

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG**



**ISO 9001:2015**

**THIẾT KẾ , CHẾ TẠO BỘ ĐIỀU CHỈNH PID  
CHO PHÉP ĐIỀU CHỈNH CÁC HỆ SỐ  
KHUYẾCH ĐẠI VÀ HẰNG SỐ THỜI GIAN**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY  
NGÀNH ĐIỆN TỰ ĐỘNG CÔNG NGHIỆP**

**HẢI PHÒNG - 2018**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG**



**ISO 9001:2015**

**THIẾT KẾ , CHẾ TẠO BỘ ĐIỀU CHỈNH PID  
CHO PHÉP ĐIỀU CHỈNH CÁC HỆ SỐ  
KHUYẾT ĐẠI VÀ HẰNG SỐ THỜI GIAN**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY  
NGÀNH ĐIỆN TỰ ĐỘNG CÔNG NGHIỆP**

Sinh viên: **Lã Mạnh Thắng**  
Người hướng dẫn: **GS.TSKH Thân Ngọc Hoàn**

**HẢI PHÒNG - 2018**

Cộng hoà xã hội chủ nghĩa Việt Nam  
**Độc lập – Tự Do – Hạnh Phúc**  
-----o0o-----  
BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

## **NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP**

Sinh viên : Lã Mạnh Thắng – MSV : 1412102018  
Lớp : ĐC1802- Ngành Điện Tự Động Công Nghiệp  
Tên đề tài : Thiết kế chế tạo bộ điều chỉnh loại PID cho phép  
điều chỉnh các hệ số khuếch đại và hằng số thời gian

## NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp (về lý luận, thực tiễn, các số liệu cần tính toán và các bản vẽ).

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. Các số liệu cần thiết để thiết kế, tính toán

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

3. Địa điểm thực tập tốt nghiệp..... :

## **CÁC CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP**

Người hướng dẫn thứ nhất:

Họ và tên	:	Thân Ngọc Hoàn
Học hàm, học vị	:	GS.TSKH
Cơ quan công tác	:	Trường Đại học dân lập Hải Phòng
Nội dung hướng dẫn	:	Toàn bộ đề tài

Người hướng dẫn thứ hai:

Họ và tên	:
Học hàm, học vị	:
Cơ quan công tác	:
Nội dung hướng dẫn	:

Đề tài tốt nghiệp được giao ngày      tháng      năm 2018.

Yêu cầu phải hoàn thành xong trước ngày.....tháng.....năm 2018

Đã nhận nhiệm vụ Đ.T.T.N  
Sinh viên

Đã giao nhiệm vụ Đ.T.T.N  
Cán bộ hướng dẫn Đ.T.T.N

Nguyễn Đăng Phương

GS.TSKH Thân Ngọc Hoàn

Hải Phòng, ngày.....tháng.....năm 2018

**HIỆU TRƯỞNG**

**GS.TS. NGUYỄN TRẦN HỮU NGHỊ**

## PHẦN NHẬN XÉT TÓM TẮT CỦA CÁN BỘ HƯỚNG DẪN

1. Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp.

.....

.....

.....

.....

.....

2. Đánh giá chất lượng của Đ.T.T.N ( so với nội dung yêu cầu đã đề ra trong nhiệm vụ Đ.T.T.N, trên các mặt lý luận thực tiễn, tính toán giá trị sử dụng, chất lượng các bản vẽ..)

[illegible]

3. Cho điểm của cán bộ hướng dẫn  
(Điểm ghi bằng số và chữ)

Ngày.....tháng.....năm 2018  
Cán bộ hướng dẫn chính  
(Ký và ghi rõ họ tên)

**NHẬN XÉT ĐÁNH GIÁ CỦA NGƯỜI CHĂM PHẢN BIỆN  
ĐỀ TÀI TỐT NGHIỆP**

1. Đánh giá chất lượng đề tài tốt nghiệp về các mặt thu thập và phân tích số liệu ban đầu, cơ sở lý luận chọn phương án tối ưu, cách tính toán chất lượng thuyết minh và bản vẽ, giá trị lý luận và thực tiễn đề tài.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. Cho điểm của cán bộ chấm phản biện  
(*Điểm ghi bằng số và chữ*)

Ngày.....tháng.....năm 2018  
Người chấm phản biện  
(*Ký và ghi rõ họ tên*)

# MỤC LỤC

## Trang

<b>LỜI NÓI ĐẦU.....</b>	<b>8</b>
<b>CHƯƠNG 1: CÁC BỘ ĐIỀU KHIỂN DÙNG TRONG HỆ THỐNG TỰ ĐỘNG TRUYỀN ĐỘNG ĐIỆN.....</b>	<b>9</b>
1.1 Khái niệm.....	9
1.2 Bộ điều khiển P.....	10
1.2.1 Hàm truyền .....	10
1.2.2 Quy luật điều chỉnh P.....	11
1.3 Bộ điều khiển PI.....	13
1.3.1 Hàm truyền .....	13
1.3.2 Kỹ thuật điều chỉnh PI.....	14
1.3.3 Quy luật điều chỉnh PI.....	15
1.4 Bộ điều khiển PID.....	18
1.4.1 Hàm truyền.....	18
1.4.2 Kỹ thuật điều khiển PID.....	19
1.4.3 Quy luật điều chỉnh PID.....	20
<b>CHƯƠNG 2: TỔNG QUAN VỀ ĐỘNG CƠ ĐIỆN MỘT CHIỀU...26</b>	
2.1 Tổng quan về động cơ điện một chiều.....	26
2.1.1 Cấu tạo động cơ điện một chiều .....	26
2.1.2 Đặc tính cơ của động cơ điện một chiều.....	29
2.2 Các phương pháp điều khiển động cơ một chiều.....	33
2.2.1 Phương pháp thay đổi điện trở phản ứng.....	33
2.2.2 Phương pháp thay đổi điện áp phản ứng.....	35
2.3 Giới thiệu một số hệ truyền động 1 chiều.....	37



2.3.1 Hệ truyền động máy phát - động cơ điện một chiều (F - Đ).....	37
2.3.2 Hệ truyền động xung áp – động cơ (XA – ĐC).....	41
2.3.3 Hệ truyền động chỉnh lưu – động cơ điện một chiều CL – ĐC .....	43
CHƯƠNG 3: THIẾT KẾ, CHẾ TẠO BỘ ĐIỀU KHIỂN ĐỘNG CƠ MỘT CHIỀU THUẬT TOÁN PID DÙNG PIC16F877A.....	55
3.1 Khái quát về mô hình.....	55
3.1.1 Giới thiệu sơ lược về các modul của mạch.....	55
3.1.2 Sơ đồ nguyên lý mạch.....	56
3.1.3 Cách vận hành mạch.....	56
3.2 Giới thiệu các linh kiện, phân tử sử dụng trong mạch.....	57
3.2.1 Vi điều khiển PIC 16F8774.....	57
3.2.1.1 Khái quát về điều khiển PIC 16F8774.....	57
3.2.1.2 Tổ chức bộ nhớ.....	62
3.2.1.3 Stack .....	66
3.2.1.4 Khái quát về chức năng của các port trong vi điều khiển PIC16F877A.....	66
3.2.1.5 Các vấn đề về Timer ().....	69
3.2.1.6 Ngắt ngoài (External Interrupt).....	78
3.2.2 Mạch cầu h ( h-bridge circuit ).....	81
3.2.3 LCD.....	82
3.3 Thiết kế mạch phần cứng , code chương trình và lưu đồ giải thuật.....	95
3.3.1 Thiết kế mạch phần cứng.....	95
3.3.2 Giới thiệu về chương trình viết code và biên dịch:.....	100
3.3.3 Lưu đồ giải thuật:.....	101
3.3.4 Một số hình ảnh thực tế của Bộ điều khiển.....	115

KẾT LUẬN .....	116
TÀI LIỆU THAM KHẢO.....	117

## LỜI MỞ ĐẦU

Động cơ một chiều được ứng dụng rất rộng rãi trong nhiều lĩnh vực thực tiễn, vì vậy có rất nhiều đề tài thiết kế bộ điều khiển cho động cơ một chiều và được đề cập rất nhiều trên các sách báo, tạp chí và internet. Việc ứng dụng động cơ DC vào sản xuất cũng như nghiên cứu khoa học đã mang lại những thành tựu nhất định. Tuy nhiên để động cơ DC hoạt động tốt thì ta phải thiết kế cho nó một bộ điều khiển giúp cho động cơ hoạt động một cách linh hoạt. Hiện nay có rất nhiều bộ điều khiển có thể làm tốt việc đó, tuy nhiên cá nhân em nhận thấy bộ điều khiển PID có thể đáp ứng tốt các yêu cầu của việc điều khiển động cơ DC, vì vậy em đã nhận đề tài **“Thiết kế, chế tạo bộ điều khiển PID điều khiển động cơ điện một chiều”** nhằm tìm hiểu kỹ hơn về bộ điều khiển đó.

Nội dung đề tài bao gồm:

CHƯƠNG 1: CÁC BỘ ĐIỀU KHIỂN DÙNG TRONG HỆ THỐNG TỰ ĐỘNG TRUYỀN ĐỘNG ĐIỆN

CHƯƠNG 2: TỔNG QUAN VỀ ĐỘNG CƠ ĐIỆN MỘT CHIỀU

CHƯƠNG 3: THIẾT KẾ, CHẾ TẠO BỘ ĐIỀU KHIỂN ĐỘNG CƠ MỘT CHIỀU THUẬT TOÁN PID DÙNG PIC16F877A

Trong quá trình làm đề tài tốt nghiệp, em luôn nhận được sự hướng dẫn, chỉ bảo tận tình và cung cấp những tài liệu cần thiết của thầy giáo **GS.TSKH Thân Ngọc Hoàn**. Em xin gửi tới thầy lời cảm ơn chân thành. Tuy nhiên, do thời gian và giới hạn của đồ án cùng với phạm vi nghiên cứu tài liệu với kinh nghiệm và kiến thức còn hạn chế nên bản đồ án này không tránh khỏi những thiếu sót rất mong sự đóng góp ý kiến của thầy cô để bản đồ án của em được hoàn thiện hơn.

**Sinh viên thực hiện**

**Lã Mạnh Thắng**

# CHƯƠNG 1 : CÁC BỘ ĐIỀU KHIỂN DÙNG TRONG HỆ THỐNG TỰ ĐỘNG ĐIỆN

## 1.1. KHÁI NIỆM.

Một bộ điều khiển vi tích phân tỉ lệ (bộ điều khiển PID) là một cơ chế phản hồi vòng điều khiển (bộ điều khiển) tổng quát được sử dụng rộng rãi trong các hệ thống điều khiển công nghiệp – bộ điều khiển PID được sử dụng phổ biến nhất trong số các bộ điều khiển phản hồi. Một bộ điều khiển PID tính toán một giá trị "sai số" là hiệu số giữa giá trị đo thông số biến đổi và giá trị đặt mong muốn. Bộ điều khiển sẽ thực hiện giảm tối đa sai số bằng cách điều chỉnh giá trị điều khiển đầu vào. Trong trường hợp không có kiến thức cơ bản về quá trình, bộ điều khiển PID là bộ điều khiển tốt nhất. Tuy nhiên, để đạt được kết quả tốt nhất, các thông số PID sử dụng trong tính toán phải điều chỉnh theo tính chất của hệ thống-trong khi kiểu điều khiển là giống nhau, các thông số phải phụ thuộc vào đặc thù của hệ thống.

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

Bằng cách điều chỉnh 3 hằng số trong giải thuật của bộ điều khiển PID, bộ điều khiển có thể dùng trong những thiết kế có yêu cầu đặc biệt. Đáp ứng

của bộ điều khiển có thể được mô tả dưới dạng độ nhạy sai số của bộ điều khiển, giá trị mà bộ điều khiển vọt lố điểm đặt và giá trị dao động của hệ thống. Lưu ý là công dụng của giải thuật PID trong điều khiển không đảm bảo tính tối ưu hoặc ổn định cho hệ thống.

Vài ứng dụng có thể yêu cầu chỉ sử dụng một hoặc hai khâu tùy theo hệ thống. Điều này đạt được bằng cách thiết đặt đội lợi của các đầu ra không mong muốn về 0. Một bộ điều khiển PID sẽ được gọi là bộ điều khiển PI, PD, P hoặc I nếu vắng mặt các tác động bị khuyết. Bộ điều khiển PI khá phổ biến, do đáp ứng vi phân khá nhạy đối với các nhiễu đo lường, trái lại nếu thiếu giá trị tích phân có thể khiến hệ thống không đạt được giá trị mong muốn.

Chú ý: Do sự đa dạng của lĩnh vực lý thuyết và ứng dụng điều khiển, nhiều qui ước đặt tên cho các biến có liên quan cùng được sử dụng.

## 1.2 BỘ ĐIỀU KHIỂN P [2]

### 1.2.1 Hàm truyền

Một dạng của mạch sớm pha được gọi là bộ điều khiển tỷ lệ (proportional controller, hay P controller), vì phương trình của nó bao gồm thành phần tỷ lệ có dạng như sau:

$$u_{ra}(t) = K_P u_{vào}(t) \quad (1.1)$$

Hàm truyền của bộ điều khiển P có dạng:

$$G_P(s) = \frac{U_{ra}}{(s)} \quad (1.2)$$

### 1.2.2 Kỹ thuật điều chỉnh P

Tín hiệu điều khiển trong quy luật tỉ lệ được hình thành theo công thức:

$$X = K_P \cdot e \quad (1.3)$$

Trong đó:  $K_P$  là hệ số khuếch đại của quy luật. Theo tính chất của khâu

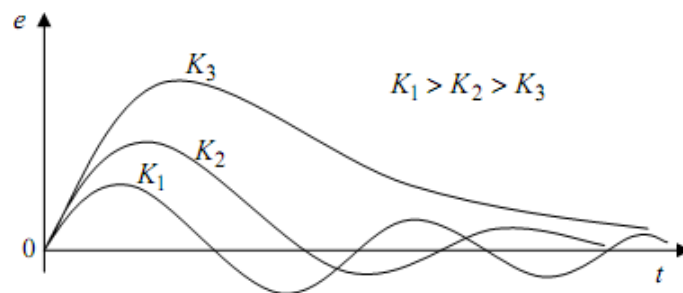
khuếch đại (hay khâu tỷ lệ) ta thấy tín hiệu ra của khâu luôn luôn trùng pha với tín hiệu vào. Điều này nói lên ưu điểm của khâu khuếch đại là có độ tác động nhanh. Vì vậy, trong công nghiệp, quy luật tỉ lệ làm việc ổn định với mọi đối tượng. Tuy nhiên, nhược điểm cơ bản của khâu tỉ lệ là khi sử dụng với các đối tượng tĩnh, hệ thống điều khiển luôn tồn tại sai lệch tĩnh. Để giảm giá trị sai lệch tĩnh thì phải tăng hệ số khuếch đại nhưng khi đó, tính dao động của hệ thống sẽ tăng lên và có thể làm hệ thống mất ổn định.

Trong công nghiệp, quy luật tỉ lệ thường được dùng cho những hệ thống cho phép tồn tại sai lệch tĩnh. Để giảm sai lệch tĩnh, quy luật tỉ lệ thường được hình thành theo biểu thức:

$$x = x_0 + K_p e \quad (1.4)$$

Trong đó  $x_0$  là điểm làm việc của hệ thống. Tác động điều khiển luôn giữ cho tín hiệu điều khiển thay đổi xung quanh giá trị này khi xuất hiện sai lệch.

Hình dưới mô tả quá trình điều khiển với các hệ số  $K_p$  khác nhau.



**Hình 1.1:** Quá trình điều khiển với các hệ số P khác nhau. Hệ số  $K_p$  càng cao thì sai số xác lập và quá điều khiển càng lớn.

### 1.2.3 Quy luật điều chỉnh P [2]

Giả sử bài toán ở đây là điều khiển tốc độ động cơ với tín hiệu đặt tốc độ là  $r = 1000$  vòng/phút,  $K_p = 15$ . Ta thử khảo sát xem sự biến thiên của tín hiệu ra của bộ điều khiển theo thời gian sẽ như thế nào.

Giả thiết tại thời điểm  $t = 0$  tín hiệu ra của hệ thống  $y = 0$ . Khi đó, tín hiệu sai lệch sẽ là  $e = r - y = 1000$ . Đầu ra của bộ điều khiển là  $u = k_p \cdot e = 15 \cdot 1000 = 1500$ . Tín hiệu này sẽ được đưa đến đầu vào của đối tượng cần điều khiển làm cho đầu ra  $y$  của nó bắt đầu tăng lên, dẫn đến  $e$  bắt đầu giảm.

Trong một số trường hợp, do quán tính của hệ thống, khi sai lệch  $e = 0$  (nghĩa là đầu ra  $y$  đã bằng với giá trị đặt  $r$ ) làm cho  $u = K_p \cdot e = 0$  nhưng tốc độ động cơ tiếp tục gia tăng.

Khi tốc độ vượt quá tốc độ đặt thì tín hiệu ra của bộ điều khiển lại lớn hơn 0, làm cho tốc độ lại tăng lên nhưng với quán tính nhỏ hơn.

Sau một vài chu kỳ dao động như trên thì tốc độ động cơ sẽ ổn định ở một giá trị nào đó, phụ thuộc vào các tham số của hệ thống.

a. Sai lệch tĩnh

Đối với quy luật điều chỉnh P, khi tốc độ động cơ bằng với tốc độ đặt  $e = 0$  thì tín hiệu điều khiển  $u = K_p \cdot e$  cũng bằng 0 và, do đó, tốc độ động cơ sẽ bị kéo giảm xuống. Vì vậy, muốn  $u$  khác 0 thì  $e$  phải khác 0. Nghĩa là phải luôn có một sai lệch giữa tín hiệu đặt và tín hiệu đầu ra thực tế của tín hiệu điều khiển.

Trong ví dụ trên, giả sử sau khi ổn định thì tốc độ động cơ đạt 970 vòng/phút thì sai lệch tĩnh sẽ là  $e = 1000 - 970 = 30$  vòng/phút và tín hiệu ra của bộ điều khiển sẽ là  $u = K_p \cdot e = 15 \cdot 30 = 450$ .

b. Giảm sai lệch tĩnh

Nếu tăng  $K_p$  lên 150 chẳng hạn thì sai lệch tĩnh  $e$  chỉ cần bằng 3 là có thể đủ để tạo ra một tín hiệu điều khiển bằng 450 để duy trì một mômen đủ lớn giữ cho động cơ quay. Rõ ràng, khi tăng  $K_p$  thì có thể làm giảm được sai

lệch tĩnh. Tuy nhiên, nếu  $K_p$  tăng quá lớn thì hệ có thể bị dao động, không ổn định.

### 1.3 BỘ ĐIỀU KHIỂN PI [2]

#### 1.3.1 Hàm truyền

Một dạng của mạch chậm pha được gọi là bộ điều khiển tỷ lệ-tích phân (proportional-integral controller, hay PI controller), vì phương trình của nó bao gồm hai thành phần, tỷ lệ và tích phân, có dạng như sau:

$$u_{ra}(t) = K_P u_{vào}(t) + K_I \int_0^t u_{vào}(\tau) d\tau \quad (1.5)$$

Hàm truyền của bộ điều khiển PI có dạng:

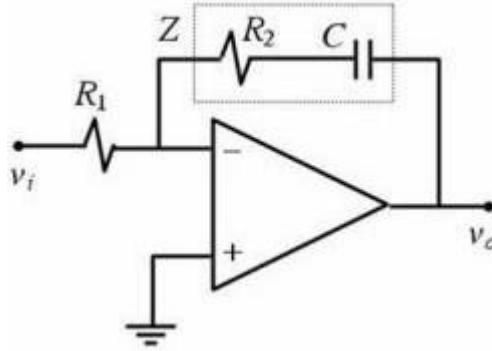
$$G_{PI}(s) = \frac{U_{ra}(s)}{U_{vào}(s)} = K_P + \frac{K_I}{s} \quad (1.6)$$

Tương tự như đối với bộ điều khiển PD, khi sử dụng mạch bù có hàm truyền  $G_{PI}(s)$  này, chúng ta có thể điều chỉnh ảnh hưởng của mạch bù, qua đó điều chỉnh đáp ứng của hệ thống bằng cách thay đổi hai tham số  $K_P$  và  $K_I$ . Chúng ta có thể sử dụng mạch chậm pha như trong Hình 1.1 để làm bộ điều khiển PI. Khi đó, các phần tử của mạch phải được chọn sao cho  $\alpha$  rất lớn để hàm truyền của mạch chậm pha có điểm cực gần bằng không. Hàm truyền của mạch chậm pha khi đó có thể xấp xỉ được như sau:

$$G_c(s) = \frac{1 + \tau s}{1 + \alpha \tau s} = \frac{1}{\alpha} \cdot \frac{1 + \tau s}{1/\alpha + \tau s} \cong \frac{1}{\alpha} + \frac{1/(\alpha \tau)}{s} \quad (1.7)$$

Đó chính là dạng của hàm truyền của bộ điều khiển PI.





**Hình 1.2:** Mạch của khâu hiệu chỉnh PI.

Mạch sớm pha được sử dụng để tạo ra một góc sớm pha, nhờ đó có được dự trữ pha như mong muốn cho hệ thống. Việc sử dụng mạch sớm pha cũng có thể biểu diễn được trên mặt phẳng  $s$  như một phương pháp làm thay đổi quỹ tích nghiệm của phương trình đặc trưng. Còn mạch chậm pha, mặc dù có ảnh hưởng làm giảm tính ổn định của hệ thống, thường được sử dụng để cung cấp sự suy giảm nhằm làm giảm sai số ở trạng thái xác lập của hệ thống.

### 1.3.2 Kỹ thuật điều chỉnh PI [2]

Để hệ thống vừa có tác động nhanh, vừa triệt tiêu được sai lệch tĩnh (là sai lệch giữa giá trị mong muốn so với giá trị ra thực tế khi hệ thống ở trạng thái xác lập) người ta kết hợp quy luật tỉ lệ với quy luật tích phân để tạo ra quy luật tỉ lệ - tích phân.

Tín hiệu điều khiển được xác định theo công thức:

$$x = K_p \cdot e + K_i \int e \cdot dt = K_p \left( e + \frac{1}{T_i} \int e \cdot dt \right) \quad (1.8)$$

Trong đó:  $K_p$  là hệ số khuếch đại

$T_i$  là hằng số thời gian tích phân

Hàm truyền của quy luật tỉ lệ tích phân có dạng:

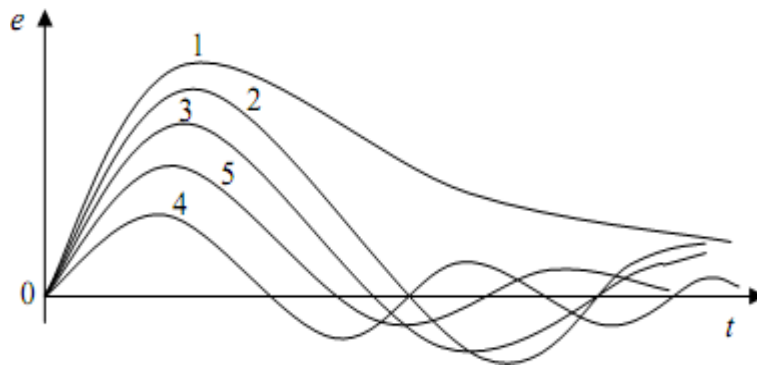
$$|W(p)| = K_p \left( 1 + \frac{1}{T_i \cdot p} \right) \quad (1.9)$$

$$W(j\omega) = K_p \left( 1 - j \frac{1}{T_i \cdot \omega} \right) \quad (1.10)$$

Đặc tính pha tần :

$$\varphi(\omega) = \arctg \left( -\frac{1}{T_i \cdot \omega} \right) \quad (1.11)$$

Rõ ràng, về tốc độ tác động thì quy luật PI chậm hơn quy luật tỉ lệ nhưng nhanh hơn quy luật tích phân. Hình dưới mô tả các quá trình quá độ của hệ thống điều khiển tự động sử dụng quy luật PI với các tham số  $K_p$  và  $T_i$  khác nhau



**Hình 1.3:** Quá trình quá độ của hệ thống điều khiển sử dụng quy luật PI.

- Đường 1 ứng với  $K_p$  nhỏ và  $T_i$  lớn. Tác động điều khiển nhỏ nên hệ thống không dao động.

- Đường 2 ứng với  $K_p$  nhỏ và  $T_i$  nhỏ. Tác động điều khiển tương đối lớn và thiên về quy luật tích phân nên hệ thống có tác động chậm, dao động với tần số nhỏ và không tồn tại sai lệch tĩnh.

- Đường 3 mô tả quá trình khi  $K_p$  lớn và  $T_i$  lớn. Tác động điều khiển tương đối lớn nhưng thiên về quy luật tỉ lệ nên hệ thống dao động với tần số lớn và tồn tại sai lệch tĩnh.

- Đường 4 tương ứng với quá trình điều khiển khi  $K_p$  lớn và  $T_i$  nhỏ. Tác động điều khiển rất lớn. Quá trình điều khiển dao động mạnh, thời gian điều khiển kéo dài và không có sai lệch tĩnh.

- Đường 5 được xem như là quá trình tối ưu khi  $K_p$  và  $T_i$  thích hợp với đối tượng điều khiển.

Trong thực tế, quy luật điều khiển PI được sử dụng khá rộng rãi và đáp ứng được chất lượng cho hầu hết các quá trình công nghệ. Tuy nhiên, do có thành phần tích phân nên độ tác động của quy luật bị chậm đi. Vì vậy, nếu đối tượng có nhiều tác động liên tục mà hệ thống điều khiển lại đòi hỏi độ chính xác cao thì quy luật PI không đáp ứng được.

### 1.3.3 Quy luật điều chỉnh PI [2]

Quy luật điều chỉnh P có ưu điểm là tác động nhanh. Tín hiệu điều khiển phụ thuộc trực tiếp vào sai lệch giữa tín hiệu đặt và tín hiệu thực. Tuy nhiên, khi sai lệch bằng 0 thì tín hiệu điều khiển cũng mất nên luôn tồn tại sai lệch tĩnh như đã nói ở trên.

Vậy làm thế nào để triệt tiêu sai lệch tĩnh?

Câu trả lời là phải đưa ra tín hiệu điều khiển cho đến khi nào sai lệch tĩnh bằng 0 thì giữ nguyên giá trị điều khiển đó.

Giả sử tại thời điểm  $k = 0$   $u_{i,0} = 0$ .

Tại thời điểm  $k=1$  thì  $u_{i,1} = K_{ie1}$ ;  $u_{i,0} = K_{ie1}$  tương tự như bộ điều khiển kiểu P.

Tại thời điểm tiếp theo  $u_{i,2} = K_{ie2} + u_{i,1}$  và cứ như vậy tín hiệu điều khiển lần sau bằng tín hiệu điều khiển ở lần trước đó cộng đại số với tích giữa hệ số tích phân và sai lệch làm cho sai lệch  $e$  (dương hoặc âm) giảm dần (hệ ổn định).

Giả thiết tại thời điểm  $t = 0$   $u_{i,0} = 0$  thời điểm  $t = 1$  mà  $y_1 = 200$  thì  $e_1 = r - y_1 = 1000 - 200 = 800$  và tín hiệu ra của bộ điều khiển sẽ là

$$u_{i,1} = K_{ie1} + u_{i,0} = 0,25.800 + 0 = 200$$

Tín hiệu này sẽ được đưa đến đầu vào của đối tượng cần điều khiển làm cho đầu ra  $y$  của nó tiếp tục tăng, dẫn đến  $e$  bắt đầu giảm.

Tại thời điểm  $t = 2$  giả sử  $y_2 = 500$  thì  $e_2 = r - y_2 = 1000 - 500 = 500$  và tín hiệu ra của bộ điều khiển sẽ là

$$u_{i,2} = K_{ie2} + u_{i,1} = 0,25.500 + 200 = 125 + 200 = 325$$

Tại thời điểm  $t = 3$  giả sử  $y_3 = 800$  thì  $e_3 = r - y_3 = 1000 - 800 = 200$  và tín hiệu ra của bộ điều khiển sẽ là

$$u_{i,3} = K_{ie3} + u_{i,2} = 0,25.200 + 325 = 50 + 325 = 375$$

(giá trị  $u_{i,2} = 325$  của chu kỳ điều khiển trước được cộng thêm 50). Đầu ra  $y$  tiếp tục tăng.

Tại thời điểm  $t = 4$  giả sử  $y_4 = 900$  thì  $e_4 = r - y_4 = 1000 - 900 = 100$  và tín hiệu ra của bộ điều khiển sẽ là

$$u_{i,4} = K_{ie4} + u_{i,3} = 0,25.100 + 375 = 25 + 375 = 400$$

(giá trị  $u_{i,3} = 375$  của chu kỳ điều khiển trước được cộng thêm 25). Đầu ra  $y$  tiếp tục tăng.

Tại thời điểm  $t = 5$  giả sử đầu ra đã bám theo đầu vào, nghĩa là  $y_5 = 1000$  thì  $e_5 = r - y_5 = 1000 - 1000 = 0$  và tín hiệu ra của bộ điều khiển sẽ là

$$u_{i,5} = K_{ie5} + u_{i,4} = 0,25.0 + 400 = 400$$

(tín hiệu ra của bộ điều khiển được giữ nguyên giá trị  $u_{i,4} = 400$  của chu kỳ điều khiển trước). Tín hiệu đầu ra bộ điều khiển không thay đổi và tốc độ được giữ nguyên.

Giả sử tại thời điểm  $t = 6$  tốc độ  $y_6 = 1100$  thì

$e_6 = r - y_6 = 1000 - 1100 = -100$  Tín hiệu ra của bộ điều khiển sẽ

$$\text{là } u_{i,6} = K_{ie6} + u_{i,5} = 0,25.(-100) + 400 = 400 - 25 = 375$$

(tín hiệu ra của bộ điều khiển đã được bớt đi giá trị -25 so với chu kỳ

điều khiển trước). Tín hiệu đầu ra bộ điều khiển giảm làm cho tốc độ động cơ cũng giảm xuống.

Như vậy, nếu tại thời điểm  $t = n$  đầu ra bám kịp tín hiệu đặt (sai lệch bằng 0) thì tín hiệu ra của bộ điều khiển  $u_i, n$  cũng sẽ không đổi. Tại bất kỳ một thời điểm nào nếu sai lệch lại khác 0 thì tín hiệu ra của bộ điều khiển lại tiếp tục thay đổi nhằm kéo đầu ra bám theo tín hiệu đặt.

Tác động chậm.

Việc "thêm, bớt" nói trên làm cho  $e_k$  nhỏ dần và giá trị "thêm, bớt"  $K_{iek}$  cũng nhỏ dần... Quá trình đó diễn ra liên tục cho đến khi đáp ứng đầu ra của hệ bằng với giá trị đặt hay  $e_k = 0$ . Khi  $K_i$  càng lớn thì đáp ứng đầu ra càng nhanh đạt đến giá trị gần với giá trị mong muốn nhưng quá trình "thêm, bớt" để cho giá trị sai lệch tiến về 0 lại diễn ra càng chậm, làm cho thời gian điều khiển kéo dài. Cần lưu ý ở đây là "chậm" tiến về giá trị đặt chứ còn tại thời điểm đầu thì đáp ứng của khâu I vẫn bám rất nhanh tới giá trị đặt nếu  $K_i$  lớn (cũng giống hệt như tác động điều chỉnh kiểu P).

Như vậy, kết hợp tác động nhanh của khâu P và khả năng triệt tiêu sai lệch tĩnh của khâu I ta sẽ có được một bộ điều khiển kiểu PI được sử dụng rất rộng rãi trong công nghiệp.

## 1.4 BỘ ĐIỀU KHIỂN PID [2]

### 1.4.1 Hàm truyền

Một dạng của mạch sớm-chậm pha được sử dụng rất phổ biến, nhất là trong các hệ thống điều khiển công nghiệp, là bộ điều khiển tỷ lệ-vi tích phân (proportional-integral-derivative controller hay PID controller), hay còn gọi là bộ điều khiển ba phương thức (three-mode controller), được biểu diễn bằng phương trình vi phân có dạng như sau:

$$u_{ra}(t) = K_P u_{vào}(t) + K_D \frac{du_{vào}(t)}{dt} + K_I \int_0^t u_{vào}(\tau) d\tau \quad (1.12)$$

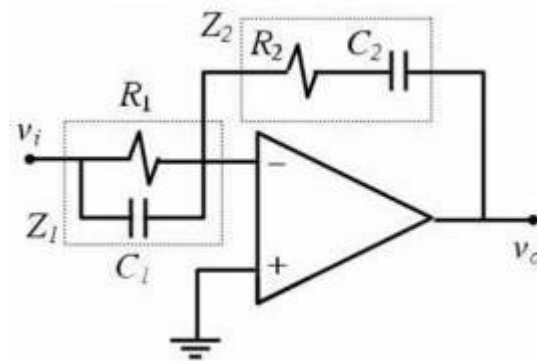
Hàm truyền của bộ điều khiển PID nói trên sẽ là:

$$G_{PID}(s) = \frac{U_{ra}(s)}{U_{vào}(s)} = K_p + K_D s + \frac{K_I}{s}$$

Error! Bookmark not defined.

Thành phần tỷ lệ ( $K_P$ ) của bộ điều khiển PID có tác dụng làm tăng tốc độ của đáp ứng và làm giảm nhưng không làm triệt tiêu sai số ở trạng thái xác lập. Thành phần tích phân ( $K_I$ ) có thể làm triệt tiêu sai số ở trạng thái xác lập, nhưng sẽ làm ảnh hưởng đến hiệu suất nhất thời theo chiều hướng không được mong muốn vì phần trăm quá mức của đáp ứng nhất thời sẽ tăng khi  $K_I$  tăng. Ngược lại với  $K_I$ , thành phần đạo hàm ( $K_D$ ) có tác dụng nâng cao tính ổn định của hệ thống và làm giảm phần trăm quá mức của đáp ứng nhất thời, nhờ đó cải thiện hiệu suất nhất thời của hệ thống vòng kín.

Đặc biệt, người ta thường sử dụng các bộ điều khiển PID để điều khiển những quá trình quá phức tạp để có thể thiết lập được các mô hình toán học chính xác, thường là các quá trình phi tuyến và đa biến. Trong những trường hợp đó, với ba tham số  $K_P$ ,  $K_I$  và  $K_D$  của bộ điều khiển PID để điều chỉnh, chúng ta vẫn có thể hy vọng đạt được hiệu suất mong muốn cho hệ thống mà không cần thực hiện nhiều bước phân tích và thiết kế phức tạp.



**Hình 1.4:** Mạch của khâu hiệu chỉnh PID.

Trong nhiều trường hợp, chúng ta có thể cần một mạch bù có thể cung cấp cả góc sớm pha như của một mạch sớm pha và sự suy giảm về độ lớn như

của một mạch chậm pha. Một mạch có đặc tính như vậy được gọi là mạch sớm-chậm pha (lead-lag network). Một mạch sớm-chậm pha sẽ có cả hai thành phần sớm pha và chậm pha, vì vậy hàm truyền của mạch sẽ có dạng như sau:

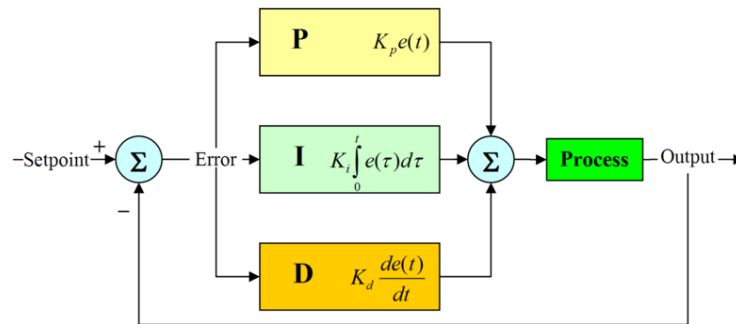
$$G_c(s) = K \frac{s - z_1}{s - p_1} \cdot \frac{s - z_2}{s - p_2} = K_1 \frac{1 + \alpha \tau_1 s}{1 + \tau_1 s} \cdot \frac{1 + \tau_2 s}{1 + \beta \tau_2 s} \quad (1.14)$$

ở đó,  $|z_1| < |p_1|$  và  $|z_2| > |p_2|$ , hay  $\alpha > 1$  và  $\beta > 1$ .

### 1.4.2 Kỹ thuật điều khiển PID

Kỹ thuật điều khiển PID (Tỉ lệ, tích phân, vi phân) được sử dụng rộng rãi trong công nghiệp. Dùng để điều khiển những quá trình phức tạp để thiết lập mô hình toán học chính xác, thường là các quá trình đa biến và phi tuyến.

Điều khiển PID là một kiểu điều khiển có hồi tiếp, ngõ ra thay đổi tương ứng với sự sai lệch giữa tín hiệu đầu ra so với đáp ứng mong muốn. Tùy theo mức độ thì người ta có thể chỉ áp dụng điều khiển P, điều khiển PI, điều khiển PD hoặc điều khiển PID.



**Hình 1.5:** Mô hình thuật toán PID.

Để tăng tốc độ tác động của quy luật PI, trong thành phần của nó người ta ghép thêm thành phần vi phân và nhận được quy luật điều khiển tỉ lệ vi tích phân. Tác động điều khiển được tính toán theo công thức:

$$x = K_p e + K_i \int + K_d \frac{de}{dt} = K_p \left( e + \frac{1}{T_i} \int e \cdot dt + T_d \frac{de}{dt} \right) \quad (1.15)$$

Trong đó:  $K_p$  là hệ số khuếch đại

$$T_i = \frac{K_p}{K_i} \text{ là hằng số thời gian tích phân}$$

$$T_d = \frac{K_d}{K_p} \text{ là hằng số thời gian vi phân}$$

Hàm truyền của quy luật tỉ lệ - vi tích phân có dạng:

$$W_p = K_p \left( 1 + \frac{1}{T_i \cdot p} + T_d \cdot p \right) \quad (1.16)$$

Hàm truyền tần số của khâu PID:

$$W_j(\omega) = K_p \left( 1 + j \left( T_d \omega - \frac{1}{T_i \omega} \right) \right) \quad (1.17)$$

Đặc tính pha tần:

$$\varphi(\omega) = \text{arctg} \left( \frac{T_i T_d \omega^2 - 1}{T_i \omega} \right) \quad (1.18)$$

Như vậy khi  $\omega = 0$  thì  $\varphi(\omega) = -\pi/2$ , còn khi  $\omega = \sqrt{\frac{1}{T_i T_d}}$  thì  $\varphi(\omega) = 0$  và khi  $\omega = \infty$  thì  $\varphi(\omega) = \pi/2$ . Rõ ràng góc lệch pha của tín hiệu ra so với tín hiệu vào nằm trong khoảng từ  $-\pi/2$  đến  $\pi/2$ , phụ thuộc vào các tham số  $K_p$ ,  $T_i$ ,  $T_d$ , và tần số của tín hiệu vào. Nghĩa là về tốc độ tác động, quy luật PID còn có thể nhanh hơn cả quy luật tỉ lệ. Quy luật PID đáp ứng được yêu cầu về chất lượng của hầu hết các quy trình công nghệ, nhưng việc hiệu chỉnh các tham số của nó rất phức tạp, đòi hỏi người sử dụng phải có một trình độ nhất định. Vì vậy, trong công nghiệp, quy luật PID chỉ sử dụng ở những nơi cần thiết, khi quy luật PI không đáp ứng được yêu cầu về chất lượng điều chỉnh.

Để hiểu rõ hơn, ta có thể xem hình so sánh về sai lệch điều khiển giữa các quy luật:

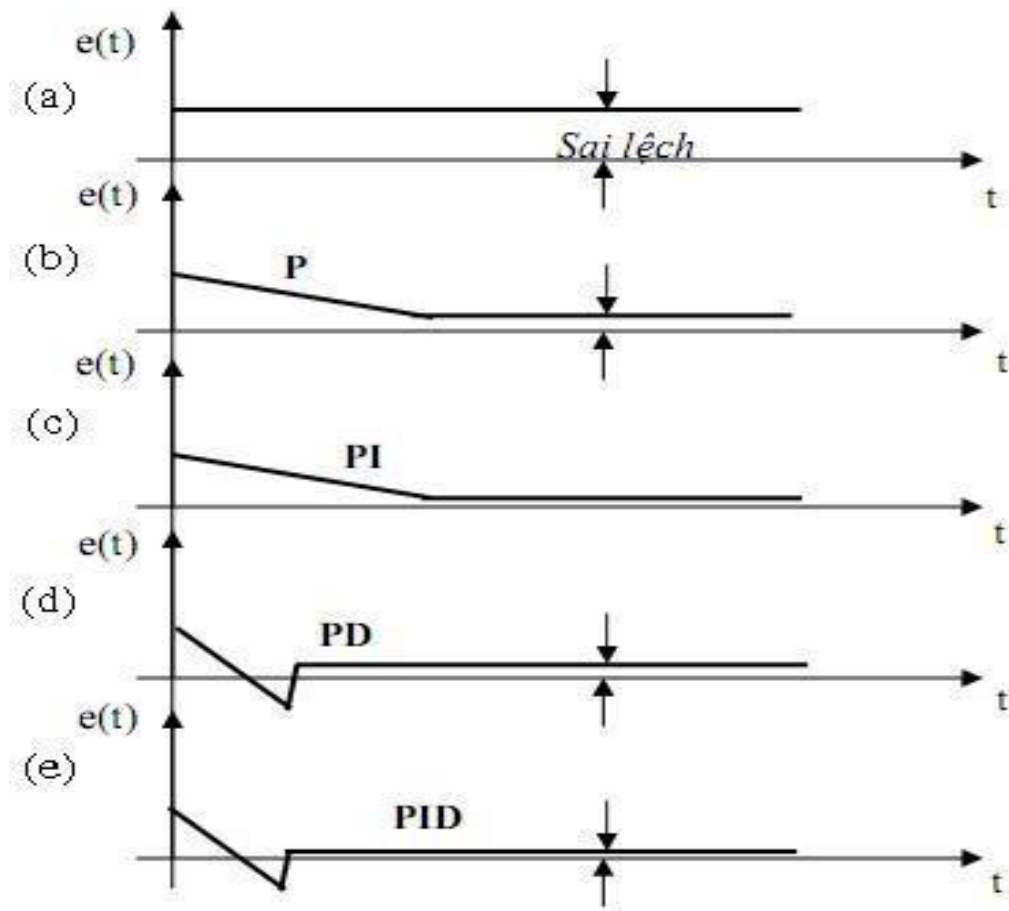
- Hình 1.6.c thể hiện sai lệch điều khiển của quy luật PI. So sánh với hình 2.4.b ta thấy khi kết hợp quy luật tích phân với quy luật tỉ lệ thì hệ có tác



động chậm, không có sai lệch tĩnh.

- Hình 1.6.d thể hiện sai lệch điều khiển của quy luật PD. So với với quy luật PI (hình 1.6.c) ta thấy quy luật PD tác động nhanh hơn, nhưng không làm giảm sai lệch tĩnh.

- Hình 1.6.e thể hiện sai lệch điều khiển của quy luật PID. Quy luật PID có tốc độ tác động nhanh và làm giảm sai lệch tĩnh.



**Hình 1.6:** Minh họa sai lệch điều khiển với các luật điều chỉnh.

### 1.4.3 Quy luật điều chỉnh PID

Rõ ràng việc phối hợp các đặc tính P, I, và D sẽ cho chúng ta khả năng thiết kế được một bộ điều khiển PID phù hợp với các đối tượng cần điều khiển khác nhau.

a. Sử dụng bộ điều khiển PID.

Một vấn đề cần được đặt ra là trong trường hợp nào thì nên dùng bộ điều khiển kiểu P, PI, PD hay PID?

b. Với các đối tượng có đáp ứng nhanh.

Giả sử một bộ điều khiển kiểu PD được dùng để điều khiển cho một đối tượng có đáp ứng nhanh như điều khiển dòng, điều khiển tốc độ động cơ... Nếu vì một lý do nào đó (như tải tăng chẳng hạn) làm cho đầu ra của hệ thống giảm nhanh về một giá trị nào đó thì do sai lệch sau đó gần như không đổi nên khâu D sẽ gần như không có tác dụng.

Trong trường hợp này, nếu thay vì sử dụng bộ điều khiển kiểu PD ta sử dụng một bộ điều khiển kiểu PI thì tín hiệu ra của khâu vi phân sẽ liên tục được cộng dồn làm cho tín hiệu đầu ra của bộ điều khiển ngày càng lớn và có thể đủ để thắng mức độ gia tăng của tải thì sẽ làm tốc độ động cơ tiếp tục tăng trở lại giá trị đặt.

Vì vậy, đối với các đối tượng có đáp ứng nhanh thì sử dụng các bộ điều khiển kiểu PI (có đáp ứng chậm) hoặc PID là phù hợp.

c. Với các đối tượng có đáp ứng chậm. Hiện tượng Windup.

Giả sử một bộ điều khiển kiểu PI được dùng để điều khiển cho một đối tượng có đáp ứng chậm như điều khiển nhiệt độ chẳng hạn. Do đáp ứng chậm nên có thể xảy ra trường hợp sai lệch giữa giá trị đặt so với giá trị thực có thể diễn ra trong thời gian dài. Ví dụ nhiệt độ đặt là  $85^{\circ}\text{C}$ , nhiệt độ hiện tại của đối tượng là  $35^{\circ}\text{C}$  và giả sử phải giữ nhiệt hết công suất thì phải sau 15 phút thì nhiệt độ mới đạt yêu cầu.

Chú ý là tín hiệu ra của khâu  $I_{ui,k} = K_{iek} + u_{i,k-1}$  được cộng dồn liên tục do chu kỳ điều khiển thường rất nhỏ (chỉ cỡ vài chục ms chẳng hạn). Kết quả là tín hiệu ra của khâu I cứ tăng lên mãi, vượt quá khả năng của mạch công suất nhưng do sai lệch vẫn còn lớn nên khâu I vẫn tiếp tục cộng dồn...

Hiện tượng như vậy còn được gọi là Windup.

Vì vậy, với bộ điều khiển có khâu I người ta có thể còn cần phải thiết kế

thêm một phần để chống lại hiện tượng này và được gọi là anti windup.

Trong ví dụ này, nếu thay vì sử dụng bộ điều khiển kiểu PI ta sử dụng một bộ điều khiển kiểu PD thì tín hiệu ra của khâu  $D u_{d,k} = K_d(e_k - e_{k-1})$  sẽ có giá trị không lớn, phù hợp với mức độ gia tăng dần dần của nhiệt độ đầu ra.

Như vậy, đối với các đối tượng có đáp ứng chậm thì sử dụng các bộ điều khiển kiểu PD (có đáp ứng nhanh) là phù hợp.

Nói chung, dựa trên các phân tích ở trên có thể thấy rằng việc sử dụng một bộ điều khiển kiểu PID và chọn được các tham số phù hợp thì có thể đáp ứng được nhu cầu điều khiển cho nhiều loại đối tượng khác nhau.

### **Kết luận:**

Các thành phần P, I, và D trong bộ điều khiển PID (số) có ý nghĩa rất cụ thể và rõ ràng. Trong bài toán điều khiển bám (theo giá trị đặt), thành phần tỷ lệ P phản ứng lại ngay với sai lệch, không cần "nhớ" đáp ứng trước đó như thế nào, nhờ vậy mà nó tạo ra đáp ứng nhanh và kịp thời. Thành phần tích phân I là thành phần "có nhớ", nó lưu lại giá trị điều khiển của vòng lặp trước sau đó điều chỉnh thêm vào hay bớt đi một lượng nào đó (do hệ số  $K_i$  và độ lớn của sai số quyết định) để tạo ra tín hiệu điều khiển cho vòng lặp tiếp theo cho đến khi sai lệch bằng 0. Thành phần D cũng là một thành phần "có nhớ", nó so sánh mức độ chênh lệch của sai lệch ở vòng lặp hiện tại và sai lệch được lưu ở vòng lặp trước đó để đưa ra tín hiệu điều khiển của riêng mình. Khi độ chênh giữa sai lệch của hai chu kỳ điều khiển kế tiếp càng lớn thì tín hiệu điều khiển ra của nó càng lớn (bản chất của đạo hàm). Còn nếu sai lệch của chu kỳ điều khiển hiện tại cũng giống như sai lệch ở chu kỳ điều khiển trước (nghĩa là sai lệch vẫn còn nhưng không thay đổi) thì tín hiệu điều khiển ra của nó bằng 0.

Như vậy, ba thành phần P, I, và D trong một mạch vòng điều khiển cũng tương tự như 7 nốt nhạc trong một bản nhạc. Bằng cách phối hợp các thành phần đó với những tỷ lệ khác nhau chúng ta cũng có thể tạo ra một "bản nhạc"

điều khiển với các "giai điệu" khác nhau. Nếu phối hợp tốt sẽ tạo ra một bản nhạc như mong muốn và êm ái. Còn nếu phối hợp không khéo thì sẽ tạo ra một bản nhạc uốn éo, giật cục với giai điệu khó có thể biết trước được.

## CHƯƠNG 2

### ĐỘNG CƠ ĐIỆN MỘT CHIỀU

#### 2.1 CẤU TẠO PHÂN LOẠI ĐỘNG CƠ ĐIỆN MỘT CHIỀU [4]

##### 2.1.1.1 Cấu tạo của động cơ điện một chiều

Động cơ điện một chiều có thể phân thành hai phần chính: Phần tĩnh và phần động.

- Phần tĩnh hay stato hay còn gọi là phần kích từ động cơ, là bộ phận sinh ra từ trường nó gồm có:

+) Mạch từ và dây cuốn kích từ lồng ngoài mạch từ (nếu động cơ được kích từ bằng nam châm điện), mạch từ được làm bằng sắt từ (thép đúc, thép đặc). Dây quấn kích thích hay còn gọi là dây quấn kích từ được làm bằng dây điện từ, các cuộn dây điện từ nay được mắc nối tiếp với nhau.

+) Cực từ chính: Là bộ phận sinh ra từ trường gồm có lõi sắt cực từ và dây quấn kích từ lồng ngoài lõi sắt cực từ. Lõi sắt cực từ làm bằng những lá thép kỹ thuật điện hay thép cacbon dày 0,5 đến 1mm ép lại và tán chặt. Trong động cơ điện nhỏ có thể dùng thép khối. Cực từ được gắn chặt vào vỏ máy nhờ các bulông. Dây quấn kích từ được quấn bằng dây đồng bọc cách điện và mỗi cuộn dây đều được bọc cách điện kỹ thành một khối, tẩm sơn cách điện trước khi đặt trên các cực từ. Các cuộn dây kích từ được đặt trên các cực từ này được nối tiếp với nhau

+) Cực từ phụ: Cực từ phụ được đặt trên các cực từ chính. Lõi thép của cực từ phụ thường làm bằng thép khối và trên thân cực từ phụ có đặt dây quấn mà cấu tạo giống như dây quấn cực từ chính. Cực từ phụ được gắn vào vỏ máy nhờ những bulông.

+) Gông từ: Gông từ dùng làm mạch từ nối liền các cực từ, đồng thời làm vỏ máy. Trong động cơ điện nhỏ và vừa thường dùng thép dày uốn và hàn lại, trong máy điện lớn thường dùng thép đúc. Có khi trong động cơ điện nhỏ dùng gang làm vỏ máy.

+ ) Các bộ phận khác:

Nắp máy: Để bảo vệ máy khỏi những vật ngoài rơi vào làm hư hỏng dây quấn và an toàn cho người khỏi chạm vào điện. Trong máy điện nhỏ và vừa nắp máy còn có tác dụng làm giá đỡ ổ bi. Trong trường hợp này nắp máy thường làm bằng gang.

Cơ cấu chổi than: Để đưa dòng điện từ phần quay ra ngoài. Cơ cấu chổi than bao gồm có chổi than đặt trong hộp chổi than nhờ một lò xo tì chặt lên cổ góp. Hộp chổi than được cố định trên giá chổi than và cách điện với giá. Giá chổi than có thể quay được để điều chỉnh vị trí chổi than cho đúng chỗ, sau khi điều chỉnh xong thì dùng vít cố định lại.

- Phần quay hay rôto: Bao gồm những bộ phận chính sau.

+ ) Phần sinh ra sức điện động gồm có:

Mạch từ được làm bằng vật liệu sắt từ (lá thép kỹ thuật) xếp lại với nhau.

Trên mạch từ có các rãnh để lồng dây quấn phản ứng.

Cuộn dây phản ứng: Gồm nhiều bó dây nối với nhau theo một qui luật nhất định. Mỗi bó dây gồm nhiều vòng dây các đầu dây của bó dây được nối với các phiến đồng gọi là phiến góp, các phiến góp đó được ghép cách điện với nhau và cách điện với trục gọi là cổ góp hay vành góp.

Tỳ trên cổ góp là cặp trỏ than làm bằng than graphit và được ghép sát vào thành cổ góp nhờ lò xo.

+ ) Lõi sắt phản ứng: Dùng để dẫn từ, thường dùng những tấm thép kỹ thuật điện dày 0,5mm phủ cách điện mỏng ở hai mặt rồi ép chặt lại để giảm tổn hao do dòng điện xoáy gây nên. Trên lá thép có dập hình dạng rãnh để sau khi ép lại thì đặt dây quấn vào. Trong những động cơ trung bình trở lên người ta còn dập những lỗ thông gió để khi ép lại thành lõi sắt có thể tạo được những lỗ thông gió dọc trục. Trong những động cơ điện lớn hơn thì lõi sắt thường chia thành những đoạn nhỏ, giữa những đoạn ấy có để một khe hở gọi là khe hở thông gió. Khi máy làm việc gió thổi qua các khe hở làm nguội dây

quấn và lõi sắt.

Trong động cơ điện một chiều nhỏ, lõi sắt phần ứng được ép trực tiếp vào trục. Trong động cơ điện lớn, giữa trục và lõi sắt có đặt giá rôto. Dùng giá rôto có thể tiết kiệm thép kỹ thuật điện và giảm nhẹ trọng lượng rôto.

+) Dây quấn phần ứng: Dây quấn phần ứng là phần phát sinh ra suất điện động và có dòng điện chạy qua, dây quấn phần ứng thường làm bằng dây đồng có bọc cách điện. Trong máy điện nhỏ có công suất dưới vài Kw thường dùng dây có tiết diện tròn. Trong máy điện vừa và lớn thường dùng dây tiết diện chữ nhật, dây quấn được cách điện cẩn thận với rãnh của lõi thép.

Để tránh khi quay bị văng ra do lực li tâm, ở miệng rãnh có dùng nêm để đè chặt hoặc đai chặt dây quấn. Nêm có thể làm bằng tre, gỗ hay bakelit.

+) Cổ góp: Cổ góp gồm nhiều phiến đồng có được mạ cách điện với nhau bằng lớp mica dày từ 0,4 đến 1,2mm và hợp thành một hình trụ tròn. Hai đầu trụ tròn dùng hai hình ốp hình chữ V ép chặt lại. Giữa vành ốp và trụ tròn cũng cách điện bằng mica. Đuôi vành góp có cao lên một ít để hàn các đầu dây của các phần tử dây quấn và các phiến góp được dễ dàng.

#### **2.1.1.2 Phân loại, ưu nhược điểm của động cơ điện một chiều**

##### **- Phân loại động cơ điện một chiều**

Khi xem xét động cơ điện một chiều cũng như máy phát điện một chiều người ta phân loại theo cách kích thích từ các động cơ. Theo đó ta có 4 loại động cơ điện một chiều thường sử dụng:

+) Động cơ điện một chiều kích từ độc lập: Phần ứng và phần kích từ được cung cấp từ hai nguồn riêng rẽ.

+) Động cơ điện một chiều kích từ song song: Cuộn dây kích từ được mắc song song với phần ứng.

+) Động cơ điện một chiều kích từ nối tiếp: Cuộn dây kích từ được mắc nối tiếp với phần ứng.

+) Động cơ điện một chiều kích từ hỗn hợp: Gồm có 2 cuộn dây kích từ,

một cuộn mắc song song với phần ứng và một cuộn mắc nối tiếp với phần ứng.

- Ưu nhược điểm của động cơ điện một chiều

Do tính ưu việt của hệ thống điện xoay chiều: dễ sản xuất, dễ truyền tải..., cả máy phát và động cơ điện xoay chiều đều có cấu tạo đơn giản và công suất lớn, dễ vận hành... mà máy điện (động cơ điện) xoay chiều ngày càng được sử dụng rộng rãi và phổ biến. Tuy nhiên động cơ điện một chiều vẫn giữ một vị trí

nhất định trong công nghiệp giao thông vận tải, và nói chung ở các thiết bị cần điều khiển tốc độ quay liên tục trong phạm vi rộng (như trong máy cán thép, máy công cụ lớn, đầu máy điện...). Mặc dù so với động cơ không đồng bộ để chế tạo động cơ điện một chiều cùng cỡ thì giá thành đắt hơn do sử dụng nhiều kim loại màu hơn, chế tạo bảo quản cở góp phức tạp hơn. Nhưng do những ưu điểm của nó mà máy điện một chiều vẫn không thể thiếu trong nền sản xuất hiện đại.

+) Ưu điểm của động cơ điện một chiều là có thể dùng làm động cơ điện hay máy phát điện trong những điều kiện làm việc khác nhau. Song ưu điểm lớn nhất của động cơ điện một chiều là điều chỉnh tốc độ và khả năng quá tải. Nếu như bản thân động cơ không đồng bộ không thể đáp ứng được hoặc nếu đáp ứng được thì phải chi phí các thiết bị biến đổi đi kèm (như bộ biến tần....) rất đắt tiền thì động cơ điện một chiều không những có thể điều chỉnh rộng và chính xác mà cấu trúc mạch lực, mạch điều khiển đơn giản hơn đồng thời lại đạt chất lượng cao.

+) Nhược điểm chủ yếu của động cơ điện một chiều là có hệ thống cổ góp - chổi than nên vận hành kém tin cậy và không an toàn trong các môi trường rung chấn, dễ cháy nổ.

## **2.1.2 Đặc tính cơ của động cơ điện một chiều [4]**

### **2.1.2.1 Nguyên lý làm việc của động cơ điện một chiều**

Khi cho điện áp một chiều vào, trong dây quấn phần ứng có điện. Các

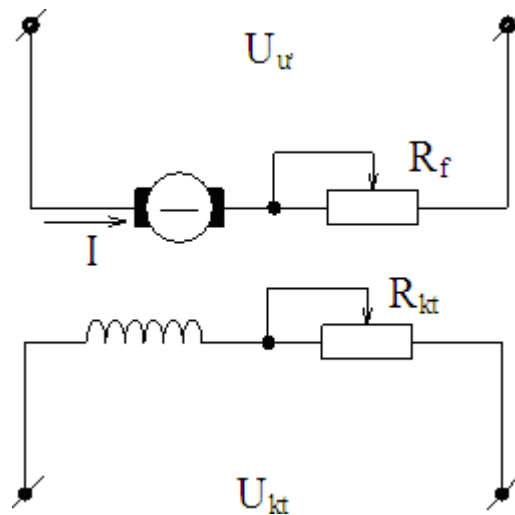


thanh dẫn có dòng điện nằm trong từ trường sẽ chịu lực tác dụng làm rôto quay, chiều của lực được xác định bằng quy tắc bàn tay trái.

Khi phần ứng quay được nửa vòng, vị trí các thanh dẫn đổi chỗ cho nhau. Do có phiều góp chiều dòng điện dũ nguyên làm cho chiều lực từ tác dụng không thay đổi. Khi quay, các thanh dẫn cắt từ trường sẽ cảm ứng với suất điện động  $E_r$  chiều của suất điện động được xác định theo quy tắc bàn tay phải, ở động cơ chiều sđđ  $E_r$  ngược chiều dòng điện  $I_r$  nên  $E_r$  được gọi là sức phản điện động. Khi đó ta có phương trình:  $U = E_r + R_r \cdot I_r$

#### 2.1.2.2 Đặc tính cơ của động cơ điện một chiều kích từ độc lập

Khi nguồn một chiều có công suất không đủ lớn thì mạch điện phần ứng và mạch điện kích từ mắc vào hai nguồn độc lập nhau. Lúc này động cơ được gọi là động cơ điện một chiều kích từ độc lập[2].



**Hình 2.1:** Sơ đồ nối dây của động cơ điện một chiều kích từ độc lập

Ta có phương trình cân bằng điện áp của mạch phần ứng như sau:

$$U_r = E_r + (R_r + R_f)I_r \quad (2.1)$$

Trong đó:

$U_r$ : Điện áp phần ứng, V

$E_r$ : Sức điện động phần ứng, V

$R_u$ : Điện trở mạch phản ứng,  $\Omega$

$I_u$ : Dòng điện của mạch phản ứng, A

Với:  $R_u = r_u + r_{cf} + r_b + r_{ct}$

$r_u$ : Điện trở cuộn dây phản ứng

$r_{cf}$ : Điện trở cuộn dây cực từ phụ

$r_{ct}$ : Điện trở tiếp xúc cuộn bù

Sức điện động  $E_u$  của phản ứng động cơ được xác định theo biểu thức:

$$E = \frac{P.N}{2\pi a} \cdot \Phi \omega = K \cdot \Phi \cdot \omega \quad (2.2)$$

Trong đó:

P: Số đôi cực từ chính

N: Số thanh dẫn tác dụng của cuộn dây phản ứng

a: Số đôi mạch nhánh song song của cuộn dây phản ứng

$\omega$ : Tốc độ góc (rad/s)

$\Phi$ : Từ thông kích từ dưới một cực từ

$K = \frac{P.N}{2\pi a}$  : Hệ số cấu tạo của động cơ

Từ (2.1) và (2.2) ta có:

$$\omega = \frac{U}{K \cdot \Phi} - \frac{R_u + R_f}{K \cdot \Phi} \quad (2.3)$$

Biểu thức trên là phương trình đặc tính cơ điện của động cơ.

Mặt khác, mô men điện từ  $M_{dt}$  của động cơ được xác định bởi

$$M_{dt} = K\Phi \cdot I_u \quad (2.4)$$

Với  $I = \frac{M_{dt}}{K \cdot \Phi}$  : thay giá trị I vào (2.3) ta có

$$\omega = \frac{U}{K \cdot \Phi} - \frac{R_u + R_f}{(K \cdot \Phi)^2} \cdot M_{dt} \quad (2.5)$$

Nếu bỏ qua tổn thất cơ và tổn thất thép thì mômen cơ trên trục động cơ

bằng mô men điện từ, ta ký hiệu là  $M$ . Nghĩa là:  $M_{dt} = M_{co} = M$

$$\omega = \frac{U_u}{K \cdot \Phi} - \frac{R_u + R_f}{(K \cdot \Phi)^2} \cdot M \quad (2.6)$$

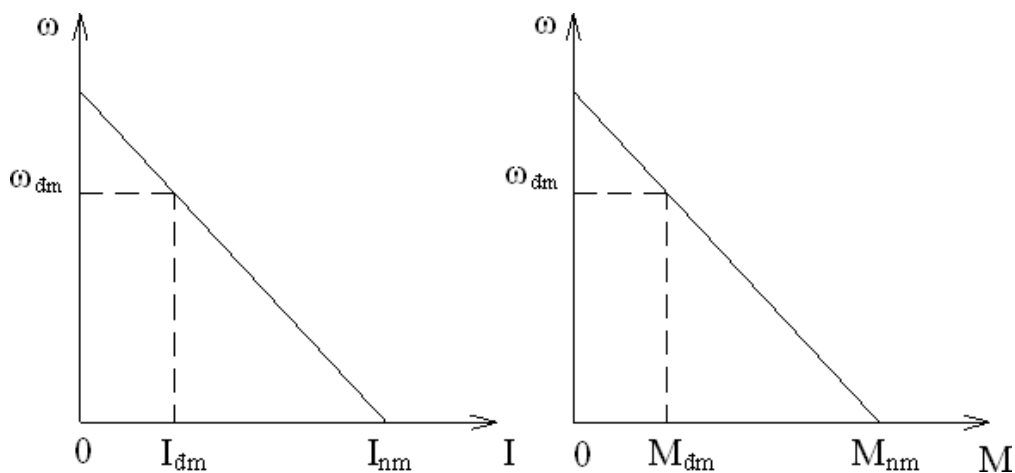
Đây là phương trình đặc tính cơ của động cơ điện một chiều kích từ độc lập.

Giả thiết phản ứng được bù đủ, từ thông  $\Phi = \text{const}$ , thì các phương trình đặc tính cơ điện (2.3) và phương trình đặc tính cơ (2.6) là tuyến tính. Đồ thị của chúng được biểu diễn trên hình 1.2 là những đường thẳng.

Theo các đồ thị, khi  $I_r = 0$  hoặc  $M = 0$  ta có:

$$\omega = \frac{U}{K \cdot \Phi} = \omega_0$$

$\omega_0$  được gọi là tốc độ không tải lý tưởng của động cơ điện một chiều kích từ độc lập.



**Hình 2.2:** Đặc tính cơ điện và đặc tính cơ của động cơ điện một chiều

Khi  $\omega = 0$  ta có:

$$I = \frac{U}{R_u + R_f} = I_{nm} \quad (2.7)$$

$$M = K \cdot \phi \cdot I_{nm} = M_{nm} \quad (2.8)$$

$I_{nm}$  và  $M_{nm}$  được gọi là dòng điện ngắn mạch và mô men ngắn mạch.

## 2.2 CÁC PHƯƠNG PHÁP ĐIỀU KHIỂN TỐC ĐỘ ĐỘNG CƠ MỘT CHIỀU

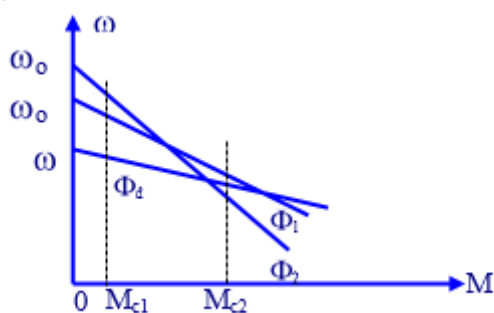
- Phương pháp thay đổi điện trở phản ứng
- Phương pháp thay đổi từ thông  $\Phi$
- Phương pháp thay đổi điện áp phản ứng

### 2.2.1 Phương pháp thay đổi điện trở phản ứng

- Nguyên lý điều khiển:

Giả thiết  $U = U_{dm}$ ,  $R_r = \text{const}$ . Muốn thay đổi từ thông động cơ ta thay đổi dòng điện kích từ, thay đổi dòng điện trong mạch kích từ bằng cách nối nối tiếp biến trở vào mạch kích từ hay thay đổi điện áp cấp cho mạch kích từ.

Bình thường khi động cơ làm việc ở chế độ định mức với kích thích tối đa ( $\Phi = \Phi_{max}$ ) mà phương pháp này chỉ cho phép tăng điện trở vào mạch kích từ nên chỉ có thể điều chỉnh theo hướng giảm từ thông  $\Phi$  tức là điều chỉnh tốc độ trong vùng trên tốc độ định mức. Nên khi giảm  $\Phi$  thì tốc độ không tri lí tưởng  $\omega_0$  tăng, còn độ cứng đặc tính cơ giảm, ta thu được họ đặc tính cơ nằm trên đặc tính cơ tự nhiên



**Hình 2.3:** Đặc tính cơ của động cơ khi giảm từ thông

- Khi tăng tốc độ động cơ bằng cách giảm từ thông thì dòng điện tăng và tăng vượt quá mức giá trị cho phép nếu mômen không đổi. Vì vậy muốn giữ cho dòng điện không vượt quá giá trị cho phép đồng thời với việc giảm từ

thông thì ta phải giảm  $M_t$  theo cùng tỉ lệ.

- Đặc điểm của phương pháp:

+ ) Phương pháp này có thể thay đổi tốc độ về phía tăng.

+ ) Phương pháp này chỉ điều khiển ở vùng tải không quá lớn so với định mức, việc thay đổi từ thông không làm thay đổi dòng điện ngắn mạch.

+ ) Việc điều chỉnh tốc độ bằng cách thay đổi từ thông là phương pháp điều khiển với công suất không đổi.

+ ) Đánh giá các chỉ tiêu điều khiển: Sai số tốc độ lớn, đặc tính điều khiển nằm trên và dốc hơn đặc tính tự nhiên. Dải điều khiển phụ thuộc vào phần cơ của máy. Có thể điều khiển trơn trong dải điều chỉnh  $D = 3 : 1$ . Vì công suất của cuộn dây kích từ bé, dòng điện kích từ nhỏ nên ta có thể điều khiển liên tục với

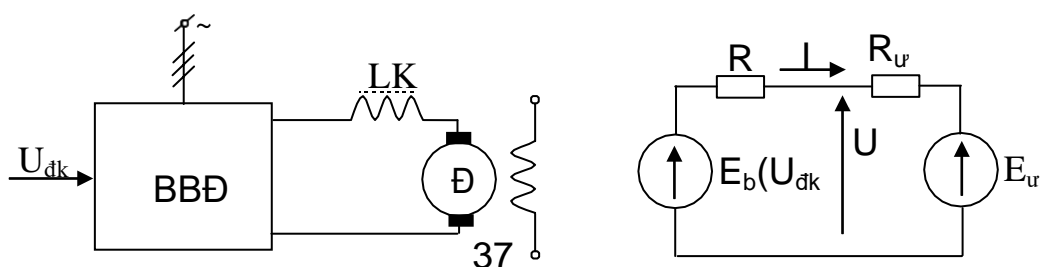
$$\Phi \approx 1.$$

+ ) Phương pháp này được áp dụng tương đối phổ biến, có thể thay đổi liên tục và kinh tế ( vì việc điều chỉnh tốc độ thực hiện ở mạch kích từ với dòng kích từ  $(1 \div 10)\% I_{dm}$  của phần ứng nên tổn hao điều chỉnh thấp).

Đây là phương pháp gần như là duy nhất đối với động cơ điện một chiều khi cần điều chỉnh tốc độ lớn hơn tốc độ điều khiển.

### 2.2.2 Phương pháp thay đổi điện áp phản ứng

- Để điều chỉnh điện áp phản ứng động cơ một chiều cần có thiết bị nguồn như máy phát điện một chiều kích từ độc lập, các bộ chỉnh lưu điều khiển ... Các thiết bị nguồn này có chức năng biến năng lượng điện xoay chiều thành một chiều có sức điện động  $E_b$  điều chỉnh nhờ tín hiệu điều khiển  $U_{dk}$ . Vì nguồn có công suất hữu hạn so với động cơ nên các bộ biến đổi này có điện trở trong  $R_b$  và điện cảm  $L_b$  khác không. Để đưa tốc độ động cơ với hiệu suất cao trong giới hạn rộng rãi 1:10 hoặc hơn nữa[3].



**Hình 2.4:** Sơ đồ dùng bộ biến đổi điều khiển điện áp phản ứng

Ở chế độ xác lập có thể viết được phương trình đặc tính của hệ thống như sau:

$$E_b - E_{ur} = I_r(R_b + R_{ud}) \quad (2.9)$$

$$\omega = \frac{E}{K_{\varphi_d}} - \frac{R_b + R_{ud}}{K_{\varphi} \cdot I} u \quad (2.10)$$

$$\omega = \omega_o U_{dk} - \frac{M}{|\beta|} \quad (2.11)$$

- Vì từ thông của động cơ được giữ không đổi nên độ cứng đặc tính cơ cũng không đổi, còn tốc độ không tải lý tưởng thì tùy thuộc vào giá trị điện áp điều khiển  $U_{dk}$  của hệ thống, do đó có thể nói phương pháp điều chỉnh này là triệt để.

Để xác định giải điều chỉnh tốc độ ta để ý rằng tốc độ lớn nhất của hệ thống bị chặn bởi đặc tính cơ cơ bản, là đặc tính ứng với điện áp phản ứng định mức và từ thông cũng được giữ ở giá trị định mức. Tốc độ nhỏ nhất của dải điều chỉnh bị giới hạn bởi yêu cầu về sai số tốc độ và về mômen khởi động. Khi mômen tải là định mức thì các giá trị lớn nhất và nhỏ nhất của tốc độ là:

$$\omega_{\max} = \omega_{o \max} - \frac{M_{dm}}{|\beta|} \quad (2.12)$$

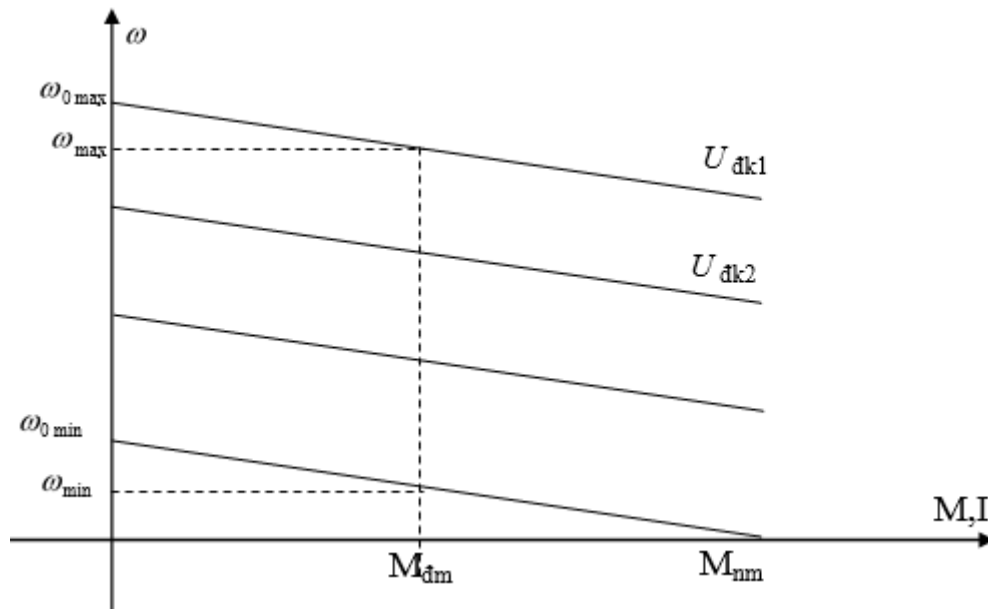
$$\omega_{\min} = \omega_{o \min} - \frac{M_{dm}}{|\beta|} \quad (2.13)$$

Để thoả mãn khả năng quá tải thì đặc tính thấp nhất của dải điều chỉnh phải có mômen ngắn mạch là:  $M_{nmmin} = M_{cmax} = K_M \cdot M_{dm}$

Trong đó  $K_M$  là hệ số quá tải về mômen. Vì họ đặc tính cơ là các đường thẳng song song nhau, nên theo định nghĩa về độ cứng đặc tính cơ có thể viết

$$\omega_{\min} = M_{nm} - M \quad \frac{1}{|\beta|} = \frac{M_d}{|\beta|} K_M - 1$$

(2.14)



**Hình 2.5:** Đặc tính cơ của động cơ khi thay đổi điện áp

- Với một cơ cấu máy cụ thể thì các giá trị  $\omega_{0\max}$ ,  $M_{\text{đm}}$ ,  $K_M$  là xác định, vì vậy phạm vi điều chỉnh  $D$  phụ thuộc tuyến tính vào giá trị của độ cứng  $\beta$ . Khi điều chỉnh điện áp phản ứng động cơ bằng các thiết bị nguồn điều chỉnh thì điện trở tổng mạch phản ứng gấp khoảng hai lần điện trở phản ứng động cơ. Do đó có thể tính sơ bộ được.

Vì thế tải có đặc tính mômen không đổi thì giá trị phạm vi điều chỉnh tốc độ cũng không vượt quá 10. Đối với các máy có yêu cầu cao về dải điều chỉnh và độ chính xác duy trì tốc độ làm việc thì việc sử dụng các hệ thống hờ như trên là không thoả mãn được.

- Trong phạm vi phụ tải cho phép có thể coi đặc tính cơ tĩnh của hệ truyền động một chiều kích từ độc lập là tuyến tính. Khi điều chỉnh điện áp phản ứng thì độ cứng có đặc tính cơ trong toàn dải là như nhau, do đó độ sụt tốc tương

đôi sẽ đạt giá trị lớn nhất tại đặc tính thấp nhất của dải điều chỉnh. Hay nói cách khác, nếu tại đặc tính cơ thấp nhất của dải điều chỉnh mà sai số tốc độ không vượt quá giá trị sai số cho phép, thì hệ truyền động sẽ làm việc với sai số luôn nhỏ hơn sai số cho phép trong toàn bộ dải điều chỉnh. Sai số tương đối của tốc độ ở đặc tính cơ thấp nhất là:

$$s = \frac{\omega_{o\min} - \omega_{\min}}{\omega_{o\min}} = \frac{\Delta\omega}{\omega_{o\min}} \quad (2.15)$$

$$s = \frac{M_{dm}}{|\beta| \cdot \omega_{o\min}} \leq s_{cp} \quad (2.16)$$

- **Nhận xét:** Cả 3 phương pháp trên đều điều chỉnh được tốc độ động cơ điện một chiều nhưng chỉ có phương pháp điều chỉnh tốc độ động cơ điện một chiều bằng cách thay đổi điện áp  $U_r$  đặt vào phần ứng của động cơ là tốt nhất và hay được sử dụng nhất vì nó thu được đặc tính cơ có độ cứng không đổi, điều chỉnh tốc độ bằng phẳng và không bị tổn hao.

## 2.3 GIỚI THIỆU MỘT SỐ HỆ TRUYỀN ĐỘNG ĐIỆN MỘT CHIỀU

- Hệ truyền động máy phát - động cơ một chiều (F - Đ)
- Hệ truyền động xung áp - động cơ (XA – ĐC)
- Hệ truyền động chỉnh lưu - động cơ (CL - ĐC)

### 2.3.1 Hệ truyền động máy phát - động cơ điện một chiều (F - Đ)

- Cấu trúc hệ F - Đ và đặc tính cơ bản:

Hệ thống máy phát - động cơ (F - Đ) là hệ truyền động điện mà bộ biến đổi điện là máy phát điện một chiều kích từ độc lập. Máy phát này thường do động cơ sơ cấp không đồng bộ ba pha kéo quay[3].

Tính chất của máy phát điện được xác định bởi hai đặc tính: Đặc tính từ hoá là sự phụ thuộc giữa sức điện động máy phát vào dòng điện kích từ và đặc tính tải là sự phụ thuộc của điện áp trên hai cực của máy phát vào dòng điện



tải. Các đặc tính này nói chung là phi tuyến do tính chất của lõi sắt, do các phản ứng của dòng điện phản ứng ... trong tính toán gần đúng có thể tuyến tính hoá các đặc tính này:

$$E_F = K_F \cdot \phi_F \cdot \omega_F = K_F \cdot \omega_F \cdot C \cdot i_{KF} \quad (2.17)$$

Trong đó:

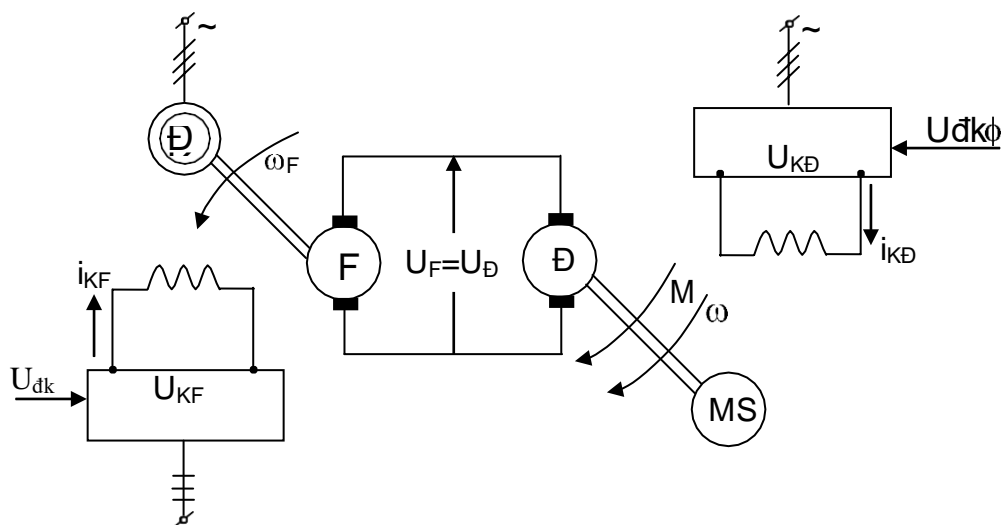
$K_F$ : là hệ số kết cấu của máy phát

$C = \Delta \phi_F / \Delta i_{KF}$  là hệ số góc của đặc tính từ hoá.

Nếu dây quấn kích thích của máy phát được cấp bởi nguồn áp lý tưởng  $U_{KF}$  thì:  $I_{KF} = U_{KF} / r_{KF}$

Sức điện động của máy phát trong trường hợp này sẽ tỷ lệ với điện áp kích thích bởi hệ số hằng  $K_F$  như vậy có thể coi gần đúng máy phát điện một chiều kích từ độc lập là một bộ khuếch đại tuyến tính:

$$E_F = K_F \cdot U_{KF}$$



**Hình 2.6 :** Sơ đồ nguyên lý hệ truyền động máy phát động cơ

Nếu đặt  $R = R_{uF} + R_{uĐ}$  thì có thể viết được phương trình các đặc tính của hệ F - Đ như sau:

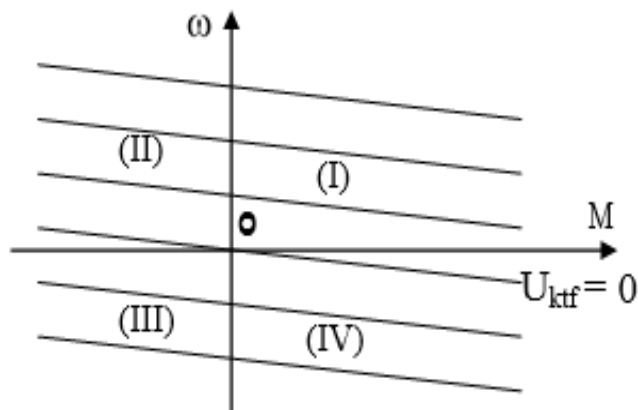
$$\omega = \frac{K_F}{K\Phi} U_{KF} - \frac{RI}{K\Phi} \quad (2.18)$$

$$\omega = \frac{K_F}{K\Phi} U_{KF} - \frac{R}{K\Phi^2} M \quad (2.19)$$

$$\omega = \omega_o \frac{U_{KF}}{U_{KD}} - \frac{M}{\beta U_{KD}} \quad (2.20)$$

Các biểu thức trên chứng tỏ rằng, khi điều chỉnh dòng điện kích thích của máy phát thì điều chỉnh được tốc độ không tải của hệ thống còn độ cứng đặc tính cơ thì giữ nguyên. Cũng có thể điều chỉnh kích từ của động cơ để có dải điều chỉnh tốc độ rộng hơn.

- Các chế độ làm việc của hệ F - Đ



**Hình 2.7:** Các trạng thái làm việc của hệ F - Đ

Trong hệ F - Đ không có phần tử phi tuyến nào nên hệ có những đặc tính động rất tốt, rất linh hoạt khi chuyển các trạng thái làm việc. Với sơ đồ cơ bản như hình 2.6 động cơ chấp hành Đ có thể làm việc ở chế độ điều chỉnh được cả hai phía: Kích thích máy phát F và kích thích động cơ Đ, đảo chiều quay bằng cách đảo chiều dòng kích thích máy phát, hãm động năng khi dòng kích thích máy phát bằng không, hãm tái sinh khi giảm tốc độ hoặc khi đảo chiều dòng kích từ, hãm ngược ở cuối giai đoạn hãm tái sinh khi đảo chiều hoặc khi làm

việc ổn định với mômen tải có tính chất thế năng ... hệ F - Đ có đặc tính cơ ở cả bốn góc phần tư của mặt phẳng toạ độ

+) Ở góc phần tư thứ I và thứ III tốc độ quay và mômen quay của động cơ luôn cùng chiều nhau, sức điện động máy phát và động cơ có chiều đối nhau và  $|E_F| > |E|$ ,  $|\omega_c| > |\omega|$ . Công suất điện từ của máy phát và động cơ là:

$$P_F = E_F \cdot I > 0$$

$$P_D = E \cdot I < 0$$

$$P_{cơ} = M \cdot \omega > 0$$

Các biểu thức này nói lên rằng năng lượng được vận chuyển thuận chiều từ nguồn  $\rightarrow$  máy phát  $\rightarrow$  động cơ  $\rightarrow$  tải.

+) Vùng hãm tái sinh nằm ở góc phần tư thứ II và thứ IV, lúc này do  $|\omega| > |\omega_o|$  nên  $\parallel > |E_F|$ , mặc dù E,  $E_F$  mắc ngược nhau, nhưng dòng điện phản ứng lại chạy ngược từ động cơ về máy phát làm cho mômen quay ngược chiều tốc độ quay. Công suất điện từ của máy phát, công suất điện từ và công suất cơ học của động cơ là:

$$P_F = E_F \cdot I < 0$$

$$P_D = E \cdot I > 0$$

$$P_{cơ} = M \cdot \omega < 0$$

Chỉ do dòng điện đổi chiều mà các bất đẳng thức trên cho ta thấy năng lượng được chuyển vận theo chiều từ tải  $\rightarrow$  động cơ  $\rightarrow$  máy phát  $\rightarrow$  nguồn, máy phát F và động cơ Đ đổi chức năng cho nhau. Hãm tái sinh trong hệ F - Đ được khai thác triệt để khi giảm tốc độ, khi hãm để đảo chiều quay và khi làm việc ổn định với tải có tính chất thế năng.

- Đặc điểm của hệ F - Đ:

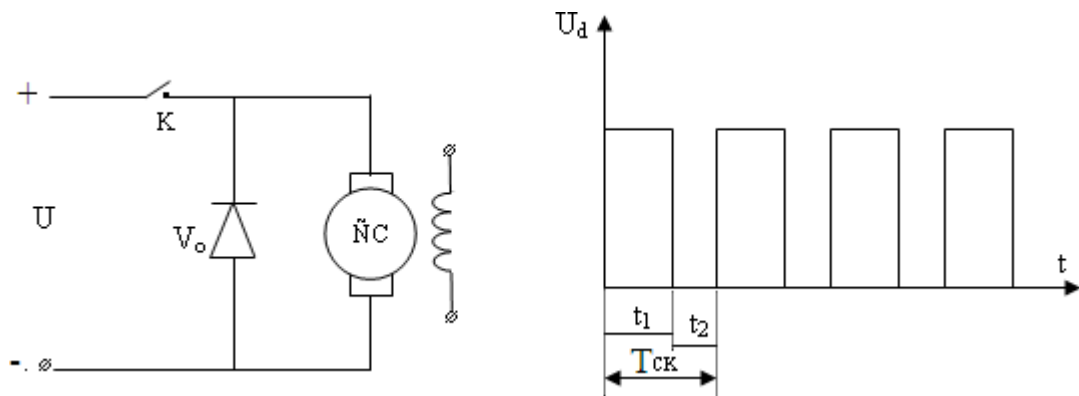
+) Các chỉ tiêu chất lượng của hệ F - Đ về cơ bản tương tự các chỉ tiêu của hệ điều áp dùng bộ biến đổi nói chung. Ưu điểm nổi bật của hệ F - Đ là sự chuyển đổi trạng thái làm việc rất linh hoạt, khả năng chịu quá tải lớn, do vậy

thường sử dụng hệ truyền động F - Đ ở các máy khai thác trong công nghiệp mỏ.

+) Nhược điểm quan trọng nhất của hệ F - Đ là dùng nhiều máy điện quay, trong đó ít nhất là hai máy điện một chiều, gây ồn lớn, công suất lắp đặt máy ít nhất gấp ba lần công suất động cơ chấp hành. Ngoài ra do các máy phát một chiều có từ dư, đặc tính từ hoá có trễ nên khó điều chỉnh sâu tốc độ.

### 2.3.2 Hệ truyền động xung áp – động cơ (XA – ĐC)

Bộ biến đổi xung áp là một nguồn điện áp dùng điều chỉnh tốc độ động cơ điện một chiều[3].



**Hình 2.8:** Sơ đồ nguyên lý và giản đồ xung

Để cải thiện dạng sóng của dòng điện phản ứng ta thêm vào mạch một van chỉnh  $V_0$ . Có thể sử dụng thyristor hoặc transistor công suất để thay cho khóa K ở trên. Khi đóng cắt khóa K, trên phần ứng động cơ sẽ có điện áp biến đổi theo dạng xung vuông. Khi ở trạng thái dòng liên tục thì giá trị trung bình của điện áp ra sẽ là:

$$U_d = \frac{1}{T_{CK}} \int_0^{t_1} U dt = \frac{t_1}{T_{CK}} U = \gamma \cdot U \quad (2.21)$$

Trong đó:

$t_1$  : Là thời gian khóa ở trạng thái đóng

$t_2$  : Là thời gian khóa ở trạng thái mở

$T_{ck}$  : Thời gian thực hiện một chu kỳ đóng mở khóa

$$\gamma = \frac{t_1}{T_C K} : \text{Là độ rộng của xung áp}$$

Vậy ta có thể coi bộ biến đổi xung đẳng trị với nguồn liên tục có điện áp ra  $U_d$  và  $U_d$  có thể thay đổi được bằng cách thay đổi độ rộng xung. Mặt khác, thời gian một chu kỳ đóng cắt của khóa  $K$  rất nhỏ so với hằng số thời gian cơ học của hệ truyền động, nên ta coi tốc độ và sức điện động phản ứng động cơ không thay đổi trong thời gian  $T_{ck}$ .

- Đặc tính điều chỉnh của hệ XA - ĐC

$$\omega = \frac{\gamma \cdot U}{K \cdot \Phi_{dm}} - \frac{R_b + R_b}{K \cdot \Phi_{dm}} \cdot I \quad (2.22)$$

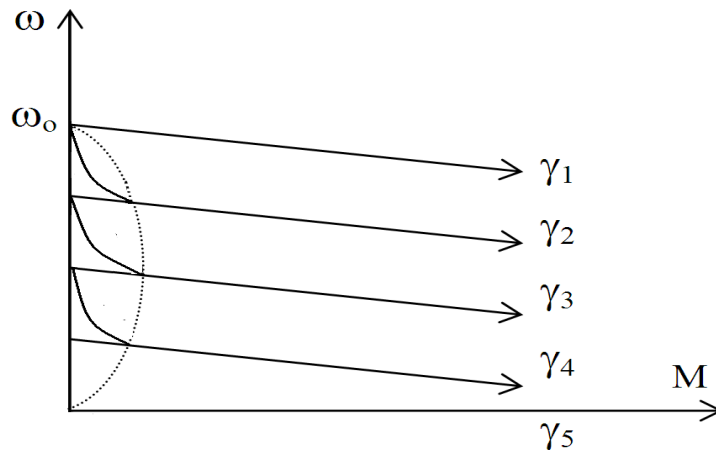
$$\theta = \frac{\gamma \cdot U}{K \cdot \Phi_{dm}} - \frac{R_b + R_b}{(K \cdot \Phi_{dm})^2} \cdot M \quad (2.23)$$

Khi thay đổi  $\gamma$  ta được họ đường thẳng song song có độ cứng  $\beta = \text{const}$  và tốc độ không tải lý tưởng  $\omega_0$  thay đổi theo  $\gamma$ . Nếu nguồn vô cùng lớn thì ta có thể bỏ qua  $R_{bd}$ , khi đó độ cứng của đặc tính cơ của hệ có độ cứng là:

$$\beta = \beta_{TN} = \frac{(K \cdot \Phi_{dm})^2}{R_b} = \text{const} \quad (2.24)$$

Tốc độ không tải lý tưởng  $\omega_0$  phụ thuộc vào  $\gamma$  chỉ là giá trị giả định. Nó có thể tồn tại nếu như dòng trong hệ là liên tục kể cả khi giá trị dòng tiến đến 0. Vì vậy hai biểu thức trên chỉ đúng với trạng thái dòng liên tục.

Khi dòng điện đủ nhỏ thì hệ sẽ chuyển trạng thái từ dòng liên tục sang trạng thái dòng gián đoạn. Khi đó các phương trình đặc tính điều chỉnh nói trên không còn đúng nữa mà lúc này đặc tính của hệ là những đường cong rất dốc.



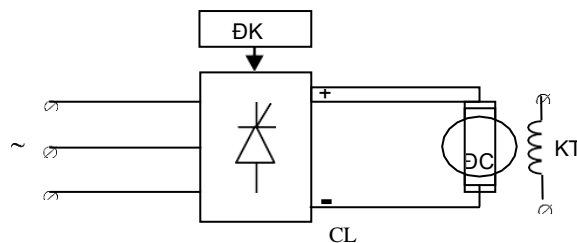
**Hình 2.9:** Đặc tính cơ của hệ

- Nhận xét:

- +) Tất cả đặc tính điều chỉnh của hệ XA – ĐC khi dòng điện gián đoạn đều có chung một giá trị không tải lý tưởng, chỉ ngoại trừ trường hợp  $= 0$ .
- +) Bộ nguồn xung áp cần ít van dẫn nên vốn đầu tư ít, hệ đơn giản chắc chắn.
- +) Độ cứng của đặc tính cơ lớn.
- +) Điện áp dạng xung nên gây ra tổn thất phụ khá lớn trong động cơ. Khi làm việc ở trạng thái dòng điện gián đoạn thì đặc tính làm việc kém ổn định và tổn thất năng lượng nhiều.

### 2.3.3 Hệ truyền động chỉnh lưu – động cơ điện một chiều (CL – ĐC)

- Sơ đồ nguyên lý:



**Hình 2.10:** Sơ đồ nguyên lý của hệ chỉnh lưu - động cơ điện một chiều

Hệ truyền động chỉnh lưu có điều khiển - động cơ điện một chiều (CL - ĐC) có bộ biến đổi là các mạch chỉnh lưu có điều khiển, có sức điện động  $E_d$  phụ thuộc vào giá trị của xung điều khiển ( tức là phụ thuộc vào góc điều khiển hay góc mở Tiristor )[3].

Điện áp chỉnh lưu  $U_d$  ( hay  $E_d$ ) là điện áp không tải ở đầu ra, có dạng đập mạch với số lần đập mạch là  $n$  trong một chu kì  $2\pi$  của điện áp thứ cấp máy biến áp.

+) Với sơ đồ chỉnh lưu tia:  $n = m$ , trong đó  $m$  là số pha

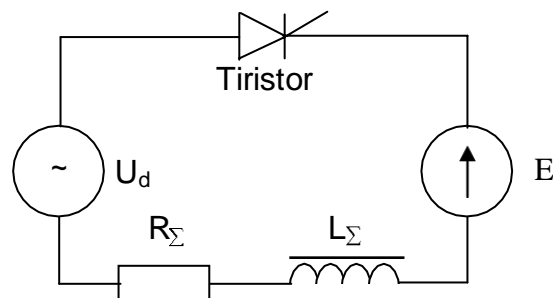
+) Với sơ đồ hình cầu:  $n = 2.m$ , trong đó  $m$  là số pha

Giả sử điện áp thứ cấp của máy biến áp có dạng hình sin với biểu thức:

$$u_2 = U_{2m} \cdot \sin \omega t = U_{2m} \cdot \sin \theta, \quad (\text{với } \theta = \omega t) \quad (2.25)$$

Trong khoảng  $\Theta = (0 - 2\pi)$  thì dạng điện áp và dòng điện lặp lại như chu kì ban đầu nên ta chỉ cần xét trong một chu kì  $T = 2\pi$

- Sơ đồ thay thế của hệ CL – ĐC



**Hình 2.11:** Sơ đồ thay thế của hệ chỉnh lưu - động cơ điện một chiều

Khi van dẫn thì ta có phương trình cân bằng điện áp như sau:

$$U_d \cdot \sin \theta - E = i_d \cdot R_\Sigma + L_\Sigma \frac{di_d}{dt} \quad (2.26)$$

$$\text{Suy ra: } U_{2m} \cdot \sin \theta - E = i_d \cdot R_\Sigma + L_\Sigma \cdot \frac{di_d}{dt} \quad (2.27)$$

Trong đó:

$$R_\Sigma = R_{ba} + R_u + R_k$$

$$L_\Sigma = L_{ba} + L_u + L_k$$

- Trạng thái dòng liên tục

Ở trạng thái dòng liên tục, khi van này chưa khóa thì van kế tiếp đã mở, việc mở van kế tiếp là điều kiện cần để khóa van đang dẫn. Do vậy, điện áp của chỉnh lưu sẽ có dạng đường bao của điện áp thứ cấp máy biến áp.

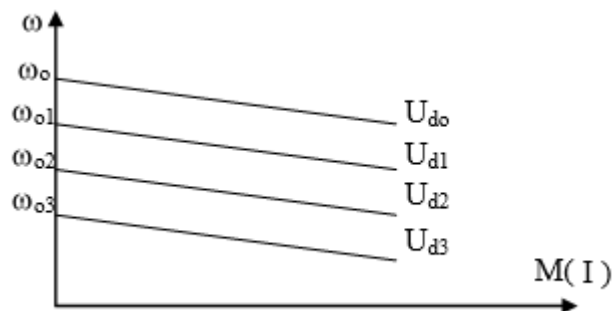
Giá trị trung bình của điện áp chỉnh lưu:

$$U_d = \frac{n}{2\pi} \int_{\alpha}^{\alpha + \frac{2\pi}{n}} u_2 \cdot dt = \frac{n}{\pi} \int_{\alpha}^{\alpha + \frac{2\pi}{n}} U_{2m} \cdot \sin\theta \cdot d\theta \quad (2.28)$$

$$= \frac{n}{\pi} \cdot \sin \frac{\pi}{n} \cdot U_{2m} \cdot \cos \alpha = U_{do} \alpha$$

+) Độ cứng của đặc tính cơ:

$$\beta = \frac{dM}{d\omega} \approx \frac{\Delta M}{\Delta \omega} = \frac{(K \cdot \Phi_{drr})^2}{R_{\Sigma}} \quad (2.29)$$

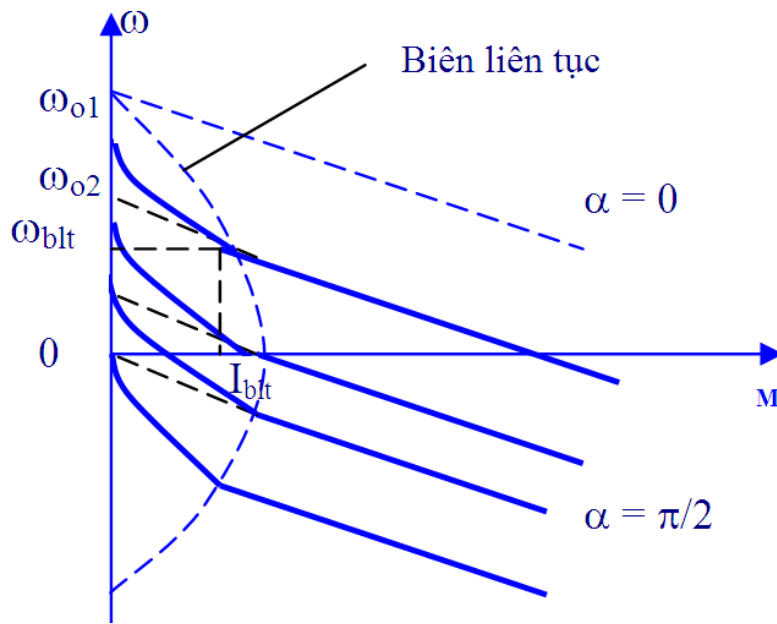


**Hình 2.12:** Đặc tính cơ của hệ chỉnh lưu - động cơ một chiều khi dòng liên tục

Khi điện kháng trong mạch không đủ lớn, nếu sức điện động của động cơ đủ lớn thì dòng điện tải sẽ trở thành gián đoạn. Ở trạng thái này thì dòng qua van bất kì sẽ bằng 0 trước khi van kế tiếp mở. Do vậy trong một khoảng dẫn của van thì sức điện động của chỉnh lưu bằng sức điện động nguồn:  $e_d = U_2$



Đặc tính cơ của hệ CL - ĐC khi dòng điện gián đoạn:



**Hình 2.13:** Đặc tính cơ của hệ chỉnh lưu - động cơ khi dòng gián đoạn

- Nhận xét:

+) Ưu điểm: Hệ truyền động chỉnh lưu - động cơ có độ tác động nhanh cao, không gây ồn và dễ tự động hóa, do các van bán dẫn có hệ số khuếch đại công suất rất cao, vì vậy rất thuận tiện cho việc thiết lập hệ thống tự động điều chỉnh để nâng cao chất lượng các đặc tính tĩnh và các đặc tính động của hệ thống. Mặt khác, việc dùng hệ chỉnh lưu - động cơ có kích thước và trọng lượng nhỏ gọn.

+) Nhược điểm: Hệ truyền động chỉnh lưu - động cơ có các van bán dẫn là các phần tử phi tuyến tính, do đó dạng điện áp chỉnh lưu ra có biên độ đập mạch cao, gây nên tổn thất phụ trong máy điện một chiều.

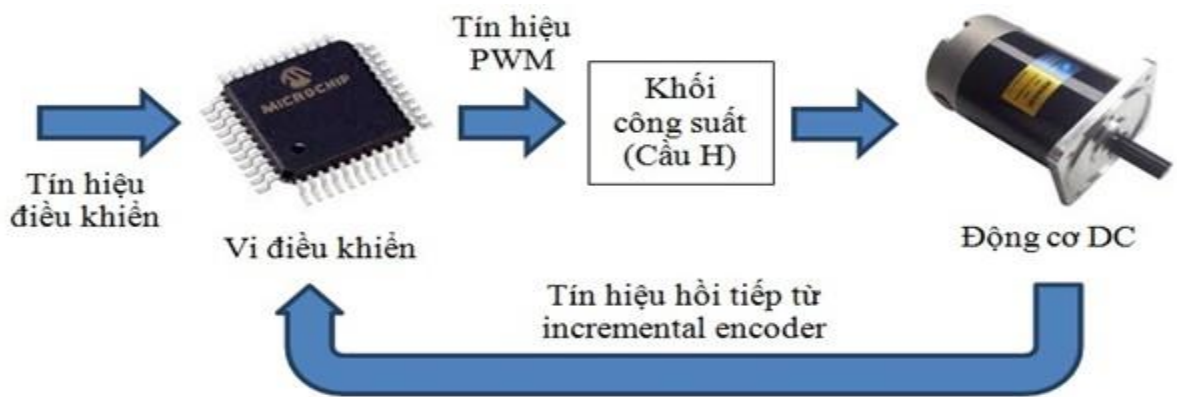
## CHƯƠNG 3

# THIẾT KẾ, CHẾ TẠO BỘ PID ĐIỀU KHIỂN ĐỘNG CƠ MỘT CHIỀU DÙNG PIC16F877A

## MỞ ĐẦU

### 3.1 KHÁI QUÁT VỀ MÔ HÌNH

Thực hiện chế tạo bộ điều khiển với ngôn ngữ lập trình C cho Pic 16F877A để điều khiển tốc độ cho động cơ DC có gắn encoder hồi tiếp tốc độ. Tốc độ được cài đặt và điều chỉnh trực tiếp với nút nhấn từ bàn phím và tốc độ tức thời hồi tiếp từ encoder được hiển thị trên màn hình LCD. Dưới đây là sơ đồ Driver PID cho động cơ DC.



*Hình 3.1* Sơ đồ khối phân cứng

#### 3.1.1 Giới thiệu sơ lược về các modul của mạch:

-Yêu cầu đặt ra: Điều khiển tốc độ cho động cơ DC có gắn encoder hồi tiếp tốc độ. Tốc độ được cài đặt từ bàn phím và tốc độ tức thời hồi tiếp từ encoder được hiển thị trên màn hình LCD 16x2.

-Tóm tắt hướng thực hiện đề tài:

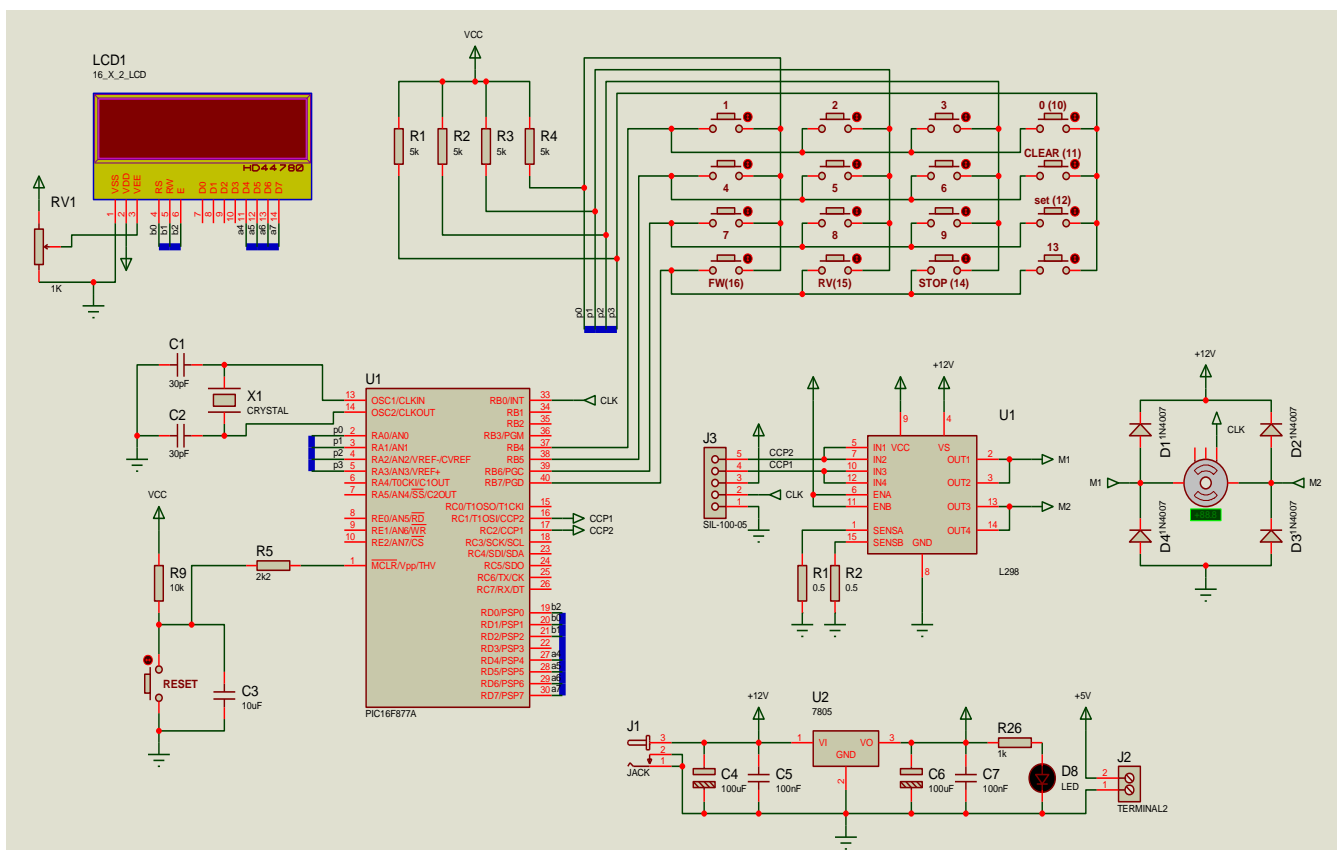
❖ Sử dụng Pic 16F877A là vi điều khiển trung tâm. Dùng chương trình CCS lập trình C và biên dịch chương trình.

❖ Xây dựng khối bàn phím gồm 16 phím để nhập tốc độ và điều khiển động cơ DC:

- 10 phím từ 0 đến 9 để cài đặt tốc độ (vòng/phúc).
- 1 phím SET (hay ENTER) để lưu tốc độ cài đặt.

- 1 phím CLEAR để xóa tốc độ cài đặt.
- 1 phím SAVE để lưu tốc độ vào eeprom.
- 3 phím điều khiển: quay thuận (FORWARD), quay nghịch (REVERSE), dừng (STOP).
- ❖ Hiện thị tốc độ dùng màn hình LCD 16x2, lập trình ở chế độ 4 bit (sử dụng 4 chân để nhận dữ liệu từ Pic).
- ❖ Sử dụng mạch cầu H là IC L298N để đảo chiều động cơ.
- ❖ Sử dụng 2 kênh PWM của vi điều khiển Pic thay đổi giá trị áp trung bình đặt vào động cơ để điều khiển tốc độ.
- ❖ Đối tượng điều khiển là động cơ DC 12V có gắn Encoder.
- ❖ Ngoài ra trên mạch còn có 1 phím nguồn (POWER) cấp điện từ adapter cho mạch và 1 phím RESET cho pic 16F877A.
- ❖ Để cấp nguồn cho mạch ta dùng adapter AC/DC (220V/12V) và khối nguồn sử dụng IC 7805 để ổn áp điện áp 5V cung cấp cho Pic.

### 3.1.2 Sơ đồ nguyên lý mạch



**Hình 3.2** Sơ đồ nguyên lý

### **3.1.3 Cách vận hành mạch:**

#### **▪ Bước 1:**

Bật nguồn (nhấn nút POWER), chờ cho Pic và màn hình LCD khởi động, màn hình hiển thị: “CHỌN CHẾ ĐỘ:”

Chương trình có 2 chế độ làm việc: chế độ 1 là bấm tốc độ đặt, có lưu tốc độ vào eeprom; chế độ 2 là bấm tốc độ có định thời gian để thay đổi chiều quay.

#### **▪ Bước 2:**

Nhập tốc độ từ bàn phím các phím từ 0 đến 9. Nếu nhập sai ta nhấn phím CLEAR con trỏ trên LCD sẽ xóa hết các số đã nhập, ta phải nhập lại từ đầu. Sau khi nhập xong, nhấn phím ENTER để lưu tốc độ đặt, tốc độ đặt được tính theo đơn vị vòng/phút.

Nếu là chế độ 2 thì ta phải nhập thêm thời gian, sau đó nhấn phím ENTER

#### **▪ Bước 3:**

Để điều khiển động cơ ta nhấn phím: quay thuận (FORWARD), quay nghịch (REVERSE), dừng (STOP).

#### **▪ Bước 4:**

Để nhập lại tốc độ ta nhấn phím CLEAR rồi tiến hành đặt tốc độ như bước 2.

-Tốc độ tức thời của động cơ sẽ được cập nhật mỗi 0,5s và sẽ được so sánh với tốc độ đặt để đưa ra tính hiệu điều khiển, đồng thời cứ mỗi 0,5s tốc độ sẽ hiển thị trên màn hình LCD.

## **3.2 GIỚI THIỆU CÁC LINH KIỆN SỬ DỤNG TRONG MẠCH**

### **3.2.1 Vi điều khiển PIC16F8774**

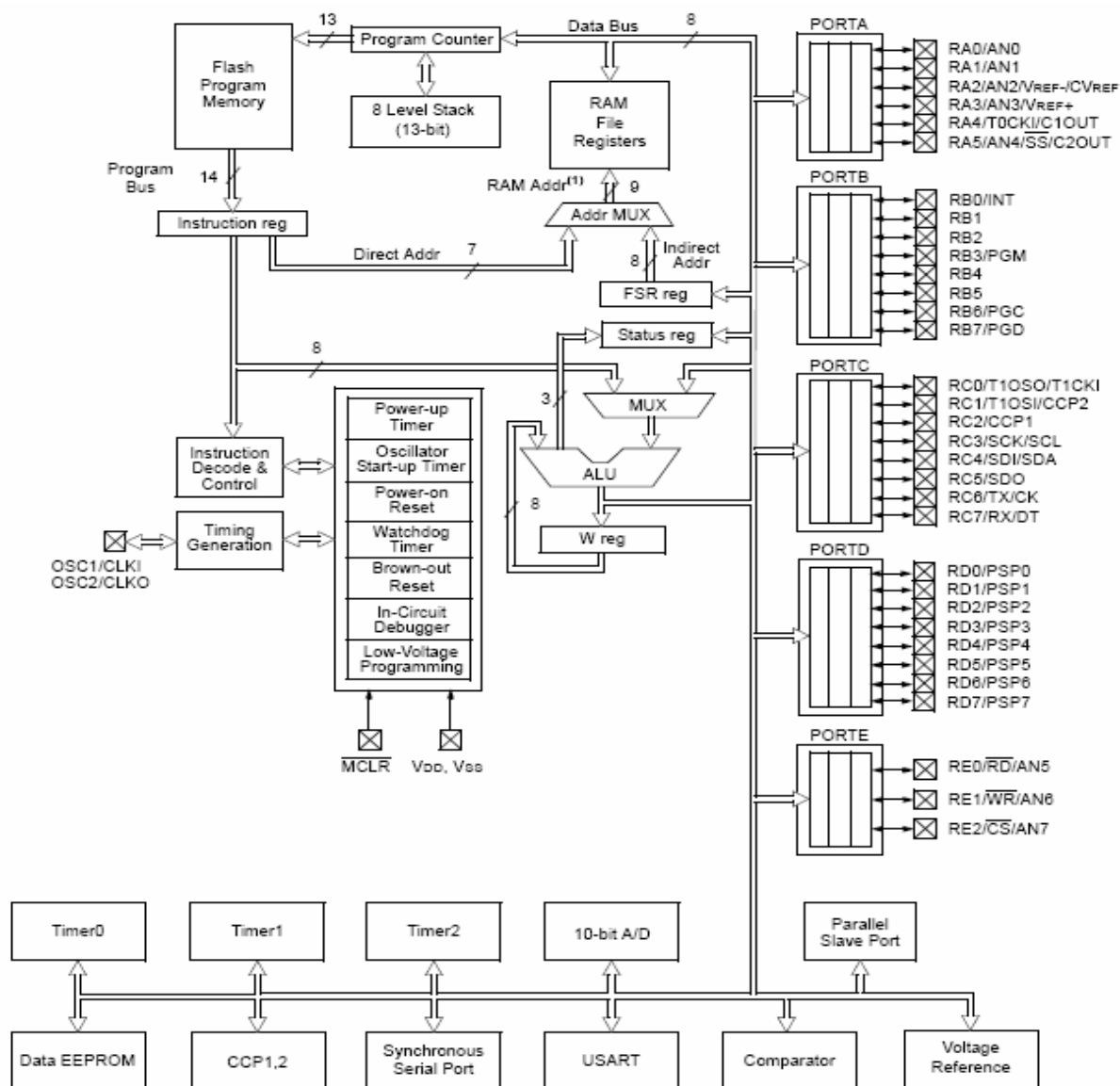
#### **3.2.1.1 Khái quát về vi điều khiển PIC16F877A**

*a/ Khái quát*

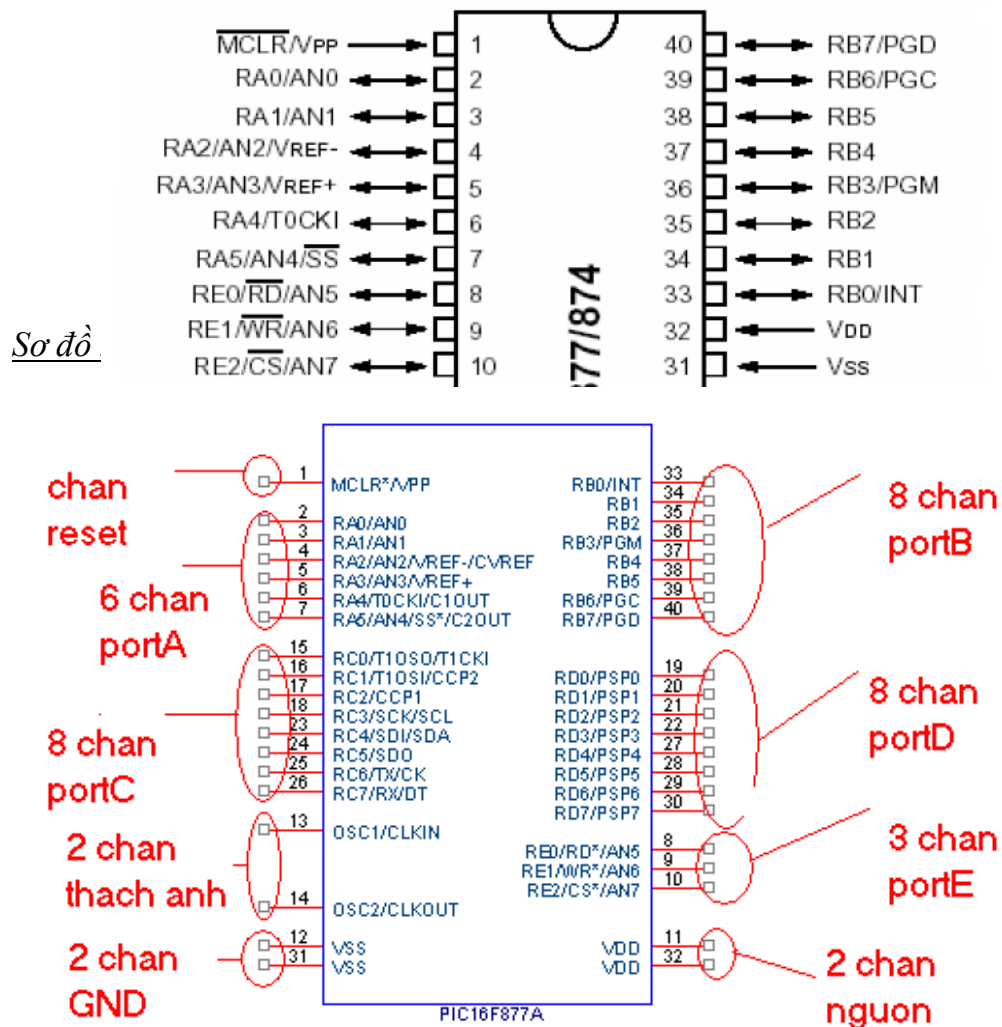
- PIC là tên viết tắt của “Programmable Intelligent computer” do hãng General Instrument đặt tên cho con vi điều khiển đầu tiên của họ. Hãng Microchip tiếp tục phát triển sản phẩm này và cho đến hàng đã tạo ra gần 100 loại sản phẩm khác nhau.
- PIC16F877A là dòng PIC khá phổ biến, khá đầy đủ tính năng phục vụ cho hầu hết tất cả các ứng dụng thực tế. Đây là dòng PIC khá dễ cho người mới làm quen với PIC có thể học tập và tạo nền tảng về họ vi điều khiển PIC của mình.
- PIC 16F877A thuộc họ vi điều khiển 16Fxxx có các đặc tính sau:
  - Ngôn ngữ lập trình đơn giản với 35 lệnh có độ dài 14 bit.
  - Tất cả các câu lệnh thực hiện trong 1 chu kỳ lệnh ngoại trừ 1 số câu lệnh rẽ nhánh thực hiện trong 2 chu kỳ lệnh. Chu kỳ lệnh bằng 4 lần chu kỳ dao động của thạch anh.
  - Bộ nhớ chương trình Flash 8Kx14 words, với khả năng ghi xóa khoảng 100 ngàn lần.
  - Bộ nhớ Ram 368x8bytes.
  - Bộ nhớ EPROM 256x8 bytes.
  - Khả năng ngắt (lên tới 14 nguồn cả ngắt trong và ngắt ngoài).
  - Ngăn nhớ Stack được chia làm 8 mức.
  - Truy cập bộ nhớ bằng địa chỉ trực tiếp hoặc gián tiếp.
  - Dải điện thế hoạt động rộng: 2.0V đến 5.5V.
  - Nguồn sử dụng 25mA.
  - Công suất tiêu thụ thấp:
    - <0.6mA với 5V, 4MHz
    - 20uA với nguồn 3V, 32 kHz.
  - Có 3 timer: timer0, 8 bit chức năng định thời và bộ đếm với hệ số tỷ lệ trước. Timer1, 16 bit chức năng bộ định thời, bộ đếm với hệ số tỷ

lệ trước, kích hoạt chế độ Sleep.Timer2, 8 bit chức năng định thời và bộ đếm với hệ số tỷ lệ trước và sau.

- Có 2 kênh Capture/ so sánh điện áp (Compare)/điều chế độ rộng xung PWM 10 bit / (CCP).
- Có 8 kênh chuyển đổi ADC 10 bit.
- Cổng truyền thông nối tiếp SSP với SPI phương thức chủ và I<sup>2</sup>C (chủ/phụ).Bộ truyền nhận thông tin đồng bộ, dị bộ (USART/SCL) có khả năng phát hiện 9 bit địa chỉ.
- Cổng phụ song song (PSP) với 8 bit mở rộng, với RD, WR và CS điều khiển.
- Do thời gian làm đồ án có hạn nên chúng em chỉ tập trung tìm hiểu các tính năng của PIC 16F877A có liên quan đến đề tài, dưới đây là 1 vài tính năng của PIC 16F877A được ứng dụng trong đồ án như:
  - Tổ chức bộ nhớ của PIC 16F877A.
  - Chức năng của các Port I/O.
  - Chức năng và cách thiết lập các tham số của 3 Timer 0,1,2.
  - Chức năng và cách thiết lập bộ điều chế độ rộng xung PWM.
  - Định nghĩa ngắt, các nguồn ngắt và tìm hiểu sâu về ngắt timer và ngắt ngoài là hai chức năng được sử dụng trong đề tài này.



**Hình 3.2:** Sơ đồ nguyên lý PIC 16F877A



**Hình 3.4:** Sơ đồ nguyên lý các Port của PIC 16F877A

### c/ Nhận xét

Từ sơ đồ chân và sơ đồ nguyên lý ở trên, ta rút ra các nhận xét ban đầu như sau :

- PIC16F877A có tất cả 40 chân



- 40 chân trên được chia thành 5 PORT, 2 chân cấp nguồn, 2 chân GND, 2 chân thạch anh và một chân dùng để RESET vi điều khiển.
- 5 port của PIC16F877A bao gồm :
  - + PORT B: 8 chân
  - + PORT D: 8 chân
  - + PORT C: 8 chân
  - + PORT A: 6 chân
  - + PORT E: 3 chân

### 3.2.1.2 Tổ chức bộ nhớ:

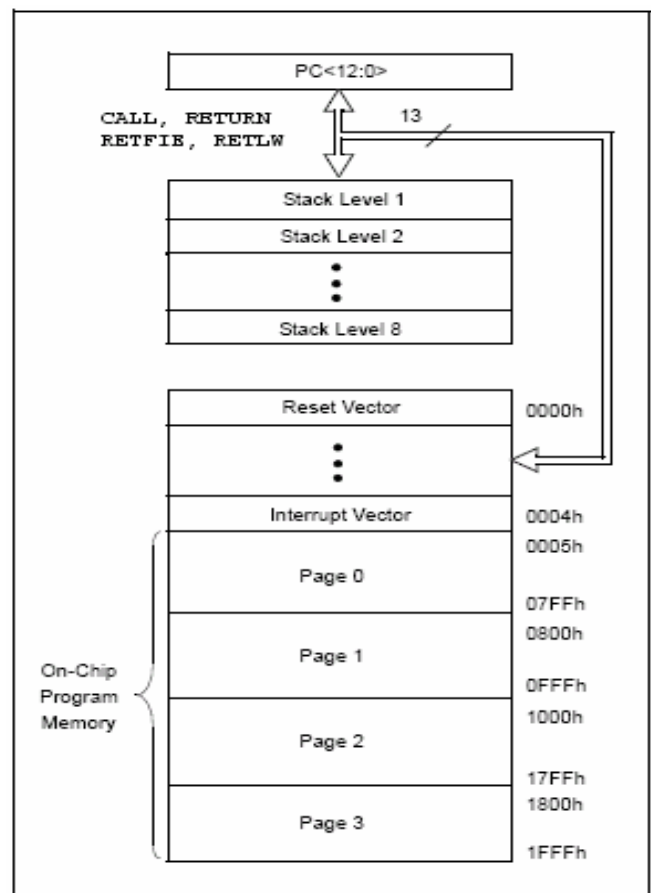
Cấu trúc bộ nhớ của vi điều khiển PIC16F877A bao gồm bộ nhớ chương trình (Program memory) và bộ nhớ dữ liệu (Data Memory).

#### a/ Bộ nhớ chương trình:

Bộ nhớ chương trình của vi điều khiển PIC16F877A là bộ nhớ flash, dung lượng bộ nhớ 8K word (1 word = 14 bit) và được phân thành nhiều trang (từ page0 đến page 3) . Như vậy bộ nhớ chương trình có khả năng chứa được  $8 \times 1024 = 8192$  lệnh (vì một lệnh sau khi mã hóa sẽ có dung lượng 1 word (14 bit)).

Để mã hóa được địa chỉ của 8K word bộ nhớ chương trình, bộ đếm chương trình có dung lượng 13 bit (PC<12:0>).

Khi vi điều khiển được reset, bộ đếm chương trình sẽ chỉ đến địa chỉ 0000h (Reset vector). Khi có ngắt xảy ra, bộ đếm chương trình sẽ chỉ



đến địa chỉ 0004h (Interrupt vector). Bộ nhớ chương trình không bao gồm bộ nhớ stack và không được địa chỉ hóa bởi bộ đếm chương trình.

**Hình 3.5:** Cấu trúc bộ nhớ chương trình PIC 16F877A

**b/ Bộ nhớ dữ liệu:**

- Bộ nhớ dữ liệu của PIC16F877A được chia thành 4 bank. Mỗi bank có dung lượng 128 byte.
- Nếu như 2 bank bộ nhớ dữ liệu của 8051 phân chia riêng biệt : 128 byte đầu tiên thuộc bank1 là vùng Ram nội chỉ để chứa dữ liệu, 128 byte còn lại thuộc bank 2 là cùng các thanh ghi có chức năng đặc biệt SFR mà người dùng không được chứa dữ liệu khác, còn 4 bank bộ nhớ dữ liệu của PIC16F877A được tổ chức theo cách khác.
- Mỗi bank của bộ nhớ dữ liệu PIC16F877A bao gồm cả các thanh ghi có chức năng đặc biệt SFR nằm ở các ô nhớ địa chỉ thấp và các thanh ghi mục đích dùng chung GPR nằm ở vùng địa chỉ còn lại của mỗi bank thanh ghi. Vùng ô nhớ các thanh ghi mục đích dùng chung này chính là nơi người dùng sẽ lưu dữ liệu trong quá trình viết chương trình. Tất cả các biến dữ liệu nên được khai báo chứa trong vùng địa chỉ này.
- Trong cấu trúc bộ nhớ dữ liệu của PIC16F877A, các thanh ghi SFR nào mà thường xuyên được sử dụng (như thanh ghi STATUS) sẽ được đặt ở tất cả các bank để thuận tiện trong việc truy xuất. Sở dĩ như vậy là vì, để truy xuất một thanh ghi nào đó trong bộ nhớ của 16F877A ta cần phải khai báo đúng bank chứa thanh ghi đó, việc đặt các thanh ghi sử dụng thường xuyên giúp ta thuận tiện hơn rất nhiều trong quá trình truy xuất, làm giảm lệnh chương trình.

Dựa trên sơ đồ 4 bank bộ nhớ dữ liệu PIC16F877A ta rút ra các nhận xét như sau :

- Bank0 gồm các ô nhớ có địa chỉ từ 00h đến 77h, trong đó các thanh ghi dùng chung để chứa dữ liệu của người dùng địa chỉ từ 20h đến 7Fh. Các thanh ghi PORTA, PORTB, PORTC, PORTD, PORTE đều chứa ở bank0, do đó để truy xuất dữ liệu các thanh ghi này ta phải chuyển đến bank0. Ngoài ra một vài các thanh ghi thông dụng khác ( sẽ giới thiệu sau) cũng chứa ở bank

- Bank1 gồm các ô nhớ có địa chỉ từ 80h đến FFh. Các thanh ghi dùng chung có địa chỉ từ A0h đến EFh. Các thanh ghi TRISA, TRISB, TRISC, TRISD, TRISE cũng được chứa ở bank1

- Tương tự ta có thể suy ra các nhận xét cho bank2 và bank3 dựa trên sơ đồ trên.

Cũng quan sát trên sơ đồ, ta nhận thấy thanh ghi STATUS, FSR... có mặt trên cả 4 bank. Một điều quan trọng cần nhắc lại trong việc truy xuất dữ liệu của PIC16F877A là : phải khai báo đúng bank chứa thanh ghi đó. Nếu thanh ghi nào mà 4 bank đều chứa thì không cần phải chuyển bank.

- **Thanh ghi chức năng đặc biệt SFR:** (Special Function Register)
- Đây là các thanh ghi được sử dụng bởi CPU hoặc được dùng để thiết lập và điều khiển các khối chức năng được tích hợp bên trong vi điều khiển. Có thể phân thanh ghi SFR làm hai loại: thanh ghi SFR liên quan đến các chức năng bên trong (CPU) và thanh ghi SRF dùng để thiết lập và điều khiển các khối chức năng bên ngoài (ví dụ như ADC, PWM, ...).
- Một số thanh ghi chức năng đặc biệt:
  - **Thanh ghi STATUS** (03h, 83h, 103h, 183h): thanh ghi chứa kết quả thực hiện phép toán của khối ALU, trạng thái reset và các bit chọn bank cần truy xuất trong bộ nhớ dữ liệu.

- **Thanh ghi OPTION\_REG (81h, 181h):** thanh ghi này cho phép đọc và ghi, cho phép điều khiển chức năng pull-up của các chân trong PORTB, xác lập các tham số về xung tác động, cạnh tác động của ngắt ngoại vi và bộ đếm Timer0.
- **Thanh ghi INTCON (0Bh, 8Bh, 10Bh, 18Bh):** thanh ghi cho phép đọc và ghi, chứa các bit điều khiển và các cờ hiệu khi timer0 bị tràn, ngắt ngoại vi RB0/INT và ngắt interrupt-on-change tại các chân của PORTB.
- **Thanh ghi PIE1 (8Ch):** chứa các bit điều khiển chi tiết các ngắt của các khối chức năng ngoại vi
- **Thanh ghi PIR1 (0Ch)** chứa cờ ngắt của các khối chức năng ngoại vi, các ngắt này được cho phép bởi các bit điều khiển chứa trong thanh ghi PIE1.
- **Thanh ghi PIE2 (8Dh):** chứa các bit điều khiển các ngắt của các khối chức năng CCP2, SSP bus, ngắt của bộ so sánh và ngắt ghi vào bộ nhớ EEPROM.
- **Thanh ghi PCON (8Eh):** chứa các cờ hiệu cho biết trạng thái các chế độ reset của vi điều khiển.

- **Thanh ghi mục đích chung GPR:** (General Purpose Register)

Các thanh ghi này có thể được truy xuất trực tiếp hoặc gián tiếp thông qua thanh ghi FSG (File Select Register). Đây là các thanh ghi dữ liệu thông thường, người sử dụng có thể tùy theo mục đích chương trình mà có thể dùng các thanh ghi này để chứa các biến số, hằng số, kết quả hoặc các tham số phục vụ cho chương trình.



**Hình 3.6:** Cấu trúc thanh ghi chức năng chung của PIC 16F877A

### 3.2.1.3 Stack

- Stack không nằm trong bộ nhớ chương trình hay bộ nhớ dữ liệu mà là một vùng nhớ đặc biệt không cho phép đọc hay ghi. Khi lệnh CALL được thực hiện hay khi một ngắt xảy ra làm chương trình bị rẽ nhánh, giá trị của bộ đếm chương trình PC tự động được vi điều khiển cất vào trong stack. Khi một trong các lệnh RETURN, RETLW hay RETFIE được thực thi, giá trị PC sẽ tự động được lấy ra từ trong stack, vi điều khiển sẽ thực hiện tiếp chương trình theo đúng qui trình định trước.

- Bộ nhớ Stack trong vi điều khiển PIC họ 16F87xA có khả năng chứa được 8 địa chỉ và hoạt động theo cơ chế xoay vòng. Nghĩa là giá trị cất vào bộ nhớ Stack lần thứ 9 sẽ ghi đè lên giá trị cất vào Stack lần đầu tiên và giá trị cất vào bộ nhớ Stack lần thứ 10 sẽ ghi đè lên giá trị cất vào Stack lần thứ 2.

- Cần chú ý là không có cờ hiệu nào cho biết trạng thái stack, do đó ta không biết được khi nào stack tràn. Bên cạnh đó tập lệnh của vi điều khiển dòng PIC cũng không có lệnh POP hay PUSH, các thao tác với bộ nhớ stack sẽ hoàn toàn được điều khiển bởi CPU.

### 3.2.1.4 Khái quát về chức năng của các port trong vi điều khiển PIC16F877A

#### a/ **PORTA:**

-PORTA (RPA) bao gồm 6 I/O pin. Đây là các chân “hai chiều” (bidirectional pin), nghĩa là có thể xuất và nhập được. Chức năng I/O này được điều khiển bởi thanh ghi TRISA (địa chỉ 85h). Muốn xác lập chức năng của một chân trong PORTA là input, ta “set” bit điều khiển tương ứng với chân đó trong thanh ghi TRISA và ngược lại, muốn xác lập chức năng của một chân trong PORTA là output, ta “clear” bit điều khiển tương ứng với chân đó trong thanh ghi TRISA. Thao tác này hoàn toàn tương tự đối với các PORT và các thanh ghi

điều khiển tương ứng TRIS (đối với PORTA là TRISA, đối với PORTB là TRISB, đối với PORTC là TRISC, đối với PORTD là TRISD và đối với PORTE là TRISE).

-Ngoài ra, PORTA còn có các chức năng quan trọng sau :

- Ngõ vào Analog của bộ ADC : thực hiện chức năng chuyển từ Analog sang Digital
- Ngõ vào điện thế so sánh
- Ngõ vào xung Clock của Timer0 trong kiến trúc phần cứng : thực hiện các nhiệm vụ đếm xung thông qua Timer0...
- Ngõ vào của bộ giao tiếp MSSP (Master Synchronous Serial Port)

- Các thanh ghi SFR liên quan đến PORTA bao gồm:

PORTA (địa chỉ 05h) : chứa giá trị các pin trong PORTA.

TRISA (địa chỉ 85h) : điều khiển xuất nhập.

CMCON (địa chỉ 9Ch) : thanh ghi điều khiển bộ so sánh.

CVRCON (địa chỉ 9Dh) : thanh ghi điều khiển bộ so sánh điện áp.

ADCON1 (địa chỉ 9Fh) : thanh ghi điều khiển bộ ADC.

### **b/ PORTB**

- PORTB (RPB) gồm 8 pin I/O. Thanh ghi điều khiển xuất nhập tương ứng là TRISB.

- Bên cạnh đó một số chân của PORTB còn được sử dụng trong quá trình nạp chương trình cho vi điều khiển với các chế độ nạp khác nhau. PORTB còn liên quan đến ngắt ngoại vi và bộ Timer0. PORTB còn được tích hợp chức năng điện trở kéo lên được điều khiển bởi chương trình.

- Các thanh ghi SFR liên quan đến PORTB bao gồm:

PORTB (địa chỉ 06h, 106h) : chứa giá trị các pin trong PORTB

TRISB (địa chỉ 86h, 186h) : điều khiển xuất nhập

OPTION\_REG (địa chỉ 81h, 181h): điều khiển ngắt ngoại vi và bộ Timer0.

#### **c/ PORTC**

PORTC có 8 chân và cũng thực hiện được 2 chức năng input và output dưới sự điều khiển của thanh ghi TRISC tương tự như hai thanh ghi trên.

Ngoài ra PORTC còn có các chức năng quan trọng sau :

- Ngõ vào xung clock cho Timer1 trong kiến trúc phần cứng
- Bộ PWM thực hiện chức năng điều xung lập trình được tần số, duty cycle: sử dụng trong điều khiển tốc độ và vị trí của động cơ v.v....
- Tích hợp các bộ giao tiếp nối tiếp I2C, SPI, SSP, USART

#### **d/ PORTD**

-PORTD có 8 chân. Thanh ghi TRISD điều khiển 2 chức năng input và output của PORTD tương tự như trên. PORTD cũng là cổng xuất dữ liệu của chuẩn giao tiếp song song PSP (Parallel Slave Port).

-Các thanh ghi liên quan đến PORTD bao gồm:

Thanh ghi PORTD: chứa giá trị các pin trong PORTD.

Thanh ghi TRISD: điều khiển xuất nhập.

Thanh ghi TRISE: điều khiển xuất nhập PORTE và chuẩn giao tiếp PSP.

#### **e/ PORTE**

-PORTE có 3 chân. Thanh ghi điều khiển xuất nhập tương ứng là TRISE. Các chân của PORTE có ngõ vào analog Bên cạnh đó PORTE còn là các chân điều khiển của chuẩn giao tiếp PSP.

-Các thanh ghi liên quan đến PORTE bao gồm:

PORTE: chứa giá trị các chân trong PORTE.

TRISE: điều khiển xuất nhập và xác lập các thông số cho chuẩn giao tiếp PSP.

ADCON1: thanh ghi điều khiển khối ADC.

### 3.2.1.5 Các vấn đề về Timer

PIC16F877A có tất cả 3 timer : timer0 (8 bit), timer1 (16 bit) và timer2 (8 bit).

#### a/ Timer0

- **Là bộ định thời hoặc bộ đếm có những ưu điểm sau:**
  - 8 bit cho bộ định thời hoặc bộ đếm.
  - Có khả năng đọc và viết.
  - Có thể dùng đồng bên trong hoặc bên ngoài.
  - Có thể chọn cạnh xung của xung đồng hồ.
  - Có thể chọn hệ số chia đầu vào (lập trình bằng phần mềm).
  - Ngắt tràn.
- **Hoạt động của Timer 0:**
  - Timer 0 có thể hoạt động như một bộ định thời hoặc một bộ đếm. Việc chọn bộ định thời hoặc bộ đếm có thể được xác lập bằng việc xoá hoặc đặt bit TOCS của thanh ghi OPTION\_REG<5>.
  - Nếu dùng hệ số chia xung đầu vào thì xoá bit PSA của thanh ghi OPTION\_REG<3>.
  - Trong chế độ bộ định thời được lựa chọn bởi việc xoá bit T0CS (OPTION\_REG<5>), nó sẽ được tăng giá trị sau một chu kỳ lệnh nếu không chọn hệ số chia xung đầu vào. Và giá trị của nó được viết tới thanh ghi TMR0.
  - Khi dùng xung clock bên ngoài cho bộ định thời Timer0 và không dùng hệ số chia clock đầu vào Timer0 thì phải đáp ứng các điều kiện cần thiết để có thể hoạt động đó là phải bảo đảm xung clock bên ngoài có thể đồng bộ với xung clock bên trong ( $T_{osc}$ ).
  - Hệ số chia dùng cho Timer 0 hoặc bộ WDT. Các hệ số nay không có khả năng đọc và khả năng viết. Để chọn hệ số chia xung cho bộ định thời của Timer0 hoặc cho bộ WDT ta tiến hành xoá hoặc đặt bit PSA của thanh ghi OPTION\_REG<3>



- Những bit PS2, PS1, PS0 của thanh ghi OPTION\_REG<2:0> dùng để xác lập các hệ số chia.

- Bộ tiền định có giá trị 1:2 chẳng hạn, có nghĩa là : bình thường không sử dụng bộ tiền định của Timer0 (đồng nghĩa với tiền định tỉ lệ 1:1) thì cứ khi có tác động của 1 xung clock thì timer0 sẽ tăng thêm một đơn vị. Nếu sử dụng bộ tiền định 1:4 thì phải mất 4 xung clock thì timer0 mới tăng thêm một đơn vị. Vô hình chung, giá trị của timer0 (8 bit) lúc này không còn là 255 nữa mà là  $255 \times 4 = 1020$ .

#### - Ngắt của bộ Timer0

Ngắt của bộ Timer 0 được phát sinh ra khi thanh ghi TMR0 bị tràn tức từ FFh quay về 00h. Khi đó bit T0IF của thanh ghi INTCON<2> sẽ được đặt. Bit này phải được xóa bằng phần mềm nếu cho phép ngắt bit T0IE của thanh ghi INTCON<5> được set. Timer0 bị dừng hoạt ở chế độ SLEEP ngắt Timer 0 không đánh thức bộ xử lý ở chế độ SLEEP.

#### - Các thanh ghi liên quan đến Timer0 bao gồm:

- Thanh ghi OPTION\_REG : điều khiển hoạt động của Timer0

OPTION\_REG REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
RBP	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

**Hình 3.7:** Cấu trúc thanh ghi OPTION\_REG REGISTER điều khiển hoạt động của Timer0

- bit 5 T0CS lựa chọn nguồn clock  
1=Clock ngoại từ chân T0CKI  
0=Clock trong Fosc/4
- bit 4 T0SE bit lựa chọn sườn xung clock  
1=Timer 0 tăng khi chân T0CKI từ cao xuống thấp(sườn xuống)  
0=Timer 0 tăng khi chân T0CKI từ thấp lên cao(sườn xuống)
- bit 3 PSA bit gán bộ chia xung đầu vào  
1=gán bộ chia Prescaler cho WDT

0=gán bộ chia Prescaler cho Timer 0

- bit 2:0 PS2:PS1 lựa chọn hệ số chia hệ số xung theo bảng sau:

PS2:PS 0	Timer 0	W DT
000	1:2	1:1
001	1:4	1:2
010	1:8	1:4
011	1:16	1:8
100	1:32	1:1 6
101	1:64	1:3 2
110	1:128	1:6 4
111	1:256	1:1 28

- Thanh ghi TMR0 đại chỉ 01h và 101h : chứa giá trị của bộ định thời Timer0
- Thanh ghi INTCON : cho phép ngắt hoạt động

Thanh ghi chứa các bit điều khiển và các bit cờ hiệu khi timer0 bị tràn, ngắt ngoại vi RB0/INT và ngắt interrupt\_on\_change tại các chân của

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF
bit 7							bit 0

**Hình 3.8:** Cấu trúc thanh ghi INTCON cho phép ngắt Timer0 hoạt động

- Bit 7 GIE Global Interrupt Enable bit

GIE = 1 cho phép tất cả các ngắt.

GIE = 0 không cho phép tất cả các ngắt.

- Bit 6 PEIE Pheripheral Interrupt Enable bit

PEIE = 1 cho phép tất cả các ngắt ngoại vi.

PEIE = 0 không cho phép tất cả các ngắt ngoại vi.

- Bit 5 TMR0IE Timer0 Overflow Interrupt Enable bit

TMR0IE = 1 cho phép ngắt Timer0.

TMR0IE = 0 không cho phép ngắt Timer0.

- Bit 4 RBIE RB0/INT External Interrupt Enable bit

RBIE = 1 cho phép tất cả các ngắt ngoại vi RB0/INT

RBIE = 0 không cho phép tất cả các ngắt ngoại vi RB0/INT

- Bit 3 RBIE RB Port change Interrupt Enable bit

RBIE = 1 cho phép ngắt RB Port change

RBIE = 0 không cho phép ngắt RB Port change

- Bit 2 TMR0IF Timer0 Interrupt Flag bit

TMR0IF = 1 thanh ghi TMR0 bị tràn (phải xóa cờ hiệu bằng chương trình).

TMR0IF = 0 thanh ghi TMR0 chưa bị tràn.

- Bit 1 INTF BR0/INT External Interrupt Flag bit

INTF = 1 ngắt RB0/INT xảy ra (phải xóa cờ hiệu bằng chương trình).

INTF = 0 ngắt RB0/INT chưa xảy ra.

- Bit 0 RBIF RB Port Change Interrupt Flag bit

RBIF = 1 ít nhất có một chân RB7:RB4 có sự thay đổi trạng thái. Bít này phải được xóa bằng chương trình sau khi đã kiểm tra lại các giá trị chân tại PORTB.

RBIF = 0 không có sự thay đổi trạng thái các chân RB7:RB4.

## **b/ TIMER1**

- **Timer1** là bộ định thời 16 bit, giá trị của Timer1 sẽ được lưu trong hai thanh ghi 8 bit TMR1H:TMR1L. Cờ ngắt của Timer1 là bit TMR1IF, bit điều khiển của Timer1 là TRM1IE. Cặp thanh ghi của TMR1 sẽ tăng từ 0000h lên đến FFFFh rồi sau đó tràn về 0000h. Nếu ngắt được cho phép, nó sẽ xảy ra khi giá trị của TMR1 tràn từ FFFFh rồi

về 0000h, lúc này TMR1IF sẽ bật lên.

- **Timer1 có 3 chế độ hoạt động :**

- Chế độ hoạt động định thời đồng bộ: Chế độ được lựa chọn bởi bit TMR1CS. Trong chế độ này xung cấp cho Timer1 là  $F_{osc}/4$ , bit T1SYNC không có tác dụng.
- Chế độ đếm đồng bộ: trong chế độ này, giá trị của timer1 sẽ tăng khi có xung cạnh lên vào chân T1OSI/RC1. Xung clock ngoại sẽ được đồng bộ với xung clock nội, hoạt động đồng bộ được thực hiện ngay sau bộ tiền định tỉ lệ xung (prescaler).
- Chế độ đếm bất đồng bộ: chế độ này xảy ra khi bit T1SYNC được set. Bộ định thời sẽ tiếp tục đếm trong suốt quá trình ngủ (Sleep) của vi điều khiển và có khả năng tạo một ngắt khi bộ định thời tràn và làm cho vi điều khiển thoát khỏi trạng thái ngủ.

- **Các thanh ghi liên quan đến Timer1 bao gồm:**

- INTCON (địa chỉ 0Bh, 8Bh, 10Bh, 18Bh): cho phép ngắt hoạt động (2 bit GIE và PEIE).
- PIR1 (địa chỉ 0Ch): chứa cờ ngắt Timer1 (TMR1IF).
- PIE1 (địa chỉ 8Ch): cho phép ngắt Timer1 (TMR1IE).

Ba thanh ghi vừa nêu trên sẽ được trình bày ở phần chương trình ngắt của PIC

- TMR1L (địa chỉ 0Eh): chứa giá trị 8 bit thấp của bộ đếm Timer1.
- TMR1H (địa chỉ 0Fh): chứa giá trị 8 bit cao của bộ đếm Timer1.

Hai thanh ghi TMR1L và TMR1H là 2 thanh ghi chứa dữ liệu 16 bit (lần lượt chứa 4 bit thấp và 4 bit cao) của bộ đếm Timer1

- T1CON (địa chỉ 10h): xác lập các thông số chi Timer

**T1CON: TIMER1 CONTROL REGISTER (ADDRESS 10h)**

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
bit 7							bit 0

**Hình 3.9:** Cấu trúc thanh ghi T1CON điều khiển hoạt động của Timer1

- bit 7,6 không sử dụng
- bit 5,4 T1CKPS1: T1CKPS0 lựa chọn hệ số chia xung vào.

	T1CKPS1	
T1CKPS0		
	00	1:1
	01	1:2
	10	1:4
	11	1:8

- bit 3 T1OSCEN bit điều khiển bộ dao động Timer1

1= Bộ dao động hoạt động

0= Bộ dao động không hoạt động

- bit 2 bit điều khiển xung clock ngoại đồng bộ

khi TMR1CS=1

- bit2=0 có đồng clock ngoại

=1 không đồng bộ clock ngoài

khi TMR1CS=0 bit này không có tác dụng

- bit 1 TMR1CS bit lựa chọn nguồn xung clock vào

TMR1CS=1 clock từ chân RC0/T1OSO/T1CKI (sườn lên)

TMR1CS=0 clock trong Fosc/4

- bit 0 bit bật tắt Timer

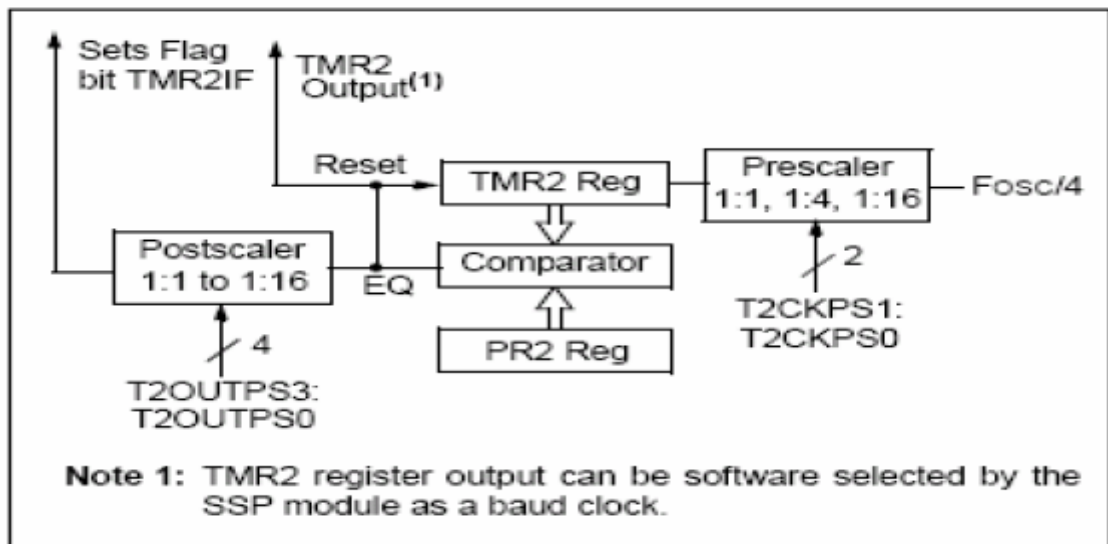
1= Timer 1 enable

0=Timer 1 Disable

### c/ Timer 2

- **Timer2:** là bộ định thời 8 bit bao gồm một bộ tiền định (prescaler), một bộ hậu định Postscaler và một thanh ghi chu kỳ

viết tắt là PR2. Việc kết hợp timer2 với 2 bộ định tỉ lệ cho phép nó hoạt động như một bộ định thời 16 bit. Module timer2 cung cấp thời gian hoạt động cho chế độ điều biến xung PWM nếu module CCP được chọn.



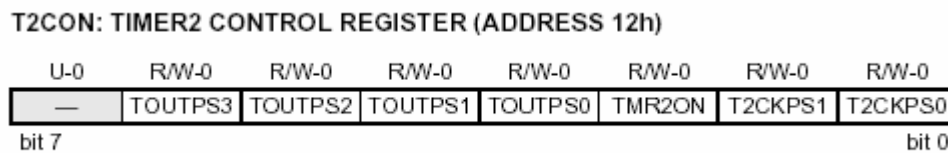
**Hình 3.10:** Sơ đồ khối Timer2

### - Hoạt động của bộ Timer2

- Timer2 được dùng chủ yếu ở phần điều chế xung của bộ CCP, thanh ghi TMR2 có khả năng đọc và viết, nó có thể xóa bằng việc reset lại thiết bị. Đầu vào của xung có thể chọn các tỷ số sau; 1:1; 1:4 hoặc 1:16 việc lựa chọn các tỷ số này có thể điều khiển bằng các bit sau T2CKPS1 và bit T2CKPS0.

- Bộ Timer2 có 1 thanh ghi 8 bit PR2 . Timer 2 tăng từ giá trị 00h cho đến khớp với PR2 và tiếp theo nó sẽ reset lại giá trị 00h và lệnh kế tiếp thực hiện. Thanh ghi PR2 là một thanh ghi có khả năng đọc và khả năng viết.

### c/ Thanh ghi T2CON: điều khiển hoạt động của Timer2



**Hình 3.11:** Cấu trúc thanh ghi T2CON điều khiển hoạt động của Timer2

- bit 7 không sử dụng
- bit 6-3 TOUTPS3: TOUTPS0 bit lựa chọn hệ số đầu ra Timer 2  
 0000=1:1  
 0001=1:2  
 0010=1:3  
 ...  
 1111=1:16
- bit 2 TMR2ON bit bật tắt hoạt động Timer 2  
 1= enable  
 0= disable
- bit 1- 0 T2CKPS1:T2CKPS0 chọn hệ chia đầu vào  
 00 = 1:1  
 01 = 1:4  
 1x=1:16

### 3.2.5 Ngắt (interrupt)

- PIC16F877A có đến 14 nguồn tạo ra hoạt động ngắt được điều khiển bởi thanh ghi INTCON (bit GIE). Bên cạnh đó mỗi ngắt còn có một bit điều khiển và cờ ngắt riêng. Các cờ ngắt vẫn được set bình thường khi thỏa mãn điều kiện ngắt xảy ra bất chấp trạng thái của bit GIE, tuy nhiên hoạt động ngắt vẫn phụ thuộc vào bit GIE và các bit điều khiển khác. Bit điều khiển ngắt RB0/INT và TMR0 nằm trong thanh ghi INTCON, thanh ghi này còn chứa bit cho phép các ngắt ngoại vi PEIE. Bit điều khiển các ngắt nằm trong thanh ghi PIE1 và PIE2. Cờ ngắt của các ngắt nằm trong thanh ghi PIR1 và PIR2.
- Trong một thời điểm chỉ có một chương trình nh ngắt được thực thi, chương trình ngắt được kết thúc bằng lệnh RETFIE. Khi chương trình ngắt được thực thi, bit GIE tự động được xóa, địa chỉ lệnh tiếp theo

của chương trình chính được cất vào trong bộ nhớ Stack và bộ đếm chương trình sẽ chỉ đến địa chỉ 0004h. Lệnh RETFIE được dùng để thoát khỏi chương trình ngắt và quay trở về chương trình chính, đồng thời bit GIE cũng sẽ được set để cho phép các ngắt hoạt động trở lại. Các cờ hiệu được dùng để kiểm tra ngắt nào đang xảy ra và phải được xóa bằng chương trình trước khi cho phép ngắt tiếp tục hoạt động trở lại để ta có thể phát hiện được thời điểm tiếp theo mà ngắt xảy ra.

- Đối với các ngắt ngoại vi như ngắt từ chân INT hay ngắt từ sự thay đổi trạng thái các pin của PORTB (PORTB Interrupt on change), việc xác định ngắt nào xảy ra cần 3 hoặc 4 chu kì lệnh tùy thuộc vào thời điểm xảy ra ngắt.

- Cần chú ý là trong quá trình thực thi ngắt, chỉ có giá trị của bộ đếm chương trình được cất vào trong Stack, trong khi một số thanh ghi quan trọng sẽ không được cất và có thể bị thay đổi giá trị trong quá trình thực thi chương trình ngắt. Điều này nên được xử lí bằng chương trình để tránh hiện tượng trên xảy ra.

- Các nguồn ngắt của Pic 16F877A:

- 1) RTCC hoặc TIMER0: ngắt tràn Timer0.
- 2) RB: ngắt khi có sự thay đổi trạng thái 1 trong các chân từ RB4 đến RB7 của PORTB.
- 3) EXT: (External Interrupt) ngắt ngoài khi có sự thay đổi trạng thái ở chân RB0 của PORTB.
- 4) AD: ngắt khi bộ chuyển đổi tính hiệu tương tự sang tính hiệu số chuyển đổi hoàn tất 1 tính hiệu.
- 5) TBE: ngắt khi bộ đệm của cổng RS232 rỗng.
- 6) RDA: ngắt khi cổng RS232 nhận tính hiệu.
- 7) TIMER1: ngắt khi timer1 bị tràn.
- 8) TIMER2: ngắt khi timer2 bị tràn.
- 9) CCP1; CCP2: ngắt khi bộ capture hoặc bộ Compare (bộ so sánh



điện áp) hoạt động; ở kênh 1 hoặc 2.

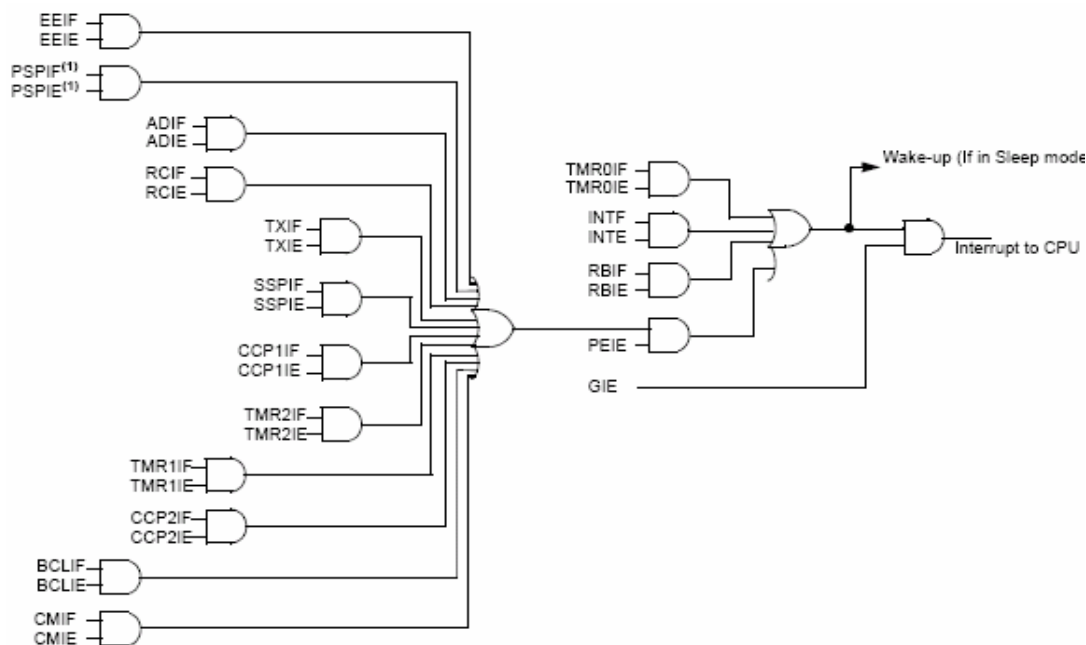
10) SSP: ngắt khi SPI hoặc I2C hoạt động.

11) PSP: ngắt khi truyền nhận dữ liệu song song.

12) BUSCOL: ngắt khi xung đột đường truyền.

13) EEPROM: ngắt khi ghi xong dữ liệu.

14) COMP: ngắt sau khi thực hiện so sánh tính hiệu.



**Hình 3.12:** Sơ đồ hoạt động ngắt của PIC 16F877A

Với đề tài “Điều khiển tốc độ động cơ DC” chúng em sử dụng hai loại nguồn ngắt là ngắt ngoài khi có sự thay đổi trạng thái ở chân RB0 của PORTB và ngắt tràn Timer1, vì thế chúng em sẽ trình bày cụ thể hơn về nguyên lý hoạt động và các cài đặt của hai loại ngắt này trong phần dưới đây:

### 3.2.1.6 Ngắt ngoài (External Interrupt):

- Ngắt này dựa trên sự thay đổi trạng thái của pin RB0/INT. Cảnh tác động gây ra ngắt, có thể là cạnh lên hay cạnh xuống và được điều khiển bởi bit INTEDG (thanh ghi OPTION\_ REG <6>). Khi có cạnh tác động thích hợp xuất hiện tại pin RB0/INT, cờ ngắt INTF được set

bất chấp trạng thái các bit điều khiển GIE và PEIE. Ngắt này có khả năng đánh thức vi điều khiển từ chế độ sleep nếu bit cho phép ngắt được set trước khi lệnh SLEEP được thực thi.

- Thanh ghi OPTION\_REG: địa chỉ 81h, 181h

Thanh ghi này cho phép điều khiển chức năng pull-up của các pin trong PORTB, xác

lập các tham số về xung tác động, cạnh tác động của ngắt ngoại vi RB0 (External Interrupt) và bộ đếm Timer0.

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

**Hình 3.13:** Cấu trúc thanh ghi OPTION\_REG cho phép ngắt ngoại và ngắt timer0

- Bit 7 PORTB pull-up enable bit

= 1 không cho phép chức năng pull-up của PORTB

= 0 cho phép chức năng pull-up của PORTB

- Bit 6 INTEDG Interrupt Edge Select bit

INTEDG = 1 ngắt xảy ra khi cạnh dương chân RB0/INT xuất hiện.

INTEDG = 0 ngắt xảy ra khi cạnh âm chân RB0/INT xuất hiện.

- Bit 5 T0CS Timer0 Clock Source select bit

T0SC = 1 clock lấy từ chân RA4/TOCK1.

T0SC = 0 dùng xung clock bên trong (xung clock này bằng với xung clock

dùng để thực thi lệnh).

- Bit 4 T0SE Timer0 Source Edge Select bit

T0SE = 1 tác động cạnh lên.

T0SE = 0 tác động cạnh xuống.

- Bit 3 PSA Prescaler Assignment Select bit

PSA = 1 bộ chia tần số (prescaler) được dùng cho WDT

PSA = 0 bộ chia tần số được dùng cho Timer0

- Bit 2:0 PS2:PS0 Prescaler Rate Select bit

Các bit này cho phép thiết lập tỉ số chia tần số của Prescaler.

Bit Value	TMR0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

#### a/ Ngắt tràn Timer1:

- Cặp thanh ghi TMR1H và TMR1L chứa giá trị đếm của Timer1, chúng tăng từ giá trị 0000h đến giá trị FFFFh đến giá trị này tiếp tục tăng thì timer1 tràn và quay lại giá trị 0000h. Và ngắt xuất hiện khi tràn quá giá trị FFFFh khi này cờ ngắt TMR1IF sẽ được đặt. Ngắt có thể hoạt động hoặc không hoạt động nhờ việc đặt hoặc xóa bit TMR1IE.

- Thanh ghi điều khiển Timer1 T1CON:

**T1CON: TIMER1 CONTROL REGISTER (ADDRESS 10h)**

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
bit 7							bit 0

- bit 7,6 không sử dụng
- bit 5,4 T1CKPS1:T1CKPS0 lựa chọn hệ số chia xung vào

T1CKPS1:T1CKPS0	tỷ lệ chia đầu vào
00	1:1
01	1:2
10	1:4

- bit 3 T1OSCEN bit điều khiển bộ dao động Timer1

1= Bộ dao động hoạt động

0= Bộ dao động không hoạt động

- bit 2 bit điều khiển xung clock ngoài đồng bộ  
khi TMR1CS=1

bit2=0 có đồng clock ngoài

=1 không đồng bộ clock ngoài

khi TMR1CS=0 bit này không có tác dụng

- bit 1 TMR1CS bit lựa chọn nguồn xung clock vào

TMR1CS=1 clock từ chân RC0/T1OSO/T1CKI (sườn lên)

TMR1CS=0 clock trong Fosc/4

- bit 0 bit bật tắt Timer

1= Timer 1 enable

0=Timer 1 Disable

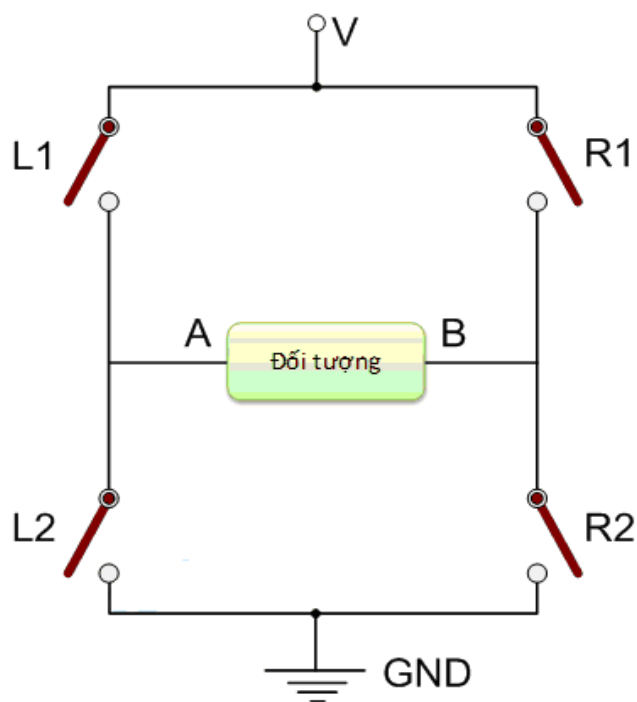
### 3.2.2 Mạch cầu h ( h-bridge circuit )

#### 3.2.2.1 Công dụng và nguyên lí hoạt động:

Mạch cầu H là một mạch điện giúp đảo chiều dòng điện qua một đối tượng .Đối tượng là động cơ DC mà chúng ta cần điều khiển .Mục đích điều khiển là cho phép dòng điện qua đối tượng theo chiều A đến B hoặc B đến A .Từ đó giúp đổi chiều quay của động cơ.

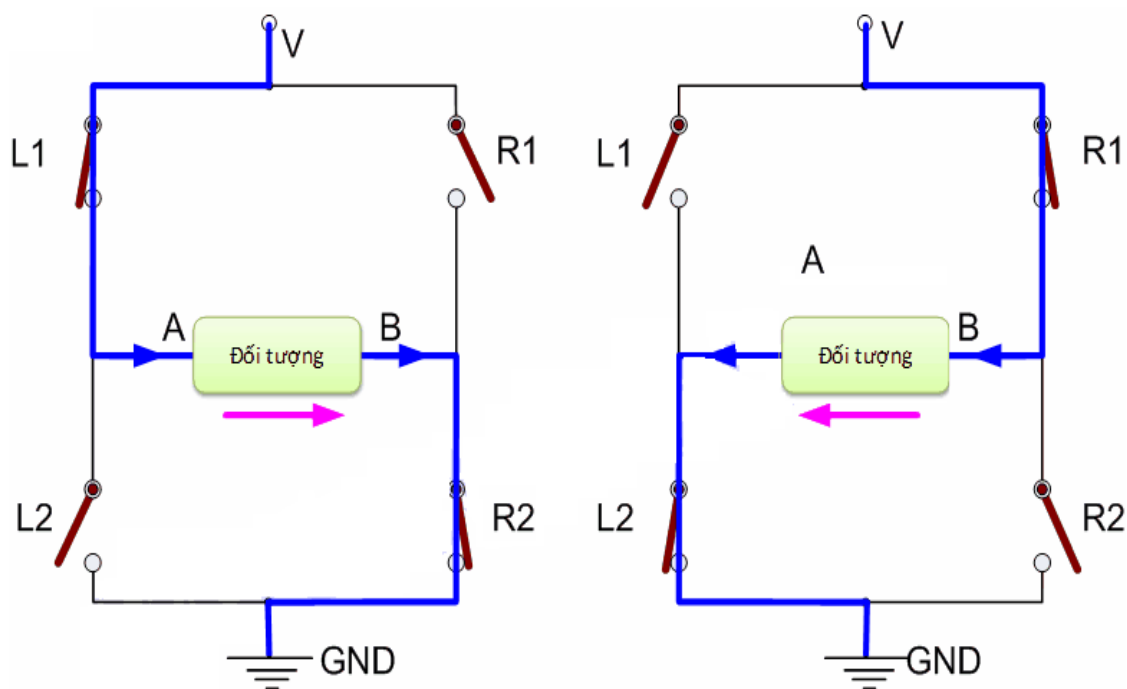
Hiện nay, ngoài loại mạch cầu H được thiết kế từ các linh kiện rời như: BJT công suất, Mosfet, ... Còn có các loại mạch cầu H được tích hợp thành các IC như: L293D và L298D. Do đối tượng điều khiển trong đề tài này là động cơ

DC có điện áp 12V và công suất nhỏ nên chúng em dùng mạch cầu H đảo chiều động cơ là IC L298.



**Hình 3.14** : Mạch cầu H

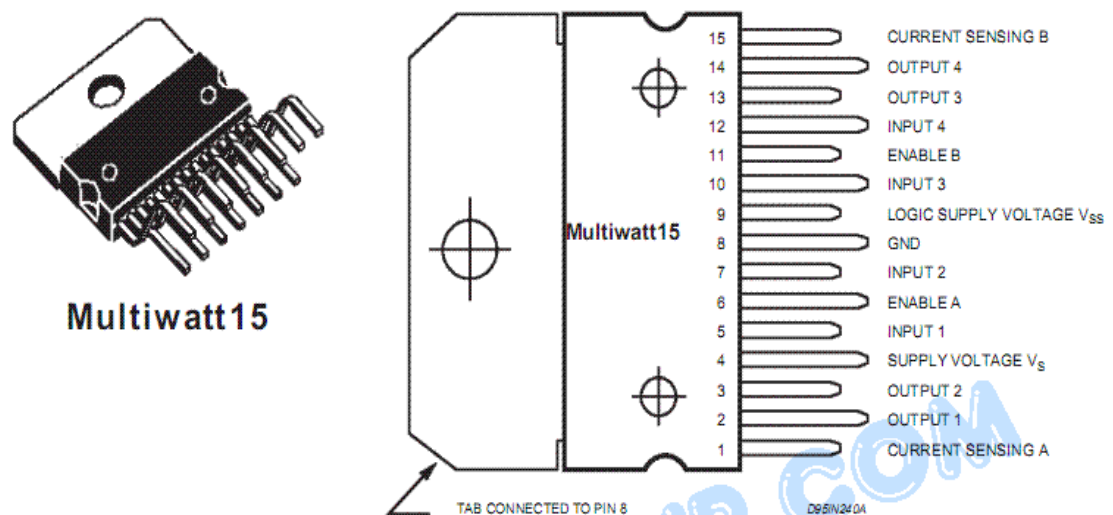
Khảo sát hoạt động của mạch cầu H



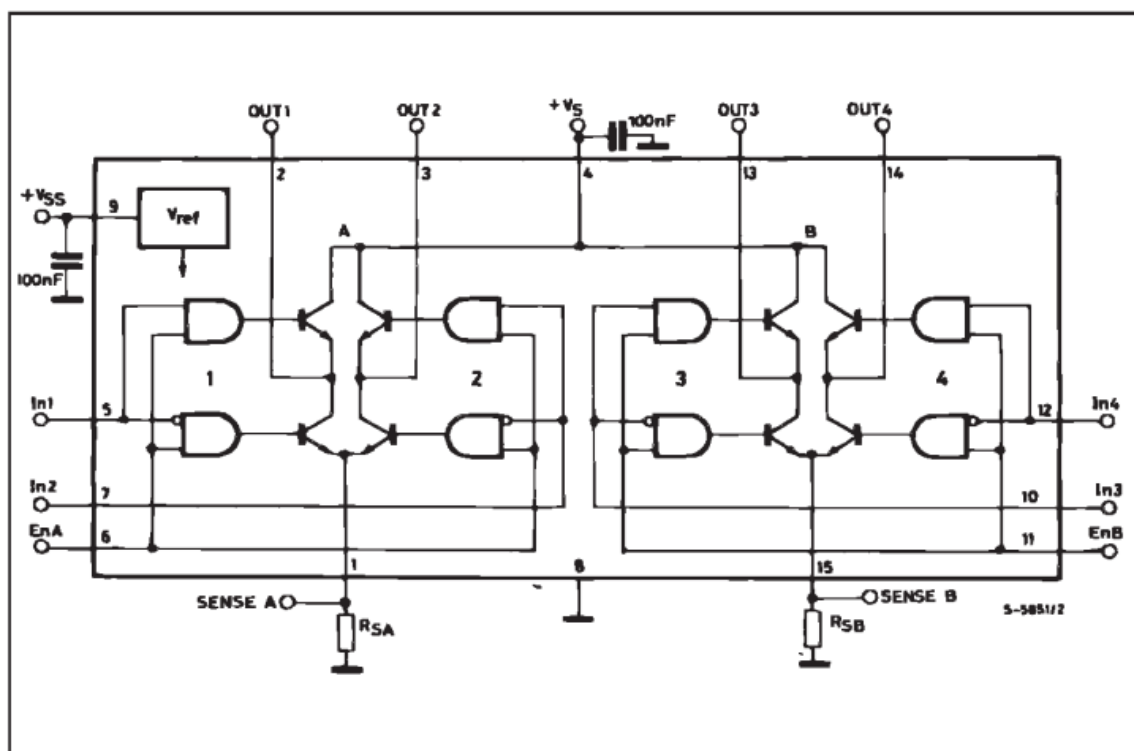
**Hình 3.15**: Nguyên lý hoạt động của mạch cầu H

### 3.2.2.2 Mạch cầu H L298D:

L298D là một chip tích hợp 2 mạch trong gói 15 chân. L298D có điện áp danh nghĩa cao (lớn hơn 50V) và dòng điện danh nghĩa lớn hơn 2A nên rất thích hợp cho các ứng dụng công suất nhỏ như các động cơ DC loại vừa và nhỏ.



**Hình 3.16:** Sơ đồ chân của IC L298D (phải)  
IC L298D (trái)



**Hình 3.17:** Sơ đồ nguyên lý của IC L298D

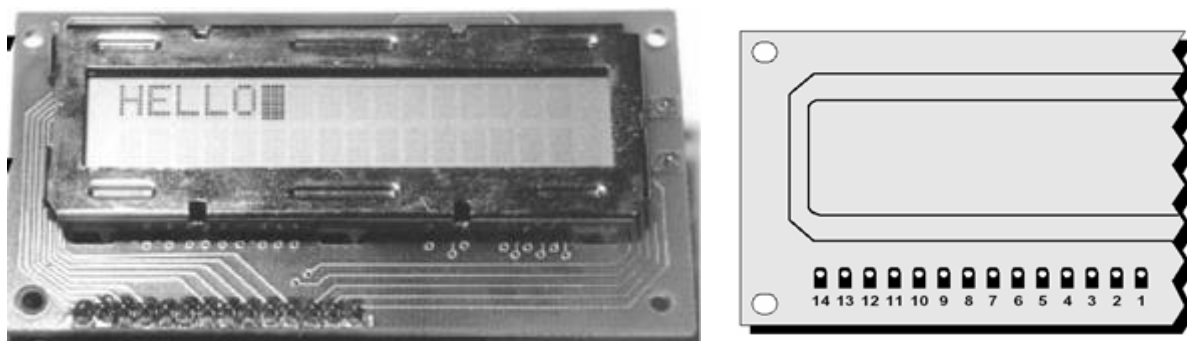
Có 2 mạch cầu H trên mỗi chip L298D nên có thể điều khiển 2 đối tượng riêng với 1 chip này. Mỗi mạch cầu H bao gồm 1 đường nguồn Vs (thật ra là đường chung cho 2 mạch cầu), một chân current sensing (cảm biến dòng) ở phần cuối của mạch cầu H, chân này không được nối đất mà bỏ trống để cho người dùng nối 1 điện trở nhỏ gọi là sensing resistor. Bằng cách đo điện áp rơi trên điện trở này chúng ta có thể tính được dòng qua điện trở, cũng là dòng qua động cơ, mục đích của việc này là để xác định dòng quá tải. Nếu việc đo lường là không cần thiết thì ta có thể nối chân này với GND. Động cơ sẽ được nối với 2 chân OUT1, OUT2 hoặc OUT3, OUT4. Chân EN (ENA và ENB) cho phép mạch cầu hoạt động, khi chân này được kéo lên mức cao.

L298D không chỉ được dùng để đảo chiều động cơ mà còn điều khiển vận tốc động cơ bằng PWM. Trong thực tế, công suất thực mà L298D có thể tải nhỏ hơn giá trị danh nghĩa của nó ( $U = 50V$ ,  $I = 2A$ ). Để tăng dòng tải của chip lên gấp đôi, chúng ta có thể nối hai mạch cầu H song song với nhau (các chân có chức năng như nhau của 2 mạch cầu được nối chung).

### **3..2.3 LCD**

#### **3.2.3.1 Chức năng và hình dạng LCD.**

Ngày nay, thiết bị hiển thị LCD (Liquid Crystal Display) được sử dụng trong rất nhiều các ứng dụng của vi điều khiển. LCD có rất nhiều ưu điểm so với các dạng hiển thị khác: nó có khả năng hiển thị kí tự đa dạng, trực quan (chữ, số và kí tự đồ họa), dễ dàng đưa vào mạch ứng dụng theo nhiều giao thức giao tiếp khác nhau, tốn rất ít tài nguyên hệ thống và giá thành rẻ ...



**Hình 3.17** : LCD và sơ đồ chân

### 3.2.3.2 Chức năng các chân

Chân số	Tên	Chức năng
1	$V_{SS}$	Chân nối đất cho LCD, khi thiết kế mạch ta nối chân này với GND của mạch điều khiển
2	$V_{DD}$	Chân cấp nguồn cho LCD, khi thiết kế mạch ta nối chân này với $V_{CC}=5V$ của mạch điều khiển
3	$V_{ee}$	Chân này dùng để điều chỉnh độ tương phản của LCD.
4	RS	Chân chọn thanh



		<p>ghi (Register select). Nối chân RS với logic “0” (GND) hoặc logic “1” (<math>V_{CC}</math>) để chọn thanh ghi.</p> <p>+ Logic “0”: Bus DB0-DB7 sẽ nối với thanh ghi lệnh IR của LCD (ở chế độ “ghi” - write) hoặc nối với bộ đếm địa chỉ của LCD (ở chế độ “đọc” - read) .</p> <p>+ Logic “1”: Bus DB0-DB7 sẽ nối với thanh ghi dữ liệu DR bên trong LCD</p>
5	R/W	<p>Chân chọn chế độ đọc/ghi (Read/Write). Nối chân R/W với logic “0” để LCD hoạt động ở chế độ ghi, hoặc nối với logic “1” để LCD ở chế độ đọc.</p>
6	E	<p>Chân cho phép (Enable). Sau khi các tín hiệu được đặt lên bus DB0-DB7, các lệnh chỉ được chấp nhận khi có 1 xung cho phép của chân E.</p> <p>+ Ở chế độ ghi: Dữ liệu ở bus sẽ được LCD chuyển vào (chấp nhận) thanh ghi bên trong khi phát hiện một xung (high-to-low transition) của tín hiệu chân E.</p> <p>+ Ở chế độ đọc: Dữ liệu sẽ được LCD xuất ra DB0-DB7 khi phát hiện cạnh lên (low-to-high</p>

		transition) ở chân E và được LCD giữ ở bus đến khi nào chân E xuống mức thấp.
7-14	DB0-DB7	<p>Tám đường của bus dữ liệu dùng để trao đổi thông tin với MPU. Có 2 chế độ sử dụng 8 đường bus này :</p> <p>+ Chế độ 8 bit: Dữ liệu được truyền trên cả 8 đường, với bit MSB là bit DB7.</p> <p>+ Chế độ 4 bit: Dữ liệu được truyền trên 4 đường từ DB4 tới DB7, bit MSB là DB7.</p>

**Bảng 1.** Chức năng các chân của LCD

\* Ghi chú: Ở chế độ “đọc”, nghĩa là MPU sẽ đọc thông tin từ LCD thông qua các chân DBx.

Còn khi ở chế độ “ghi”, nghĩa là MPU xuất thông tin điều khiển cho LCD thông qua các chân DBx.

### **3.2.3.3 Đặc tính điện của các chân giao tiếp:**

LCD sẽ bị hỏng nghiêm trọng, hoặc hoạt động sai lệch nếu bạn vi phạm khoảng đặc tính điện sau đây:

Chân cấp nguồn (Vcc-GND)	Min:-0.3V , Max+7V
Các chân ngõ vào (DBx, E, ...)	Min:-0.3V , Max:(Vcc+0.3V)
Nhiệt độ hoạt động	Min:-30C , Max:+75C
Nhiệt độ bảo quản	Min:-55C , Max:+125C

### 3.2.3.4 Tập lệnh của LCD

Các lệnh của LCD có thể chia thành 3 nhóm như sau:

□ Các lệnh về kiểu hiển thị.

VD: Kiểu hiển thị (1 hàng / 2 hàng), chiều dài dữ liệu (8 bit / 4 bit), ...

□ Chỉ định địa chỉ RAM nội.

□ Nhóm lệnh truyền dữ liệu trong RAM nội.

Với mỗi lệnh, LCD cần một khoảng thời gian để hoàn tất, thời gian này có thể khá lâu đối với tốc độ của MPU, nên ta cần kiểm tra cờ BF hoặc đợi (delay) cho LCD thực thi xong lệnh hiện hành mới có thể ra lệnh tiếp theo.

**Bảng 2.** Các lệnh của LCD

Tên lệnh	Hoạt động	T <sub>exe</sub> (max)
<b>Clear Display</b>	<p><b>Mã lệnh :</b> DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0</p> <p style="text-align: center;">DBx = 0 0 0 0 0 0 0 0</p> <p style="text-align: center;">1</p> <p>Lệnh Clear Display (x a hiển thị) sẽ ghi một khoảng trống-blank (mã hiển kí tự 20H) vào tất cả ô nhớ trong DDRAM, sau đó trả bộ đếm địa chỉ AC=0, trả lại kiểu hiển thị gốc nếu nó bị thay đổi. Nghĩa là : Tắt hiển thị, con trỏ dời về góc trái (hàng đầu tiên), chế độ tăng AC.</p>	
<b>Return home</b>	<p><b>Mã lệnh :</b> DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0</p> <p style="text-align: center;">DBx = 0 0 0 0 0 0 0 1</p> <p style="text-align: center;">*</p> <p>Lệnh Return home trả bộ đếm địa chỉ AC về 0, trả lại kiểu</p>	1.52 ms

	hiển thị gốc nếu n bị thay đổi. Nội dung của DDRAM không thay đổi.	
<b>Entry mode set</b>	<p><b>Mã lệnh :</b> DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0</p> <p style="text-align: center;">DBx = 0 0 0 0 0 1 [I/D]</p> <p>[S]</p> <p><b>I/D :</b> Tăng (I/D=1) hoặc giảm (I/D=0) bộ đếm địa chỉ hiển thị AC 1 đơn vị mỗi khi có hành động ghi hoặc đọc vùng DDRAM. Vị trí con trỏ cũng di chuyển theo sự tăng giảm này.</p> <p><b>S :</b> Khi S=1 toàn bộ nội dung hiển thị bị dịch sang phải (I/D=0) hoặc sang trái (I/D=1) mỗi khi c hành động ghi vùng DDRAM. Khi S=0: không dịch nội dung hiển thị. Nội dung hiển thị không dịch khi đọc DDRAM hoặc đọc/ghi vùng CGRAM.</p>	37 us
<b>Display on/off control</b>	<p><b>Mã lệnh :</b> DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0</p> <p style="text-align: center;">DBx = 0 0 0 0 1 [D] [C]</p> <p>[B]</p> <p><b>D:</b> Hiển thị màn hình khi D=1 và ngược lại. Khi tắt hiển thị, nội dung DDRAM không thay đổi.</p> <p><b>C:</b> Hiển thị con trỏ khi C=1 và ngược lại. Vị trí và hình dạng con trỏ, xem hình 8</p> <p><b>B:</b> Nhấp nháy kí tự tại vị trí con trỏ khi B=1 và ngược lại. Xem thêm hình 8 về kiểu nhấp nháy. Chu kì nhấp nháy khoảng 409,6ms khi mạch dao động nội LCD là 250kHz</p>	37 us

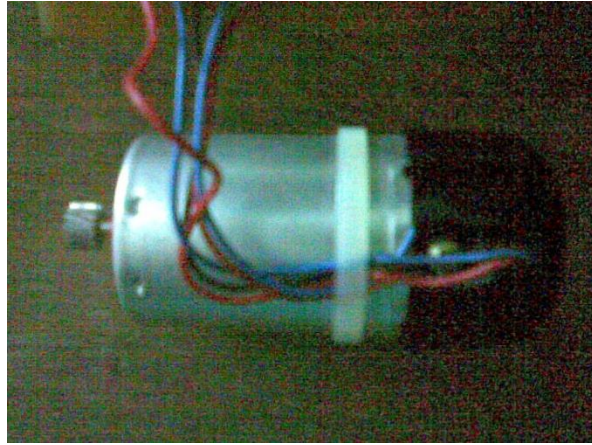


	<i>thị 2 hàng</i>	
<b>Set CGRAM address</b>	<p><b>Mã lệnh :</b> DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0</p> <p>DBx = 0 1</p> <p>[ACG][ACG][ACG][ACG][ACG][ACG]</p> <p>Lệnh này ghi vào AC địa chỉ của CGRAM. Kí hiệu [ACG] chỉ 1 bit của chuỗi dữ liệu 6 bit. Ngay sau lệnh này là lệnh đọc/ghi dữ liệu từ CGRAM tại địa chỉ đã được chỉ định</p>	37 us
<b>Set DDRAM address</b>	<p><b>Mã lệnh :</b> DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0</p> <p>DBx = 1 [AD] [AD] [AD] [AD] [AD] [AD] [AD]</p> <p>Lệnh này ghi vào AC địa chỉ của DDRAM, dùng khi cần thiết lập tọa độ hiển thị mong muốn. Ngay sau lệnh này là lệnh đọc/ghi dữ liệu từ DDRAM tại địa chỉ đã được chỉ định.</p> <p>Khi ở chế độ hiển thị 1 hàng: địa chỉ c thể từ 00H đến 4FH.</p> <p>Khi ở chế độ hiển thị 2 hàng, địa chỉ từ 00h đến 27H cho hàng thứ nhất, và từ 40h đến 67h cho hàng thứ 2.</p> <p>Xem chi tiết ở hình 4.</p>	37 us
<b>Read BF and address</b>	<p><b>Mã lệnh :</b> DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0</p> <p>DBx = [BF] [AC] [AC] [AC] [AC] [AC] [AC] [AC] (RS=0, R/W=1)</p> <p>Như đã đề cập trước đây, khi cờ BF bật, LCD đang làm việc và lệnh tiếp theo (nếu có) sẽ bị bỏ qua nếu cờ BF chưa về mức thấp. Cho nên, khi lập trình điều khiển, bạn phải kiểm tra cờ BF trước khi ghi dữ</p>	0 us

	<p>liệu vào LCD.</p> <p>Khi đọc cờ BF, giá trị của AC cũng được xuất ra các bit [AC]. N là địa chỉ của CG hay DDRAM là tùy thuộc vào lệnh trước đọc</p>	
<p><b>Write data to CG or DDRAM</b></p>	<p><b>Mã lệnh :</b> DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0</p> <p>DBx = [Write data]</p> <p>(RS=1, R/W=0)</p> <p>Khi thiết lập RS=1, R/W=0, dữ liệu cần ghi được đưa vào các chân DBx từ mạch ngoài sẽ được LCD chuyển vào trong LCD tại địa chỉ được xác định từ lệnh ghi địa chỉ trước đó (lệnh ghi địa chỉ cũng xác định luôn vùng RAM cần ghi)</p> <p>Sau khi ghi, bộ đếm địa chỉ AC tự động tăng/giảm 1 tùy theo thiết lập Entry mode.</p> <p>Lưu ý là thời gian cập nhật AC không tính vào thời gian thực thi lệnh.</p>	37 us
<p><b>Read data from CG</b></p>	<p><b>Mã lệnh :</b> DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0</p> <p>DBx = [Read data] (RS=1, R/W=1)</p>	37 us

### 3.2.4 Đối tượng điều khiển: Động cơ DC

- Đây là động cơ sử dụng trong đề tài:

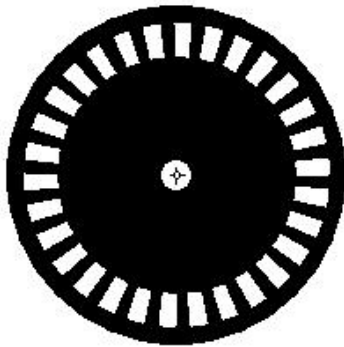


**Hình 3.18:** Động cơ DC có gắn encoder sử dụng trong đồ án

- Bên trong động cơ có gắn một encoder đồng trục với nó dùng để xác định tốc độ và vị trí của động cơ.
- Các thông số của động cơ như sau:
  - + Điện áp DC cấp cho động cơ: 12VDC
  - + Tốc độ tối đa 2000 vòng/phút
  - + Số xung của encoder 60xung/vòng
  - + Điện cảm  $L=102\text{mH}$
- Động cơ có tất cả 5 dây ra:
  - + 2 dây cung cấp nguồn 12 V cho động cơ
  - + 2 dây nguồn 5V cung cấp nguồn cho encoder
  - + 1 dây tín hiệu đưa xung encoder ra ngoài
- Phương pháp điều khiển: Thay đổi tốc độ động cơ bằng cách thay đổi áp cấp vào cho động cơ.
- Nguyên lý hoạt động của cảm biến encoder: có nhiều loại encoder khác nhau. Mỗi loại lại có một nguyên lý hoạt động khác nhau, trong khuôn khổ báo cáo đồ án, em xin trình bày phần nguyên lý loại encoder trong đề tài mà em sử dụng: incremental encoder.

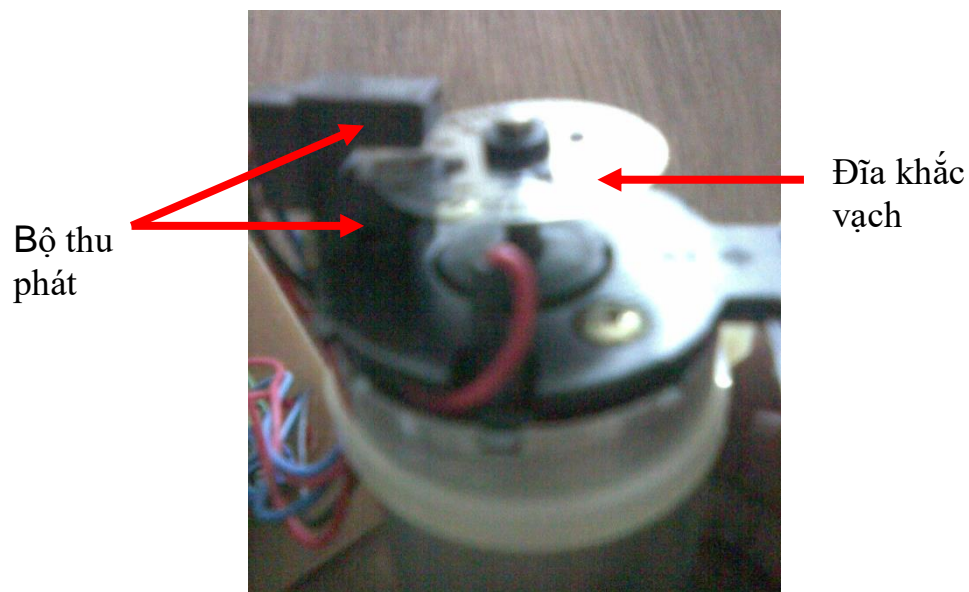


### Mô hình thứ 1



**Hình 3.19:** Encoder

Incremental encoder về cơ bản là một đĩa tròn quay quanh một trục được đục lỗ như hình trên.



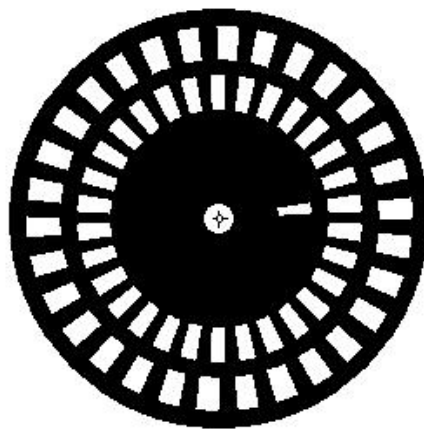
**Hình 3.20:** Cấu tạo Encoder

Ở 2 bên mặt của cái đĩa tròn đó, sẽ có một bộ thu phát quang. Trong quá trình encoder quay quanh trục, nếu gặp lỗ rỗng thì ánh sáng chiếu qua được, nếu gặp mảnh chắn thì tia sáng không chiếu qua được. Do đó tín hiệu nhận được từ sensor quang là một chuỗi xung. Mỗi encoder được chế tạo sẽ biết sẵn số xung trên một vòng. Do

đó ta có thể dùng vi điều khiển đếm số xung đó trong một đơn vị thời gian và tính ra tốc độ động cơ.

Encoder mà em sử dụng trong đồ án của mình, hoàn toàn giống với mô hình ở trên. Tuy nhiên, mô hình trên có nhược điểm lớn là: ta không thể xác định được động cơ quay trái hay quay phải, vì có quay theo chiều nào đi nữa thì chỉ có một dạng xung đưa ra. Ngoài ra điểm bắt đầu của động cơ, ta cũng không thể nào biết được.

Cải tiến mô hình 1 bằng mô hình 2 như sau:



**Hình 3.21:** Đĩa khắc 2 vòng vạch lệch pha nhau  $90^\circ$

- Trong mô hình này, người ta đục tất cả là 2 vòng lỗ. Vòng ngoài cùng giống như mô hình 1, vòng giữa lệch pha so với vòng ngoài là  $90^\circ$ .

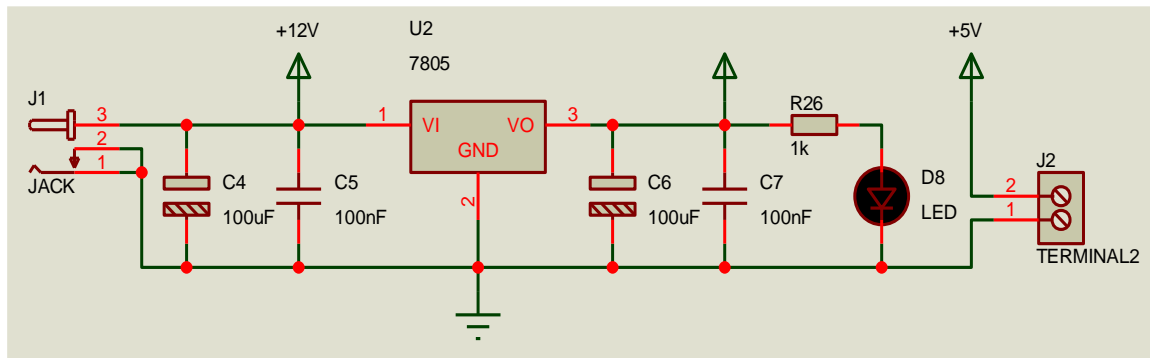
### **3.3 THIẾT KẾ MẠCH PHẦN CỨNG CODE CHƯƠNG TRÌNH VÀ LƯU ĐỒ GIẢI THUẬT**

#### **3.3.1 Thiết kế mạch phần cứng**

Mạch được thiết kế gồm có các khối như sau: khối nguồn, khối bàn phím, khối hiển thị, khối mạch công suất, khối điều khiển.

##### **3.3.1.1 Khối nguồn:**

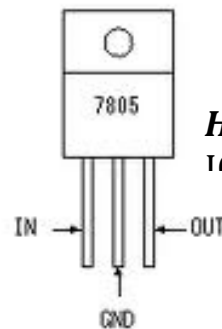
Mạch lấy nguồn xoay chiều qua adapter AC/DC 220VAC/12VDC, và được ổn áp nhờ IC 7805. Sơ đồ nguyên lý mạch:



**Hình 3.22:** Khối mạch ổn áp

Chức năng của các phần tử trong mạch:

- IC 7805: chức năng ổn áp điện áp 5V

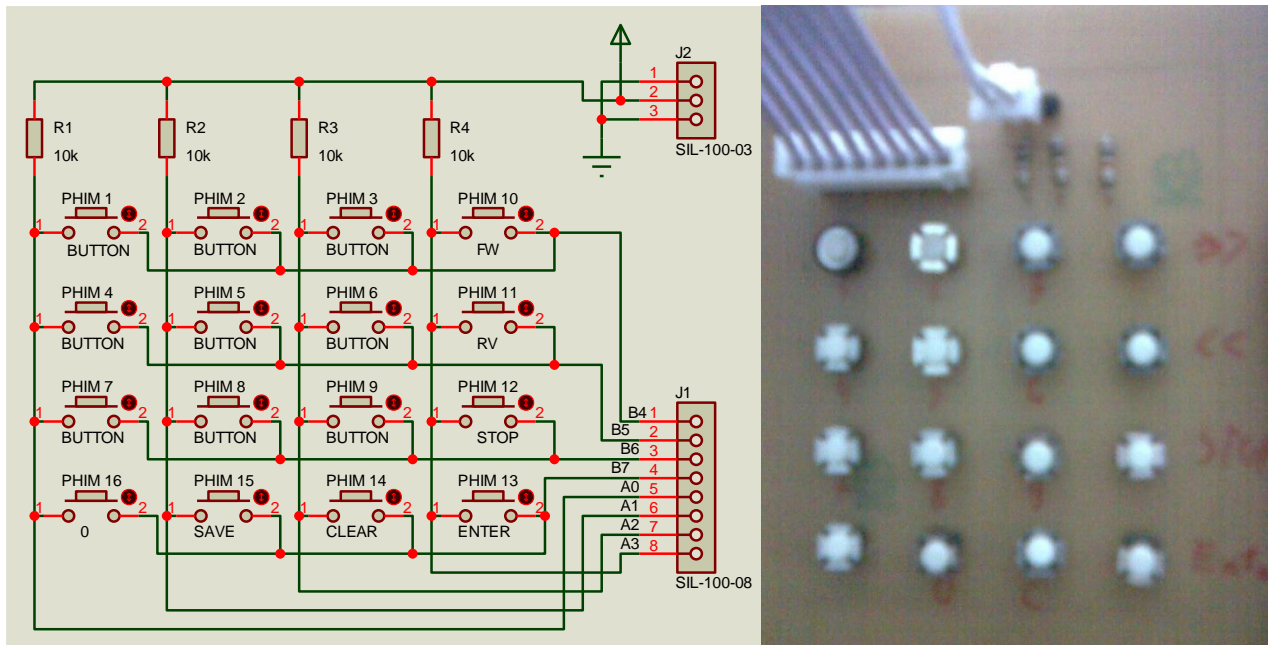


**Hình 3.23:**  
IC 7805

- C4 tụ hóa (có phân cực) ổn áp ngõ vào, điện dung của tụ này càng lớn thì điện áp vào IC 7805 càng phẳng.
- C5 và C7 tụ giấy (không phân cực) là hai tụ lọc nhiễu tần số cao ở ngõ vào và ngõ ra.
- C6 tụ hóa có tác dụng dập dao động tự kích khi sử dụng IC ổn áp dòng 78xx.

### 3.3.1.2 Khối mạch bàn phím:

Mạch bàn phím gồm 16 phím, được bố trí thành 4 hàng và 4 cột như hình vẽ:



**Hình 3.24:** Khối mạch bàn

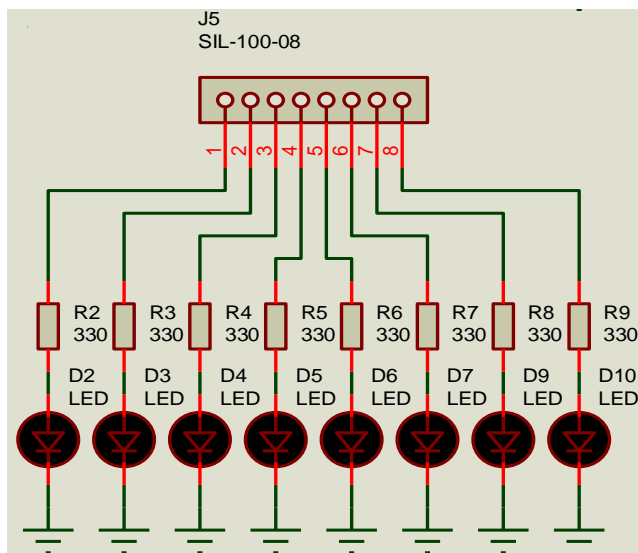
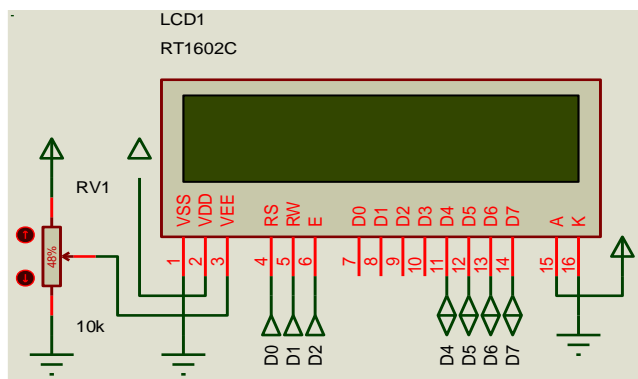
Bốn hàng được nối với Port B (từ B4 đến B7) của vi xử lý, bốn cột được nối với Port A (từ A0 đến A3) của vi xử lý và được nối với nguồn VCC qua điện trở (nghĩa là các chân của Port A từ A0 đến A3 sẽ luôn nhận mức 1 khi không có phím nhấn), việc làm này nhằm phục vụ cho giải thuật quét phím sẽ được trình bày ở phần sau.

Trong số 16 phím có 10 phím để nhập dữ liệu số từ 0 đến 9, 6 phím điều khiển (FW, RV, STOP, ENTER, CLEAR, SAVE).

Nguồn VCC trong sơ đồ mạch bàn phím là 5V, được cấp từ khối mạch ổn áp ở trên. Vì mạch bàn phím truyền nhận dữ liệu trực tiếp với PIC nên cần điện áp ổn định.

### 3.3.1.3 Khối mạch hiển thị:

Mạch hiển thị bao gồm màn hình LCD giao tiếp với PIC qua Port D với giao thức 4 bit, ngoài ra còn có biến trở để điều chỉnh độ sáng của LCD.

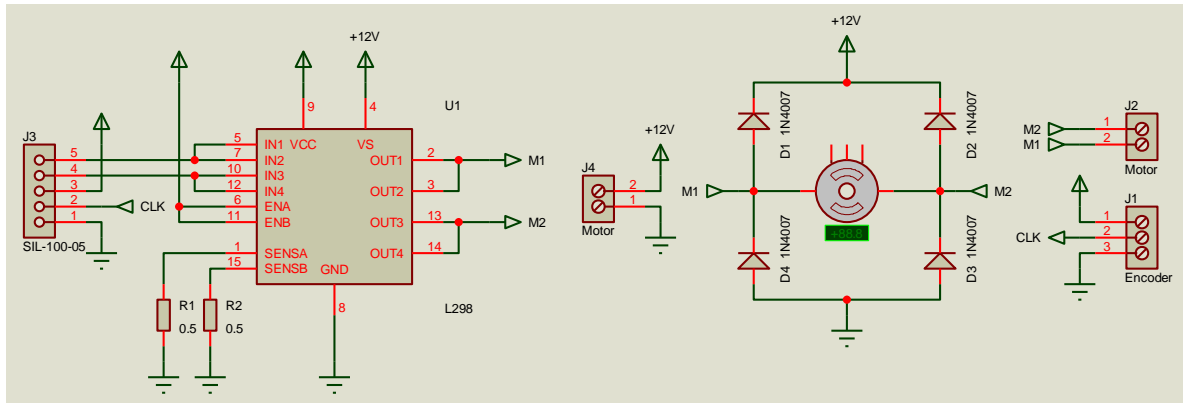


**Hình 3.25:** Khối mạch LCD và LED đơn

Ngoài ra, còn có khối hiển thị gồm 8 led đơn được hạn dòng bằng điện trở  $220\Omega$ , có tác dụng để text chương trình, các khối khác và cũng được dùng để báo hiệu đảo chiều trong chế độ có cài đặt thời gian.

#### 3.3.1.4 Khối mạch công suất:

– Mạch công suất sử dụng IC cầu H L298, với 2 kênh A và B, mỗi kênh với điện áp định mức 50V và dòng định mức cho tải là 2A. Khi đấu song song 2 kênh ta được dòng cấp cho tải lên đến 4A (gấp đôi). Điện áp điều khiển 5V.



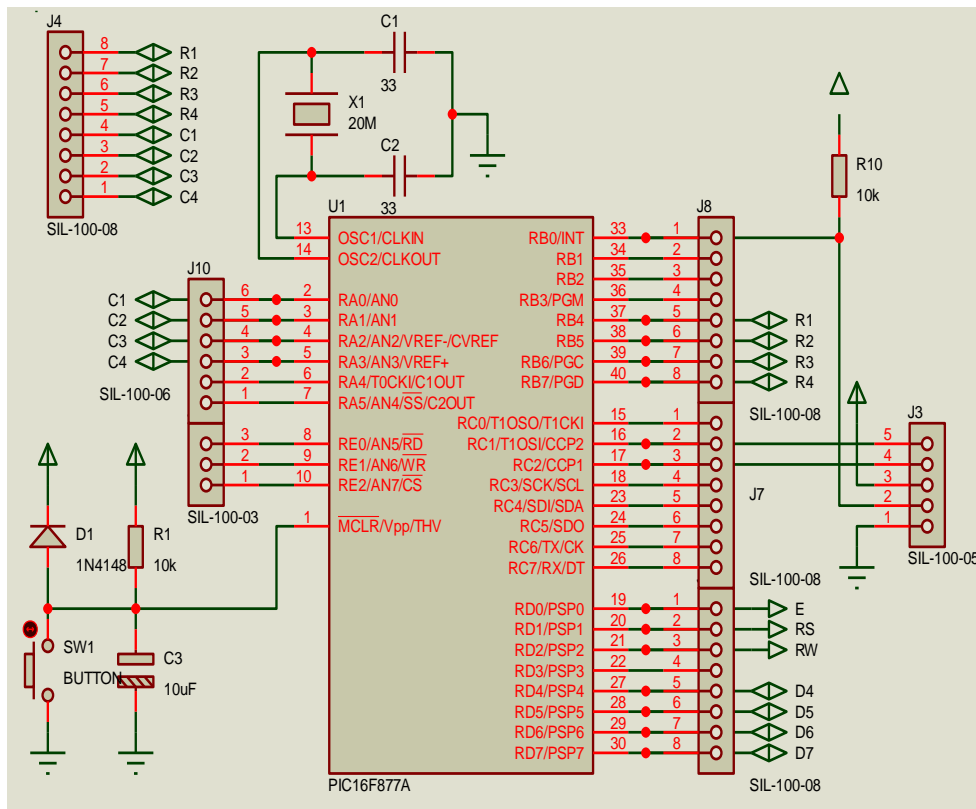
**Hình 3.26:** Khối mạch công suất sử dụng IC L298

- Cầu Diode dùng để chống dòng điện ngược, do tải động cơ có tính chất cảm kháng. Nguồn cấp cho động cơ 12V
- Sử dụng IC cầu H này, không những dùng để đảo chiều động cơ mà còn điều khiển tốc độ động cơ bằng phương pháp băm xung (PWM).

### 3.3.1.5 Khối mạch điều khiển:

Vì điều khiển trung tâm là PIC 16F877A. Với chức năng của các port như sau:

- Port A (từ A0 đến A3) được set là ngõ vào để nhận tín hiệu từ 4 cột của bàn phím.
- Port B (từ B3 đến B7) là ngõ ra xuất tín hiệu ra 4 hàng của khối bàn phím, chân RB0 nhận tín hiệu xung từ Encoder (vì ngắt ngoài xảy ra khi tín hiệu thay đổi trên chân RB0);
- Ta sử dụng 2 chân RC1(CCP2) và RC2 (CCP1) của Port C để xuất tín hiệu PWM điều khiển động cơ.
- Port D (trừ chân RD3) gửi tín hiệu đến khối hiển thị LCD. Ba chân từ RD0 đến RD2 nối với 3 chân điều khiển của LCD. Bốn chân từ RD4 đến RD7 nối với 4 bit cao của các chân nhận dữ liệu của LCD.



**Hình 3.27:** Khối mạch điều khiển PIC 16F877A

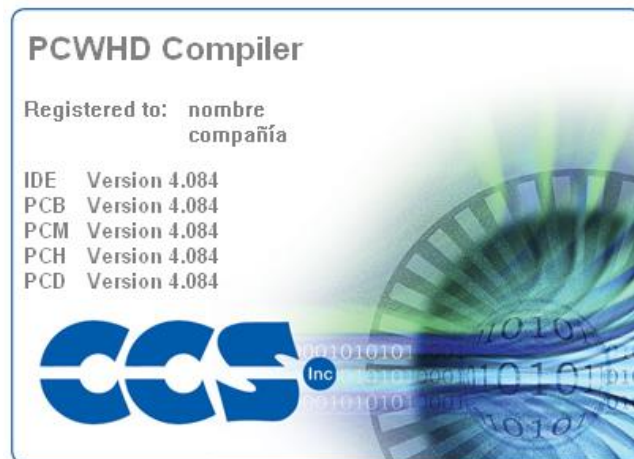
Thạch anh dùng trong mạch có giá trị 20MHz, với 2 tụ C1,C2 là tụ giấy với điện dung 33uF. Pic reset khi chân số 1 MCLR nối mass. Tụ C3 có tác dụng chống nhiễu ở chân số 1.

### 3.3.2 Giới thiệu về chương trình viết code và biên dịch:

Trong đồ án này nhóm chúng em sử dụng chương trình viết code CCS, chương trình cho phép lập trình ngôn ngữ C cho vi điều khiển PIC của Microchip.



Chương trình này có các câu lệnh đơn giản, dễ hiểu, hỗ trợ biên dịch với chức năng hiển thị lỗi và cảnh báo chương trình không khả dụng. Ngoài ra nó còn có thể biên dịch từ code C ra file.hex và cả code Assemble



Cấu trúc 1 chương trình trong CCS:

- Đầu tiên là các chỉ thị tiền xử lý : ( # . . . ) có nhiệm vụ báo cho CCS cần sử dụng những gì trong chương trình C như dùng vì xử lý gì , có dùng giao tiếp PC không , ADC không , DELAY không , có sử dụng ngắt hay không...
- Các khai báo biến.
- Các hàm con.
- Các hàm phục vụ ngắt theo sau bởi 1 chỉ thị tiền xử lý cho biết dùng ngắt nào.
- Chương trình chính.

### 3.3.3 Lưu đồ giải thuật:

Lưu đồ giải thuật gồm: lưu đồ chương trình chính, chương trình quét phím, chương trình ngắt của timer1.

- Chương trình chính là 1 vòng lặp vô hạn có những chương trình con như: quét phím, check phím, chọn chế độ, tính PWM, nhập dữ liệu tốc độ, nhập dữ liệu thời gian, save vào eeprom.



- Chương trình quét phím thể hiện giải thuật nhận phím nhấn và nhận biết giá trị của phím (phím nhấn là phím nào) .
- Chương trình ngắt timer 1 có tác dụng cập nhật giá trị tốc độ, tính toán giá trị PWM, và xuất tín hiệu đến khối hiển thị.

### CODE Chương Trình

```
#include"E:/TL/DOAN2\cauH\MACH\15\main.h"
#include <LCD.C>
//#include <math.h> //dung cac ham toan hoc
#include delay(clock=20000000)
#include fast_io(a)
#include fast_io(b)
#include fast_io(c)
#include fast_io(d)
//=====khai bao ham con=====
int quetphim();
int checkphim(b);
void pwm();
void ghi_tocdo();
void ghi_thoigian();
void clear();
void read_rom();
//=====khai bao bien=====
int8 i,t,a,b,c,d,sttphim,duty,ct,l,m;
int16 s_xung,s_vong,setpoint,error,luu,tg,tg1,dem;
//===== bien luu eeprom=====
int8 e0,e1,e2,e3,j,k;
int16 e;
//=====chuong trinh quet phim=====
//quet phim so
int quetphim()
{
output_b(0xe0);// B4=0
a=0;
b=1;
checkphim(b);
if (a!=0)
{delay_ms(200);
return (sttphim);}

output_b(0xd0);// B5=0
a=0;
```

```

b=2;
checkphim(b);
if (a!=0)
{ delay_ms(200);
return (sttphim);}
output_b(0xb0);// B6=0
a=0;
b=3;
checkphim(b);
if (a!=0)
{ delay_ms(200);
return (sttphim);}
output_b(0x70);// B6=0
a=0;
b=4;
checkphim(b);
if (a!=0)
{ delay_ms(200);
return (sttphim);} }

//=====chuong trinh check phim=====
int checkphim(b)
{
switch (b)
{
case 1:
if(!input(pin_a0))
{ sttphim=1;
a=1;}
else if(!input(pin_a1))
{ sttphim=2;
a=1;}
else if(!input(pin_a2))
{ sttphim=3;
a=1;}
else if(!input(pin_a3))
{ sttphim=10;//thuan
a=1;}

else { }
break;

case 2:

```

```

if(!input(pin_a0))
{ sttphim=4;
a=1;}
else if(!input(pin_a1))
{ sttphim=5;
a=1;}
else if(!input(pin_a2))
{ sttphim=6;
a=1;}
else if(!input(pin_a3))
{ sttphim=11;//nghich
a=1;}
else { }
break;
case 3:
if(!input(pin_a0))
{ sttphim=7;
a=1;}
else if(!input(pin_a1))
{ sttphim=8;
a=1;}
else if(!input(pin_a2))
{ sttphim=9;
a=1;}
else if(!input(pin_a3))
{ sttphim=12;//stop
a=1;}
else { }
break;

case 4:
if(!input(pin_a0))
{ sttphim=0;//0
a=1;}
else if(!input(pin_a1))
{ sttphim=15;//save

a=1;}
else if(!input(pin_a2))
{ sttphim=14;//clear
a=1;}
else if(!input(pin_a3))
{ sttphim=13;//set

```

```

a=1;}
else {}
break;}
return (sttphim);
}
// chuong trinh nhan xung tu encoder
//ngat ngoai, nhan xung tu encoder
#include
void RB0_isr()
{
s_xung++;//dem so xung o chan RB0
}
//ngat timer1, tinh toan pwm va hien thi
#include
void timer1_isr()
{
set_timer1(-62500);
if (t==5){
S_vong=s_xung*2;//xung tren phut
//ht
lcd_gotoxy(1,1);
printf(lcd_putc,"          ");
if(ct==1)
{
lcd_gotoxy(1,1);
printf(lcd_putc,"TD_dat=%lu v/p",luu);}
else {
lcd_gotoxy(1,1);
printf(lcd_putc,"TG_dat=%lu s",tg);}
lcd_gotoxy(1,2);
printf(lcd_putc,"          ");
if (d==2)
{lcd_gotoxy(1,2);
printf(lcd_putc,"TD_tt=-%luv/p",s_vong);}

if (d==1)
{lcd_gotoxy(1,2);
printf(lcd_putc,"TD_tt= %luv/p",s_vong);}
//=====
//xuat pwm
pwm();
//=====
s_xung=0;

```

```

s_vong=0;
t=0;
set_timer1(-62500);
}
else
t++;
set_timer1(-62500);
}
//ngat timer0, dat thoi gian
#int_timer0
void time0_irs()
{
set_timer0(-235);
if (dem>=tg1)
{
output_high(pin_e0);
delay_us(100);
output_low(pin_e0);
dem=0;
if (d==1)
{d=2;}
else if (d==2)
{d=1;}
else{ }
set_timer0(-235);
}
else {
dem++;
set_timer0(-235);}
}
//chuong trinh chinh
void main()

{
i=0;
sttphim=0;
setpoint=0;tg=0;
duty=0;
c=0;
d=0;
ct=0;l=0;m=0;
e0=0;e1=0;e2=0;e3=0;e=0;k=0;

```

```

//=====
=====
//1:ngo vao;0: la ngo ra
set_tris_b(0b00001111);//4 chan RB4-RB7 xuất dữ liệu ra bàn phím
set_tris_a(0b00001111);//4 chan RA0-RA3 nhận dữ liệu từ bàn phím
set_tris_c(0b00000000);//2 chan RC0 và RC1 xuất PWM
set_tris_D(0b00000000);//port D là port xuất dữ liệu ra LCD
//=====
=====
setup_timer_1(T1_INTERNAL|T1_DIV_BY_8);
/* timer1 là bộ định thời sử dụng xung nội,bo chia 1:8 thay đổi mỗi 1600ns
Dùng timer1 để ngắt mỗi 0.1s do vậy ta đặt giá trị cho timer1 là :
0.1s/1600ns=62500(D)=F424(H) => giá trị nạp là FFFF-F424=BDB*/
setup_timer_0(RTCC_INTERNAL|RTCC_DIV_256);
enable_interrupts(int_ext);//khởi động ngắt ngoài
ext_int_edge(H_TO_L); // xung từ cao xuống thấp
enable_interrupts(global);// khởi động bit ngắt GIE
setup_timer_2(T2_DIV_BY_4,249,1);
/*timer2 dùng định thời cho bộ PWM
mode: bo chia thời gian (prescale) của timer2 1:4
period: giá trị nạp chỉ thành ghi PR2
postscale : bo chỉ ra, chọn 1:1 PWM không dùng
Thạch anh 20MHz, PWM fre: 10000Hz, thay đổi duty cycle(%) để thay đổi
tốc độ*/
setup_ccp1(CCP_PWM);
setup_ccp2(CCP_PWM);
set_pwm1_duty(0);
set_pwm2_duty(0);
//=====
=====
lcd_init()
lcd_send_byte(0,0x01);
lcd_gotoxy(1,1);
printf(lcd_putc,"CHUONG TRINH DK ");

lcd_gotoxy(1,2);
printf(lcd_putc," TD DONG CO DC ");
delay_ms(1000);
lcd_gotoxy(1,1);
printf(lcd_putc," ");
lcd_gotoxy(1,2);
printf(lcd_putc," ");
read_rom();

```

```

lcd_gotoxy(1,1);
printf(lcd_putc,"CHON CHE DO :_ ");
while (true)
{
while (c==0)
{
quetphim();
if (a!=0)
{
if (sttphim==1)
{ ct=1;
c=1;
lcd_gotoxy(1,1);
printf(lcd_putc,"CHON CHE DO :_%u",ct);}
if (sttphim==2)
{ ct=2;
c=1;
lcd_gotoxy(1,1);
printf(lcd_putc,"CHON CHE DO :_%u",ct);}
}
}
if (ct==1)
{lcd_gotoxy(1,1);
printf(lcd_putc,"TD_dat=_ v/p");
lcd_gotoxy(1,2);
printf(lcd_putc,"TD_luu=%lu v/p",e);}
else {
lcd_gotoxy(1,1);
printf(lcd_putc,"TD_dat=_ v/p");
lcd_gotoxy(1,2);
printf(lcd_putc,"TG_dat=_ s");}
While (c==1)

{
quetphim();

if (a!=0)
{
if (sttphim>=0 && sttphim<=9)//ban phim tu 0--->9
{
if (m==0)
{ ghi_tocdo();}
if((ct==2)&&(m==1))

```

```

{ ghi_thoigian();}
}
if (sttphim==14)
{clear();}
if ((sttphim==13)&&(i==0)&&(e==0))
{
lcd_gotoxy(1,1);
printf(lcd_putc,"          ");
lcd_gotoxy(1,1);
printf(lcd_putc,"Phai nhap TD_dat");
delay_ms(1000);
lcd_gotoxy(1,1);
printf(lcd_putc,"          ");
lcd_gotoxy(1,1);
printf(lcd_putc,"TD_dat=_");
}
if ((sttphim==15)&&(i!=0)) //luu vao eeprom
{
write_eeprom(0,e0);
delay_ms(100);
write_eeprom(1,e1);
delay_ms(100);
write_eeprom(2,e2);
delay_ms(100);
write_eeprom(3,e3);
delay_ms(100);
write_eeprom(4,i);
delay_ms(100);
lcd_gotoxy(1,2);
printf(lcd_putc,"          ");

lcd_gotoxy(1,2);
printf(lcd_putc," LUU THANH CONG ");
}
if ((sttphim==13)&&((i!=0)||(e!=0)))
{
if (ct==1)
{
c=2;
if (e!=0 && i==0)
luu=e;}
if ((ct==2)&&(l>=2))
{

```



```

c=2;}
m=1;
}}}
while (c==2)
{
quetphim();
if(a!=0)
{
if(sttphim==10)
{
lcd_gotoxy(1,2);
printf(lcd_putc,"          ");
lcd_gotoxy(1,2);
printf(lcd_putc,"  QUAY THUAN  ");
if (ct==2)
{enable_interrupts(int_timer0);
set_timer1(-235);}
enable_interrupts(int_timer1);
enable_interrupts(global);
set_timer1(-62500);
set_pwm1_duty(duty);
d=1;
}
if (sttphim==11)
{
lcd_gotoxy(1,2);
printf(lcd_putc,"          ");
lcd_gotoxy(1,2);

printf(lcd_putc,"  QUAY NGHICH  ");
if (ct==2)
{enable_interrupts(int_timer0);
set_timer1(-235);}
enable_interrupts(int_timer1);
enable_interrupts(global);
set_timer1(-62500);
set_pwm2_duty(duty);
d=2;
}

if (sttphim==12)//stop
{
disable_interrupts(int_timer0);
set_timer0(0);

```

```

disable_interrupts(int_timer1);
set_pwm1_duty(0);
set_pwm2_duty(0);
lcd_gotoxy(1,2);
printf(lcd_putc,"          ");
lcd_gotoxy(1,2);
printf(lcd_putc,"    STOP    ");
duty=0;
d=0;
}
if (sttphim==14)
{
clear();
c=0;
lcd_gotoxy(1,1);
printf(lcd_putc,"          ");
lcd_gotoxy(1,2);
printf(lcd_putc,"          ");
lcd_gotoxy(1,1);
printf(lcd_putc,"CHON CHE DO :_ ");
}}}}

```

```

void pwm()
{
if ((luu>s_vong)&& duty<250)
{error=luu-s_vong;
if (error>1000)

duty=duty+50;
else if (error>100)
duty=duty+20;
else if (error>30)
{duty=duty+5;}
else if (error>20)
duty=duty+1.5;
else if (error>10)
duty=duty+(0.05*error);
else {duty=duty+(0.025*error);}
}
if (luu<(s_vong-2)&& (duty>0))
{error=s_vong-luu;
if (error>=10)

```

```

duty=duty-(0.05*error);
else (duty=duty-(0.02*error));
}
if(luu==s_vong)
duty=duty;
if((duty>=250)&&(luu>s_vong))
{ lcd_gotoxy(1,1);
printf(lcd_putc,"          ");
lcd_gotoxy(1,1);
printf(lcd_putc," TD_tt=MAX  ");}
if (d==0)
{ set_pwm1_duty(0);
set_pwm2_duty(0);}
else if (d==1)
{ set_pwm1_duty(duty);
set_pwm2_duty(0);}
else
{ set_pwm1_duty(0);
set_pwm2_duty(duty);}
}
//=====chuong trinh con nhap toc do=====
void ghi_tocdo()
{
if(i>=0&&i<=3)
{
// giai thuat luu eeprom
if (i==0)
e0=sttphim;
else if (i==1)

e1=sttphim;
else if (i==2)
e2=sttphim;
else {e3=sttphim;}
//=====
setpoint*=10;
setpoint+=sttphim;
i++;
lcd_gotoxy(1,1);
printf(lcd_putc,"          ");
lcd_gotoxy(1,1);
printf(lcd_putc,"TD_dat=%lu v/p",setpoint);
luu=setpoint;

```

```

}
else {
lcd_gotoxy(1,1);
printf(lcd_putc,"          ");
lcd_gotoxy(1,1);
printf(lcd_putc,"0<TD_dat<=9999");
delay_ms(1000);
lcd_gotoxy(1,1);
printf(lcd_putc,"          ");
lcd_gotoxy(1,1);
printf(lcd_putc,"TD_dat=%lu s",setpoint);
}}
//=====chuong trình con nhap thoi gian=====
void ghi_thoigian()
{
if(l>=0&&l<=3)
{
tg*=10;
tg+=sttphim;
l++;
lcd_gotoxy(1,2);
printf(lcd_putc,"          ");
lcd_gotoxy(1,2);
printf(lcd_putc,"TG_dat=%lu s",tg);
tg1=tg/0.12;
}
else {
lcd_gotoxy(1,2);
printf(lcd_putc,"          ");
lcd_gotoxy(1,2);

printf(lcd_putc," 0<TG_dat<=999 ");
delay_ms(1000);
lcd_gotoxy(1,2);
printf(lcd_putc,"          ");
lcd_gotoxy(1,2);
printf(lcd_putc,"TG_dat=%lu s",tg);
}}
//=====chuong trình con clear=====
void clear()
{
setpoint=0;
tg=0;

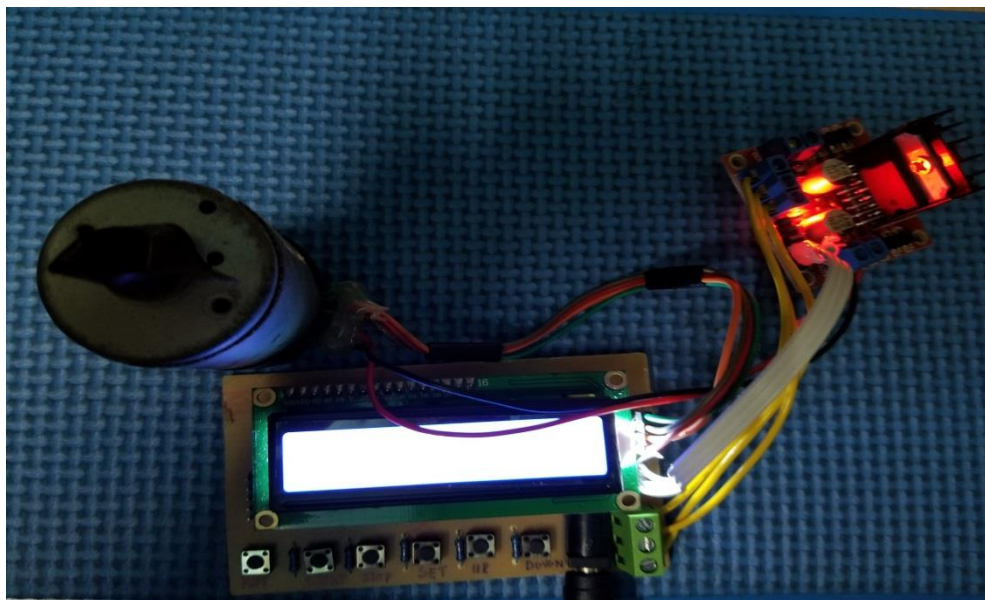
```

```

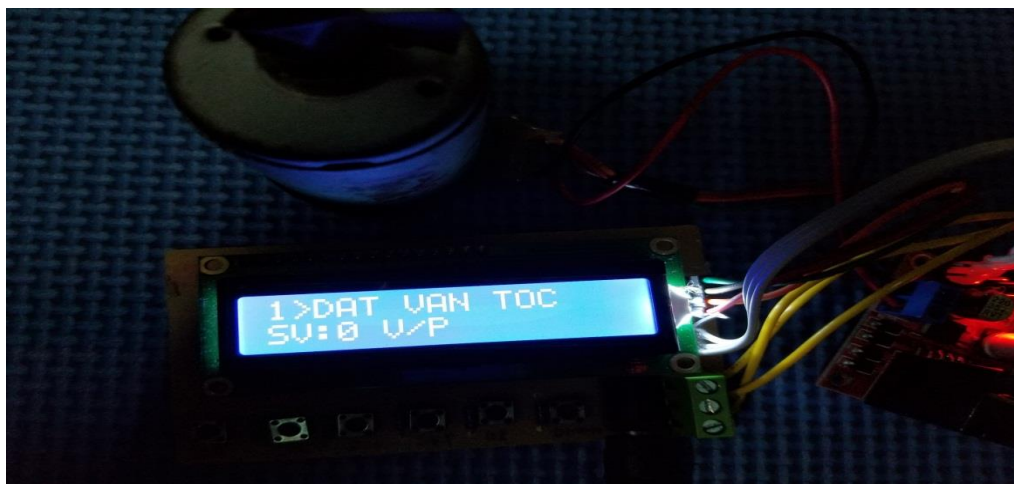
i=0;l=0;m=0;
e0=0;e1=0;e2=0;e3=0;
lcd_gotoxy(1,1);
printf(lcd_putc,"          ");
lcd_gotoxy(1,1);
printf(lcd_putc,"TD_dat=_");
lcd_gotoxy(1,2);
printf(lcd_putc,"          ");
lcd_gotoxy(1,2);
printf(lcd_putc,"TG_dat=_");
}
//=====chuong trinh con luu toc do vao eeprom=====
void read_rom()
{
k=read_eeprom(4);
for (j=0;j<k;j++)
{
e=e*10;
e=e+read_eeprom(j);
}}

```

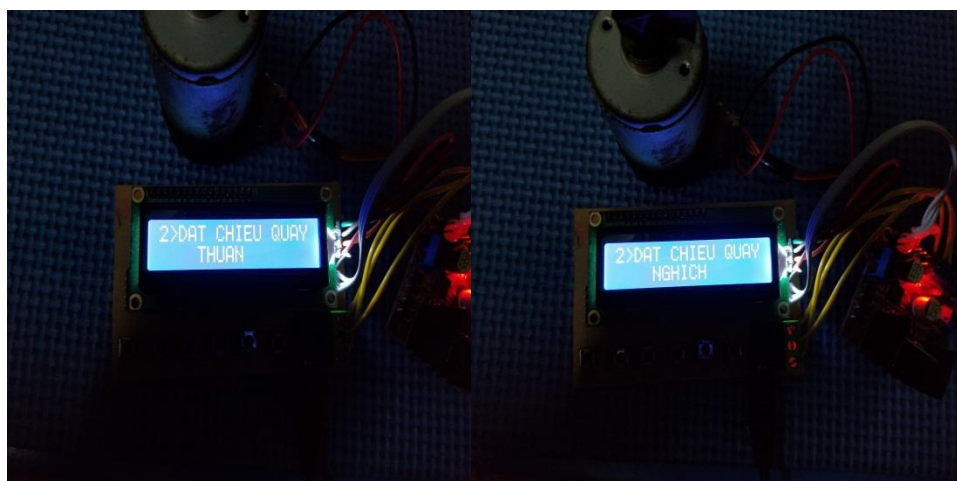
### 3.3.4 Một số các hình ảnh thực tế của bộ điều khiển



Hình 3.28 Bộ điều khiển



Hình 3.29 Đặt vận tốc



Hình 3.30 Đặt chiều quay



Hình 3.31 Màn hình khi vận hành bộ điều khiển

## KẾT LUẬN

Đề tài điều khiển động cơ một chiều sử dụng bộ điều khiển PID tuy không phải là một đề tài mới, nhưng qua đó đã phản ánh được tính nghiêm túc trong việc học hỏi và vận dụng các kiến thức vào việc thực hiện đề tài.

Sau thời gian ba tháng nghiên cứu em đã hoàn thành đề tài với các kết quả đạt được như sau:

- Tìm hiểu về các bộ điều khiển dùng trong hệ thống tự động điều khiển điện.
- Tìm hiểu tổng quan về động cơ một chiều, các phương pháp điều khiển tốc độ động cơ một chiều.
- Thiết kế, chế tạo bộ điều khiển tốc độ động cơ điện một chiều thuật toán PID dùng PIC16F8774.

Tuy nhiên do thời gian và kiến thức có hạn, nên còn một số hạn chế như sau:

- Mạch cầu H sử dụng IC L298 chỉ điều khiển được động cơ DC có công suất nhỏ.
- Đối với khối hiển thị, do tính chất của màn hình LCD nên bị hạn chế quan sát giá trị hiển thị ở khoảng cách xa.

Hướng phát triển đề tài:

- Cải thiện ổn định tốc độ động cơ bằng phương pháp PID hay điều khiển mờ.
- Tính toán thiết kế mạch công suất để có thể điều khiển được động cơ có công suất lớn hơn.
- Sử dụng led 7 đoạn để tăng khả năng quan sát của khối hiển thị.
- Kết nối với máy tính, sử dụng visual basic lập trình để điều khiển tốc độ động cơ DC.

## TÀI LIỆU THAM KHẢO

1. Nguyễn Bính (1996), ***Điện tử công suất***. NXB Khoa Học Kỹ Thuật
2. Bùi Quốc Khánh, Nguyễn Văn Liễn, Phạm Quốc Hải, Dương Văn Nghi (2008), ***Điều chỉnh tự động truyền động điện***. NXB Khoa học và kỹ thuật
3. Bùi Quốc Khánh, Nguyễn Văn Liễn (2005), ***Cơ Sở Truyền Động Điện***. NXB Khoa học và kỹ thuật
4. GS.TSKH Thân Ngọc Hoàn (2005), ***Máy Điện***, Nhà xuất bản Xây Dựng.
5. Lê Văn Doanh – Nguyễn Thế Công – Trần Văn Thịnh(2007), ***Điện tử công suất Lý thuyết thiết kế ứng dụng***, Nhà xuất bản Khoa học và kỹ thuật.
6. Dương Thiên Tử (2006), ***Kỹ thuật lập trình C***, Nhà xuất bản Thanh Niên