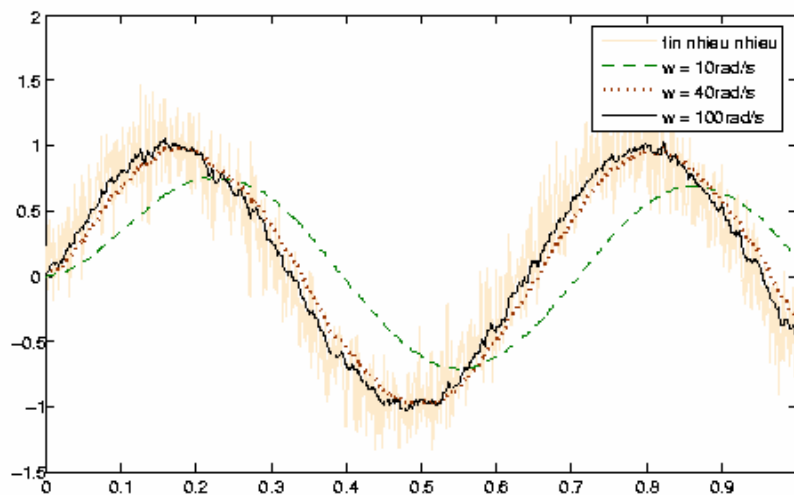
$$angle_err = q_bias_err \quad [3-115]$$

3.3.3 So sánh các bộ lọc với bộ lọc Kalman

Đối với bộ lọc thông thấp, thông cao hoặc thông dải (lọc thụ động) xấp xỉ Butterworth, Bessel và Chebychev hay elliptic: thường được sử dụng cho một tín hiệu vào và một tín hiệu ra, với tần số làm việc xác định. Ngoài dải tần này, tín hiệu sẽ bị lệch pha, hoặc độ lợi không còn là hằng số mà bị tối thiểu hóa. Do vậy trong tình huống này, ta dùng hai cảm biến để đo một giá trị là góc (cũng như vận tốc góc), nên việc chỉ dùng một bộ lọc thụ động tỏ ra không phù hợp.



Hình 3.13 Giản đồ Bode của bộ lọc thông thấp bậc nhất $\omega_0 = 100 \text{ rad/s}$

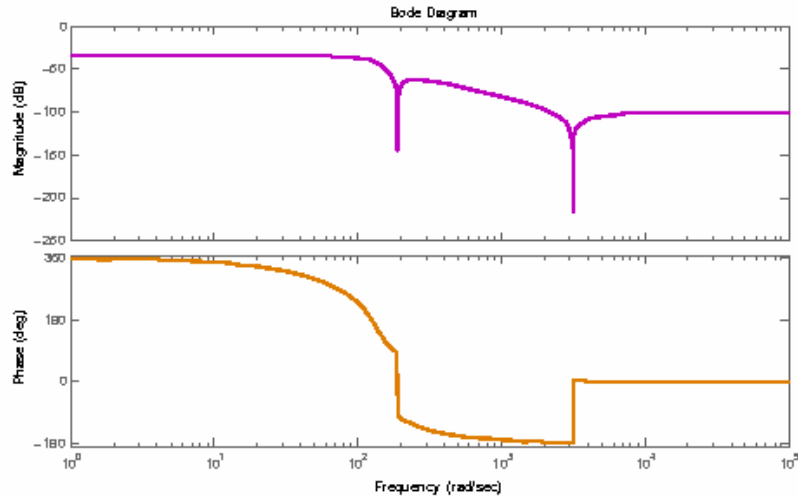


Hình 3.14 Giản đồ Bode và pha của bộ lọc bậc nhất

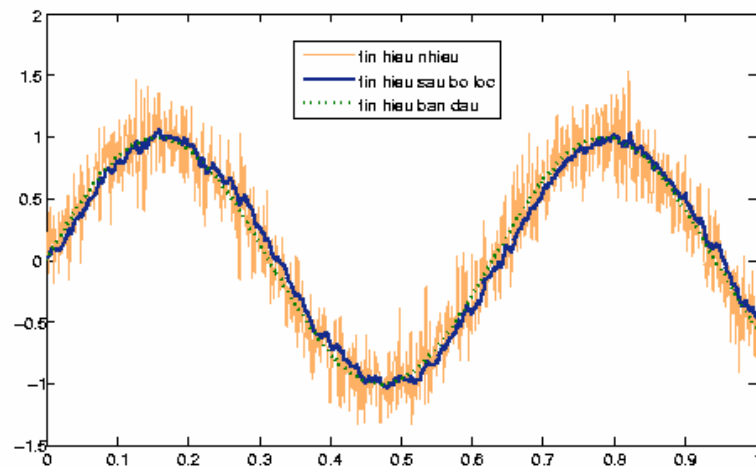
$$G(s) = \frac{\omega_0}{s + \omega_0} \text{ với } \omega_0 = 10, 40 \text{ và } 100 \text{ rad/s}$$

Ta có thể dùng bộ lọc bổ phụ (complementary filter) để kết nối hai tín hiệu từ accelerometer và gyro thành một tín hiệu duy nhất. Accelerometer được qua một bộ lọc thông thấp, còn gyro được qua một bộ lọc thông cao, từ đó, hai tín hiệu đã được lọc sẽ được nối với nhau thành một tín hiệu thống nhất. Ưu điểm của bộ lọc bổ phụ là tính toán nhanh, dễ thiết kế. Nhược điểm của bộ lọc này là bản chất vẫn của bộ lọc

thông cao và thông thấp, có nghĩa độ lợi tín hiệu không bằng nhau trong toàn dải đo, bị lệch pha rõ rệt tại vùng nổi tần số. Hơn nữa giá trị gyro_bias không được cập nhật thường xuyên, dễ làm cho bộ lọc mất tác dụng khi làm việc ở những môi trường rung động hay có nhiệt độ khác nhau. Ngoài ra, cũng phải kể đến việc chuẩn trực bộ lọc này khá khó khăn nếu không có thiết bị quan sát.



Hình 3.15 Giản đồ Bode của bộ lọc elliptic



Hình 3.16 So sánh các tín hiệu ở bộ lọc elliptic

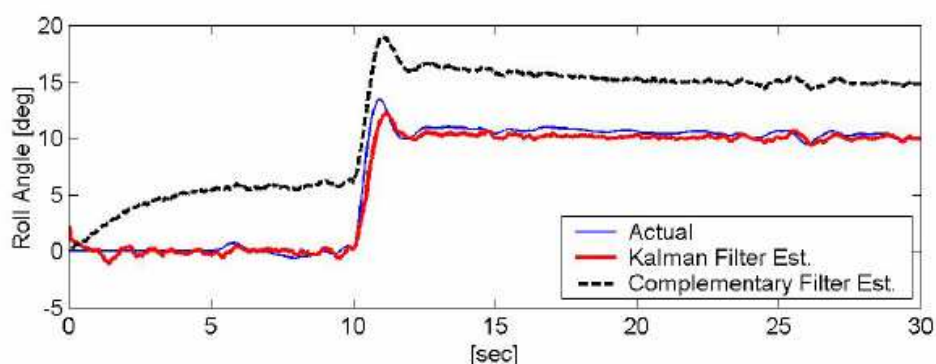
Giản đồ Bode và pha của bộ lọc xấp xỉ elliptic có các tham số trạng thái như sau:

$$A = \begin{bmatrix} -153.34 & -94.989 & 0 & 0 \\ 94.989 & 0 & 0 & 0 \\ -153.34 & 104940 & -62.975 & -138.22 \\ 0 & 0 & 138.33 & 0 \end{bmatrix}; B = \begin{bmatrix} 120 \\ 0 \\ 120 \\ 0 \end{bmatrix}$$

$$C = [-1.2726e-5 \ 8.7089e-3 \ -5.2265e-6 \ 1.0191e-5]; D = 9.9592e-6$$

Nói tóm lại, các bộ lọc thông thường là một kỹ thuật dùng phần cứng (các mạch điện tử R,L,C) hoặc phần mềm (lọc FIR, lọc IIR, cửa sổ Hamming ... trong xử lý tín hiệu số) là nhằm giữ lại các tín hiệu trong một khoảng thông dải tần số nào đó và loại bỏ tín hiệu ở các dải tần số còn lại. Đối với việc xây dựng bộ lọc bằng phần cứng, ra đời trước khi dùng các bộ lọc phần mềm, nhưng việc hiệu chỉnh đặc tính, thay đổi các tham số của bộ lọc phức tạp hơn rất nhiều so với sử dụng giải thuật xử lý tín hiệu số.

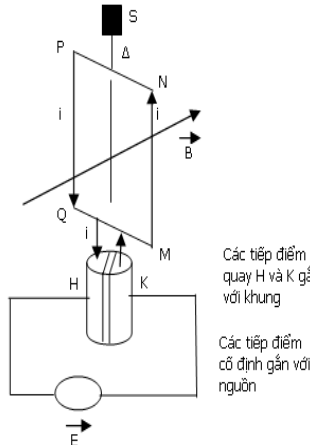
Trong các bộ lọc này, nếu tồn tại các tín hiệu nhiễu trong dải thông tần thì kết quả tín hiệu trở nên kém đi rất nhiều để có thể xử lý và điều khiển hệ thống một cách ổn định. Điều này càng tỏ ra rất thực tế đối với các bộ lọc phần cứng, vốn rất dễ bị nhiễu bởi các tín hiệu điện trong lúc hoạt động do sự kém chính xác của các linh kiện và sự bất thường của dòng điện ngõ vào.



Hình 3.17 So sánh các tín hiệu quay thực ở ngõ vào(lam), kết quả ngõ ra ở bộ lọc bổ phụ (gạch đen) và bộ lọc Kalman(đỏ)

Đối với bộ lọc Kalman, thuật ngữ “lọc” không có ý nghĩa như các bộ lọc trên. Đây là một giải thuật tính toán và ước lượng thống kê tối ưu tất cả các thông tin ngõ vào được cung cấp tới nó để có được một giá trị ra đáng tin cậy nhất cho việc xử lý tiếp theo. Do vậy lọc Kalman có thể sử dụng để loại bỏ các tín hiệu nhiễu mà được mô hình là những tín hiệu nhiễu trắng trên tất cả dải thông mà nó nhận được từ ngõ vào, dựa trên các thông kê trước đo và chuẩn trực lại giá trị ước lượng bằng các giá trị đo hiện tại với độ lệch pha gần như không tồn tại và có độ lợi tối thiểu xấp xỉ 0 đối với những tín hiệu ngõ vào không đáng tin cậy. Mặc dù phải tốn khá nhiều thời gian xử lý lệnh, nhưng với tốc độ hiện tại của các vi điều khiển làm việc tính toán ước lượng tối ưu của bộ lọc này trở nên đơn giản và đáng tin cậy rất nhiều. Nhờ có cơ chế tự cập nhật các giá trị cơ sở (bias) tại mỗi thời điểm tính toán, cũng như xác định sai lệch của kết quả đo trước với kết quả đo sau nên giá trị đo luôn được ổn định, chính xác, gần như không bị sai số về độ lợi và độ lệch pha của các tín hiệu. Hơn thế, được xây dựng bởi hàm trạng thái, do vậy bộ lọc Kalman có thể kết hợp không chỉ hai tín hiệu từ hai cảm biến, mà có thể kết hợp được nhiều cảm biến đo ở những dải tần khác nhau của cùng một giá trị đại lượng vật lý. Chính vì điều này, làm bộ lọc Kalman trở nên phổ dụng hơn tất cả những bộ lọc khác trong việc xử lý tín hiệu chính xác của các cảm biến tọa độ, như cảm biến la bàn, GPS, góc, gyro...

Hiện nay, với sự phát triển của lĩnh vực trí tuệ nhân tạo (AI), các ứng dụng AI được sử dụng trong nhiều lĩnh vực, đặc biệt là định hướng trong hàng không vũ trụ, nhưng không vì thế mà bộ lọc Kalman giảm vai trò, mà ngược lại, đó chính là một ngõ ra đáng tin cậy để cho mạng nơron xử lý và ra các quyết định trong các tàu thám hiểm tự hành.



Hình 3.18 Mô hình khung dây động cơ đặt trong từ trường

3.4 Mô hình lý thuyết động cơ DC ^[10]

Động cơ DC có thể được mô hình hóa một cách đơn giản như hình vẽ dưới đây ^[9].

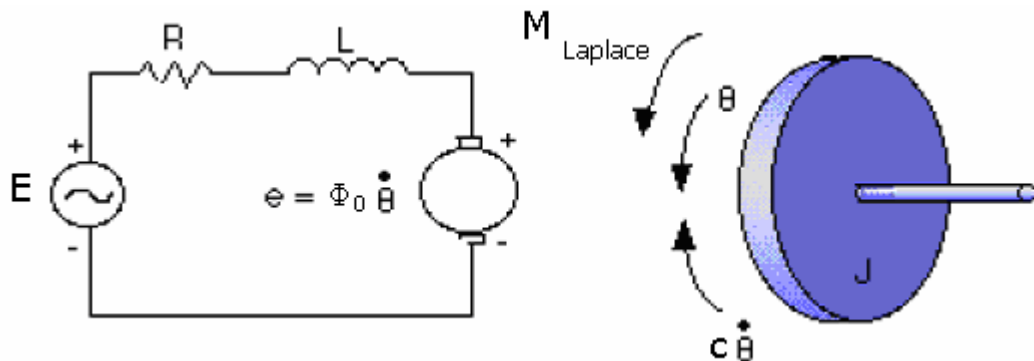
Ở đây, động cơ gồm một cuộn dây gồm N vòng dây quấn quanh một khung chữ nhật cạnh a, b có thể quay xung quanh một trục Δ . Vị trí của nó được xác định bởi góc quay θ . Cuộn dây có điện trở tổng cộng R và độ tự cảm L.

Hệ chuyển động có momen quán tính J với trục Δ . Một nam châm vĩnh cửu tạo ra một từ trường B xuyên qua khung dây.

Hệ cơ S tác dụng lên trục một ngẫu lực cản, kí hiệu $(-\Gamma)$, được tính theo công thức:

$$\Gamma = c\dot{\theta} \quad [3-116]$$

với c là hằng số giảm chấn của kết cấu cơ khí (bộ truyền động ở trục động cơ).



Hình 3.19 Mô hình điện cơ

Khi ta đặt điện áp một chiều E vào 2 đầu chổi điện H và K thì trong cuộn dây phần ứng có dòng điện I_U . Theo hiện tượng cảm ứng điện từ khung dây sẽ chịu tác dụng của lực điện trường (lực Laplace) và quay quanh trục Δ . Chú ý rằng cứ mỗi nửa vòng lại chuyển mạch, ở đây H và K là 2 vành bán nguyệt. Điều này bảo đảm cho khung luôn quay theo 1 chiều nhất định với vận tốc góc $\omega = \dot{\theta}$.

$E(t)$, $I(t)$ là tổng điện áp cảm ứng và dòng dẫn trong động cơ. $C(t)$, $\theta'(t)$ là ngẫu lực điện từ và vận tốc xoay của rotor. Ta có công thức:

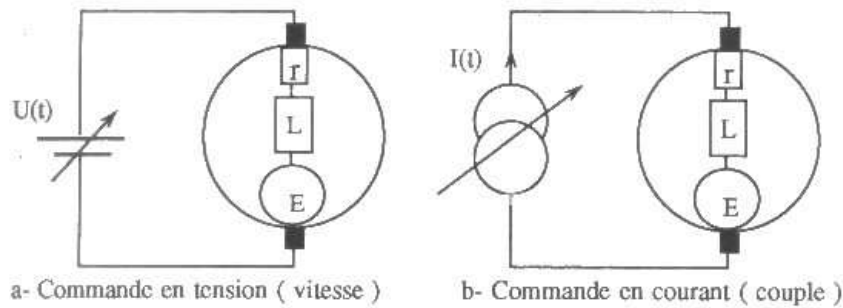
$$E(t) = k.\theta'(t) \quad [3-117]$$

$$C(t) = k.I(t) \quad [3-118]$$

Với k là hằng số phụ thuộc vào đặc tính của động cơ, loại cuộn dây, giá trị từ trường, thường là hằng số với nam châm vĩnh cửu.

Cuộn dây dẫn được mô hình hóa bởi điện trở R và độ dẫn L , được áp vào điện áp $U(t)$, có phương trình dưới đây:

$$U(t) = r.I(t) + L.\frac{d}{dt}I(t) + E(t) \quad [3-119]$$



Hình 3.20 Sơ đồ động cơ điện

Phương trình cơ học ^[11]:

Trong mỗi quan hệ ở trên, sự phụ thuộc vận tốc góc $\theta'(t)$ theo $C(t)$ là đặc tính cơ học của quán tính vật rắn và ma sát xoắn.

Với J : momen quán tính, là hằng số; $F_s(t)$ là ngẫu lực xoay và f_v là hệ số ma sát quay, ta có phương trình cơ học:

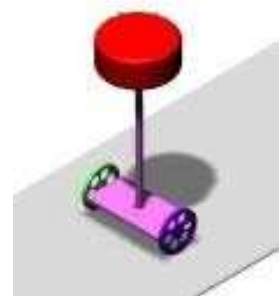
$$J.\frac{d}{dt}\theta(t) = C(t) - F_s(t) - f_v.\theta_v.\theta(t) \quad [3-120]$$

Chương 4

TÍNH TOÁN MÔ PHỎNG

4.1 Các thông số khối lượng, công suất dự tính trong mô hình mô phỏng được xây dựng

- Tổng trọng lượng người sử dụng mô hình: 60kg trở xuống.
- Pin: Ba accu khô 12V-7Ah, nặng 2,2kg/bình. Một pin 12V – 1.6Ah điều khiển phần tín hiệu và kích thích nặng 560g.
- Bánh xe: nặng 7,2kg/bánh. Đường kính bánh xe: >40cm, chu vi khi bơm căng là 1,33m.
- Khung xe: bằng inox nặng 3kg.
- Sàn xe: bằng nhôm 5mm, nặng 1,4kg.
- Tay điều khiển: bằng nhôm hộp và nhựa, nặng 2,5kg.
- Ốc vít: tối đa 2kg.
- Phần điện tử:
 - Bộ phận tản nhiệt cho MOSFET: 300g/tấm x 4 = 1,2kg
 - Bo mạch in, cảm biến: tối đa 2kg



Hình 4.1 Mô hình xe scooter

❖ Tổng trọng lượng xe:

$$2,2 \times 3 + 0,56 + 7,2 \times 2 + 3 + 1,4 + 2,5 + 2 + 1,2 + 2 + 60 = 93,66 \text{ kg} \quad [4-1]$$

Hàm trạng thái của mô hình: như đã tính toán ở chương 3, ta có:

$$\begin{bmatrix} \dot{x}_{bánh} \\ \ddot{x}_{bánh} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & g \left(1 - \frac{4}{3} L \frac{M_{thân}}{X} \right) \\ 0 & 0 & 0 \\ 0 & 0 & + \frac{g M_{thân}}{X} \end{bmatrix} \begin{bmatrix} x_{bánh} \\ \dot{x}_{bánh} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \left(\frac{4LY}{3X} - \frac{1}{M_{thân}L} \right) \\ 0 \\ -\frac{Y}{X} \end{bmatrix} C_{\theta} \quad [4-2]$$

$$\begin{bmatrix} x_{bánh} \\ \theta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_{bánh} \\ \dot{x}_{bánh} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} C_{\theta} \quad [4-3]$$

$$\text{với } X = \left(\frac{4}{3} M_{thân} L - \frac{M_{thân}}{(2M_{bánh} + M_{thân})} (M_{thân} L + M_{bánh} R) \right) \quad [4-4]$$

$$Y = \frac{M_{thân}}{(2M_{bánh} + M_{thân})} \frac{1}{R} + \frac{1}{L} \quad [4-5]$$

4.2 Mô phỏng MatLAB

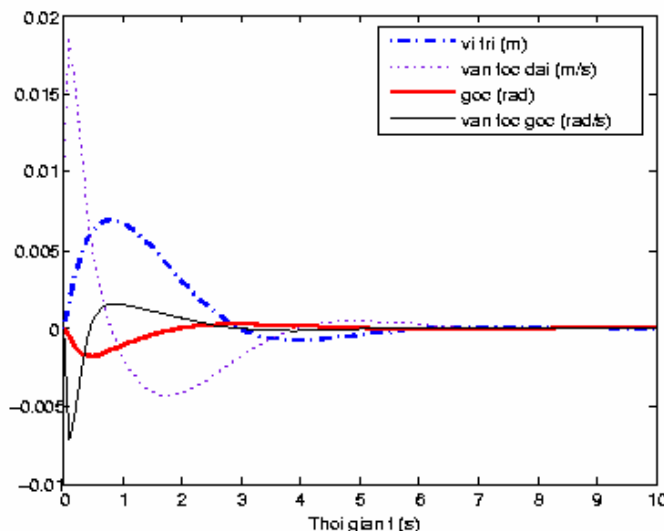
4.2.1 Giới thiệu về phần mềm MatLAB, công cụ Simulink

MatLAB là một môi trường tính toán đa ứng dụng, viết tắt của “Matrix Laboratory” – tối ưu trong việc tính toán ma trận. MatLAB là công cụ không thể thiếu đối với những người nghiên cứu trong lĩnh vực toán, điện – điện tử, cơ khí và các ngành nghề thuộc lĩnh vực khoa học ứng dụng. Với khả năng tính toán cao và được nhúng với các phần mềm mô phỏng, cũng như tự bản thân đã có sẵn các công cụ mô phỏng, MatLAB giúp tiết kiệm thời gian và công sức trong tính toán cũng như thử nghiệm các mô hình trước khi xây dựng mô hình thực tế. Các tính toán về số và đồ họa được thực hiện dễ dàng trong MatLAB với cấu trúc của ngôn ngữ thủ tục, nghĩa là tổ hợp một cấu trúc chương trình với hàng loạt các hàm toán học đã được lập trình trước, vì vậy lập trình bằng MatLAB rất đơn giản.

4.2.2 Kết quả tính bằng MatLAB

a) Chọn cực $V = [-0.7+i; -0.7-i; -8.2+3i; -8.2-3i]$, ta được:

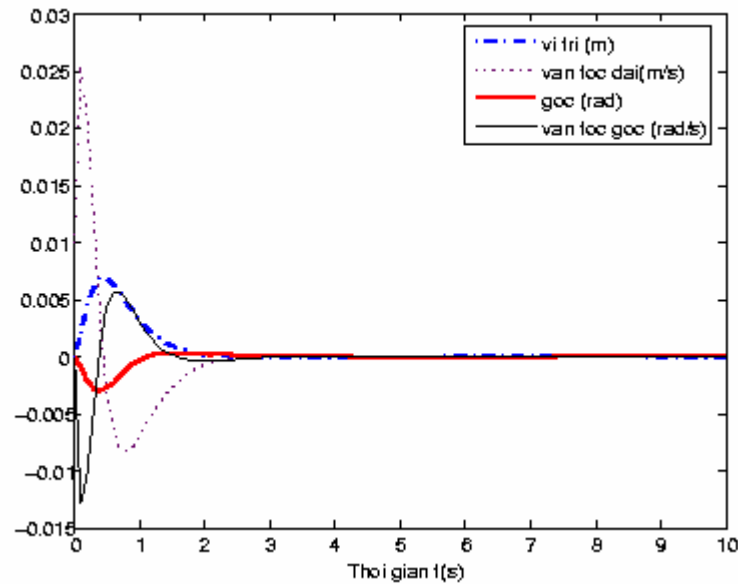
$$K = [-87.4505 \ -100.9797 \ -834.0498 \ -232.1682]$$



Hình 4.2 Kết quả mô phỏng với mong muốn cực của hệ thống là $V = [-0.7+i; -0.7-i; -8.2+3i; -8.2-3i]$

b) Chọn cực $V = [-2 - i; -2 + i; -5.4 + 3i; -5.4 - 3i]$, ta được

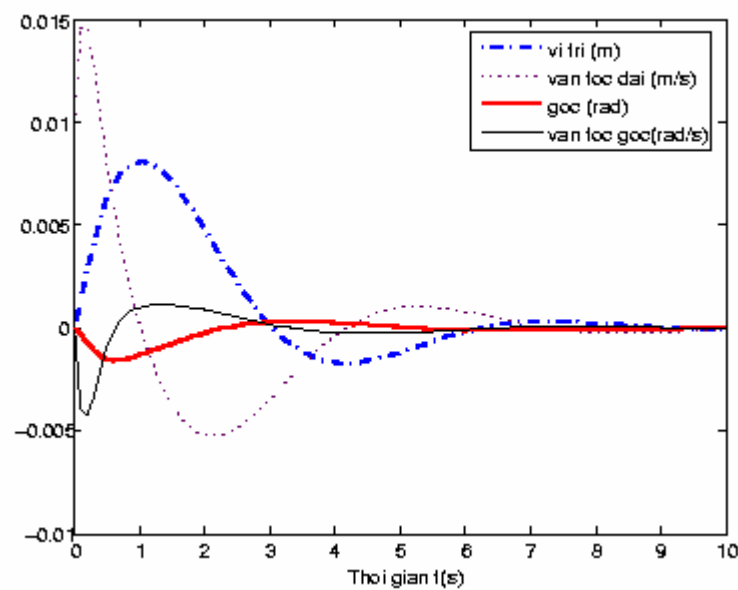
$$K = [-146.8829 \ -159.0770 \ -820.5842 \ -285.6180]$$



Hình 4.3 Kết quả mô phỏng với mong muốn cực của hệ thống là $V = [-2+i; -2-i; -5.4+3i; -5.4-3i]$

c) Chọn cực $V = [-0.5 - i; -0.5 + i; -5.4 + 3i; -5.4 - 3i]$, ta được

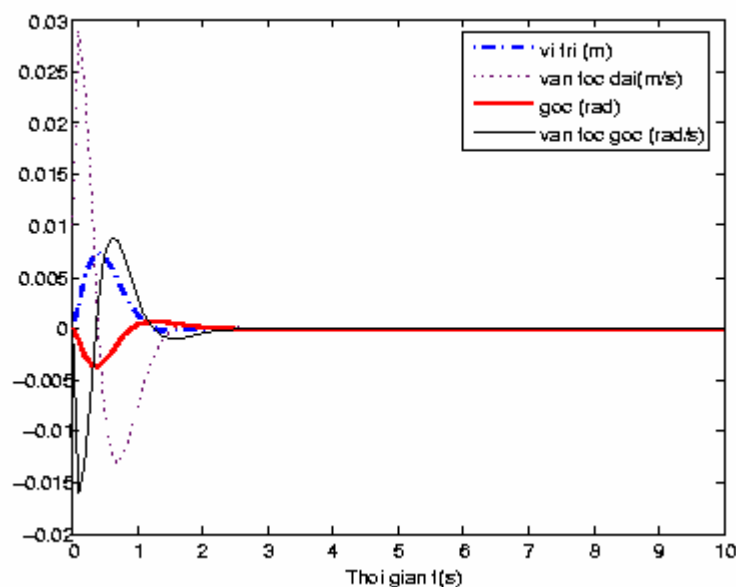
$$K = [-36.7207 \ -39.7692 \ -466.0578 \ -120.4199]$$



Hình 4.4 Kết quả mô phỏng với mong muốn cực của hệ thống là $V = [-0.5+i; -0.5-i; -5.4+3i; -5.4-3i]$

d) Chọn cực $V = [-3.6; -3.6; -3.6+3i; -3.6-3i]$, ta được

$$K = [-219.0939 \ -193.5529 \ -912.0032 \ -325.6883].$$



Hình 4.5 Kết quả mô phỏng với mong muốn cực của hệ thống là $V = [-3.6; -3.6; -3.6+3i; -3.6-3i]$

☞ Nhận xét trên các biểu đồ, về mặt lý thuyết, cả bốn giá trị K trên đều có thể dùng được vì tất cả đều làm hệ thống ổn định. Về mặt thực tế khi thực hiện mô hình, ta chọn giá trị V thứ ba, vì độ khuếch đại cho ngõ ra không nhiều, ít làm động cơ quá tải khi đảo chiều quay hơn các kết quả kia (đặc tính của động cơ khi bị đảo chiều tức thời ở vận tốc lớn sẽ tạo ra một dòng điện ngược rất lớn, dễ làm hư hỏng các mạch điều khiển), mặc dù thời gian ổn định có lâu hơn các cách chọn cực khác (7 giây so với gần 3 giây). Cũng vì đặc tính của động cơ về quan hệ giữa ngẫu lực và vận tốc, việc chọn kết quả V thứ ba cũng sẽ duy trì ngẫu lực tốt và ổn định hơn các giá trị V khác.



Hình 4.6 Mô hình mô phỏng từ Visual Nastran

4.3 Mô phỏng VisualNastran và Simulink

4.3.1 Giới thiệu về phần mềm VisualNastran

Đối với những người thiết kế trong các lĩnh vực thiết bị công nghiệp, thiết bị cho xây dựng, thể thao hay giải trí, thiết bị y tế, chi tiết động cơ máy, sản phẩm tiêu dùng... thì có lẽ quan tâm hàng đầu của họ là muốn biết sản phẩm của mình vận hành như thế nào trong điều kiện thực tế. Như vậy, nhu cầu về một phần mềm có thể mô phỏng thực nhất hoạt động của các thiết bị cơ khí là một nhu cầu có thực, có khá nhiều phần mềm có khả năng thực hiện các chức năng mô phỏng, tuy nhiên đáng quan

tâm hơn cả là phần mềm VisualNastran Desktop của hãng MSC. Working Knowledge. Một phần mềm kết hợp kỹ thuật sáng tạo, quan sát và mô phỏng trên môi trường CAD. VisualNastran Desktop kết hợp mô phỏng chuyển động tiên tiến và kỹ thuật phân tích phần tử hữu hạn tinh vi nhưng sử dụng lại rất dễ dàng.

VisualNastran Desktop gồm năm module: VisualNastran View, VisualNastran Studio, VisualNastran Motion, VisualNastran Desktop FEA và VisualNastran 4D (hay VisualNastran Desktop-vN4D, WorkingModel3D). Mô phỏng trong đề tài được sử dụng chủ yếu bằng phần mềm vNastran Desktop 4D.

Phần mềm VisualNastran có khả năng: mô phỏng chuyển động với độ chính xác cao và lập trình điều khiển chuyển động của các đối tượng.

4.3.2 Cách thực hiện mô phỏng bằng vN Desktop 4D

Vẽ mô hình scooter bằng phần mềm Solidworks theo đúng kích thước thật của xe.

Chuyển hình vẽ này cùng các thông số kích thước sang phần mềm VisualNASTRAN, vì hai phần mềm này có khả năng kết hợp được với nhau.

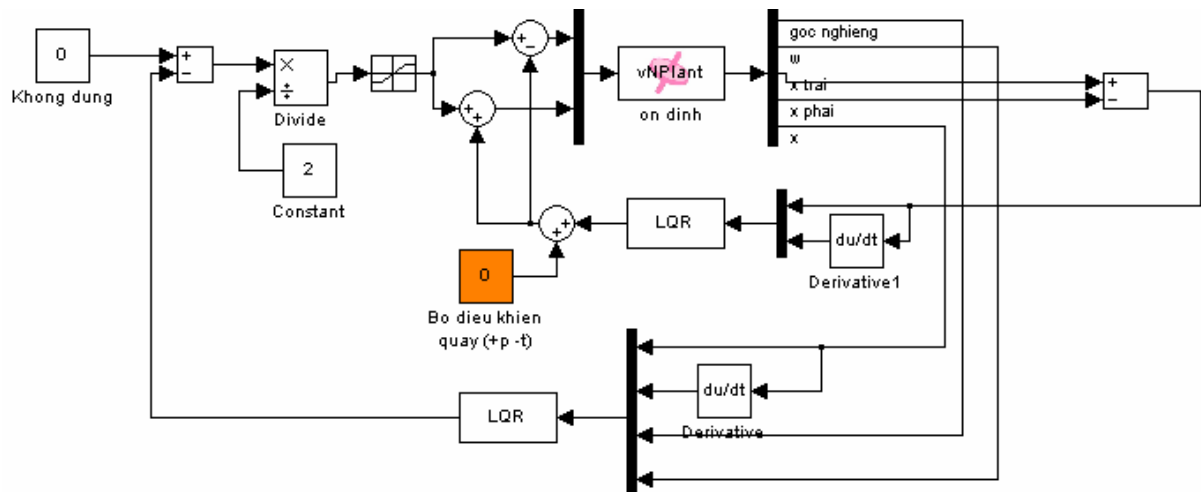
Xác định các đại lượng vật lý cho mô hình cơ khí, chẳng hạn về các tham số khối lượng, độ nhớt, độ ma sát, vận tốc gắn vào các động cơ vào bánh xe.

Dùng vN Plant Block của phần mềm VisualNastran Desktop 4D nhúng vào trong Simulink của MatLAB, các thông số về góc nghiêng, vị trí sẽ được nhúng qua bộ xử lý toán học của MatLAB đồng thời MatLAB sẽ trả giá trị điều khiển về cho vN Nastran nhờ block này. Nói cách khác, các thông số mô phỏng sẽ được hai phần mềm này tương tác với nhau tạo ra.

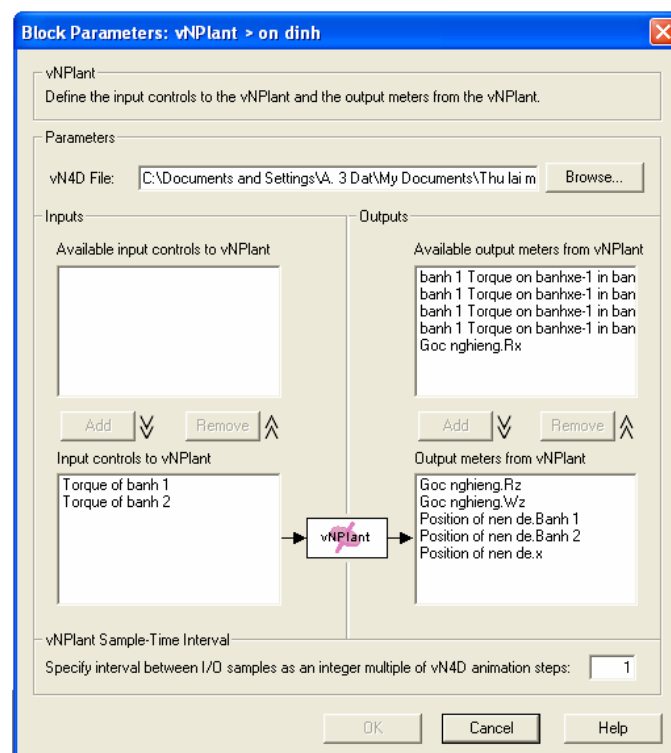
Dùng MatLAB để vẽ mô hình liên kết Simulink, xác định tham số hồi tiếp K điều khiển như trên chương 3 trình bày đồng thời hiệu chỉnh sao cho được kết quả tốt nhất.

4.3.3 Sơ đồ khối trong Simulink:

Mô tả hoạt động của khối Simulink: ứng với mỗi khoảng thời gian nhất định, các thông số vật lý của mô hình xe hai bánh sẽ được cập nhật từ vN Nastran 4D vào MatLAB thông qua khối chức năng vN Plant Block. Các giá trị vật lý này sẽ được đưa vào bộ hồi tiếp LQR có chứa tham số hồi tiếp K để trả ngược về cho mô hình xe 2 bánh tự cân bằng dưới đại lượng là ngẫu lực. Do giá trị ngẫu lực được tính ra là ngẫu lực tổng C_θ , mà ngẫu lực ở mỗi bánh chỉ bằng một nửa C_θ (theo mô hình toán ở chương 3), do vậy cần thêm một bộ chia đôi giá trị ngẫu lực trước khi đưa trở về mô hình qua vN Plant Block. Một bộ hồi tiếp khác được dùng để chỉnh lại vận tốc của hai bánh xe cho bằng nhau trong trường hợp vận tốc quay của hai bánh xe không đồng đều.



Hình 4.7 Mô hình Simulink để mô phỏng vòng hồi tiếp K tính toán momen cho động cơ

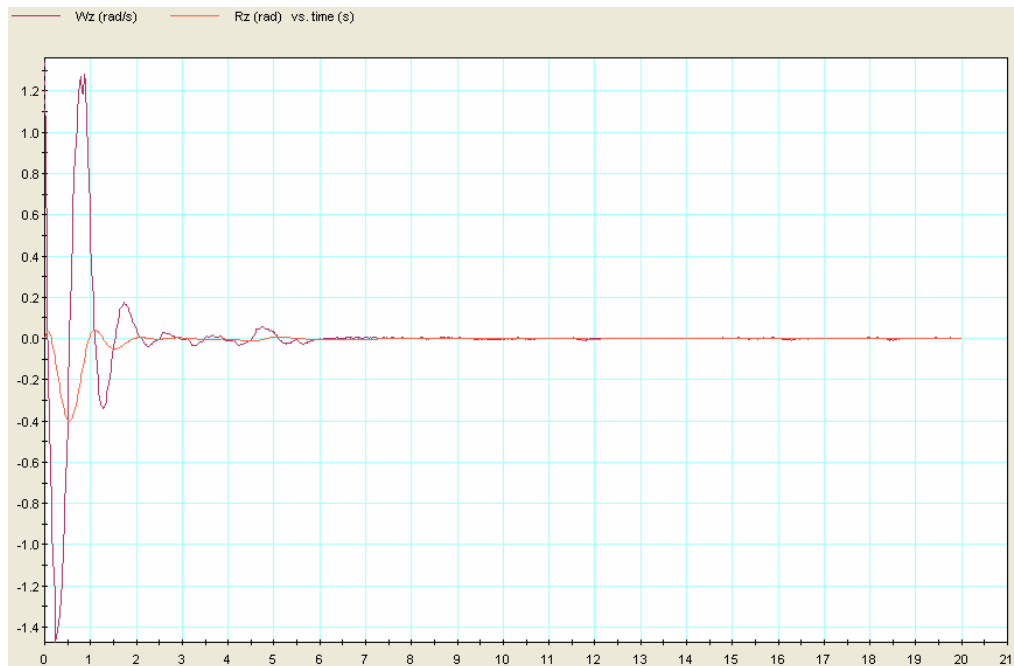


Hình 4.8 Bộ giao diện liên kết các tín hiệu giữa MatLAB-Simulink và Visual Nastran 4D

4.3.4 Kết quả mô phỏng

- ❖ Ổn định, giữ thăng bằng tại một vị trí cố định

$$K = [-36.7207 \quad -39.7692 \quad -466.0578 \quad -120.4199]$$



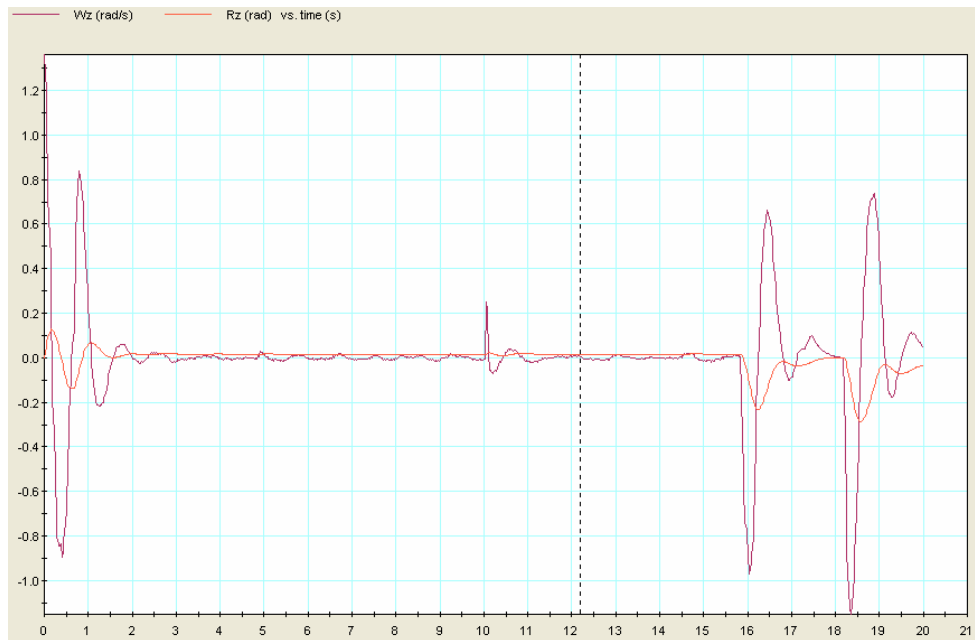
Hình 4.9 Biểu đồ đo góc nghiêng



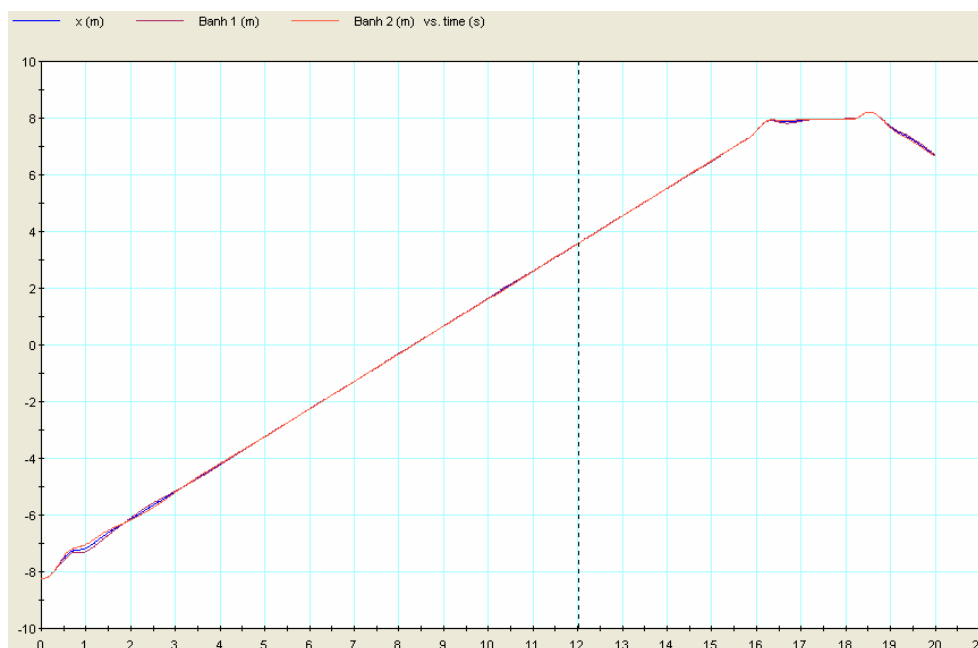
Hình 4.10 Biểu đồ đo vị trí

☞ *Nhận xét kết quả mô phỏng:* việc ổn định góc của mô hình được hoàn tất trong thời gian rất nhanh (6 giây), sau đó mới tiến về vị trí ban đầu mất một thời gian dài hơn (10 giây), tuy nhiên trong thời gian tiến về vị trí ban đầu góc đứng của mô hình luôn được đảm bảo xấp xỉ 0° , do đó nếu vận chuyển người, thì sẽ không gây cảm giác lắc lư khó chịu cho người sử dụng.

Giữ cân bằng góc, vị trí di chuyển (điều khiển PD về góc nghiêng, P vận tốc) với $K = [0 \ -39.7692 \ -466.0578 \ -120.4199]$



Hình 4.11 Biểu đồ góc nghiêng



Hình 4.12 Biểu đồ vị trí góc nghiêng

☞ *Nhận xét kết quả mô phỏng:* việc ổn định góc của mô hình cũng được hoàn tất trong thời gian rất nhanh (2 giây), tuy nhiên mỗi lần thay đổi vận tốc lại có dao động về góc nghiêng trong thời gian quá độ vận tốc, do vậy có lẽ cần phải có ổn định góc tự bản thân mô hình trước khi thay đổi có thể tiếp tục chuyển động.

Với hai kết quả mô phỏng trên, ta nhận thấy hoàn toàn có thể điều khiển mô hình xe hai bánh tự cân bằng khi di chuyển cũng như đứng yên bằng một dạng bộ điều khiển hồi tiếp ổn định K như đã mô tả.

Chương 5

THỰC HIỆN

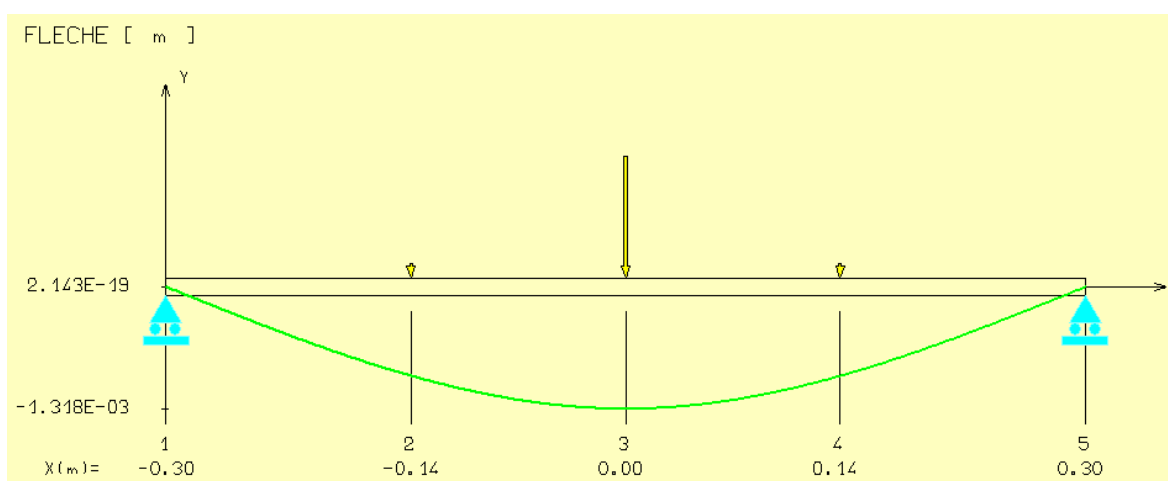
5.1 Thiết kế cơ khí

5.1.1 Tóm tắt thiết kế

Cấu trúc cơ khí của đề tài được chia làm bốn phần: khung, sàn, tay lái và hai động cơ. Khung được làm bằng inox $\varnothing 25$ và inox vuông $25 \times 25 \text{ mm}$. Sàn được làm bằng nhôm dày 5mm, vững chắc. Tay lái cũng được làm bằng inox vuông có kích thước $25 \times 25 \text{ mm}$ và tay cầm của tay lái được làm bằng ống nhựa có $\varnothing 21$, dài 250mm. Hai động cơ đặt trên hai bánh xe có đường kính 40cm. Cấu trúc cơ khí được mô tả chi tiết trên năm bản vẽ, bao gồm một bản vẽ A0 là bản vẽ lắp, ba bản vẽ A3, đó là các bản vẽ khung, bản vẽ sàn, bản vẽ tay lái và một bản vẽ A4 là bản vẽ trục của động cơ.

5.1.2 Tính toán sức bền

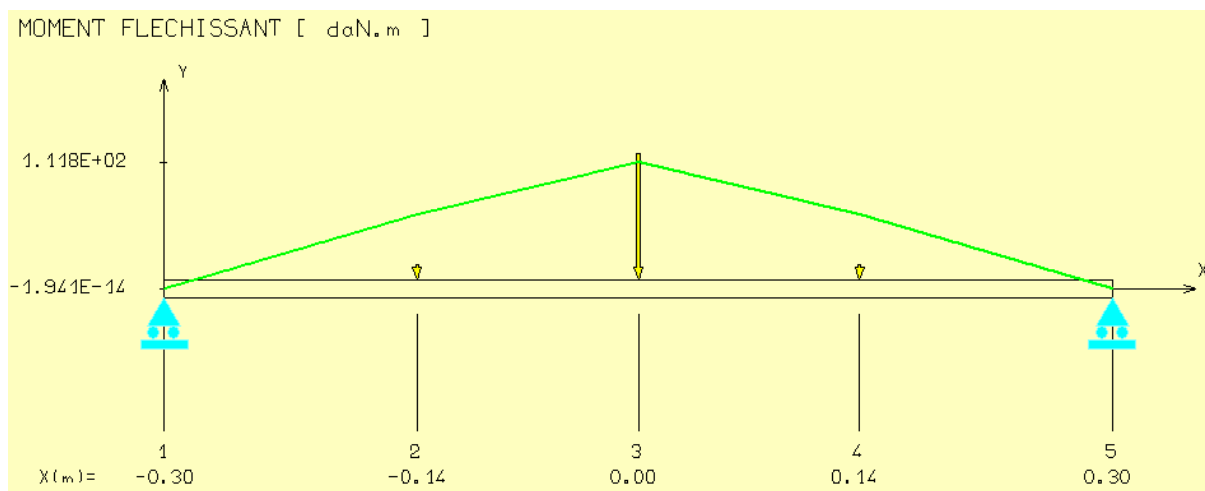
Do cấu trúc của mô hình, sàn xe được đỡ bằng hai thanh inox vuông với kích thước là $25 \text{ mm} \times 25 \text{ mm}$, dày 1mm, ta trước hết thử tính ước lượng liệu một thanh inox như vậy có chịu được toàn bộ mô hình hay không, nếu được thì ta sẽ không cần tính cụ thể mô hình, vì mô hình đã dư bền, nếu một thanh inox không chịu được, ta sẽ tính cụ thể cho toàn bộ mô hình. Để thuận tiện và nhanh chóng trong tính toán, ta sử dụng phần mềm tính toán phần tử hữu hạn cho kết cấu RDM6. Lưu ý rằng, biểu đồ momen uốn của phần mềm theo quy định của Pháp ngược chiều với tiêu chuẩn của Việt Nam.



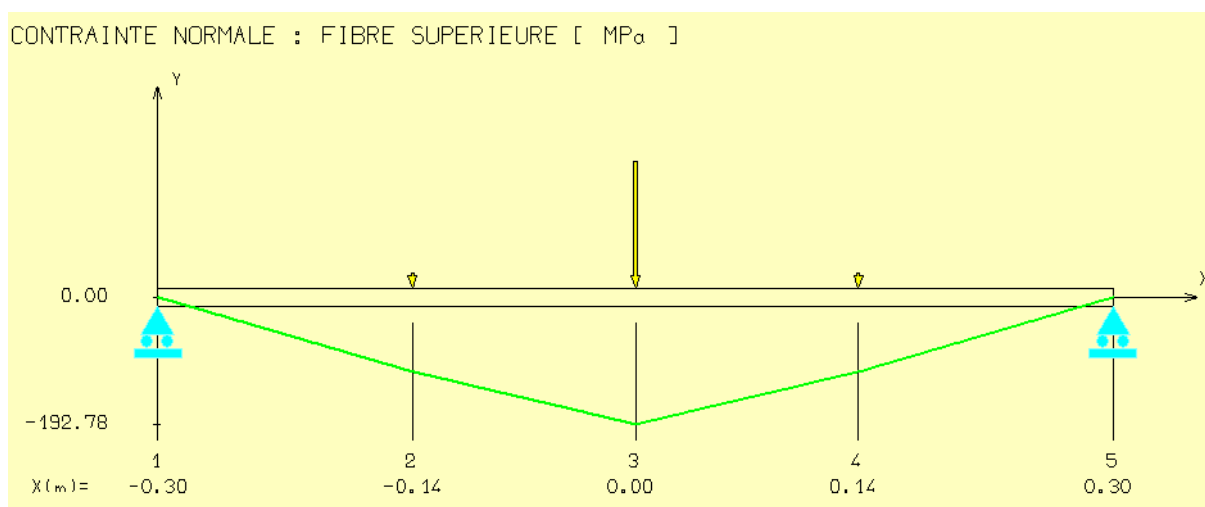
Hình 5.1 Chuyển vị của một thanh inox nếu toàn bộ tải trọng đặt lên thanh

Mô hình tính toán là một thanh dầm inox hình vuông kích thước như trên đã mô tả: lực tập trung chính giữa có giá trị là 660N, tương đương với một người 60kg và phần mạch điện điều khiển; hai lực tập trung hai bên, cách điểm giữa là 0,14m có giá

trị là 80N, tương đương trọng lượng của các bình ắc quy, ốc vít, và những linh kiện khác. Hai bánh xe đóng vai trò như hai gối đỡ, trọng lượng của hai bánh xe khi di chuyển chỉ tác dụng lên sàn, không ảnh hưởng lên khung trong mô hình tính sức bền.



Hình 5.2 Momen uốn nhìn theo phương ngang



Hình 5.3 Ứng suất uốn nhìn theo phương ngang

Theo bảng tra của phần mềm RDM6, giới hạn ứng suất chảy của inox là 200-220MPa, theo sơ đồ trên ta thấy ứng suất tối đa của mô hình tác dụng là 192,78MPa nếu toàn bộ tải trọng mô hình tập trung vào một cây inox. Mà mô hình thật có 2 thanh inox cùng một sàn bằng nhôm đỡ toàn bộ tải trọng, do vậy ứng suất trên một thanh inox này sẽ nhỏ hơn một nửa giá trị tính ở trên, cụ thể $\sigma_{\text{thanh}} \leq 192,78/2 = 96,39$ MPa. Vì vậy, phần thiết kế cơ khí đủ bền và không bị chảy dẻo trong suốt toàn bộ quá trình hoạt động.

Một số hình ảnh về phần cơ khí của scooter:



Hình 5.4 Khung xe khí chưa lắp bánh



Hình 5.5 Sàn xe



Hình 5.6 Gầm xe được đỡ bởi hai thanh inox vuông



Hình 5.7 Mỗi lắp bulông giữa khung xe và bánh xe



(a)



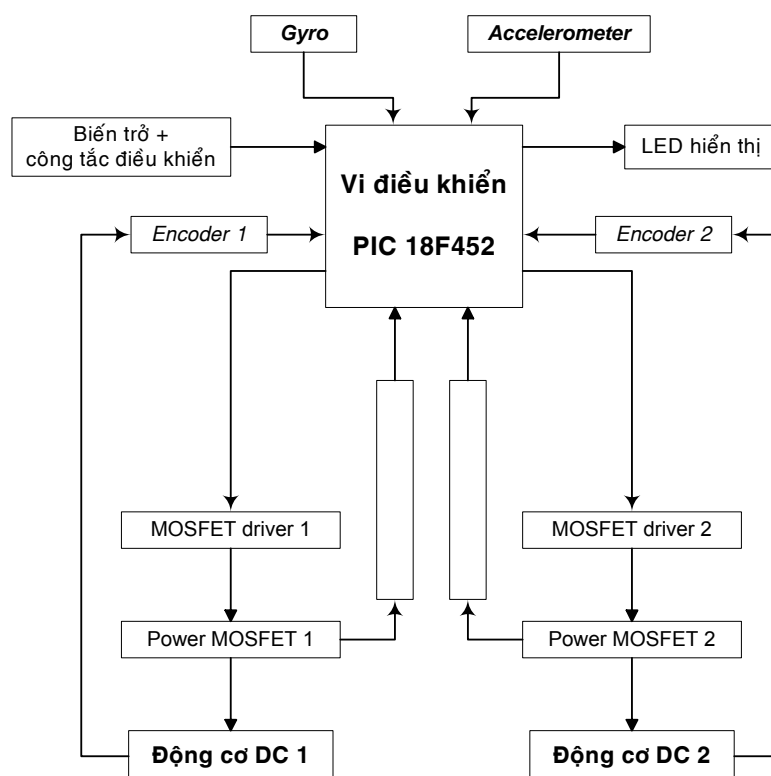
(b)

Hình 5.8 Phần cơ khí của xe hoàn chỉnh (a) và đã lắp mạch (b)

5.2 Mạch điện tử

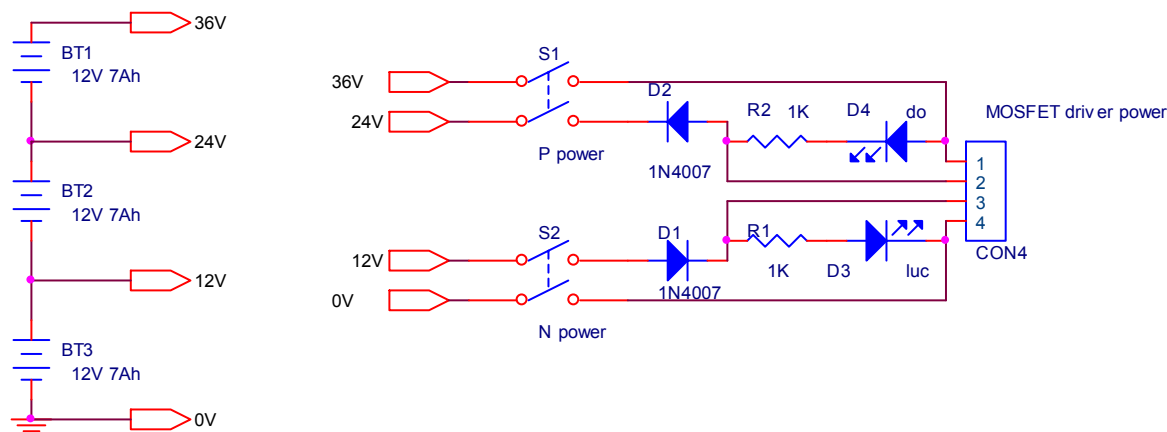
Hệ thống điện tử được xây dựng trên 4 bảng mạch in chính, nối với nhau bằng cáp điện và kết nối với những nguồn điện khác nhau để cách ly xung nhiễu, trong đó hai bảng mạch đóng vai trò là bộ lái công suất MOSFET của mỗi bánh xe với công suất cao 30A, điện áp từ 12 – 60V, một bảng mạch kết hợp lắp giữa cảm biến gia tốc góc và gia tốc đồng thời đo góc tĩnh với cảm biến vận tốc góc gyro và một bảng mạch điều khiển trung tâm và giao tiếp giữa máy tính qua cổng nối tiếp để hiệu chỉnh (calibre) các sensor. Ngoài ra còn có các PCB thành phần khác để đo vận tốc bánh xe, bảng điều khiển, đo quá dòng...

Sơ đồ khối làm việc của toàn bộ các thành phần điện – điện tử của mô hình có thể được mô tả theo hình dưới đây: tín hiệu từ hai cảm biến gyro và accelerometer được thu nhận vào vi điều khiển PIC18F452, được xử lý cùng với các giá trị về vận tốc và vị trí thu nhận được từ hai encoder của bánh xe để có được các lệnh điều khiển tốc độ bánh xe phù hợp. Các lệnh này được mã hoá thông qua hai mạch MOSFET driver, đưa tới bộ MOSFET công suất để cung cấp năng lượng phù hợp (PWM duty cycle) cho hai động cơ hoạt động. Trong lúc hoạt động, các cảm biến đo dòng quá tải luôn được kiểm tra và sẵn sàng để bắt vi điều khiển dừng các động cơ nếu có hiện tượng quá dòng xảy ra trên một trong hai bộ MOSFET công suất. Dòng điện tối đa được cài đặt được phép trong mạch MOSFET công suất là 30A.



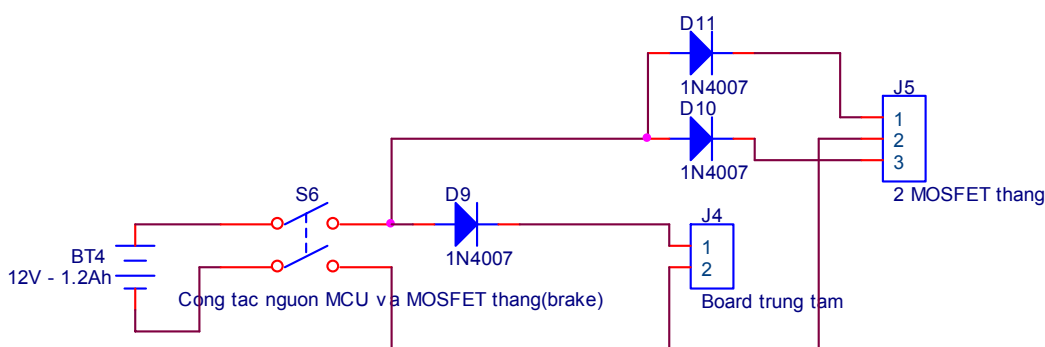
Hình 5.9 Sơ đồ khối của toàn bộ mô hình

5.2.1 Nguồn điện



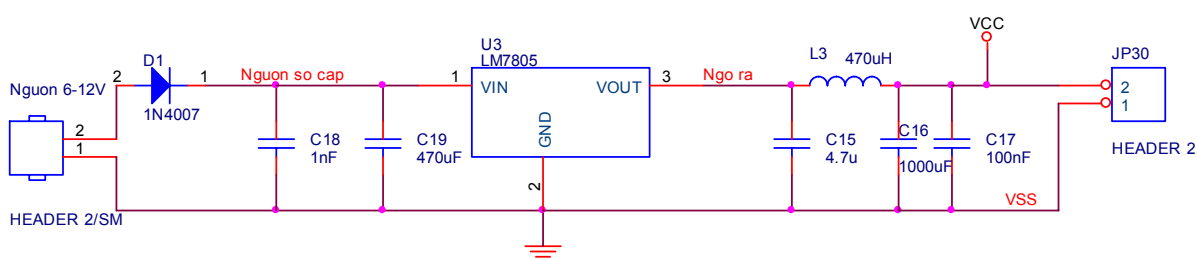
Hình 5.10 Sơ đồ nguồn điện cho động cơ và MOSFET driver

Bộ nguồn gồm 3 bình ắc quy 12V mắc nối tiếp nhau để có được một nguồn điện tương đương 36V cung cấp cho động cơ. Phần điện thấp 0-12V và 24-36V cũng được lấy thêm ra để cấp cho mạch MOSFET driver. Các diode trong đường cấp nguồn MOSFET driver dùng để tránh dòng điện truyền theo ngõ 0-24V cũng như 12-36V trong mạch.



Hình 5.11 Sơ đồ nguồn điện cho MOSFET driver thẳng và vi điều khiển

Bộ nguồn cho vi điều khiển và các MOSFET tải được tách rời so với bộ nguồn cung cấp cho động cơ để hạn chế nhiễu tín hiệu từ động cơ lên hoạt động của vi điều khiển và các cảm biến. Các diode được dùng để tránh shock xung điện về nguồn.



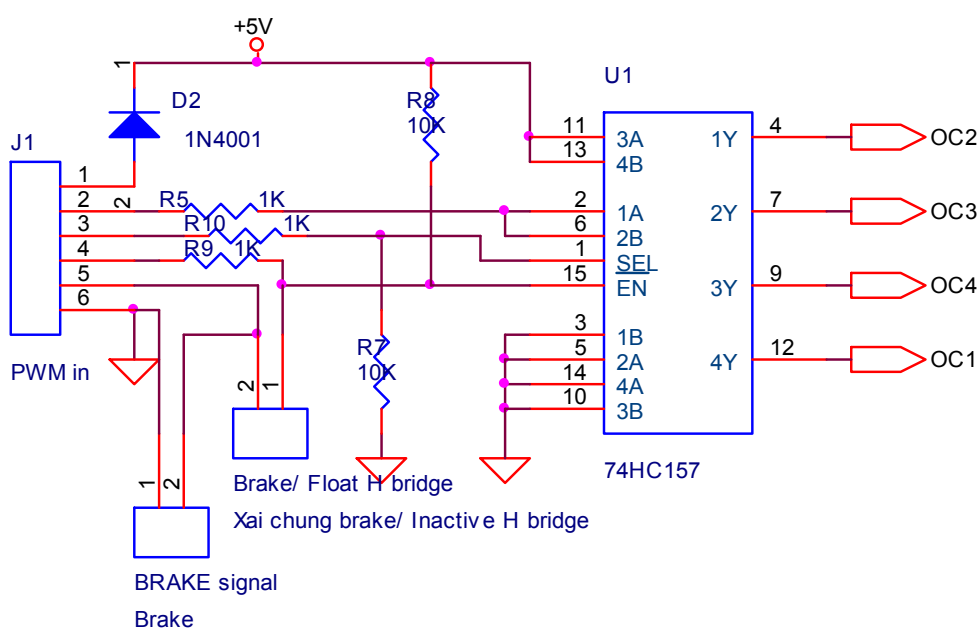
Hình 5.12 Sơ đồ nguồn điện 5V cho vi điều khiển

Sau khi nguồn điện 12V được phân phối đến mạch điều khiển trung tâm, điện áp 12V được qua một IC ổn áp LM7805 để chuyển thành điện áp 5V cung cấp cho vi điều khiển cũng như một số mạch cảm biến như hình vẽ ở trên.

5.2.2 Mạch công suất điều khiển động cơ

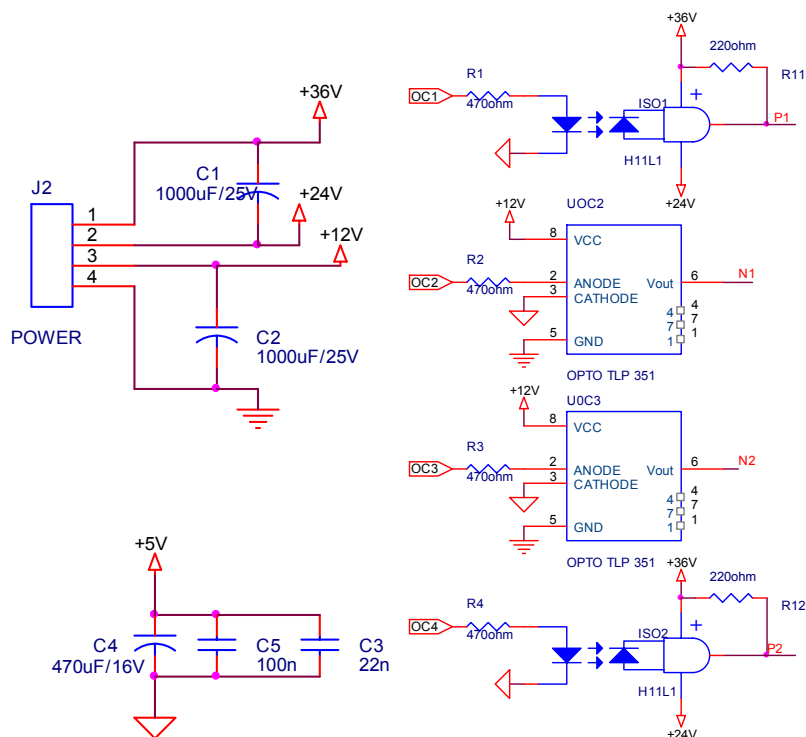
Mạch điều khiển động cơ có điện áp 36V, dòng điện hoạt động 5-10A, dòng điện khởi động tối đa 40A, điều khiển tốc độ bằng PWM với tần số 9.7665 kHz, độ phân giải 10bit/hướng. Bộ phận thắng để đảo chiều có tốc độ thực hiện 100ms.

5.2.2.1 Bộ đệm (MOSFET driver)



Hình 5.13 Bộ mã hóa tín hiệu PWM cho MOSFET driver

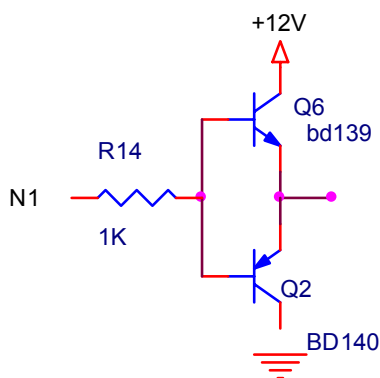
Tín hiệu từ vi điều khiển được chuyển tới IC logic trung gian như hình vẽ, nhằm mục đích tổ hợp các tín hiệu logic ngõ vào thành các tín hiệu logic ngõ ra phù hợp với hoạt động đóng mở các MOSFET driver và MOSFET công suất ở tầng kế. Tín hiệu ở tầng này được cách ly với các tín hiệu vào MOSFET driver bằng các opto quang.



Hình 5.14 Nguồn và bộ cách ly trong MOSFET driver

☞ Mạch Push-Pull:

Nguyên lý hoạt động của mạch:

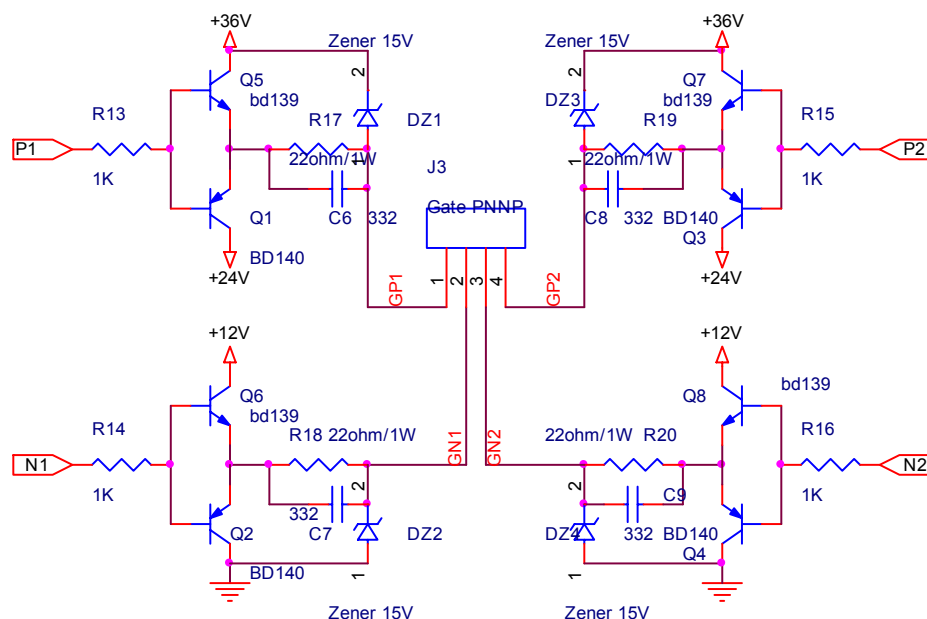


Hình 5.15 Sơ đồ mạch Push-Pull

Khi ngõ vào N1 ở mức cao, nếu ngõ ra được nối với một mạch điện tương đương một điện trở, Q6 sẽ dẫn, Q2 sẽ tắt, do đó ngõ ra của mạch Push-Pull sẽ ở mức điện áp cao.

Ngược lại, khi N1 ở mức thấp, Q6 sẽ ngưng dẫn, Q2 dẫn, kéo ngõ ra của mạch xuống mức thấp.

Tóm lại, mạch có giá trị logic và điện áp bằng với giá trị ngõ vào, nhưng qua hai transistor kênh NPN và PNP mắc bổ phụ như trên sẽ kéo dòng điện ở ngõ ra được khuếch đại lên nhiều lần (h_{FE}) so với ngõ vào, giúp MOSFET công suất đóng ngắt đủ nhanh.



Hình 5.16 Driver MOSFET công suất

5.2.2.2 MOSFET công suất – mắc bổ phụ

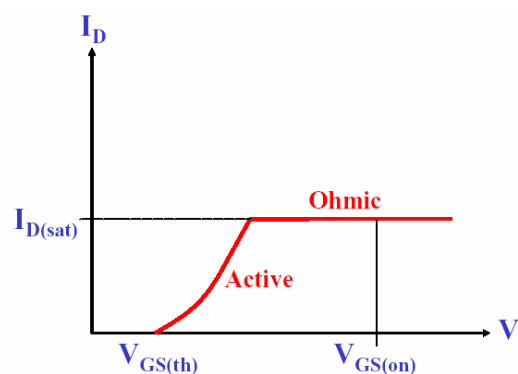
MOSFET có một lớp mỏng silicon dioxide, là lớp cách ly để ngăn ngừa cực G dương và cực âm chạm nhau. Lớp cách ly này mỏng có thể dùng để kiểm soát cực D. Và cũng vì lớp cách ly này quá mỏng nên dễ dàng bị phá hủy bởi nguồn điện G quá ngưỡng.

Cụ thể trong đề tài, IRF P140 có $V_{GS(max)} = \pm 20V$. Nếu nguồn điện G dương hơn +20V hay âm hơn -20V, lớp cách ly này sẽ bị phá hủy.

Ngoài điện áp quá ngưỡng V_{GS} , MOSFET cũng có thể bị phá hủy lớp cách ly mỏng này bằng nhiều cách khác nhau. Nếu MOSFET được lấy ra hay gắn vào vi mạch trong khi nguồn mở, điện áp thoáng qua gây ra bởi dẫn ngược (kickback) sẽ vượt quá giá trị V_{GS} . Kể cả khi vận chuyển MOSFET cũng tạo tĩnh điện đủ vượt quá giá trị V_{GS} . Vì vậy, khi vận chuyển MOSFET, người ta phải gắn vào nó một sợi dây chì hay bọc nó lại trong miếng foil thiếc hay đựng nó trong miếng foam dẫn.

Một vài MOSFET được bảo vệ bằng cách lắp diode zener song song với cực G và nguồn. Điện áp zener thấp hơn giá trị V_{GS} . Do đó, diode zener dẫn trước khi đạt giá trị V_{GS} mà có thể gây nguy hiểm cho lớp cách ly mỏng. Bất lợi của những diode zener bên trong là chúng làm giảm tổng trở ngõ vào của MOSFET. Vì những MOSFET đắt tiền dễ dàng bị hỏng nếu không được zener bảo vệ nên các làm này được đánh giá cao trong ứng dụng. Do vậy, những thiết bị MOSFET rất tinh tế nhưng lại dễ dàng bị hư hỏng.

Hầu hết các ứng dụng của MOSFET ở



Hình 5.17 Vùng hỗ dẫn (transconductance) của MOSFET

vùng ohmic. Khi phân cực ở vùng ohmic, giá trị E-MOSFET bằng với điện trở. Khi phân cực ở vùng active, giá trị của nó lại bằng với nguồn điện. Mặc dù, E-MOSFET có thể hoạt động trong vùng active nhưng ứng dụng chính của nó là trong vùng ohmic.

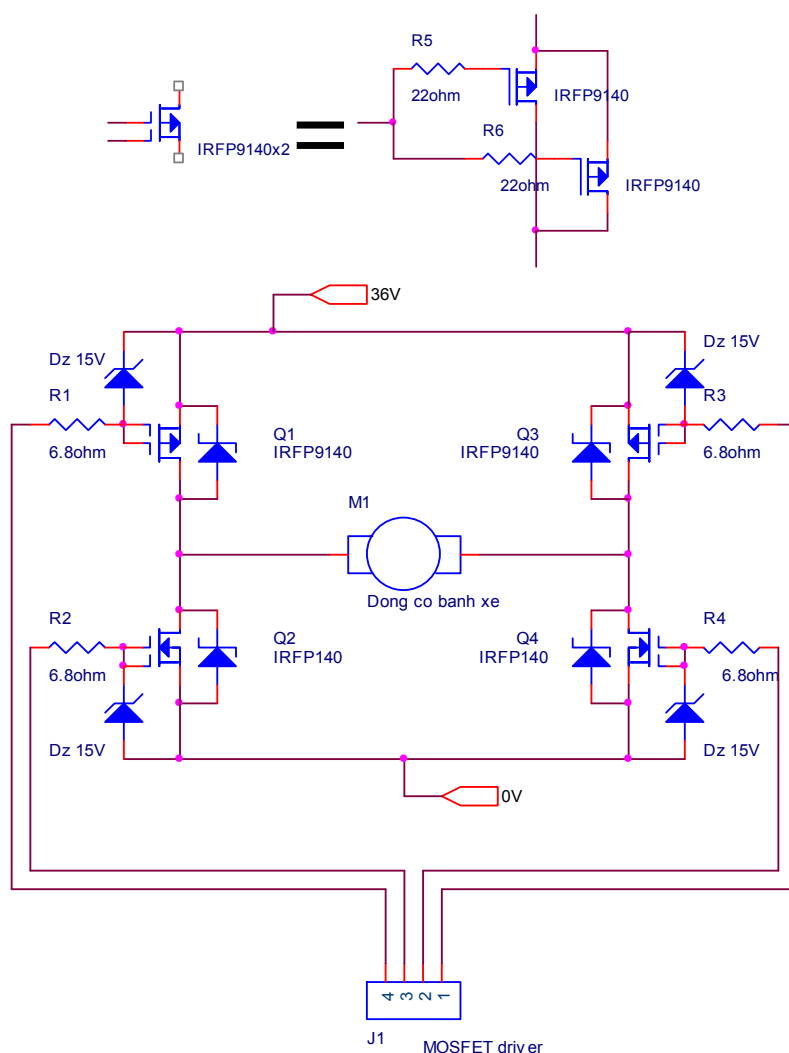
MOSFET công suất gồm: mỗi động cơ được điều khiển bởi hai cặp MOSFET kênh P, và hai cặp MOSFET kênh N, với mỗi đầu được mắc song song. Bộ nguồn cho MOSFET driver được cấp gồm 4 tín hiệu vào: 0 – 12 – 24 – 36 V.

Tín hiệu giữa mạch lái MOSFET với MOSFET công suất trước đó có điện trở shunt mắc nối tiếp với dòng nguồn 36V để đo dòng điện.

Mạch cảm biến luôn được cấp nguồn 5V cho cảm biến gia tốc và cảm biến vận tốc góc, tín hiệu lấy ra bao gồm tín hiệu analog và tín hiệu xung từ cảm biến gia tốc tại mạch cảm biến, tín hiệu của vận tốc góc được dịch xuống 2.5 thành 2.3V. Tín hiệu A/D tại giá trị cân bằng là 470.

Mạch cầu H:

- Dạng biểu đồ.



Hình 5.18 Sơ đồ mạch cầu H của MOSFET công suất

- Hoạt động.
- Giải mã.
- Bộ phận lái sườn cao.

Các nhu cầu của MOSFET driver.

C_{gs} gây ra $\sim C_{iss}$ cho IRFP140 là 1400pF, phải nạp cho $V_{th} \Rightarrow ON$.

ΔQ_g tổng số nạp cho $G \Rightarrow ON$ cho IRFP140 là 97nC.

t_f : thời gian fall của V_{DS} (Δt) cho IRFP140 là 20ns.

$\Delta Q_g < 97nC$ cho IRFP140.

$\Delta t \sim 50ns$ cho IRFP140.

$$i(t) = \frac{dq}{dt} \approx \frac{\Delta Q}{\Delta t} \quad [5-1]$$

$$i(t) = \frac{97nC}{50ns} = 1.94 \text{ cho } 50nsecond.$$

$$\text{slew rate} = \frac{dv}{dt} \approx \frac{\Delta V_{gs}}{\Delta t_{fall}} \quad [5-2]$$

$$\Delta V_{gs \max} \approx 12V$$

$$\text{slew rate} = \frac{12V}{50ns} = 240 \text{ V}/\mu s$$

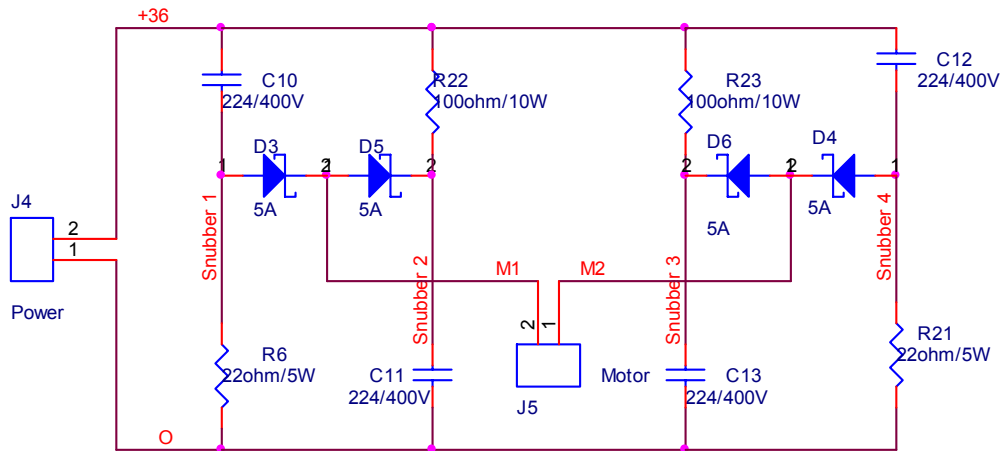
Bộ phận lái sườn thấp mạch Push-Pull BD139-BD140.

- Sườn thấp kênh N.
- V_{dd} 12V.
- Không đảo ngược 0V \sim +12V.
- 25mV bão hòa.
- I_{out} 3A
- $f_{max} < 6MHz$.

Bộ phận lái sườn cao mạch Push-Pull BD139-BD140.

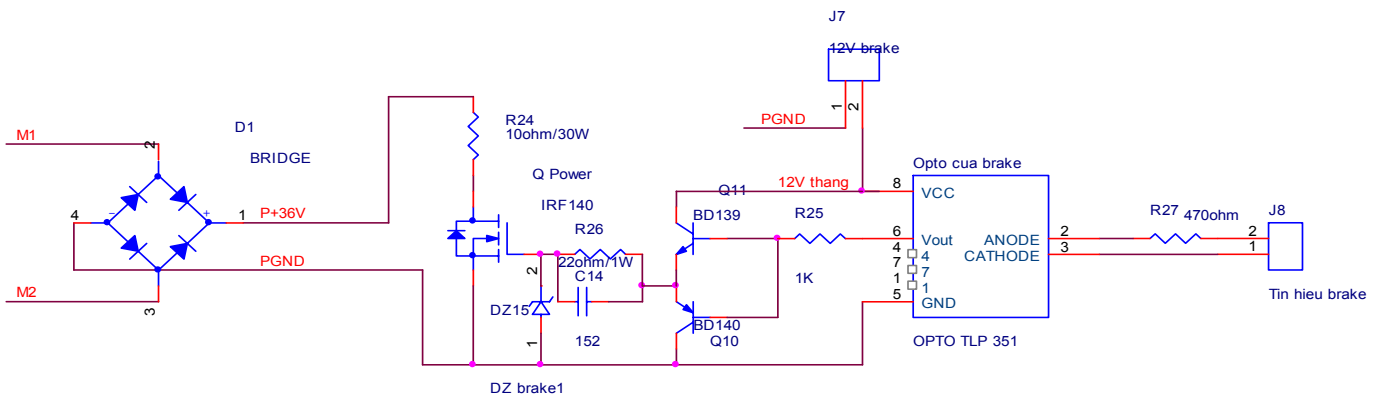
- Sườn cao kênh P.
- $V_{gate} \sim V_{source} + 24V \sim +36V$
- $< 24V$.
- I_{out} 3A
- $f_{max} < 6MHz$.

5.2.2.3 Mạch Snubber



Hình 5.19 Sơ đồ mạch Snubber

5.2.2.4 MOSFET thắng



Hình 5.20 Sơ đồ MOSFET thắng

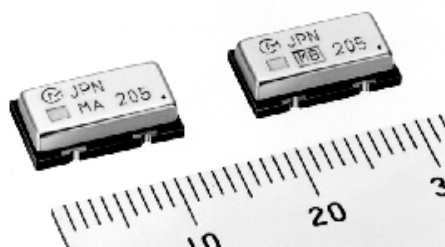
5.2.3 Cảm biến

Theo sơ đồ khối được giới thiệu ở đầu mục 5.2, các tín hiệu vào vi điều khiển PIC 18F452 gồm: tín hiệu tương tự và xung từ bảng điều khiển, tín hiệu đo góc và vận tốc góc của sàn xe so với chiều vuông góc mặt đất, đo quá tải dòng điện qua động cơ và đo vận tốc cũng như vị trí bánh xe. Nhưng khi hệ là một hệ kín điều khiển tự động, chỉ có các thành phần sau đóng vai trò tín hiệu tham khảo điều khiển, gồm các cảm biến đo góc nghiêng: gyro và accelerometer; cảm biến đo vận tốc bánh xe: incremental encoder; và cảm biến đo dòng điện: mạch điện trở shunt nối tiếp với động cơ.

Cảm biến đo góc nghiêng sử dụng hai dạng là accelerometer để đo góc nghiêng tĩnh và gyro dùng để đo góc nghiêng động. Ở đề tài này, encoder, resolver

hay một biến trở không thể dùng để đo góc với trọng lực vì những cảm biến này chỉ có thể đo góc tương đối giữa hai bộ phận mà khớp quay giữa chúng chính là các cảm biến. Chỉ có các cảm biến đo không tiếp xúc cho một giá trị tuyệt đối về góc với vector trọng lực mới khả thi để dùng đo góc nghiêng trong đề tài. Gyro được dùng để đo vận tốc góc nghiêng, và gia tốc kế accelerometer dùng để đo góc tĩnh. Gia tốc kế đưa ra giá trị góc tĩnh rất chính xác. Tuy nhiên, một mình nó không thể cung cấp cho chúng ta khả năng đáp ứng đủ nhanh vì bộ lọc 10Hz được đặt ở ngõ ra để loại bỏ độ nhiễu cơ học, đồng thời rất dễ bị nhiễu vì bản thân là một gia tốc kế. Gyro được sử dụng để bổ sung đo góc với vận tốc nhanh vì nó có thời gian đáp ứng rất nhanh, điều này đền bù cho sự đáp ứng chậm và nhiễu khi đo nhanh của gia tốc kế.

5.2.3.1 Thiết bị đo góc gyro Murata ENC-03^[31]

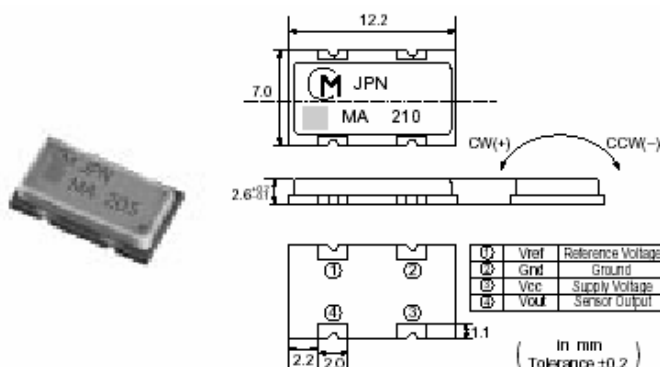


Hình 5.21 Cảm biến gyro
Murata ENC-03

Gyro Murata ENC-03 là cảm biến vận tốc góc dùng hiện tượng của lực Coriolis, được tạo ra khi vận tốc góc quay được áp trên bộ rung.

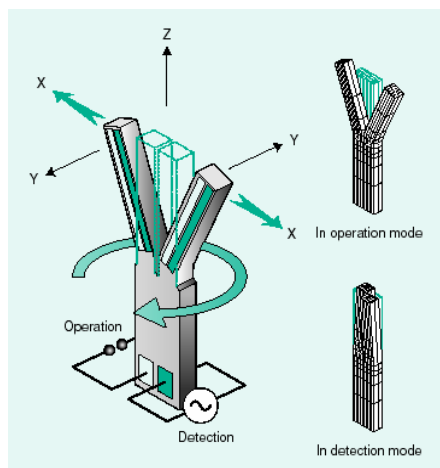
Đối với gyro, Murata ENC-03 có ngõ ra analog được chọn. Đó là một con quay hồi chuyển tuyệt vời, có các tính năng sau:

- Khả năng loại bỏ độ rung cao.
- Vận tốc góc cao $\pm 300^\circ/\text{s}$.
- Độ nhạy cao $0.67\text{mV}/\text{deg}/\text{s}$.
- Được cài đặt sẵn tín hiệu $V_{\text{ref}} = 1.35\text{V}$.



Hình 5.22 Sơ đồ chân và kích thước của cảm biến gyro Murata ENC-03

Vì hệ thống phát ra độ nhiễu cơ học, nên Murata ENC-03 được sử dụng với mục đích loại bỏ độ rung cao. Một tính năng khác là cài đặt sẵn tín hiệu V_{ref} . Điều này có nghĩa là Murata ENC-03 có thể dễ dàng ghép tương thích với một vi điều khiển. Tuy nhiên, gyro Murata ENC-03 phát ra độ nhiễu ở ngõ ra. Do vậy, bộ lọc thông thấp 1 kHz được thêm vào giữa gyro và vi điều khiển để loại bỏ độ nhiễu này.

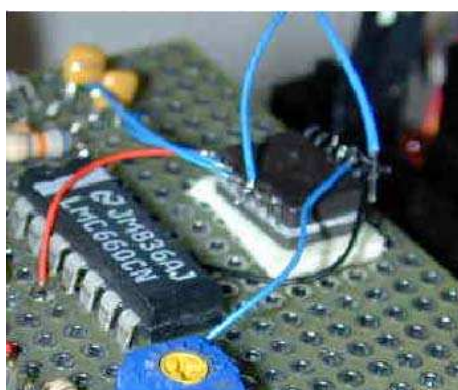


Hình 5.23 Hoạt động chính của cảm biến gyro áp điện

Hình 5.23 minh họa nguyên lý hoạt động của cảm biến gyro áp điện. Khi nhân tố piezoelectric thay đổi hình dạng thành mẫu làm đổi hướng, quay quanh trục Z trong khi chuyển động qua hướng trục Z, dao động xảy ra qua hướng X do lực Coriolis. Bằng cách dò tìm dao động điện hướng X, có thể kiểm tra giá trị của vận tốc góc.

5.2.3.2 ADXL202A^[30]

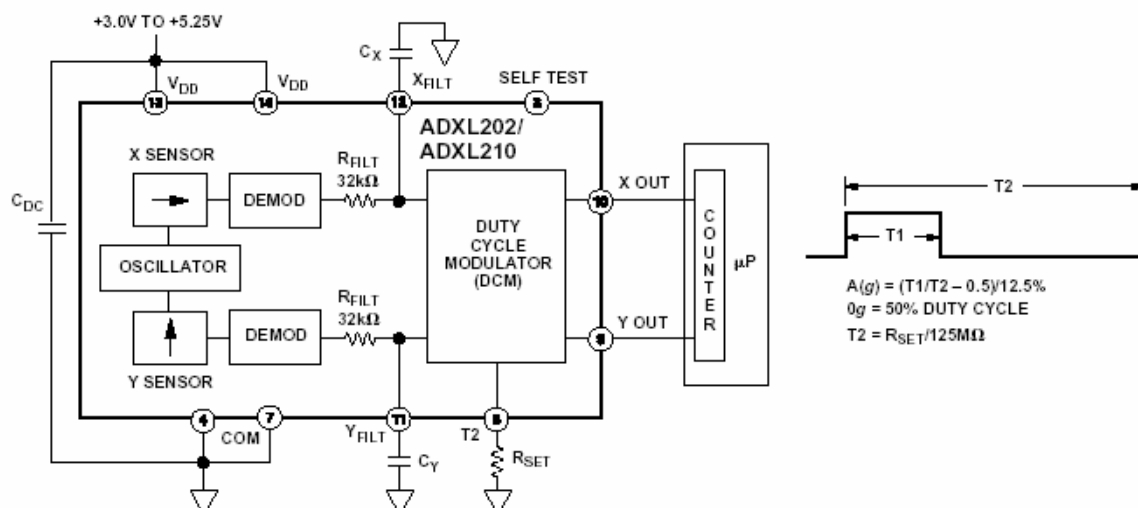
❖ Giới thiệu:



Hình 5.24 Hình chụp ADXL202^[21]

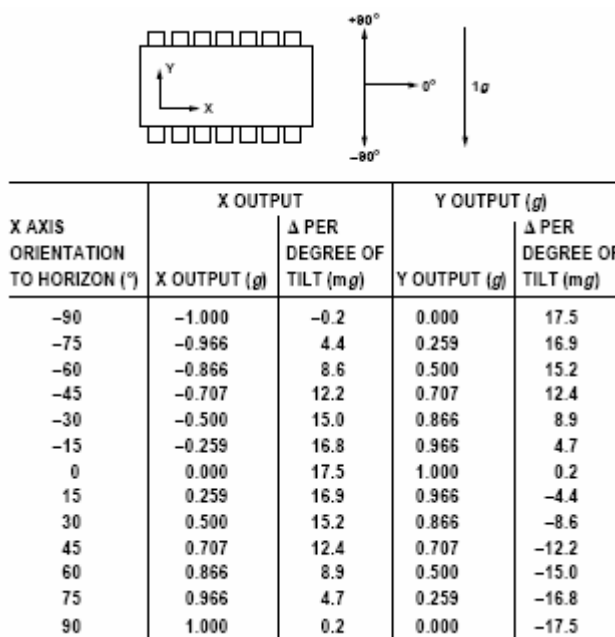
Những ưu điểm kỹ thuật của cảm biến gia tốc ngày nay, đặc biệt với những loại vi cơ silicon, làm giá thành của sản phẩm giảm đáng kể. Giờ đây, gia tốc kế có giá thấp hơn 5 USD cho mỗi trục có trong cảm biến. Do gia tốc hay một trong các đặc tính vật lý như dao động, va chạm hay độ nghiêng góc trở nên rất phổ biến trong một loạt các sản phẩm.

Có rất nhiều dạng cảm biến dùng đo lường gia tốc, dao động, va chạm hay nghiêng góc. Những cảm biến này bao gồm lớp màng áp điện (piezo-film), servo cơ điện, áp điện, góc chất lỏng, điện trở vi cơ áp điện (piezo micromachined) và cảm biến điện dung, cũng như là cảm biến vi cơ điện dung bề mặt. Mỗi cảm biến này có đặc tính khác biệt nhau về các tín hiệu ngõ ra, chi phí sản xuất, và môi trường hoạt động. Cảm biến có khả năng đo lường gia tốc có thể cho giá trị vận tốc bằng tích phân đơn và giá trị vị trí bằng tích phân bậc hai. Khả năng dao động hay va chạm có thể được dùng cho việc kiểm tra độ chịu lực của máy cũng như kiểm tra sự chuyển động và va chạm để cấp báo cho xe. Gia tốc tĩnh đo lực hấp dẫn có thể dùng để kiểm tra góc và độ nghiêng, miễn là cảm biến đáp ứng nhanh với gia tốc tĩnh.



Hình 5.25 Sơ đồ nguyên lý của cảm biến gia tốc ADXL202A

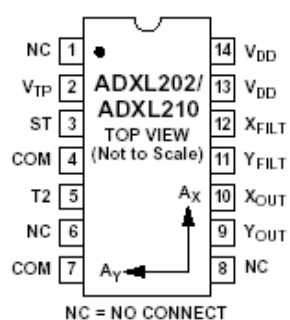
Đối với cảm biến gia tốc ADXL202A, tín hiệu góc được lấy từ giá trị xung PWM hay tín hiệu analog, trong đề tài này, tín hiệu từ ADXL202A được lấy ra từ ngõ analog.



Hình 5.26 Trục X và Y đáp ứng với những thay đổi khác nhau của góc nghiêng

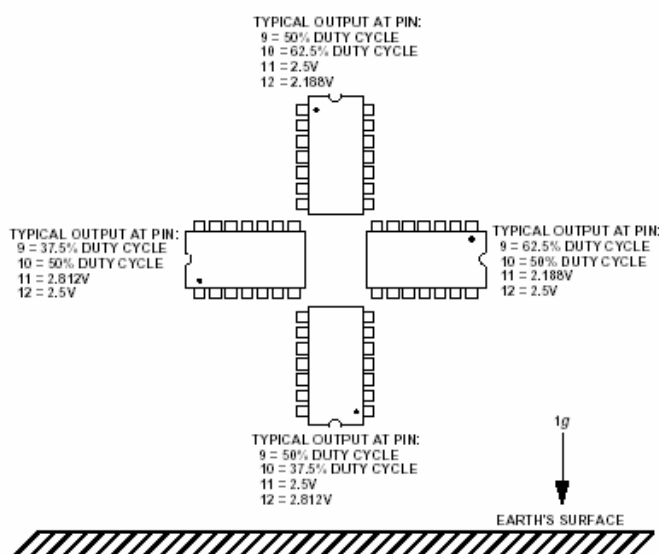
Độ nhạy của ADXL202A tạo ra khả năng thay đổi ngõ ra rất lớn trên mỗi g. Một g của khả năng thay đổi này là bằng với góc nghiêng 90°. Khả năng thay đổi ngõ ra lớn này làm ADXL202A dễ dàng đo góc sử dụng module A/D của vi điều khiển. Vì gia tốc kế rất nhạy đối với độ nhiễu cơ học, nên bộ lọc 50Hz được đặt ở ngõ ra.

Mô tả chức năng chân:



Hình 5.27 Sơ đồ chân của cảm biến ADXL202

- 1,6,8 : không dùng
- 2 : V_{TP} kiểm tra, không dùng
- 3 : ST tự kiểm tra
- 4,7 : V_{ss} masse
- 5 : R_{set} xác định tần số PWM
- 9,10 : Y,X duty Cycle output
- 11,12 : tụ lọc ngõ tín hiệu Y,X
- 13,14 : V_{DD}

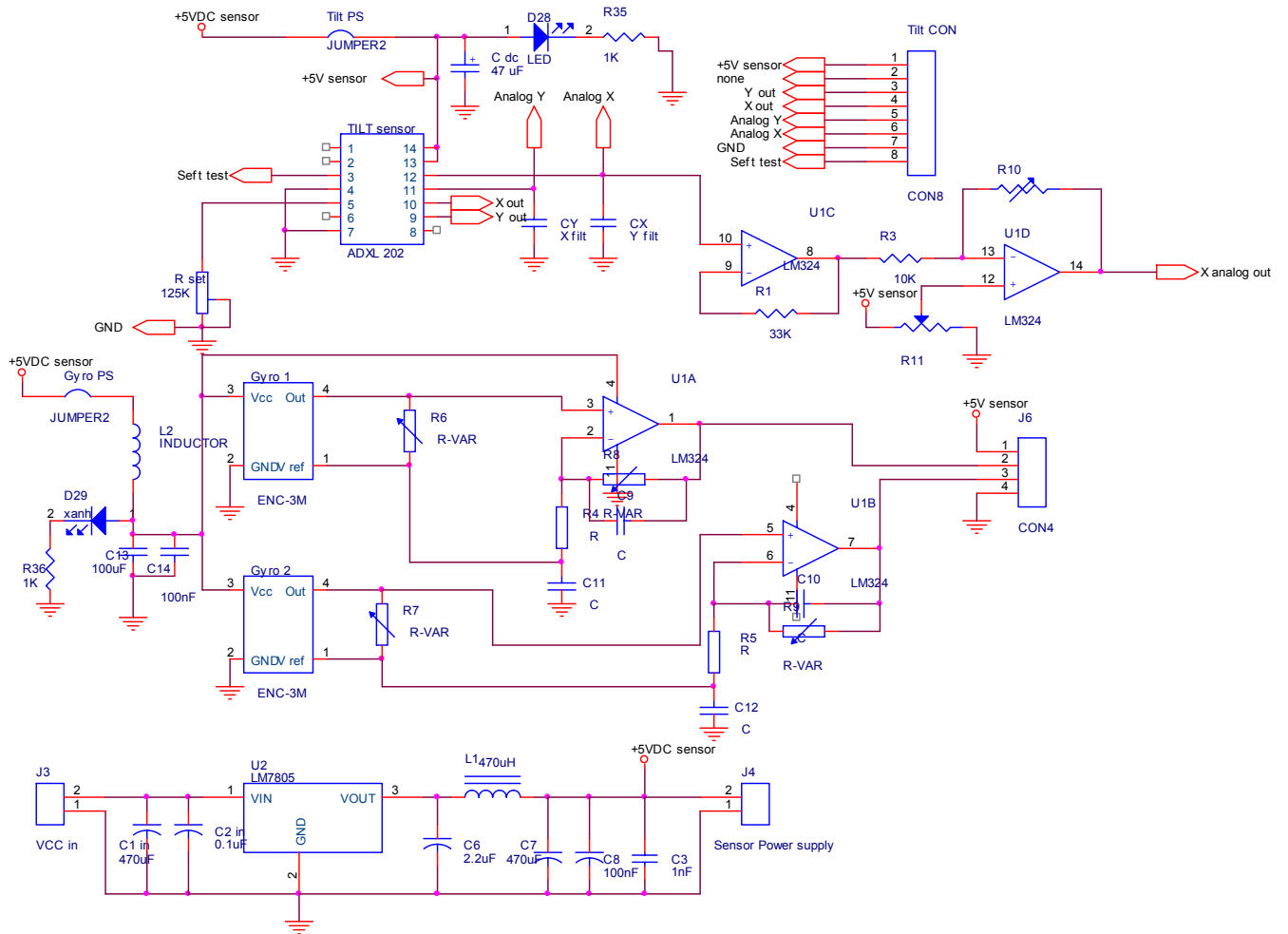


Hình 5.28 Khả năng đáp ứng của ADXL202 với trọng lực

❖ Những tính năng của ADXL202A:

- Độ nhạy cao (312mV/g).
- Điện áp tại mức 0g là cố định (2.5V đối với nguồn cung cấp 5V).
- Cảm biến gia tốc 2 trục trên chip IC đơn.
- Đo gia tốc tĩnh cũng như là gia tốc động lực.
- Ngõ ra PWM được điều chỉnh tần số theo người sử dụng bằng R_{SET} .
- Công suất thấp < 0,6mA.
- Khả năng đáp ứng nhanh hơn những cảm biến điện phân, thủy phân hay góc nghiêng nhiệt.
- Khả năng điều chỉnh bằng thông rộng với tụ điện đơn trên mỗi trục.
- Độ phân giải 5mg tại băng thông rộng 60Hz.
- Hoạt động cấp đơn +3V đến +5.25V.
- Chịu được shock 1000 g.

Sơ đồ mạch điện kết hợp cảm biến gyro Murata ENC-03 và cảm biến accelerometer:



Hình 5.29 Sơ đồ mạch điện cảm biến

Các công thức tính giá trị khuếch đại điện áp ngõ ra trên cảm biến gyro Murata ENC-03 và accelerometer bằng OPAMP LM324:

➤ Gyro Murata ENC-03:

$$\frac{V_o - V_g}{R_8} = \frac{V_g - V_{ref}}{R_4}$$

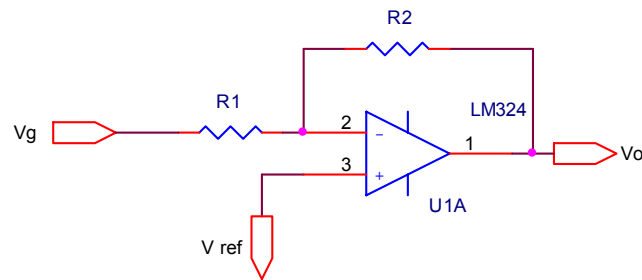
$$V_o = V_g + \frac{R_8}{R_4}(V_g - V_{ref})$$

$$V_o = V_g + \frac{R_8}{R_4}(V_g - V_{ref})$$

[5-3]

➤ Accelerometer:

$$\frac{V_o - V_-}{R_2} = \frac{V_- - V_g}{R_1}$$



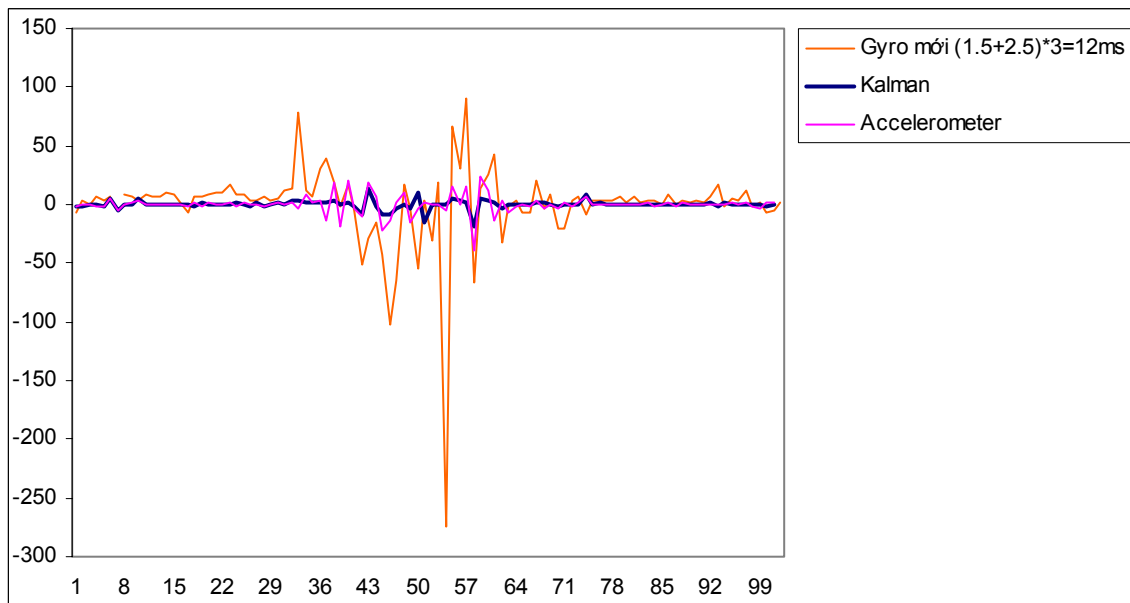
Hình 5.30 Bộ khuếch đại điện áp cho ngõ ra accelerometer

$$\frac{V_o - V_{ref}}{R_2} = \frac{V_{ref} - V_{acc}}{R_1}$$

$$V_o = V_{ref} + \frac{V_{ref} - V_g}{1} \frac{R_2}{R_1}$$

$$V_o = V_{ref} + (V_{ref} - V_g) \frac{R_2}{R_1} \quad [5-4]$$

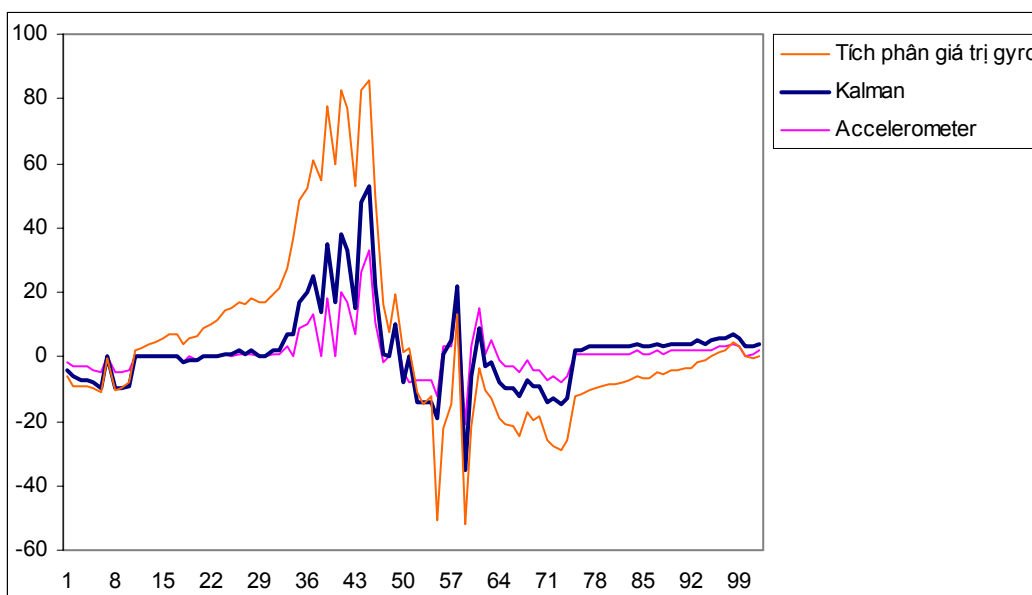
Mạch điện tử thực hiện ở phần trên được tiến hành chạy thử, calibre giá trị và áp bộ lọc Kalman vào cho hai tín hiệu từ accelerometer và gyro (chương 3, bộ lọc Kalman), kết quả được thu nhận về máy tính để kiểm tra qua giao tiếp RS-232 như sau:



Hình 5.31 So sánh giá trị vận tốc góc trên gyro, đạo hàm accelerometer và vận tốc góc sau bộ lọc Kalman

Tín hiệu trên gyro cực kỳ nhạy với các chuyển động dù là nhỏ nhất, trong khi accelerometer tỏ ra kém nhạy với các chuyển động quay tròn. Chính vì quá nhạy,

những rung động nhỏ cũng làm gyro thay đổi giá trị. Qua bộ lọc Kalman, tín hiệu vận tốc góc được chỉnh lại cho phù hợp và loại bỏ đi những xung động nhiễu của gyro và accelerometer.



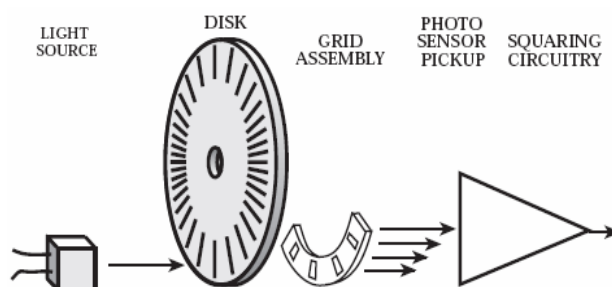
Hình 5.32 So sánh giá trị góc trên gyro (tích phân vận tốc góc), accelerometer và sau bộ lọc Kalman

Tín hiệu đo góc tĩnh trên accelerometer là rất tốt, nếu accelerometer không chịu một gia tốc tịnh tiến theo chiều nhạy nhận dạng. Tín hiệu góc thu được từ gyro (tích phân vận tốc góc thu được từ gyro) có dạng giống accelerometer, nhưng bị trôi theo thời gian tại những thời điểm không có chuyển động quay. Kết quả góc thu được sau khi qua bộ lọc Kalman đã tỏ ra rất tốt để khắc phục nhược điểm của hai cảm biến trên.

5.2.3.3 Cảm biến đo vị trí- encoder

Encoder được dùng để đo vận tốc và vị trí của bánh xe, là một trong các tham số điều khiển mô hình. Đĩa cảm biến được làm bằng mica, ghép với một tấm film trong đã in sẵn các vạch đen trắng xen kẽ.

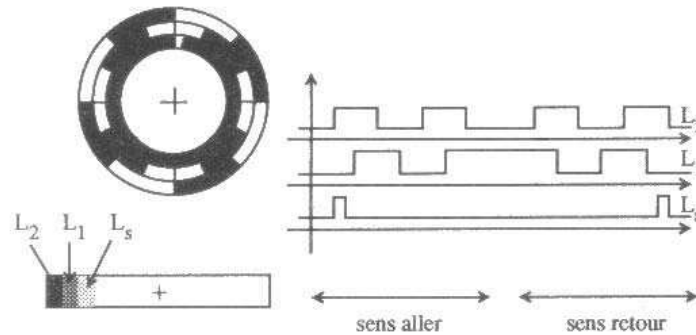
❖ Incremental encoder



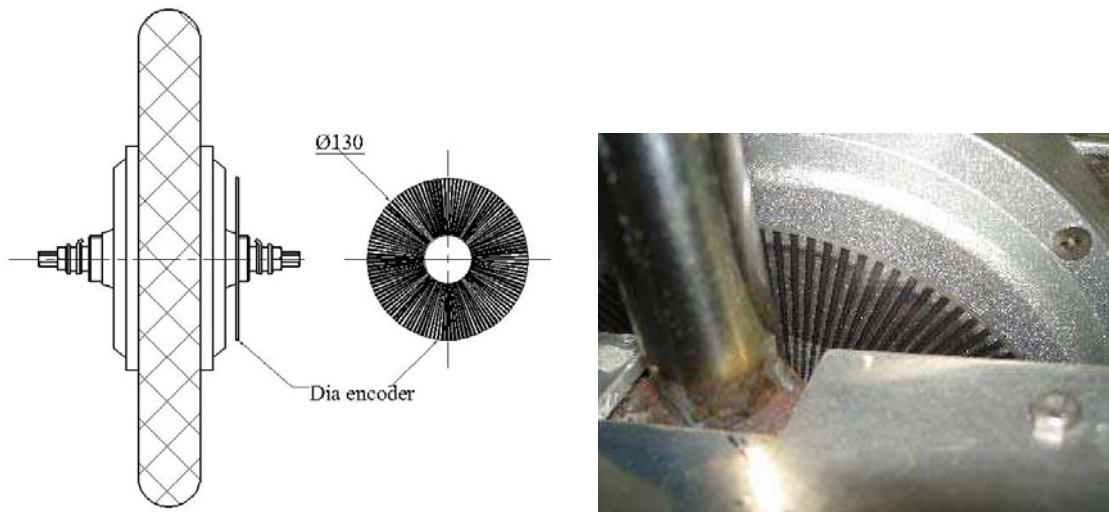
Hình 5.33 Sơ đồ nguyên lý hoạt động của incremental encoder

Cấu tạo gồm một đĩa đục có những rãnh trong suốt, một nguồn phát quang thông qua các rãnh trên đĩa đến một đầu thu. Tín hiệu ra có thể dạng xung vuông

hoặc hình sin. Việc đọc các tín hiệu ra này sẽ xác định được vị trí cũng như vận tốc của động cơ. Lưu ý rằng khi sử dụng loại cảm biến này có ba thông số quan trọng, đó là: số vạch chia, dạng tín hiệu ra và số kênh ra.



Hình 5.34 Sơ đồ tín hiệu ra của incremental encoder



Hình 5.35 Dĩa encoder

Bánh xe bằng cao su, có đường kính 40cm như hình vẽ, trên bánh xe có gắn một mâm nhôm, và trên mâm nhôm có 120 khe để đặt một bộ cảm biến quang để đếm số xung, từ đó ta có thể xác định khoảng cách mà robot đó đi được.

Tính toán khoảng cách di chuyển theo số xung đếm được trên bộ encoder.

Chu vi bánh xe:

$$P = 2\pi R = \pi d = 1,2 \text{ (cm)} \quad [5-5]$$

Số xung trên mâm encoder: 120

Góc mở trên một xung:

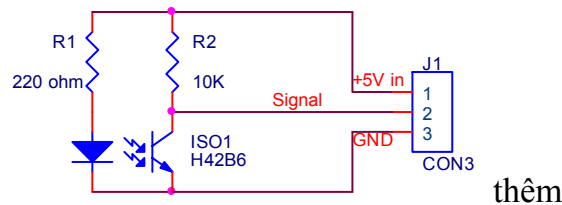
$$\alpha = \frac{2\pi}{120} = \frac{\pi}{60} \text{ (rad)} \quad [5-6]$$

Quãng đường đi được trên một xung:

$$S = \alpha R = \frac{\pi}{60} \times 20 = \frac{\pi}{3} = 1(\text{cm}) \quad [5-7]$$

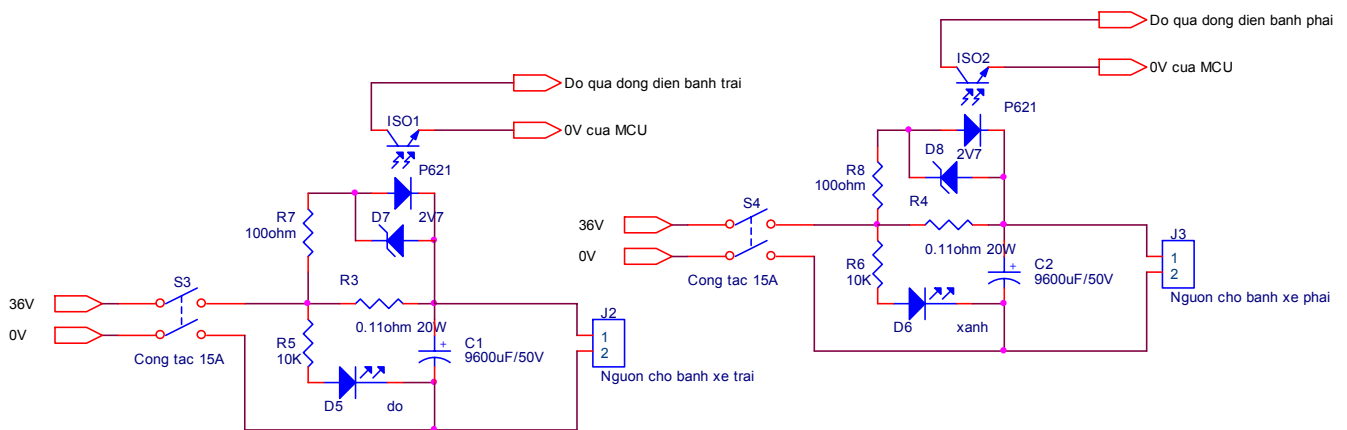
Nếu cần di chuyển một quãng đường S thì số xung di chuyển cần phải có:

$$n = \frac{S}{s} (\text{xung}) \quad [5-8]$$



Hình 5.36 Mạch đếm encoder giao tiếp tới vi điều khiển

5.2.3.4 Cảm biến đo dòng hồi tiếp (Điện trở shunt)



Hình 5.37 Sơ đồ mạch cảm biến đo dòng hồi tiếp và công tắc nguồn cho động cơ

Bộ đo dòng được thiết kế nhằm tham khảo thời điểm dòng qua động cơ bị quá tải. Giá trị dòng điện tối thiểu qua động cơ để opto bắt đầu dẫn được tính như sau:

$$V_{optoON} = 1V = V_{shunt} - V_R \quad [5-9]$$

$$V_{optoON} = V_{shunt} - I \cdot R \quad [5-10]$$

$$R_{shunt} \cdot I_{motor} = V_{shunt} = V_{optoON} + I \cdot R \quad [5-11]$$

$$I_{motor} = \frac{V_{optoON} + I \cdot R}{R_{shunt}} = \frac{1 + 0.02 \cdot 100}{0.1} = 30A \quad [5-12]$$

Nguồn điện 36V từ bộ nguồn ắc quy ghép được đưa qua công tắc, tới mạch đo quá tải dòng điện bằng một điện trở shunt mắc nối tiếp tại cực dương nguồn. Nếu bị quá tải, tín hiệu quá tải thông qua bộ cách ly opto sẽ được đưa đến vi điều khiển để tắt toàn bộ hoạt động của động cơ. Tụ điện 9600uF dùng để bù điện tích khi động cơ tăng tốc đột ngột.

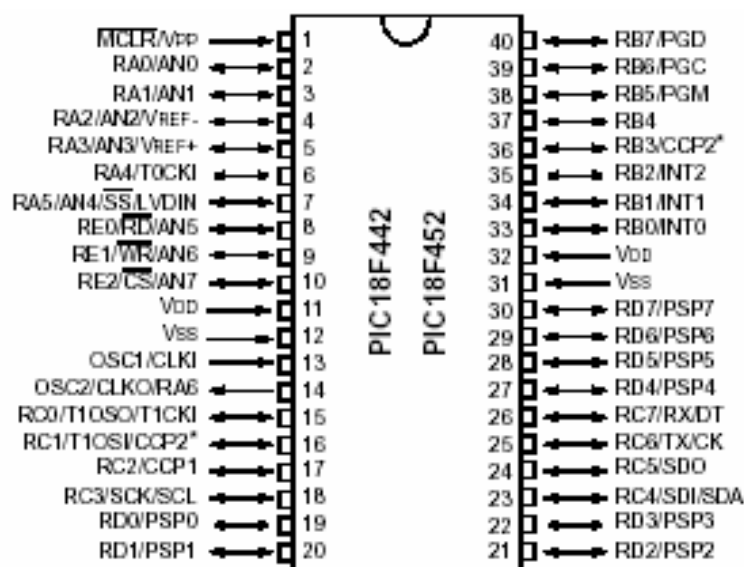
5.2.4 Bộ xử lý trung tâm - vi điều khiển PIC 18F452

Mạch điều khiển động cơ là PIC18F452 giao tiếp từ mạch điều khiển cân bằng tới những chip điều khiển H-bridge. Bằng cách gọi các bit chỉnh hướng và tín hiệu PWM đến chip điều khiển H-bridge, PIC kiểm soát động cơ điều khiển tốc độ và hướng của động cơ.

Vi điều khiển PIC 18F452 có dạng chip 40 chân. Chúng được phân biệt với họ 16Fxxx qua bốn đặc điểm:

1. Những thiết bị PIC18FX52 có bộ nhớ chương trình cực nhanh gấp hai lần và dữ liệu RAM của những thiết bị PIC18FXX2 lần lượt là 32Kb và 1536 byte, 16Kb và 768 byte.
2. Những thiết bị cung cấp kênh 8 A/D 10 bit.
3. Những thiết bị PIC18F4X2 cung cấp 5 port I/O.
4. Chỉ những thiết bị PIC18F4X2 cung cấp module CCP nâng cao, I2C, USART, bộ biến đổi A/D và cổng slave song song.

Tất cả tính năng của các thiết bị trong họ PIC18FXX2, bao gồm những module thông tin chuỗi giống hệt nhau.



Hình 5.38 Sơ đồ chân 18F452 [22]

5.2.4.1 Các khả năng của vi điều khiển Microchip PIC 18F452:

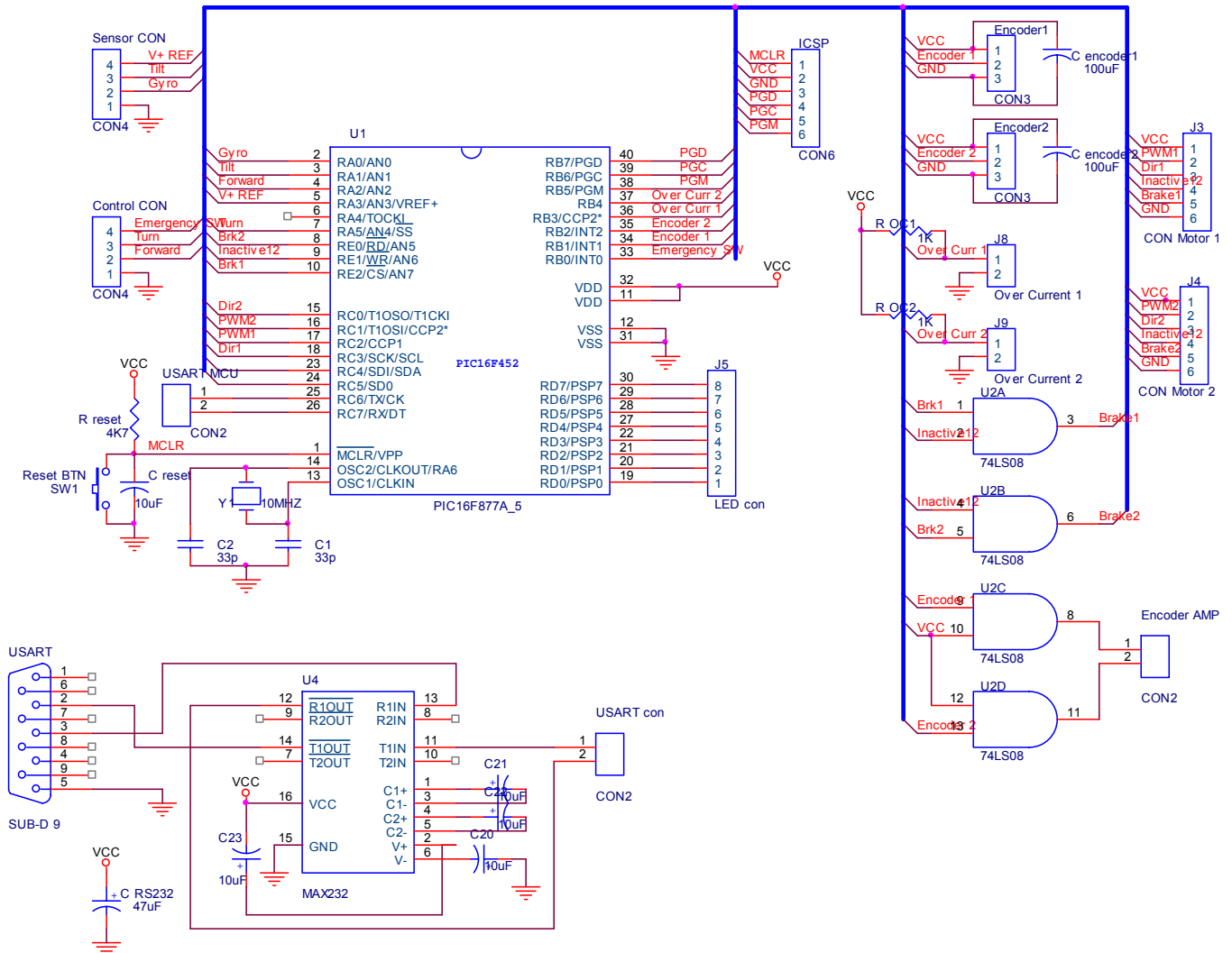
❖ Sự thực thi RISC CPU cao

- Cấu trúc/ lệnh tối ưu chương trình biên dịch C:
 - Mã nguồn tương thích với tập lệnh PIC16 và PIC17.
- Bộ nhớ chương trình tuyến tính chiếm 32kb.
- Bộ nhớ dữ liệu tuyến tính chiếm 1,5kb.

- Nhiệm vụ hoạt động 10MIPs:
 - DC – 40 MHz osc./ ngõ vào clock.
 - 4 MHz – 10 MHz osc./ngõ vào clock với PLL active.
- Lệnh dài 16-bit, đường dẫn dữ liệu dài 8-bit.
- Những mức ưu tiên để ngắt.
- Số nhân 8x8bit phần cứng chu kỳ đơn.
- ❖ Tính năng ngoại vi cao
- Sink/ dòng điện cao 25mA/ 25mA.
- Ba chân ngắt ngoại vi.
- Module Timer0: Bộ đếm/ bộ ghi 8-bit/16-bit với prescaler chương trình.
- Module Timer1: Bộ đếm/ bộ ghi 16-bit.
- Module Timer2: Bộ đếm/ bộ ghi 8-bit với thanh ghi đoạn 8-bit (thời gian dựa vào PWM).
- Module Timer3: Bộ đếm/ bộ ghi 16-bit.
- Tùy chọn clock oscillator thứ hai – Timer1/Timer3.
- Hai module Capture/Compare/PWM (CCP). Những chân CCP có thể xem là:
 - Ngõ vào capture: capture là 16-bit, độ phân giải cao nhất 6,25 ns ($T_{CY}/16$).
 - Compare là 16-bit, độ phân giải cao nhất 100 ns (T_{CY}).
 - Ngõ ra PWM: độ phân giải PWM là -1 đến 10-bit, tần số PWM lớn nhất @: Độ phân giải 8-bit = 156 kHz.
Độ phân giải 10-bit = 39 kHz.
- Module port serial đồng bộ (MSSP), hai mode hoạt động:
 - 3-wire SPITM (hỗ trợ cả 4 mode SPI).
 - Mode Slave và Master I²CTM.
- Module USART định vị:
 - Hỗ trợ RS-485 và RS-232.
- Module port Slave song song (PSP).
- ❖ Tính năng A/D
- Module biến đổi A/D 10-bit tương thích với:
 - Tốc độ mẫu nhanh.
 - Có khả năng biến đổi trong suốt thời gian Sleep.
 - Độ tuyến tính $\leq 1LSb$.

- Dò điện áp thấp chương trình (PLVD).
 - Hỗ trợ ngắt trên dò điện áp thấp.
- Xác lập lại khoảng thời gian chương trình (BOR).
 - ❖ Tính năng đặc biệt của vi điều khiển
- Bộ nhớ chương trình điện hình FLASH tăng cường chu kỳ xóa/ghi 100 000.
- Bộ nhớ EEPROM dữ liệu chu kỳ xóa/ghi 1 000 000 .
- Khả năng nhớ EEPROM dữ liệu/ FLASH: > 40 năm.
- Khả năng tự lập chương trình dưới bộ điều khiển phần mềm.
- Xác lập lại nguồn mở (POR), thời gian nguồn tắt (PWRT) và thời gian khởi động oscillator (OST).
- Thời gian watchdog (WDT) với oscillator RC on-chip của riêng nó cho hoạt động xác thực.
- Mã bảo vệ chương trình.
- Mode Sleep tiết kiệm công suất.
- Tùy chọn oscillator được chọn bao gồm:
 - Vòng lặp khóa chu kỳ 4x (của chính oscillator).
 - Ngõ vào clock oscillator thứ hai (32 kHz).
- Nguồn cung cấp 5V đơn trong vi mạch chuỗi chương trình (ICSPTM) qua hai chân.
- In-circuit Debug (ICD) qua hai chân.
 - ❖ Kỹ thuật CMOS
- Kỹ thuật FLASH/EEPROM tốc độ cao, công suất thấp.
- Thiết kế tĩnh hoàn chỉnh.
- Khoảng điện áp hoạt động rộng (2V đến 5.6V).
- Khoảng nhiệt độ rộng và industrial.
- Tiêu thụ điện thấp:
 - <1.6mA điện hình @ 5V, 4 MHz.
 - 25 μ A điện hình @ 3V, 32 kHz.
 - < 0,2 μ A dòng điện dự phòng, điện hình.

5.2.4.2 Mạch điều khiển trung tâm



Hình 5.39 Sơ đồ mạch điều khiển trung tâm

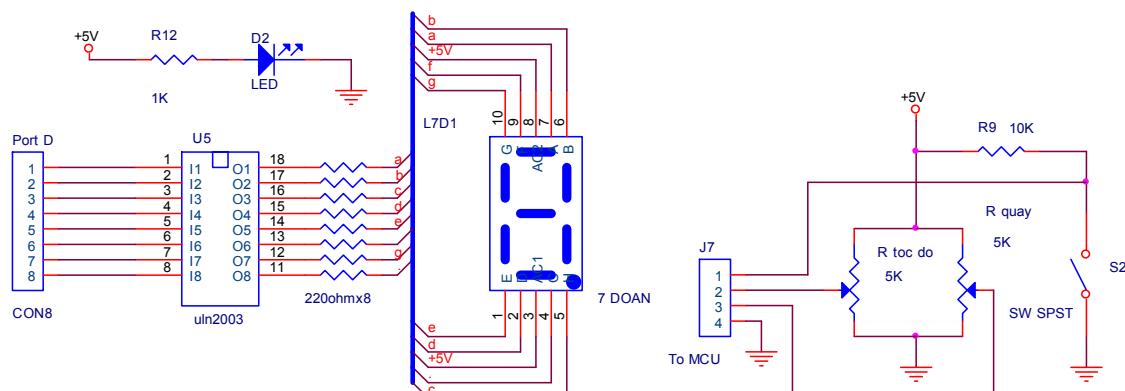
Port A được sử dụng như bộ biến đổi A/D với bốn kênh để nhận thông tin từ các cảm biến khác nhau: cảm biến accelerometer, gyro, hai bộ điện trở biến đổi có thể được nối với nhau. Chân AN3 được dùng làm ngõ vào điện áp tham khảo (V_{ref})

Port B như những nguồn ngắt để nhận tín hiệu từ cảm biến quang – incremental encoder quang (120 xung/vòng), dòng cực đại, và công tắc từ mạch điều khiển.

Port C có module CCP/PWM, do đó 4 chân port C và 3 chân port E là những tín hiệu logic để điều khiển động cơ. RC6, RC7 của port C giao tiếp với máy vi tính qua USART.

Port D không làm nhiệm vụ để điều khiển hay thu nhận tín hiệu cho hoạt động cân bằng của xe, do vậy port D được dùng để thể hiện các trạng thái của cảm biến và hoạt động của xe scooter bằng cách hiển thị ra một LED 7 đoạn qua giao diện chip ULN2003.

5.2.5 Bảng điều khiển và hiển thị:



Hình 5.40 Mạch bảng điều khiển và hiển thị

Bảng điều khiển là phần mạch duy nhất không được gắn ở dưới sàn xe, mà ngay trên tay lái, hướng về người điều khiển. Điện áp của hai biến trở được chuyển về vi điều khiển để xác định tốc độ di chuyển và góc quay mong muốn của xe, còn bảng LED 7 đoạn hiển thị tất cả các trạng thái của các cảm biến đo nghiêng, cảm biến đo vị trí, hoặc tiến trình đang hoạt động của mô hình.

5.2.6 Động cơ

5.2.6.1 Điều khiển động cơ bằng mạch lái MOSFET cầu H

Bảng mạch bộ phận lái động cơ nhận thông tin và kiểm soát H-bridge tương ứng. H-bridge điều khiển cường độ và hướng của dòng điện thông qua động cơ.



Hình 5.41 Động cơ được tích hợp bên trong bánh xe

H-bridge có khả năng thay đổi hướng. Để phòng chống shoot-through, H-bridge cần có thời gian nghỉ để thay đổi hướng, trong suốt thời gian này toàn bộ H-bridge được treo lên mức không hoạt động. Cả PIC kiểm soát động cơ và mạch điều khiển H-bridge tạo giai đoạn trì hoãn này để bảo vệ H-bridge khi gặp shoot-through dư thừa ($10\mu s$).

PIC kiểm soát liên lạc trực tiếp với những chip mạch điều khiển H-bridge, từ đó đường điều khiển được đưa đến MOSFET driver thông qua những đường tách opto.

Mã dữ liệu được thêm vào trên PIC kiểm soát động cơ để chắc chắn rằng không có tín hiệu shoot-through trên mạch điều khiển động cơ ở bộ phận MOSFET driver.

Để đo dòng điện qua mỗi động cơ, điện trở được đặt nối tiếp với H-bridge theo như mô tả ở 5.2.3.4 .



Hình 5.42 Stator đồng thời là trục động cơ



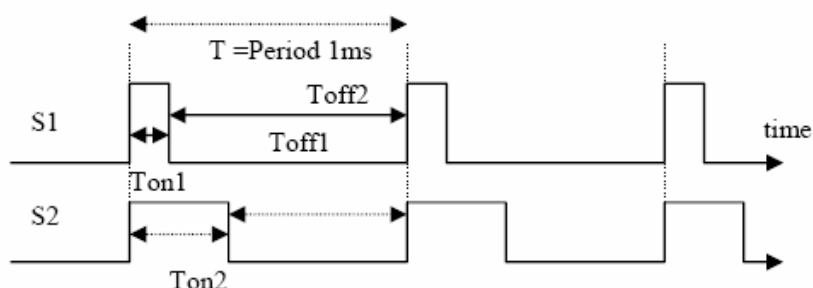
Hình 5.43 Rotor được gắn chặt vào bánh

5.2.6.2 Phương pháp điều độ rộng xung (PWM)

Có ba cách để điều khiển tốc độ động cơ DC servo trong kỹ thuật: Phương pháp điều áp (Linear Power Amplification), phương pháp điều độ tần số xung (PFM) và phương pháp điều độ rộng xung (PWM). Đề tài này áp dụng kỹ thuật điều độ rộng xung (PWM).

❖ Nguyên lý

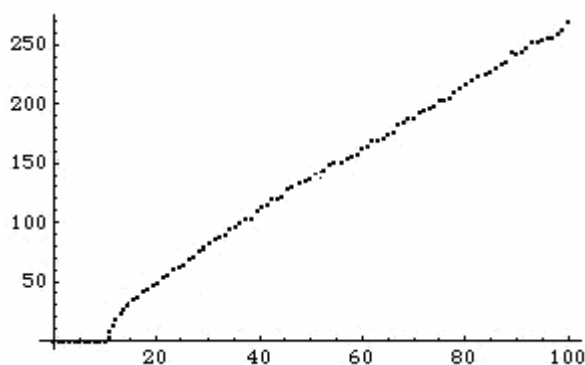
Nhìn trên hình vẽ, T_{ON} và T_{OFF} lần lượt là khoảng thời gian kích (trạng thái HIGH) và ngắt tín hiệu (trạng thái LOW) trong một chu kỳ. Việc làm này sẽ tạo ra một mức điện áp trung bình V_{TB} cấp cho động cơ tương ứng với một mức tốc độ động cơ.



Hình 5.44 Điều chế xung PWM

Như vậy, với hai tín hiệu S1 và S2 có cùng chu kỳ, nhưng chỉ có tỉ lệ T_{ON}/T_{OFF} giữa hai tín hiệu là khác nhau. Lúc này tương ứng hai điện áp trung bình khác nhau sẽ được thu và cho hai tốc độ khác nhau. Phương pháp này được gọi là điều rộng xung, cho phép điều khiển tốc độ động cơ như mong muốn.

Cũng cần chú ý rằng vì đây là điện áp trung bình của động cơ tương ứng với một tỉ lệ T_{ON}/T_{OFF} nào đó, nên mối quan hệ giữa vận tốc động cơ và điện áp trung bình này là không tuyến tính.



Hình 5.45 Ví dụ về quan hệ giữa vận tốc không tải của động cơ DC và PWM cycle cho động cơ DC Hitachi ^[27]

Điều này dễ hiểu bởi điện áp trung bình được tính bằng tích phân trong một chu kỳ của điện áp ngõ vào, cũng chính là trong khoảng T_{ON} .

Chu kỳ của một tín hiệu ra PWM là khoảng thời gian mà sau đó mẫu tín hiệu được lặp lại.

- Tần số PWM = $1/\text{chu kỳ}$.
- Hệ số duty là tỉ lệ bề rộng của mức '1' so với bề rộng chu kỳ.
- Độ phân giải (resolution) tín hiệu PWM là phân mịn nhất là hệ số duty có thể được điều chế.
- Có hai cách tạo ra xung PWM:
 - Mạch tương tự (analog): có giá trị thay đổi liên tục và có độ phân giải vô hạn cả về thời gian và biên độ, có thể được dùng để điều khiển trực tiếp nhiều thứ, như độ lớn âm được phát ra từ một radio, tốc độ của động cơ. Tuy nhiên, việc xây dựng và thiết kế mạch tương tự thường tốn kém. Có thể tạo ra một mạch PWM nhưng độ chính xác thường không cao và dễ điều khiển như mạch số.
 - Mạch số: bằng cách điều khiển các tín hiệu analog một cách số hóa, giá thành hệ thống và tiêu hao năng lượng hệ thống có thể giảm một cách đáng kể. Hơn nữa, nhiều vi điều khiển và DSPs hiện nay có thêm các bộ điều khiển PWM tích hợp, đưa đến việc điều khiển trở nên đơn giản hơn rất nhiều. Thông qua một bộ đếm có độ phân giải cao, hệ số duty của một sóng vuông được mã hóa thành một mức tín hiệu analog đặc trưng. Tín hiệu PWM vẫn là tín hiệu số vì tại một thời điểm bất kỳ, nguồn DC qua tải là mở hết hoặc ngắt hết.

❖ Cài đặt PWM trong các bộ vi điều khiển

Đối với điều khiển PWM bằng phần cứng từ các vi điều khiển có tích hợp bộ PWM trong các Timer, để bắt đầu PWM thì phần mềm của chip phải thực hiện các nhiệm vụ sau:

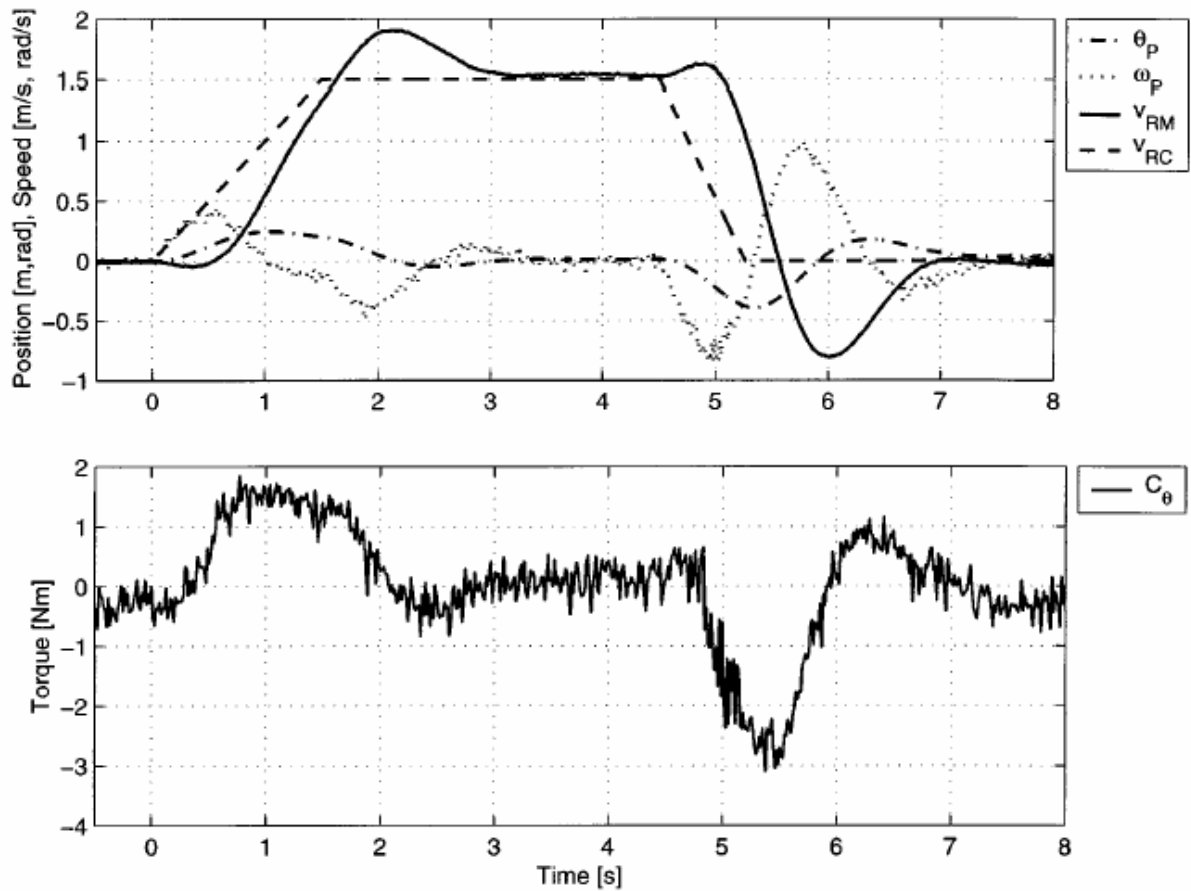
- Đặt chu kỳ của timer/counter điều chế xung vuông.
- Đặt thời gian của thanh ghi điều khiển PWM.
- Đặt hướng ra của tín hiệu PWM, tín hiệu có đảo hay không đảo.
- Khởi động timer.
- Mở khả năng của bộ điều khiển PWM.

❖ Ưu điểm

Một trong những ưu điểm của PWM là khả năng chống nhiễu giá trị từ bộ xử lý đến hệ thống điều khiển, do tín hiệu bản chất vẫn là tín hiệu số nên nhiễu chỉ xảy ra nếu nó đủ mạnh hơn mức tín hiệu số để đổi logic từ 1 thành 0 hoặc ngược lại.

Đây là một ưu điểm mà đôi khi được dùng cho truyền thông: tăng bề rộng của kênh liên lạc. Ngoài ra không phải tốn chi phí cho một bộ biến đổi D/A, cũng như tiết kiệm trong gian trong các thiết bị nhúng (embed system).

5.2.6.3 Đặc tuyến đáp ứng vận tốc và ngẫu lực động cơ DC có chổi than



Hình 5.46 Phản ứng ngẫu lực đối với tín hiệu vào là vận tốc

Kết quả trên được thu thập của Viện công nghệ Federal, Lausanne. Điều này hoàn toàn phù hợp với quan hệ vật lý của cơ học vật rắn về ngẫu lực và định luật Newton như sau:

$$C_\theta = J \frac{d\theta}{dt} \quad [5-13]$$

$$\theta = \frac{v}{R_{bánh}} \rightarrow \frac{d\theta}{dt} = \frac{1}{R_{bánh}} \frac{dv}{dt} \quad [5-14]$$

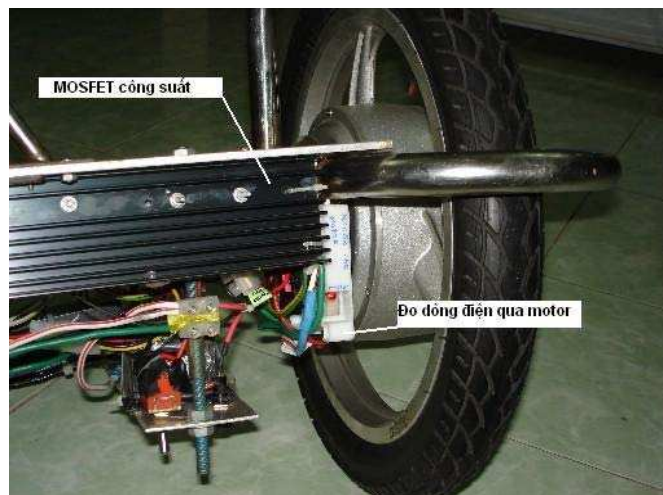
Từ đó ta hoàn toàn có cơ sở để điều khiển hệ thống theo tín hiệu PWM với hệ số chuyển đổi không tải là 0,72.

5.2.7

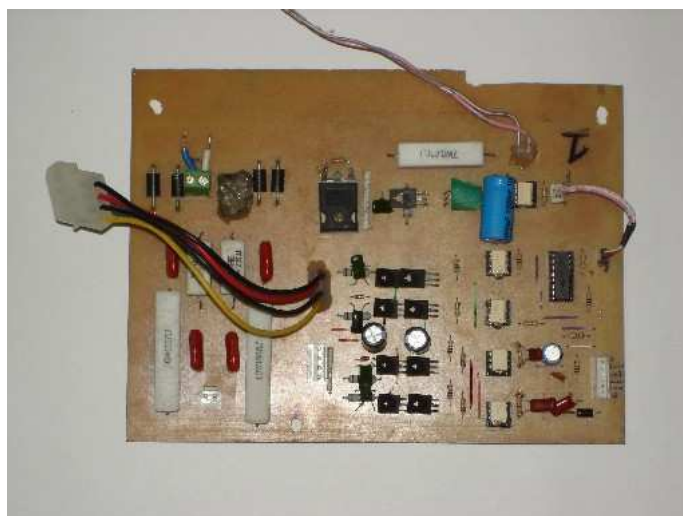
Hình chụp các mạch điện tử



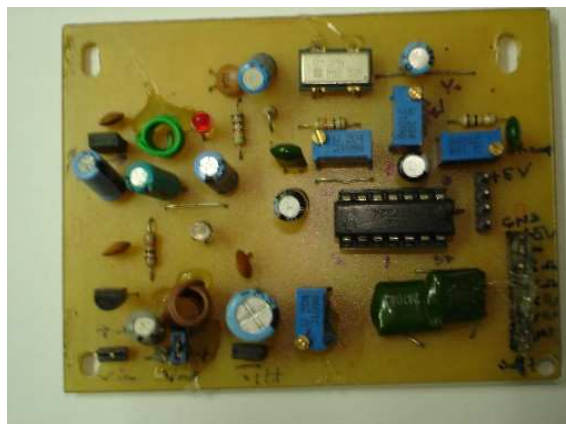
Hình 5.47 Nguồn và công tắc cho động cơ



Hình 5.48 Mạch MOSFET công suất và điện trở đo dòng



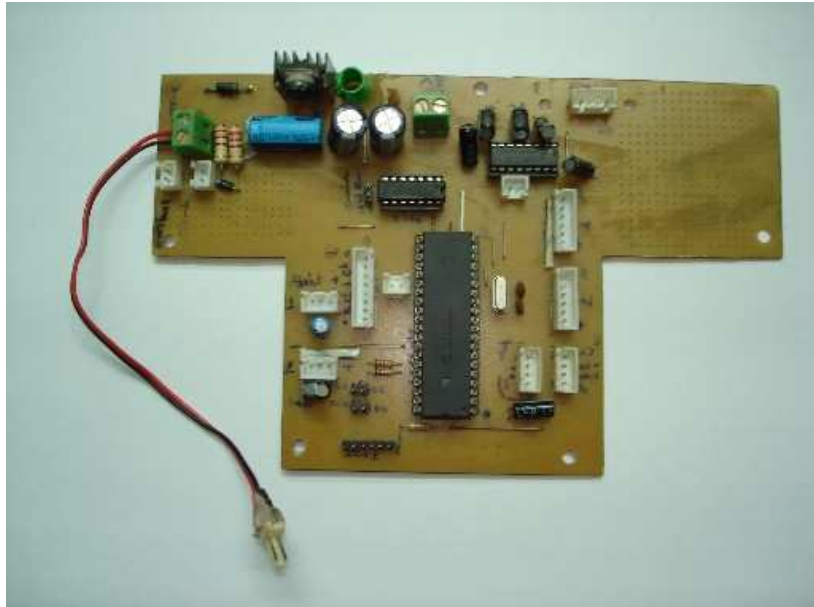
Hình 5.49 Mạch MOSFET driver



Hình 5.50 Mạch cảm biến và khuếch đại thuật toán



Hình 5.51 Bảng điều khiển và hiển thị



Hình 5.52 Mạch giao tiếp và điều khiển trung tâm

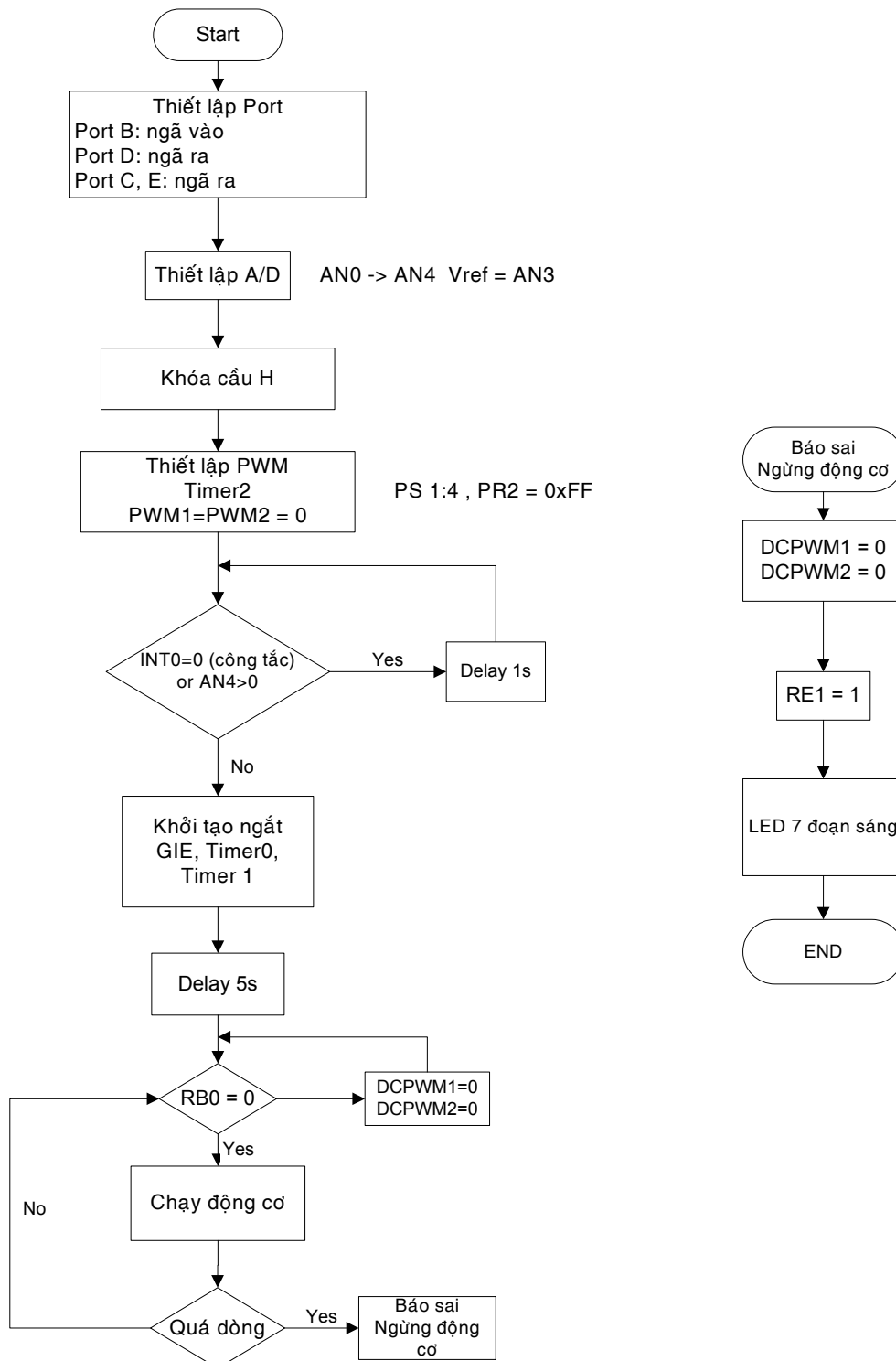


Hình 5.53 Lắp các mạch điện tử dưới sàn xe

5.3 Giải thuật - Lưu đồ chương trình

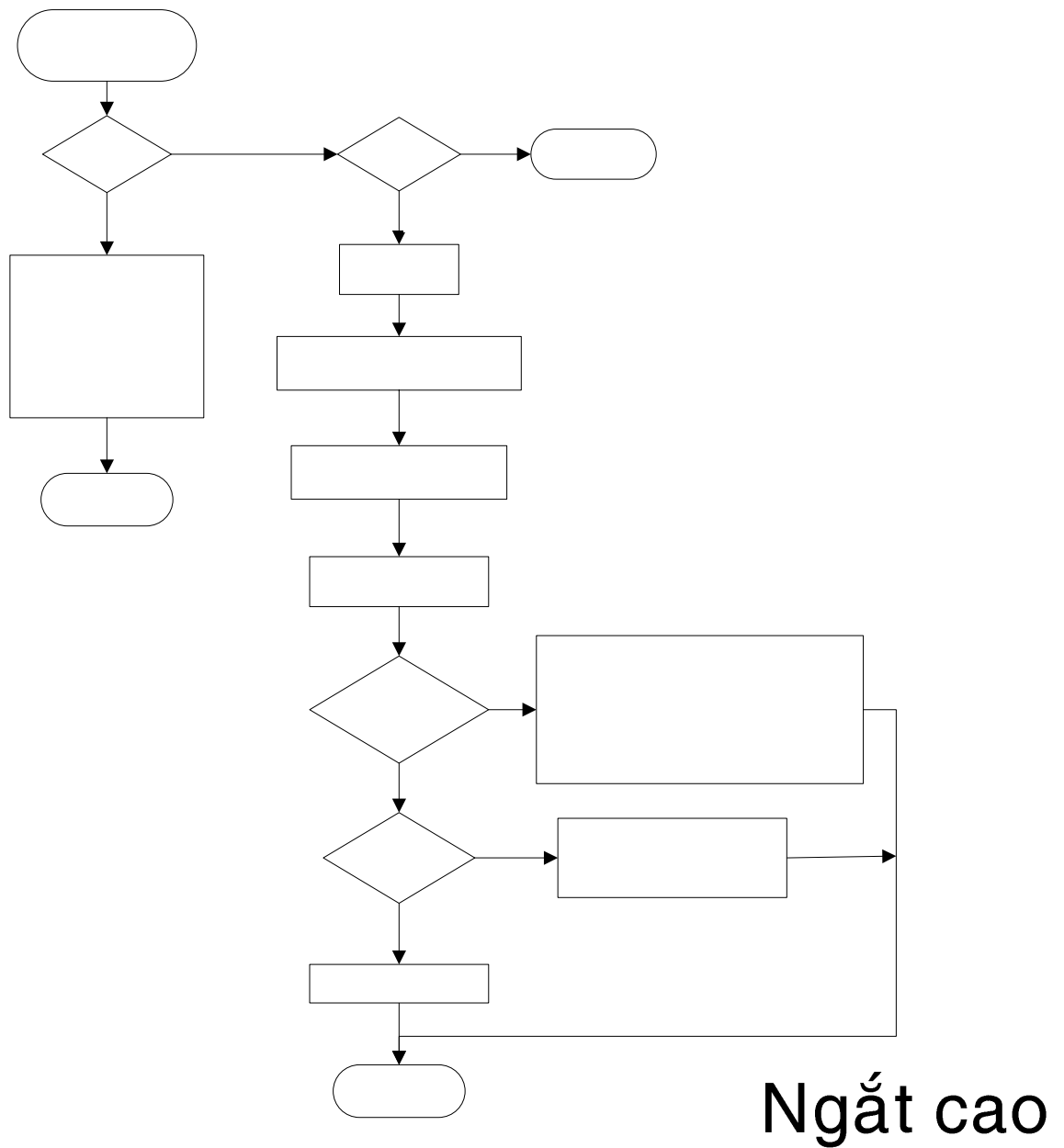
Chương trình điều khiển xe hai bánh tự cân bằng được chia ra làm 4 phần như nhau: sườn chương trình chính; 2 module ngắt cao và thấp; giải thuật điều khiển động cơ và cập nhật encoder. Việc tính toán và thu nhận giá trị các cảm biến được thực hiện ngay trong các ngắt.

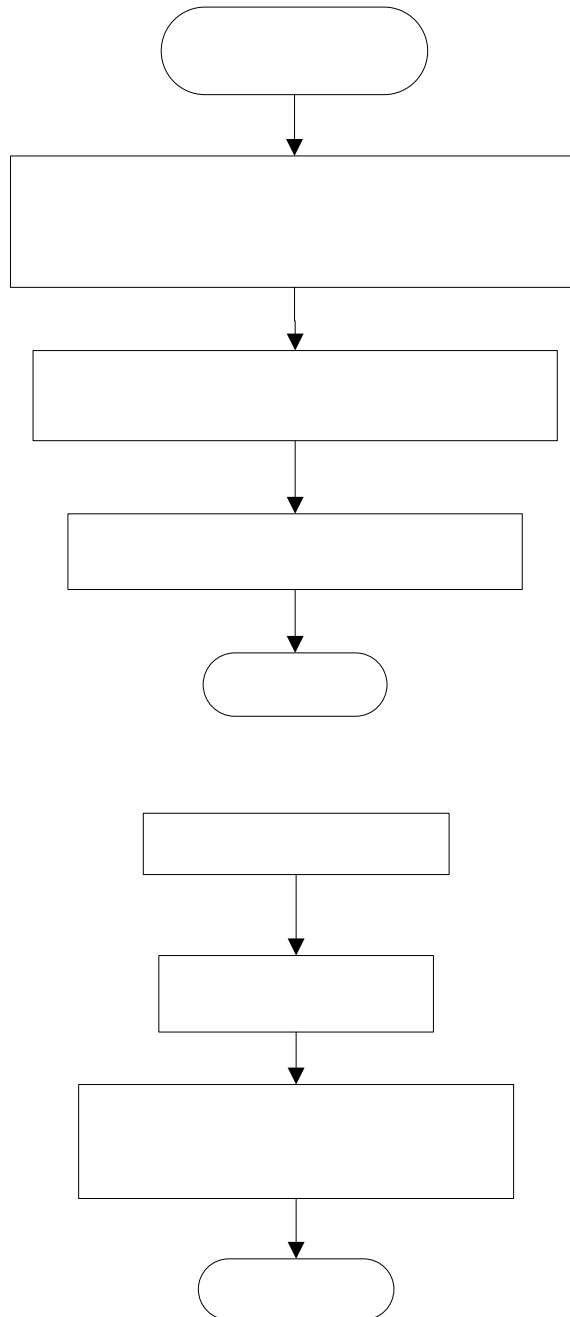
5.3.1 Chương trình chính



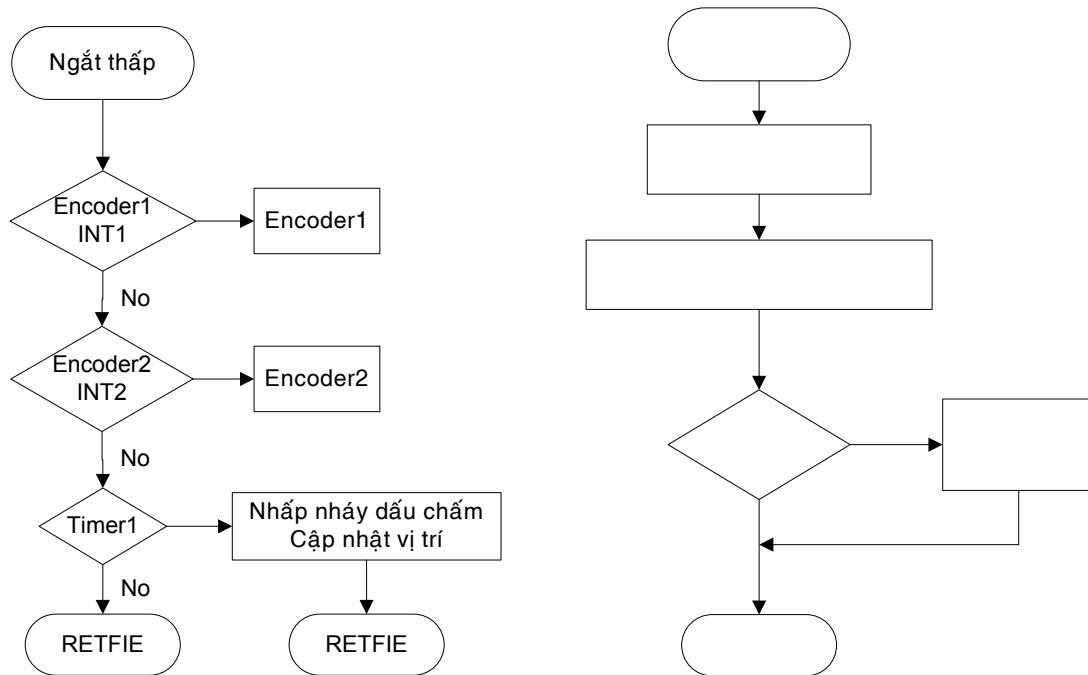
5.3.2 Chương trình ngắt

5.3.2.1 Ngắt cao (High priority ISR)

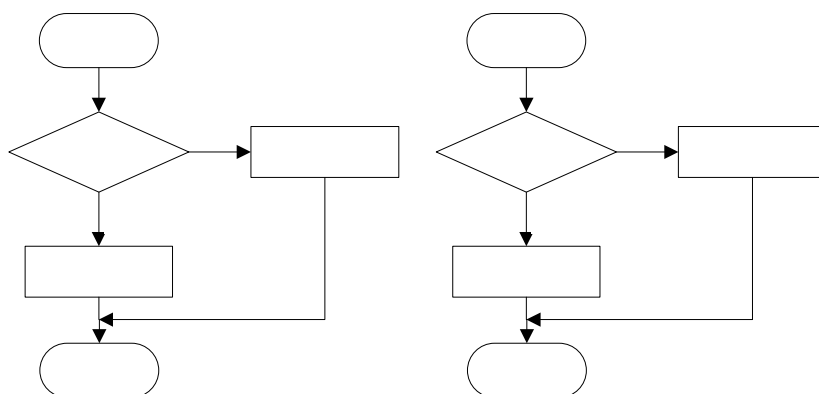




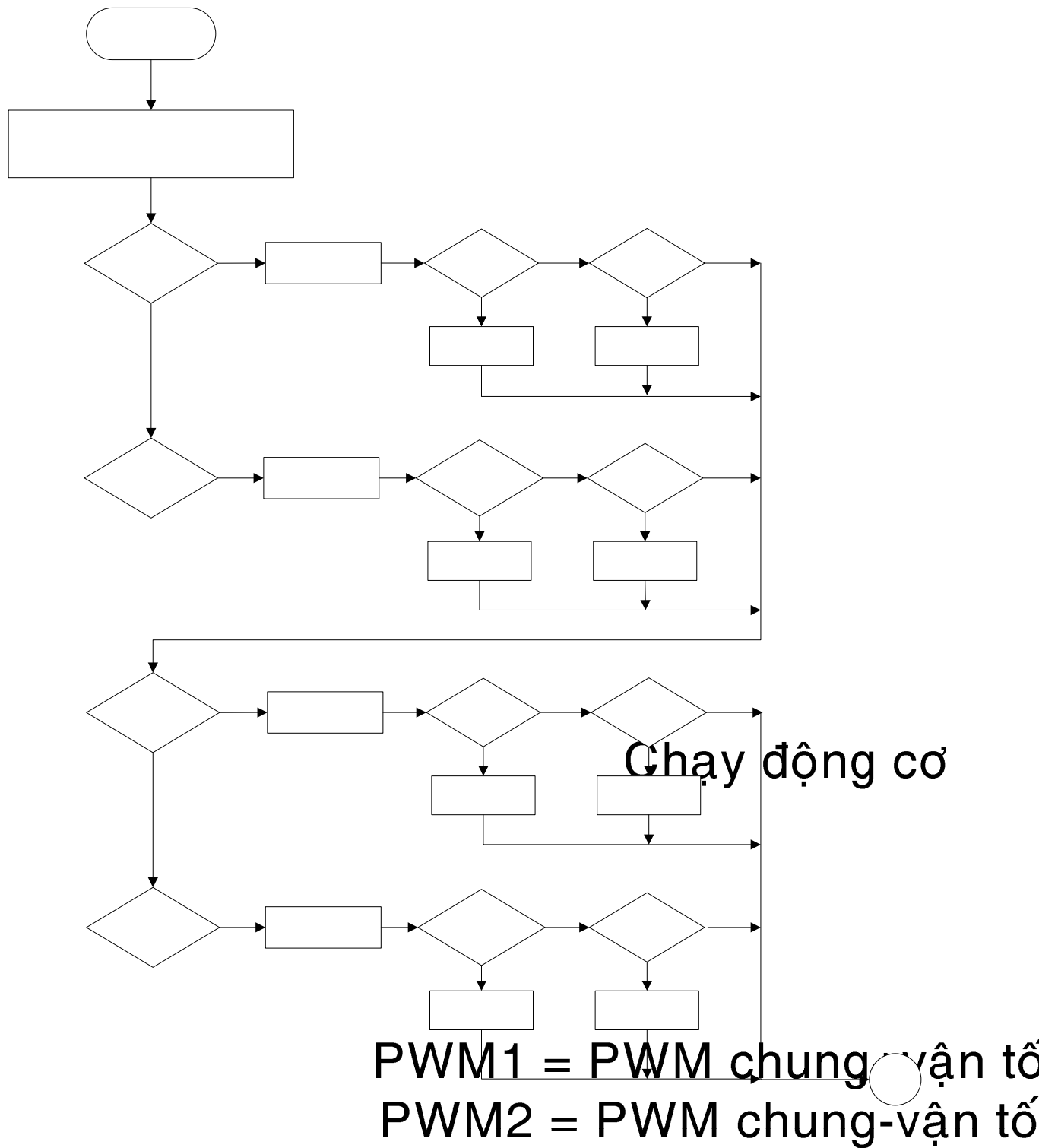
5.3.2.2 Ngắt thấp (Low priority ISR)



5.3.3 Cập nhật encoder

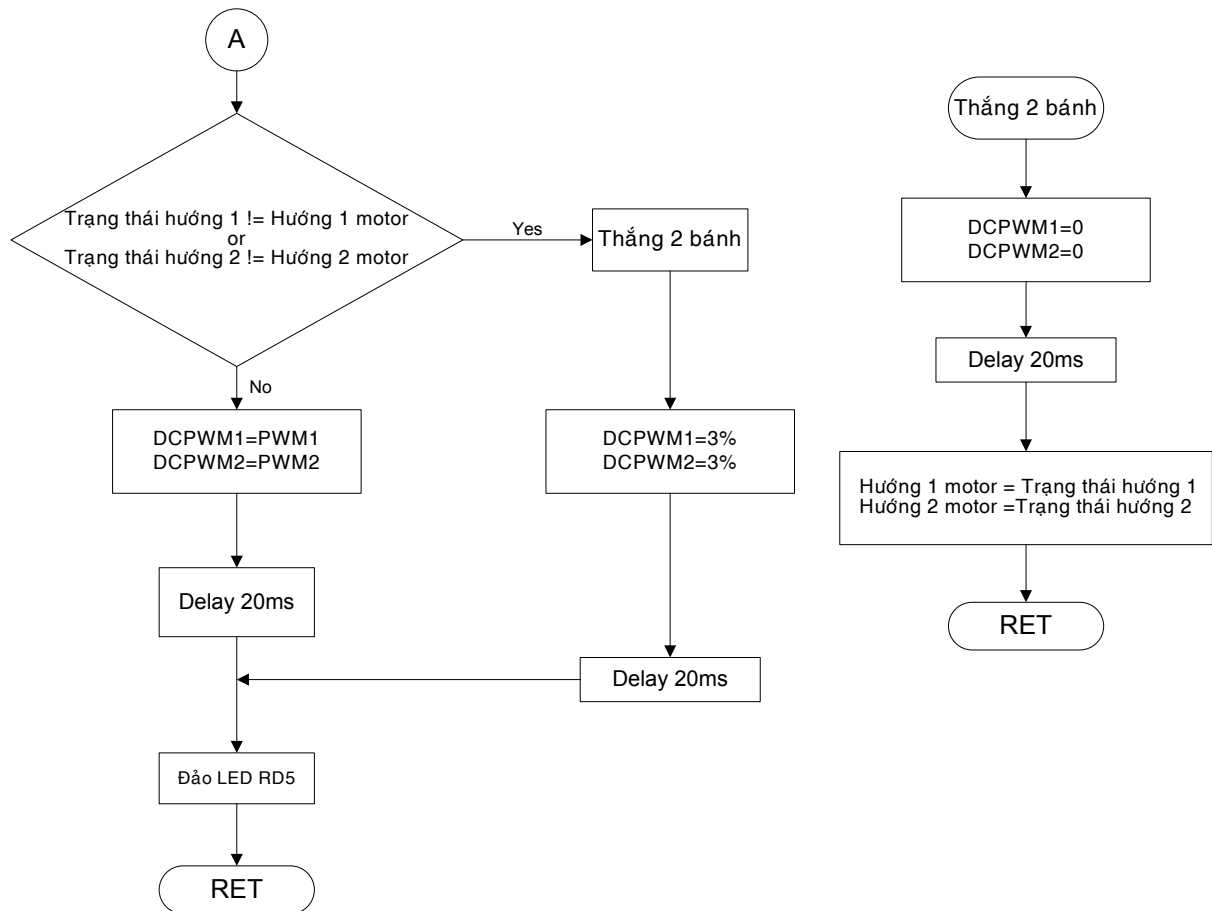


5.3.4 Điều khiển động cơ



PWM1 = PWM chung vận tốc

PWM2 = PWM chung-vận tốc



5.4 Kết quả

Ngoài các kết quả lý thuyết đạt được, trong thời gian làm luận văn, đề tài còn được thực hiện phần mô hình để có thể hoạt động được. Sau đây là một số kết quả của việc thực hiện mô hình:

- Thiết kế và chế tạo hoàn chỉnh mô hình cơ khí, bao gồm khung xe, sàn xe, tay lái.
- Thiết kế và chế tạo hoàn chỉnh các mạch điện tử, bao gồm mạch điều khiển trung tâm, mạch cảm biến đo góc, mạch cảm biến tốc độ dùng encoder, mạch MOSFET driver, mạch snubber, mạch MOSFET công suất, bảng điều khiển và hiển thị.
- Thiết kế và lắp đặt các mạch điện nguồn, các nguồn accu, các bộ phận bảo vệ quá dòng cho mạch công suất
- Lập trình cho mô hình tự thăng bằng, di chuyển tịnh tiến và quay trên địa hình phẳng.

Chương 6

CÁCH VẬN HÀNH

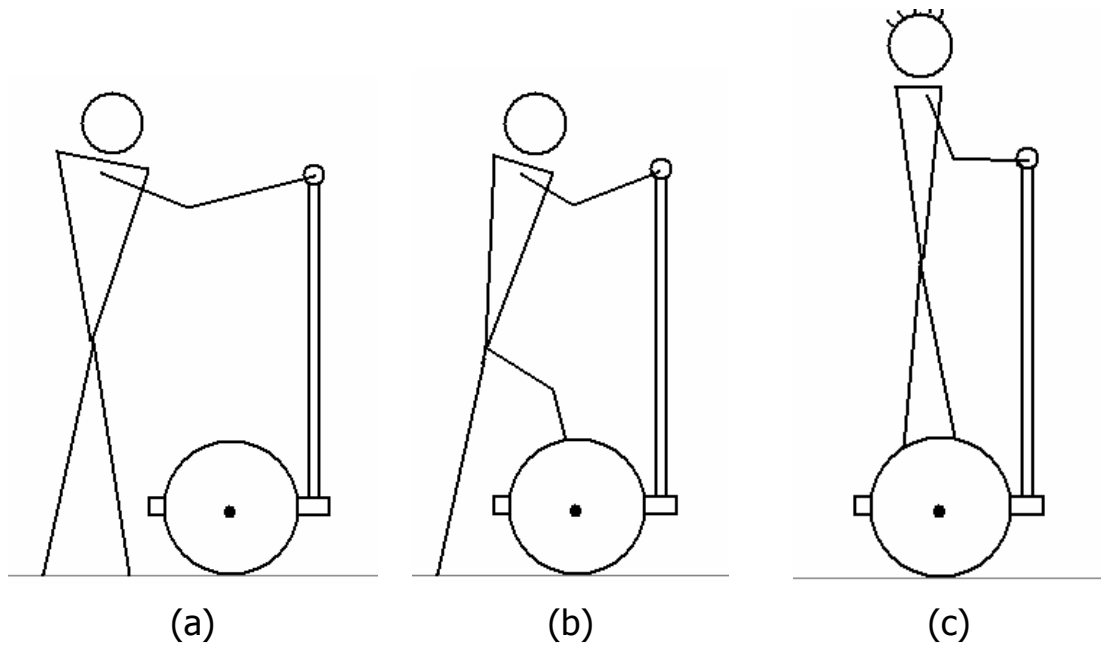
6.1 Cách sử dụng

- Kiểm tra tình trạng bánh xe: không có vết rạn, không bị bể, bánh xe được bơm căng.
- Kiểm tra bình điện còn đầy hay không, chỉ nên sử dụng khi bình còn hơn 30% trữ lượng điện.
- Kiểm tra tình trạng các cáp nguồn.



Hình 6. 1 Kiểm tra tình trạng xe trước khi sử dụng

- Bật công tắc điều khiển.
- Bật công tắc nguồn cho hai động cơ.
- Bật công tắc nguồn cho MOSFET driver, chú ý trước khi bật công tắc thì các biến trở và công tắc điều khiển phải đặt ở mức áp là 0.
- Giữ xe ở vị trí thẳng đứng. Bật công tắc giữ thăng bằng, sau khi thấy đèn nhấp nháy, tức là hệ thống giữ thăng bằng được (hình 6.2a).
- Từ từ bước một chân lên trọng tâm của xe (hình 6.2b).
- Nếu xe vẫn ổn định, bước chân còn lại lên xe (hình 6.2c).

**Hình 6.2** Cách bắt đầu dùng xe

- Một tay giữ tay lái, một tay điều khiển biến trở để thay đổi tốc độ. Biến trở bên phải điều khiển tốc độ, biến trở bên trái điều khiển tốc độ quay, quẹo.

**Hình 6.3** Đang sử dụng xe

- Nghiêng người về phía trước để tiến về phía trước.
- Nghiêng người ra sau để giảm tốc, dừng lại hoặc đi lùi.
- Thân người thẳng đứng sẽ giữ cho xe đứng yên tại chỗ.

6.2 Bảo dưỡng

- Khi thấy tín hiệu bình yếu thì phải ngừng sử dụng xe và sạc thêm cho bình.
- Khi không sử dụng trong thời gian 1 tháng, vẫn cần phải sạc điện cho bình ắc quy.
- Khi sạc bình, gỡ hết đầu các dây nguồn ra khỏi bình. Sạc đầy bình mất khoảng 20 giờ nếu bình cạn hoàn toàn, 10-12 giờ nếu bình còn từ 30% trở lên.
- Sau khi sạc xong, cắm lại các dây nguồn như ban đầu.
- Để xe ở nơi khô ráo, thoáng mát, tránh tiếp xúc với nước.

Chương 7

KẾT LUẬN

7.1 Những kết quả đạt được

- Thiết kế và hoàn thiện mô hình cơ khí xe tự cân bằng trên hai bánh.
- Thiết lập mô hình toán và hàm trạng thái cho mô hình.
- Xây dựng chương trình mô phỏng hoạt động bằng Visual Nastran và MatLAB-Simulink.
- Tất cả phần cứng thiết kế cho xe hoạt động theo yêu cầu:
 - Thiết kế mạch lái MOSFET và bộ MOSFET cầu H có thể chịu được công suất cao tối đa 45A ở 25°C, 30A ở 100°C, điện áp từ 12 – 60V.
 - Thiết kế mạch điện phối hợp giữa cảm biến gia tốc và cảm biến vận tốc góc, đồng thời đo góc tĩnh với cảm biến vận tốc góc gyro.
 - Thiết kế mạch điều khiển trung tâm giao tiếp với cảm biến, với bộ lái MOSFET và với máy tính.
- Giao tiếp giữa máy tính qua cổng nối tiếp (RS-232) USART để hiệu chỉnh các sensor và thử nghiệm.
- Thiết lập module lọc Kalman cho cảm biến đo góc (accelerometer và gyro) trên họ vi điều khiển PIC 18Fxxx.
- Phần mềm điều khiển được viết cho vi điều khiển PIC 18F452 kiểm soát sự cân bằng và chuyên chở một người.
- Xây dựng được mô hình đi trên địa hình bằng phẳng theo hướng thẳng và có thể quẹo góc nhỏ, có thể quay tròn tại chỗ.

7.2 Những kết quả chưa đạt được

- Không thực hiện được công suất MOSFET đủ lớn để scooter có thể di chuyển trên những bề mặt dốc.
- Việc quẹo khi vận chuyển người còn gặp khó khăn trong điều khiển, bởi việc điều khiển quẹo bằng biến trở không cân bằng là một giải pháp không tốt. Nếu được thay bằng một biến trở tự về vị trí cân bằng hoặc một cảm biến khoảng cách để xác định tư thế của người điều khiển xe muốn quẹo thì sẽ tốt hơn rất nhiều.

7.3 Những vấn đề chưa giải quyết

- Khối lượng của mô hình được xây dựng nặng hơn nhiều so với khối lượng tải được mô phỏng, cần phải giảm thêm.
- Cấu trúc kín và bền, chịu được va đập cho gầm xe để scooter có thể đi ở địa hình gập ghềnh và rung động cao cũng như vượt qua các vũng nước nhỏ.
- Làm đơn giản hóa quy trình khởi động xe và sạc bình.
- Chưa tạo hệ thống nạp điện lại cho bình khi giảm tốc hay thắng.
- Mạch công suất chưa an toàn khi quay ngược chiều quay của động cơ do dòng điện của động cơ trong phức tạp, đồng thời phải cải tiến thêm.
- Hoạt động còn hạn chế dưới 2 tiếng, nghĩa là chạy được khoảng chừng tối đa 20km.

7.4 Hướng phát triển

- Tạo giao diện giữa vi điều khiển của scooter với một vi điều khiển/ máy tính khác, để scooter có thể đóng vai trò là một platform cho mobile robot.
- Đối với xe dùng để di chuyển, cần thiết kế lại giao diện điều khiển đơn giản hơn trong lúc queo cua.
- Tăng khả năng tải trọng của xe lên hơn 100kg, đó là phải giải quyết vấn đề điện tử công suất.
- Làm nhẹ bớt trọng lượng của xe, bằng cách chuyển đổi khung inox sang khung sợi cacbon hoặc vật liệu tổng hợp.
- Thay đổi động cơ bánh xe từ có chổi than thành loại brushless để tăng ngẫu lực khi dùng cùng công suất.
- Nâng cao khả năng thắng gấp khi di chuyển ở vận tốc cao.

Tài liệu tham khảo

❖ Tài liệu trong nước:

- [1]. Dương Minh Trí, *Sơ đồ chân linh kiện bán dẫn*, Nhà xuất bản khoa học và kỹ thuật (tái bản lần 4), **1998**, trang 51,357,371.
- [2]. Jean-Marie BRÉBEC, *Cơ học vật rắn*, HACHETTE Supérieur – PFIEV - Nhà xuất bản Giáo dục, **2001**.
- [3]. Ngô Diên Tập, *Đo lường và điều khiển bằng máy tính*, Nhà xuất bản khoa học và kỹ thuật, **1997**, trang 19.
- [4]. Nguyễn Hữu Lộc (cb) và cộng sự, *Cơ sở thiết kế máy (phần 1)*, Trường Đại học Bách Khoa Tp.Hồ Chí Minh, **1999**, trang 302.
- [5]. Nguyễn Văn Đạo, *Stability of Dynamic systems*, Nhà xuất bản Đại học quốc gia tại Hà Nội, **1998**, trang 103.
- [6]. Trịnh Chất- Lê Văn Uyển, *Thiết kế hệ dẫn động cơ khí*, Nhà xuất bản Giáo dục, **2003**.
- [7]. Nguyễn Anh Kiệt, Nguyễn Minh Trung, *Luận văn tốt nghiệp “Nhận dạng và bám đối tượng bằng hai camera số”*, PFIEV Bộ môn Cơ điện tử, Trường Đại học Bách Khoa Hồ Chí Minh, **2004**, trang 62.

❖ Tài liệu nước ngoài:

- [8]. Georges ASCH et al., *Acquisition de données du capteur à l'ordinateur*, DUNOD, **1999**, p. 330.
- [9]. J.BLOT, *Les transistors*, DUNOD, **1995**, p. 51-57.
- [10]. Max GIODANO et Jaques LOTTIN, *Cours de Robotique – Description et fonctionnement des robots industriels*, Armand Colin Éditeur, **1990**, p. 68,69,82,83.
- [11]. Philippe de LARMINAT, *Automatique commande des systèmes linéaires (2^e édition revue et augmentée)*, HERMES Science Éditeur, **2000**, p. 167-211.
- [12]. Albert Paul MALVINO, Ph.D, E.E (1999), *Electronic principles*, Glencoe Mc Graw–Hill, (tái bản lần 6), **2001**, p. 468-473.

- [13]. William D.STANLEY, *Bộ khuếch đại xử lý và IC tuyến tính*, Nhà xuất bản khoa học và kỹ thuật, **1990**, p. 143.
- [14]. Ashish TEWARI, *Modern Control Design*, Indian Institute of Technology Kanpur, India. John Wiley and sons LTD., **2003**, p.219-236, 325-361.
- [15]. Greg WELCH and Gary BISHOP, *An Introduction to the Kalman Filter*, University of North Carolina at Chapel Hill, **2004**, p.1-16.

❖ **Tài liệu Website:**

- [16]. <http://www.geology.smu.edu/~dpa-www/robo/nbot/index.html>
- [17]. <http://www.tlb.org/scooter.html>
- [18]. <http://leiwww.epfl.ch/joe>
- [19]. <http://www.elex.camosun.bc.ca/Projects/2004/hvt/hvt.htm>
- [20]. <http://www.fusionglobal.net/gallery.htm>
- [21]. <http://www.barello.net/robots/gyrobot>
- [22]. <http://www.microchip.com>
- [23]. <http://www.rotomotion.com>
- [24]. <http://www.dprg.org/projects/index.html>
- [25]. <http://autopilot.sourceforge.net>
- [26]. <http://www.wulabs.org/bbot.html>
- [27]. <http://homepage.mac.com/sigfpe/Robotics/equibot.html>
- [28]. http://web.mit.edu/sanghyuk/www/balance_bot.html
- [29]. <http://home.earthlink.net/~botronics/index/balibot.html>
- [30]. http://www.analog.com/Analog_Root/productPage/productHome/adxl202,00.html
- [31]. <http://murata.com/enc-03>
- [32]. <http://www.picotech.com/applications/pwm-drivers/>
- [33]. <http://www.segway.com>
- [34]. http://www.ee.usyd.edu.au/tutorials_online/matlab/examples/pend/invpen.htm