

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG.....**

# **Luận văn**

**Xây dựng hệ thống  
điều chỉnh truyền động điện dùng  
bộ điều chỉnh vạn năng**

## LỜI MỞ ĐẦU

Lý thuyết điều khiển tự động đã trải qua hơn 3 thế kỷ hình thành và phát triển, ngày nay cùng với sự phát triển và khoa học công nghệ đã cho phép ứng dụng được những kết quả của Lý Thuyết Điều Khiển Tự Động hiện đại như điều khiển tối ưu, điều khiển mờ, trí tuệ nhân tạo... để tạo nên các bộ điều khiển chất lượng cao sử dụng trong công nghiệp.

Hiện nay trên thế giới đã có bán sẵn những bộ điều khiển số ứng dụng lý thuyết điều khiển kinh điển, đó là các bộ điều khiển PID, với các bộ điều khiển này đã đáp ứng được phần lớn các yêu cầu điều khiển thực tế. Tuy nhiên sử dụng các bộ điều khiển này muốn đạt được chất lượng điều khiển cao thì yêu cầu người sử dụng phải có kiến thức tốt về kỹ thuật điều khiển và với mỗi đối tượng điều khiển thì tham số tối ưu của bộ điều khiển PID là khác nhau và người sử dụng phải có hiểu biết tốt mới tìm được các giá trị của tham số này. Vậy đây chính là hạn chế của bộ điều khiển PID kinh điển, nên vấn đề phải xây dựng một bộ điều khiển hiện đại đa năng, thân thiện với người sử dụng tức là các đầu nối và cài đặt thiết bị điều khiển dễ dàng là rất cần thiết.

Xuất phát từ thực tế trên em thực hiện đề tài:” **Xây dựng hệ thống điều chỉnh truyền động điện dùng bộ điều chỉnh vạn năng**”. Với bộ điều khiển này người sử dụng không cần hiểu biết nhiều về lý thuyết điều khiển tự động mà vẫn tổng hợp lên một hệ thống điều khiển vòng kín với chất lượng cao một cách dễ dàng.

Đề án tập trung vào đi sâu nghiên cứu về lý thuyết điều khiển tự động thông thường và hiện đại đặc biệt là lý thuyết điều khiển mờ và bộ điều khiển PID kinh điển, đồng thời cũng tìm hiểu kỹ các công nghệ vi xử lý, vi điều khiển hiện đại đặc biệt là họ vi điều khiển AVR của hãng Atmel để chọn được chip làm thiết bị điều khiển trung tâm cho bộ điều khiển.

Do vậy trong đề án này em thực hiện :” **Xây dựng hệ truyền động điện sử dụng bộ điều khiển vạn năng điều khiển tốc độ động cơ**”.

**Cụ thể đề án gồm có 3 chương trong đó:**

Chương 1: Lý thuyết mờ và các luật điều khiển của bộ điều khiển.

Chương 2: Thiết kế bộ điều chỉnh sử dụng chip AVR

Chương 3: Xây dựng hệ truyền động điện sử dụng bộ điều khiển vạn năng

# CHƯƠNG 1

## LÍ THUYẾT MỜ VÀ CÁC LUẬT ĐIỀU KHIỂN CỦA BỘ ĐIỀU KHIỂN MỜ

---

Đề tài này với mục đích xây dựng nên một bộ điều khiển PID với các tham số có thể tự chỉnh định theo luật điều khiển mờ, vì vậy trước khi đi vào xây dựng bộ điều khiển ta phải đi sâu vào nghiên cứu về lý thuyết điều khiển mờ và thuật toán PID số.

### 1.1. LÝ THUYẾT ĐIỀU KHIỂN MỜ

Hệ logic mờ được sử dụng khi ta hiểu biết về đối tượng không nhiều (thậm chí không có). Xây dựng hệ logic mờ trên cơ sở kinh nghiệm điều khiển hệ thống. Ưu điểm của bộ điều khiển mờ là thiết kế đơn giản, cài đặt đơn giản.

#### 1.1.1. Khái quát chung về logic mờ - Fuzzy logic

Điều khiển mờ là ngành kỹ thuật do nhà toán học người Mỹ Zahde định hướng phát triển vào thập niên 60 của thế kỷ trước. Khi đó Zahde chỉ đưa ra lý thuyết mờ nhằm thay thế, đơn giản hoá các khái niệm đầy tính lý thuyết của xác suất, của quá trình ngẫu nhiên.

Ngày nay, lý thuyết điều khiển đã, đang phát triển rất mạnh mẽ và ứng dụng trong nhiều lĩnh vực cuộc sống. Các phương pháp điều khiển truyền thống thường đòi hỏi người ta phải hiểu biết rõ bản chất của đối tượng cần điều khiển thông qua mô hình toán học, và trong nhiều ứng dụng chúng là các phương trình toán phức tạp với bậc phi tuyến cao. Ngoài ra các đối tượng điều khiển thường nằm trong môi trường có tác động gây nhiễu và người ta rất khó xác định được các đặc tính của đối tượng điều khiển. Những đối tượng phức tạp như vậy thường nằm ngoài khả năng giải quyết của các phương pháp điều khiển truyền thống và trong quá trình tự động hoá người ta phải nhờ vào khả

năng xử lý tình huống của con người. Việc con người có khả năng điều khiển các quá trình như vậy chứng tỏ các quá trình đó đã được phản ánh và mô phỏng đúng đắn bằng mô hình nào đó trong đầu óc của kỹ sư thiết kế hệ thống. Như vậy, mối quan hệ trong các quá trình điều khiển này không phải được biểu thị bằng các mô hình toán học mà bằng mô hình ngôn ngữ với các thông tin không chính xác, không chắc chắn hay nói cách khác là những thông tin “mờ” có tính ước lệ hay định tính cao. Đó chính là cơ sở cho sự ra đời của lý thuyết mờ hiện đại.

Trong rất nhiều bài toán điều khiển, khi mà đối tượng không thể mô tả bởi một mô hình toán học hoặc có thể mô tả song mô hình của nó lại quá phức tạp, cồng kềnh, không ứng dụng được, thì điều khiển mờ chiếm ưu thế rõ rệt. Ngay cả ở những bài toán thành công theo nguyên tắc kinh điển thì việc áp dụng điều khiển mờ vẫn mang lại cho hệ thống sự cải tiến về tính đơn giản, gọn nhẹ.

Một số ưu điểm của phương pháp điều khiển mờ:

- Chỉ dựa trên các thông tin vào ra quan sát được trên các đối tượng điều khiển, không đòi hỏi phải hiểu bản chất để mô hình hoá toán học đối tượng như trong lý thuyết điều khiển truyền thống.
- Miền ứng dụng rộng lớn, đa dạng.
- Khối lượng công việc thiết kế giảm đi nhiều do ta không cần sử dụng mô hình đối tượng, nhờ đó mà trong hầu hết các bài toán ta có thể giảm khối lượng tính toán, thời gian thiết kế và hạ giá thành sản phẩm.
- Ứng dụng tương đối rộng rãi trong nhiều lĩnh vực của cuộc sống, dễ dàng thay đổi phần lập trình.
- Trong hầu hết các trường hợp, bộ điều khiển mờ làm việc ổn định, bền vững và có chất lượng điều khiển tốt.

### **1.1.2. Một số khái niệm về tập mờ - Bộ điều khiển mờ**

#### **1.1.2.1. Định nghĩa**

Tập mờ là một phần mở rộng của tập hợp kinh điển. Tập mờ mô tả các khái niệm mơ hồ, chưa xác định được các giá trị chính xác.

Mỗi phần tử cơ bản  $x$  của tập mờ được gán thêm một giá trị thực  $\mu(x)$  thuộc đoạn  $[0,1]$  để chỉ độ phụ thuộc của phần tử đó vào tập đã cho. Khi độ phụ thuộc bằng 0 thì phần tử cơ bản đó sẽ hoàn toàn không thuộc tập mờ đã chọn ngược lại với độ phụ thuộc bằng 1, phần tử cơ bản đó sẽ thuộc tập hợp với xác suất 100%.

Như vậy, tập mờ là tập của các cặp  $(x, \mu(x))$ . Tập kinh điển  $X$  của phần tử  $x$  được gọi là tập nền của tập mờ. Cho  $x$  chạy khắp trong tập hợp  $X$  ta sẽ có hàm  $\mu(x)$  có giá trị là số bất kỳ trong đoạn  $[0,1]$ , tức là:

$$\mu_F : X \rightarrow [0,1].$$

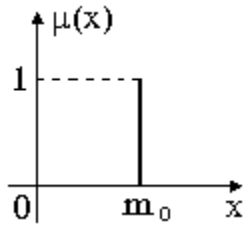
ánh xạ  $\mu_F$  được gọi là hàm liên thuộc hay hàm phụ thuộc của tập mờ  $F$ . Hàm liên thuộc là một đường cong xác định giá trị  $\mu_F$  biến thiên trong đoạn  $[0,1]$ .

Vì hàm thuộc đặc trưng cho tập mờ nên dùng hàm thuộc  $\mu(x)$  đặc trưng cho tập mờ.

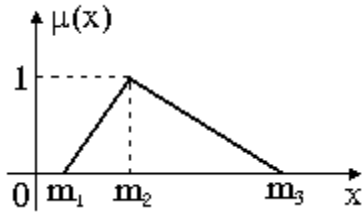
Khi xây dựng bộ điều khiển mờ thì dạng  $\mu(x)$  do người điều khiển tự định đoạt theo kinh nghiệm điều khiển. Về nguyên tắc có thể sử dụng bất kỳ hàm nào thuộc đoạn  $[0,1]$  để làm hàm thuộc. Chẳng hạn hàm  $\text{trapmf}$ ,  $\text{gbellmf}$ ,  $\text{gaussmf}$ ,  $\text{gauss2mf}$ ,  $\text{pimf}$ ,  $\text{dsigmf}$ ,  $\text{psigmf}$ ...

Tuy nhiên, trên thực sử dụng ba dạng hàm phổ biến sau: Hàm Singleton (Hàm Kronecker  $H$ ), Hàm  $\text{trimf}$  (Hàm hình tam giác), Hàm  $\text{trmpf}$  (Hàm hình thang  $H$ ).

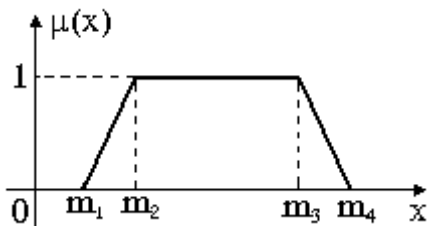
- Hàm Singleton (Hàm Kronecker  $H$ ).



- Hàm trimf (Hàm hình tam giác).



- Hàm trapezf (Hàm hình thang).



Ta có thể sử dụng các dạng hàm  $\mu(x)$  sẵn có hoặc tạo ra dạng hàm liên thuộc mới sao cho quá trình điều khiển là tối ưu. Tuy nhiên trong điều khiển mục đích sử dụng các hàm liên thuộc sao cho khả năng tích hợp chúng là đơn giản. Việc  $\mu(x)$  có giá trị là số bất kỳ trong đoạn  $[0,1]$  là điều khác biệt cơ bản giữa tập kinh điển so với tập mờ.

Đối với tập kinh điển  $A$ , hàm thuộc  $\mu(x)$  chỉ có hai giá trị

$$\mu_A(x) = \begin{cases} 1 & \text{nếu } x \in A \\ 0 & \text{nếu } x \notin A \end{cases} \quad (1-1)$$

### a) Một số thuộc tính của tập mờ

Độ cao của tập mờ: Trong thực tế không phải tập mờ nào cũng có phần tử có hàm liên thuộc bằng 1, do vậy ta có khái niệm về độ cao như sau:

Độ cao của tập mờ F (định nghĩa trên tập nền X) là giá trị:

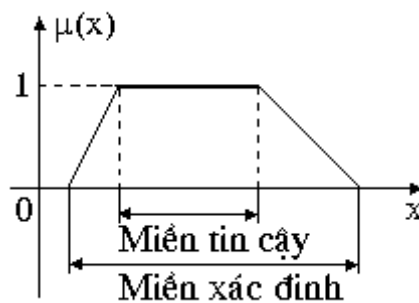
$$h = \sup_{x \in X} \mu_F(x) \quad (1 - 2)$$

Ký hiệu  $\sup_{x \in X} \mu_F(x)$  chỉ giá trị nhỏ nhất trong tất cả các giá trị chặn trên của hàm  $\mu(x)$ . Một tập mờ với ít nhất một phần tử có độ phụ thuộc bằng 1 được gọi là tập mờ chính tắc tức là  $h = 1$ , ngược lại một tập mờ F với  $h < 1$  được gọi là tập mờ không chính tắc.

Miền xác định của tập mờ: Miền xác định của tập mờ F (định nghĩa trên nền X), được ký hiệu bởi S .

$$S = \text{supp } \mu_F(x) = \{x \in X \mid \mu(x) > 0\}. \quad (1 - 3)$$

Ký hiệu  $\text{Supp } \mu_F(x)$  viết tắt của từ tiếng Anh Support, như công thức (1 - 3) đã chỉ rõ, là tập con trong X chứa các phần tử x mà tại đó hàm  $\mu_F(x)$  có giá trị dương.



**Hình 1.1: Miền tin cậy và miền xác định của tập mờ**

Miền tin cậy: Miền tin cậy của tập mờ F (định nghĩa trên nền  $X^{\otimes}$ ), được ký hiệu bởi T là tập con của X thỏa mãn:

$$T = \{x \in X \mid \mu(x) = 1\}. \quad (1 - 4)$$

## b) Biến ngôn ngữ



Trong thực tế xã hội và đặc biệt trong hoạt động tư duy của con người, nhiều vấn đề hoặc hiện tượng không được đánh giá bằng một số cụ thể mà lại đánh giá bằng một khái niệm ước lượng, tuy vậy lại rất phù hợp với thực tế. Về mặt toán học khái niệm ước lượng này được gọi là biến ngôn ngữ. Với mỗi biến ngôn ngữ ta có một khoảng xác định của biến vật lý (giá trị cụ thể). Mỗi một giá trị ngôn ngữ của một biến vật lý sẽ được xác định bằng một tập mờ với các hàm liên thuộc tương ứng.

Như vậy, với mỗi biến ta thấy có hai miền giá trị khác nhau là miền các giá trị rõ (giá trị vật lý) và miền các giá trị ngôn ngữ. Với mỗi giá trị ngôn ngữ lại được mô tả bằng một tập mờ có tập xuất xứ là các giá trị vật lý.

#### **1.1.2.2. Các phép toán trên tập mờ**

Những phép toán cơ bản trên tập mờ là phép hợp, phép giao, và phép bù. Giống như định nghĩa về tập mờ, các phép toán trên tập mờ cũng sẽ được định nghĩa thông qua các hàm thuộc, được xây dựng như các hàm thuộc của các phép hợp, giao, bù giữa hai tập hợp kinh điển. Nói cách khác, khái niệm xây dựng những phép toán trên tập mờ được hiểu là việc xác định các hàm thuộc cho phép hợp (tuyển)  $A \cup B$ , giao  $A \cap B$ , bù (phủ định)  $A^C$ ... từ những tập mờ  $A, B$ .

Một nguyên tắc cơ bản trong việc xây dựng các phép toán trên tập mờ là không được mâu thuẫn với những phép toán đã có trong lý thuyết tập hợp kinh điển. Mặc dù không giống tập hợp kinh điển, hàm thuộc của các tập mờ  $A \cup B$ , giao  $A \cap B$ , bù  $A^C$ ... được định nghĩa cùng với tập mờ, song sẽ không mâu thuẫn với các phép toán tương tự của tập hợp kinh điển nếu như chúng thỏa mãn như tính chất tổng quát được phát biểu như “tiên đề” của lý thuyết tập hợp kinh điển.

##### **a) Phép hợp hai tập mờ**

Do trong định nghĩa về tập mờ hàm thuộc giữ vai trò như một thành

phần cấu thành tập mờ nên các tính chất của phép hợp giữa hai tập hợp kinh điển sẽ không là điều hiển nhiên nữa. Thay vào đó chúng được sử dụng như những tiên đề để xây dựng phép hợp trên tập mờ.

### Định nghĩa 1

Hợp của hai tập mờ A và B có cùng tập nền X là một mờ  $H \cup B$  cũng xác định trên tập nền X có hàm thuộc  $\mu_{A \cup B}(x)$  thỏa mãn các tính chất sau:

$$1- \mu_{A \cup B}(x) \text{ chỉ phụ thuộc vào } \mu_A(x) \text{ và } \mu_B(x). \quad (1 - 5a)$$

$$2- \mu_B(x) = 0 \text{ với mọi } x \Rightarrow \mu_{A \cup B}(x) = \mu_A(x). \quad (1 - 5b)$$

$$3- \mu_{A \cup B}(x) = \mu_{B \cup A}(x), \text{ tức là có tính giao hoán.} \quad (1 - 5c)$$

$$4- \mu_{(A \cup B) \cup C}(x) = \mu_{A \cup (B \cup C)}(x). \text{ tức là có tính kết hợp.} \quad (1 - 5d)$$

$$5- \text{Nếu } A_1 \subseteq A_2 \text{ thì } A_1 \cup B \subseteq A_2 \cup B, \text{ hay } \mu_{A \cup B}(x) \text{ có tính không giảm}$$

$$\mu_{A_1}(x) \leq \mu_{A_2}(x) \text{ suy ra } \mu_{A_1 \cup B}(x) \leq \mu_{A_2 \cup B}(x). \quad (1 - 5e)$$

Có thể thấy được sẽ có nhiều công thức khác nhau được dùng để tính hàm thuộc  $\mu_{A \cup B}(x)$  cho hợp hai tập mờ. Chẳng hạn năm công thức sau đều có thể được sử dụng để định nghĩa hàm thuộc  $\mu_{A \cup B}(x)$  của phép hợp giữa hai tập mờ:

$$1- \mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\} \quad (\text{Luật lấy max}). \quad (1 - 6a)$$

$$2- \mu_{A \cup B}(x) = \begin{cases} \max\{\mu_A(x), \mu_B(x)\} & \text{khi } \min\{\mu_A(x), \mu_B(x)\} = 0 \\ 1 & \text{khi } \min\{\mu_A(x), \mu_B(x)\} \neq 0 \end{cases} \quad (1 - 6b)$$

$$3- \mu_{A \cup B}(x) = \min\{1, \mu_A(x), \mu_B(x)\} \quad (\text{Phép hợp Lukasiewicz}) \quad (1 - 6c)$$

$$4- \mu_{A \cup B}(x) = \frac{\mu_A(x) + \mu_B(x)}{1 + \mu_A(x) + \mu_B(x)} \quad (\text{Tổng Einstein}) \quad (1 - 6d)$$

$$5 - \mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x) \quad (\text{Tổng trực tiếp}) \quad (1 - 6e)$$

**\*) Hợp hai tập mờ theo luật Max**

Hợp của hai tập mờ A với hàm thuộc  $\mu_A(x)$  (định nghĩa trên tập nền M) và B với hàm thuộc  $\mu_B(x)$  (định nghĩa trên tập nền N) theo luật Max là một tập mờ xác định trên tập nền  $M \times N$  với hàm thuộc

$$\mu_{\underline{A} \cup \underline{B}}(x, y) = \max \mu_{\underline{A}}(x, y), \mu_{\underline{B}}(x, y) \quad (1 - 7)$$

Trong đó:

$$\mu_{\underline{A}}(x, y) = \mu_A(x) \text{ với mọi } y \in N$$

$$\mu_{\underline{B}}(x, y) = \mu_B(y) \text{ với mọi } x \in M$$

**\*) Hợp hai tập mờ theo luật Sum (Lukasiewicz)**

Hợp của hai tập mờ A với hàm thuộc  $\mu_A(x)$  (định nghĩa trên tập nền M) và B với hàm thuộc  $\mu_B(x)$  (định nghĩa trên tập nền N) theo luật Sum là một tập mờ xác định trên tập nền  $M \times N$  với hàm thuộc

$$\mu_{\underline{A} \cup \underline{B}}(x, y) = \min \{1, \mu_{\underline{A}}(x, y) + \mu_{\underline{B}}(x, y)\} \quad (1 - 8)$$

Trong đó:

$$\mu_{\underline{A}}(x, y) = \mu_A(x) \text{ với mọi } y \in N$$

$$\mu_{\underline{B}}(x, y) = \mu_B(y) \text{ với mọi } x \in M$$

Một cách tổng quát, do hàm thuộc  $\mu_{\underline{A} \cup \underline{B}}(x, y)$  của hai tập mờ A, B không cùng không gian nền chỉ phụ thuộc vào  $\mu_A(x) \in [0, 1]$  và  $\mu_B(x) \in [0, 1]$  nên ta có thể xem  $\mu_{\underline{A} \cup \underline{B}}(x, y)$  là hàm của hai biến  $\mu_A, \mu_B$  được định nghĩa như sau:

$$\mu_{\underline{A} \cup \underline{B}}(x, y) = \mu(\mu_A, \mu_B) : [0, 1]^2 \rightarrow [0, 1]. \quad (1 - 9)$$

Ta đi đến định nghĩa về hàm thuộc  $\mu(\mu_A, \mu_B)$  của hai tập mờ không cùng không gian nền:

### **Định nghĩa 2**

Hàm thuộc của hợp giữa hai tập mờ A với  $\mu_A(x)$  định nghĩa trên tập nền M và B với  $\mu_B(y)$  định nghĩa trên tập nền N là một hàm hai biến  $\mu(\mu_A, \mu_B) : [0,1]^2 \rightarrow [0,1]$  xác định trên nền MxN thỏa mãn:

$$1- \mu_B = 0 \Rightarrow \mu(\mu_A, \mu_B) = \mu_A \quad (1 - 10a)$$

$$2- \mu(\mu_A, \mu_B) = \mu(\mu_B, \mu_A) \quad \text{tức là có tính giao hoán.} \quad (1 - 10b)$$

$$3- \mu(\mu_A, \mu(\mu_B, \mu_C)) = \mu(\mu(\mu_A, \mu_B), \mu_C) \quad \text{tức là có tính kết hợp.} \quad (1-10c)$$

$$4- \mu(\mu_A, \mu_B) \leq \mu(\mu_C, \mu_D), \forall \mu_A \leq \mu_C, \mu_B \leq \mu_D \quad \text{tức là có tính không giảm} \quad (1-10d)$$

Một hàm hai biến  $\mu(\mu_A, \mu_B) : [0,1]^2 \rightarrow [0,1]$  thỏa mãn các điều kiện của *định nghĩa 2* còn được gọi là hàm *t- đối chuẩn* (t- conorm).

### **b) Phép giao hai tập mờ**

Như đã đề cập, phép giao  $A \cap B$  trên tập mờ phải được định nghĩa sao cho không mâu thuẫn với phép giao của tập hợp kinh điển và yêu cầu này sẽ được thỏa mãn nếu chúng có được các tính chất tổng quát của tập kinh điển.

Tương tự như đã làm với phép hợp hai tập mờ, phép giao hai tập mờ trên tập nền tổng quát hóa những tính chất cũng chỉ được thực hiện trực tiếp nếu hai tập mờ đó có cùng tập nền. Trong trường hợp chúng không cùng tập nền phải đưa chúng về một tập nền mới là tích của hai tập nền đã cho.

### **Định nghĩa 3**

Giao của hai tập mờ A và B có cùng tập nền X là một tập mờ cũng xác định trên tập nền X với hàm thuộc thỏa mãn:

$$1- \mu_{A \cap B}(x) \text{ chỉ phụ thuộc vào } \mu_A(x) \text{ và } \mu_B(x). \quad (1-11a)$$

$$2- \mu_B(x) = 1 \text{ với mọi } x \text{ suy ra } \mu_{A \cap B}(x) = \mu_A(x). \quad (1-11b)$$

$$3- \mu_{A \cap B}(x) = \mu_{B \cap A}(x), \text{ tức là phép giao có tính chất giao hoán.} \quad (1-11c)$$

$$4- \mu_{(A \cap B) \cap C}(x) = \mu_{A \cap (B \cap C)}(x), \text{ tức là có tính kết hợp.} \quad (1-11d)$$

$$5- \mu_{A1}(x) \leq \mu_{A2}(x) \Rightarrow \mu_{A1 \cap B}(x) \leq \mu_{A2 \cap B}(x), \text{ tính không giảm.} \quad (1-11e)$$

Giống như đã trình bày về phép hợp hai tập mờ, có nhiều công thức khác nhau để tính hàm thuộc  $\mu_{A \cap B}(x)$  của giao hai tập mờ và bất cứ một ánh xạ

$$\mu_{A \cap B}(x): X \rightarrow [0,1]$$

nào thỏa mãn 5 tiêu chuẩn đã nêu trong *định nghĩa 3* đều được xem như là hàm thuộc của giao hai tập mờ A và B có chung một tập nền X.

Các công thức thường dùng để tính hàm thuộc  $\mu_{A \cap B}(x)$  của phép giao gồm:

$$1- \mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\} \quad (1-12a)$$

$$2- \mu_{A \cap B}(x) = \begin{cases} \min\{\mu_A(x), \mu_B(x)\} & \text{nếu } \max\{\mu_A(x), \mu_B(x)\} = 1 \\ 0 & \text{nếu } \max\{\mu_A(x), \mu_B(x)\} \neq 1 \end{cases} \quad (1-12b)$$

$$3, \mu_{A \cap B}(x) = \max\{0, \mu_A(x) + \mu_B(x) - 1\} \text{ (Phép giao Lukasiewicz)} \quad (1-12c)$$

$$4, \mu_{A \cap B}(x) = \frac{\mu_A(x) \cdot \mu_B(x)}{2 - (\mu_A(x) + \mu_B(x)) + \mu_A(x) \cdot \mu_B(x)} \text{ (Tổng Einstein)} \quad (1-12d)$$

$$5, \mu_{A \cap B}(x) = \mu_A(x) \cdot \mu_B(x) \text{ (Tích đại số)} \quad (1-12e)$$

Tuy nhiên luật Min (1-12a) và tích đại số là hai loại luật xác định hàm thuộc của giao hai tập mờ được ưa dùng hơn cả trong kỹ thuật điều khiển mờ.

**\*) Giao hai tập mờ theo luật Min**

Giao của tập mờ A có hàm thuộc  $\mu_A(x)$  định nghĩa trên tập nền M với tập mờ B có hàm thuộc  $\mu_B(x)$  định nghĩa trên tập nền N là một tập mờ xác định trên tập nền M x N có hàm thuộc:

$$\mu_{\underline{A} \cap \underline{B}}(x, y) = \min\{\mu_A(x), \mu_B(y)\} = \min\{\mu_A(x, y), \mu_B(x, y)\} \quad (1-13a)$$

**Trong đó:**

$$\mu_{\underline{A}}(x, y) = \mu_A(x), \forall y \in N$$

**và** 
$$\mu_{\underline{B}}(x, y) = \mu_B(x), \forall x \in M$$

**\*) Giao của hai tập mờ theo luật tích đại số**

Giao của tập mờ A có hàm thuộc  $\mu_A(x)$  định nghĩa trên tập nền M với tập mờ B có hàm thuộc  $\mu_B(x)$  định nghĩa trên tập nền N là một tập mờ xác định trên tập nền M x N có hàm thuộc:

$$\mu_{\underline{A} \cap \underline{B}}(x, y) = \mu_A(x, y) \cdot \mu_B(x, y) \quad (1-13b)$$

**Trong đó:**

$$\mu_{\underline{A}}(x, y) = \mu_A(x), \forall y \in N$$

**và** 
$$\mu_{\underline{B}}(x, y) = \mu_B(x), \forall x \in M$$

Ta thấy hàm thuộc của giao hai tập mờ  $\mu_{\underline{A} \cap \underline{B}}(x, y)$  của hai tập mờ A, B không cùng nền chỉ phụ thuộc vào giá trị các hàm  $\mu_A(x) \in [0,1]$  và  $\mu_B(x) \in [0,1]$ . Do đó không mất tính tổng quát nếu ta xem  $\mu_{\underline{A} \cap \underline{B}}(x, y)$  như một hàm hai biến  $\mu_A$  và  $\mu_B$  :

$$\mu_{\underline{A} \cap \underline{B}}(x, y) = \mu(\mu_A, \mu_B) : [0,1]^2 \rightarrow [0,1] \quad (1-14)$$

và đi đến định nghĩa về hàm thuộc  $\mu(\mu_A, \mu_B)$  của giao hai tập mờ không cùng không gian nền như sau:

#### **Định nghĩa 4**

Hàm thuộc của giao hai tập mờ A với  $\mu_A(x)$  định nghĩa trên tập nền M với tập mờ B có hàm thuộc  $\mu_B(x)$  định nghĩa trên tập nền N là một hàm hai biến  $\mu(\mu_A, \mu_B) : [0,1]^2 \rightarrow [0,1]$  xác định trên tập nền  $M \times N$  thỏa mãn:

$$1- \mu_B = 1 \Rightarrow \mu(\mu_A, \mu_B) = \mu_A. \quad (1-15a)$$

$$2- \mu(\mu_A, \mu_B) = \mu(\mu_B, \mu_A) \text{ Tức là có tính giao hoán. } (1-15b)$$

$$3- \mu(\mu_A, \mu(\mu_B, \mu_C)) = \mu(\mu(\mu_A, \mu_B), \mu_C) \text{ tức là có tính kết hợp. } (1-15c)$$

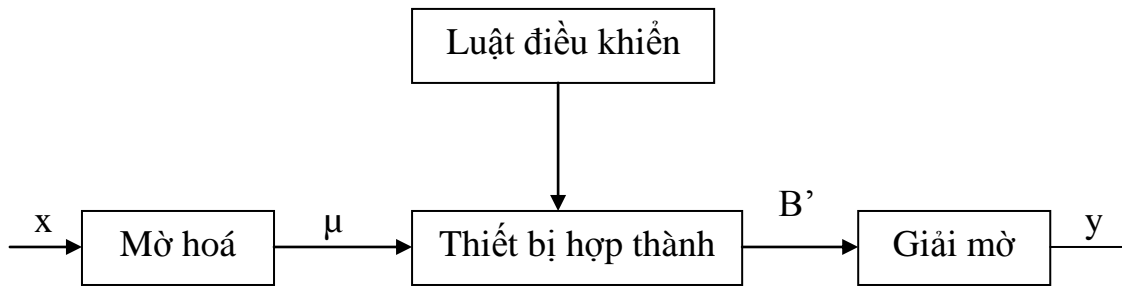
$$4- \mu(\mu_A, \mu_B) \leq \mu(\mu_C, \mu_D), \forall \mu_A \leq \mu_C \text{ và } \mu_B \leq \mu_D, \text{ tính không giảm. } (1-15d)$$

Một hàm hai biến  $\mu(\mu_A, \mu_B) : [0,1]^2 \rightarrow [0,1]$  thỏa mãn các điều kiện trên được gọi là hàm chuẩn *t*-chuẩn (*t-norm*).

#### **1.1.2.3. Bộ điều khiển mờ**

Một bộ điều khiển mờ bao gồm 3 phần chính:

- **Khâu mờ hoá:** Làm nhiệm vụ chuyển đổi từ giá trị rõ đầu vào xác định sang trạng thái đầu vào mờ. Đây là giao diện đầu vào của bộ điều khiển mờ.
- **Thiết bị hợp thành:** Triển khai luật hợp thành trên cơ sở luật điều khiển **IF...THEN**.
- **Khâu giải mờ:** Chuyển đổi từ giá trị mờ nhận được của thiết bị hợp thành sang giá trị thực để điều khiển đối tượng. Đây là giao diện đầu ra của bộ điều khiển mờ.



**Hình 1.2: Cấu trúc bộ điều khiển mờ cơ bản**

**Trong đó:**

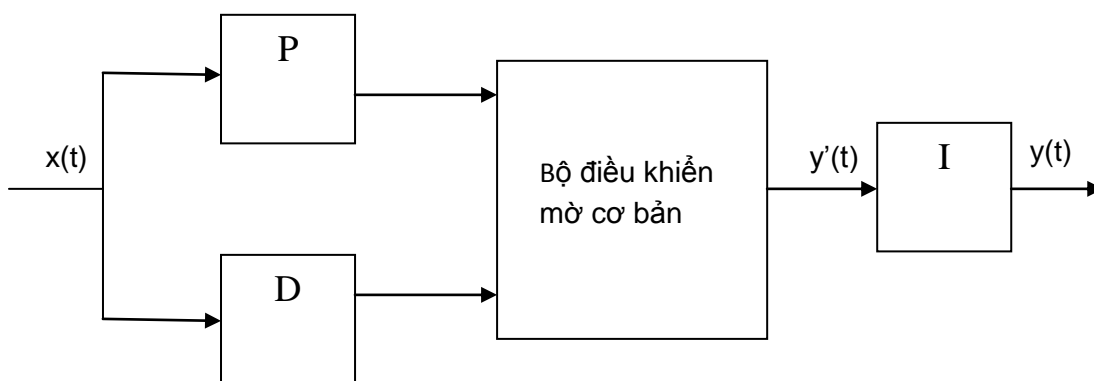
x: Là tập giá trị thực cần điều khiển đầu vào

m: Tập mờ của giá trị đầu vào.

B: Tập giá trị mờ của giá trị điều khiển thực.

y: Giá trị điều khiển thực.

Bộ điều khiển mờ cơ bản là một bộ điều khiển mờ tĩnh, nó chỉ có khả năng xử lý các giá trị hiện thời. Để giải quyết được các bài toán điều khiển động, bộ điều khiển mờ cơ bản phải được nối thêm các khâu động học thích hợp. Ví dụ, khâu tỷ lệ, vi phân hoặc tích phân.

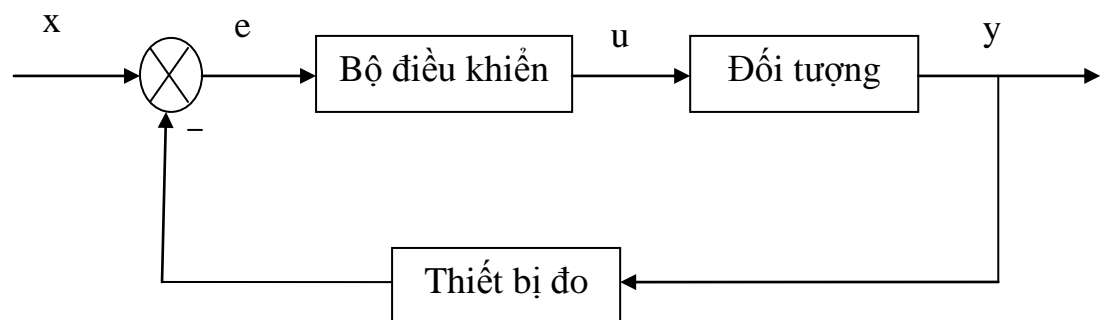


**Hình 1.3: Cấu trúc bộ điều khiển mờ động**



Hệ thống điều khiển mờ đảm nhiệm chức năng như một hệ thống điều khiển thông thường. Sự khác biệt chủ yếu ở chỗ: khi hệ thống điều khiển truyền thông

thông dựa vào logic kinh điển  $\{0,1\}$ , thì hệ thống điều khiển mờ thực hiện chức năng điều khiển dựa trên kinh nghiệm và những kết luận theo tư duy của con người, quá trình xử lý đó thông qua bộ logic mờ.



**Hình 1.4: Hệ thống điều khiển với bộ điều khiển mờ**

Để thực hiện được quá trình điều khiển, đối tượng phải được điều khiển bằng các tín hiệu rõ. Do vậy, tín hiệu ra của bộ điều khiển mờ phải được giải mờ trước khi đưa vào đối tượng. Cũng tương tự như vậy, tín hiệu ra của đối tượng qua các bộ cảm biến đo lường phải được mờ hoá trước khi đưa vào bộ điều khiển mờ.

Nguyên tắc tổng hợp một bộ điều khiển mờ hoàn toàn dựa vào những phương pháp toán học trên cơ sở định nghĩa các biến ngôn ngữ vào ra, và sự lựa chọn những luật điều khiển trong bộ điều khiển mờ. Thiết bị hợp thành triển khai các luật điều khiển theo một nguyên tắc nhất định (MAX – MIN, MAX – PROD,...), đây là phần cốt lõi của bộ điều khiển mờ.

Để cho thiết bị thực hiện luật điều khiển làm việc đúng chế độ thì phải chọn các biến ngôn ngữ sao cho phù hợp. Các đại lượng vào ra chuẩn và phù

hợp với luật điều khiển. Tất cả các vấn đề đó được hình thành trên quá trình thử nghiệm và thiết kế.

Tuy thiết bị hợp thành là bộ phận quan trọng nhất của bộ điều khiển mờ, nhưng khi giải quyết các bài toán động, trong nhiều trường hợp nó cần các thông tin về đạo hàm hay tích phân của sai lệch. Khi đó tín hiệu vào phải được xử lý sơ qua bằng các khâu động học. Đối với một bài toán có độ phức tạp cao, đôi lúc còn cần đến nhiều bộ điều khiển mờ với các khâu mắc nối tiếp hoặc song song theo kiểu mang.

### **a) Quá trình mờ hoá**

Mờ hoá là một ánh xạ từ một giá trị rõ  $x \in U \in \mathbb{R}^n$  sang một tập mờ  $A$  trong tập nền  $U$ . Mờ hoá phải đảm bảo: Độ phụ thuộc là lớn nhất, đảm bảo tính khử nhiễu, tính toán đơn giản.

Trong điều khiển, với mục đích sử dụng các hàm thuộc sao cho khả năng tích hợp chúng đơn giản, người ta chỉ quan tâm đến 3 kiểu mờ hoá cơ bản sau:

- Hàm Singleton (còn gọi là hàm Kronecker).
- Hàm hình tam giác.
- Hàm hình thang.

Trong ba cách trên, mờ hoá theo tam giác đảm bảo khử nhiễu nhưng tính toán và khử nhiễu khó, lâu. Chỉ có mờ hoá theo kiểu Singleton là được sử dụng nhiều nhất mặc dù nó không có tính khử nhiễu nhưng tính toán đơn giản và nhanh.

### **b) Thiết bị hợp thành**

Thiết bị hợp thành được hiểu là sự ghép nối chung giữa bản thân nội dung luật hợp thành và thuật toán xác định giá trị mờ của luật hợp thành khi biết trước giá trị rõ của tín hiệu vào.

Trọng tâm của hệ mờ chính là mệnh đề hợp thành IF...THEN. Ta xét hệ MISO (n đầu vào, 1 đầu ra), mệnh đề hợp thành mô tả hệ MISO là:

$$R_i: \quad \text{IF } x_1 = A_i^1 \text{ and } \dots \text{ and } x_n = A_n^1 \text{ THEN } y = B_j^1 \quad (*)$$

Với:  $\underline{x} = (x_1, \dots, x_n)^T$  là véctor đầu vào.

y là đầu ra.

$A_i^1$  là các tập mờ của biến đầu vào ( $i = 1 \div n$ ).

$B_j^1$  là các tập mờ của biến đầu ra.

Dạng (\*) là dạng chuẩn của mệnh đề hợp thành vì tất cả các dạng mô tả khác đều có thể đưa về dạng này. Chẳng hạn nếu hệ mờ là MMO thì nó chính là tổng của các hệ con MISO mà chúng được mô tả dưới dạng (\*).

Gọi R là luật hợp thành cho các mệnh đề  $R_i$  ( $i = 1 \div n$ ) ở trên:

$$R = \bigcup_i R_i \text{ (phép tích hợp các tập mờ } R_i)$$

Thiết bị hợp thành được gọi bằng tên của quy tắc thực hiện luật hợp thành. Trong điều khiển có 4 thiết bị chính sau:

Thiết bị hợp thành MAX – MIN

+ Phép suy diễn được thực hiện với luật MIN:  $\mu_{A \rightarrow B}(y) = \min\{H, \mu_B(y)\}$ .

+ Phép hợp mờ được thực hiện theo luật MAX:  $\mu_{A \cup B}(y) = \max\{\mu_A(y), \mu_B(y)\}$ .

- Thiết bị hợp thành MAX – Prod

+ Phép suy diễn được thực hiện với luật Prod:  $\mu_{A \rightarrow B}(y) = H \cdot \mu_B(y)$ .

+ Phép hợp mờ được thực hiện theo luật MAX:  $\mu_{A \cup B}(y) = \max\{\mu_A(y), \mu_B(y)\}$ .

- Thiết bị hợp thành Sum – Prod

+ Phép suy diễn được thực hiện với luật Prod:  $\mu_{A \rightarrow B}(y) = H, \mu_B(y)$ .

+ Phép hợp mờ được thực hiện theo luật MAX:

$$\mu_{A \cup B}(y) = \min\{1, \mu_A(y) + \mu_B(y)\}.$$

- Thiết bị hợp thành Sum - Min

+ Phép suy diễn được thực hiện với luật Min:  $\mu_{A \rightarrow B}(y) = \min\{H, \mu_B(y)\}$

+ Phép hợp mờ được thực hiện theo luật Max:

$$\mu_{A \cup B}(y) = \min\{1, \mu_A(y) + \mu_B(y)\}.$$

### c) Giải mờ

Giải mờ là quá trình xác định một giá trị rõ  $y'$  nào đó có thể chấp nhận được từ hàm liên thuộc  $\mu_{B'}(y)$  của giá trị mờ  $B'$  (tập mờ). Có hai phương pháp giải mờ chính là phương pháp cực đại và phương pháp điểm trọng tâm sẽ được trình bày dưới đây, trong đó cơ sở của tập mờ  $B'$  được ký hiệu thống nhất là  $Y$  [3].

#### *Phương pháp cực đại*

Giải mờ theo phương pháp cực đại gồm hai bước:

Xác định miền chứa giá trị rõ  $y'$ . Giá trị rõ  $y'$  là giá trị mà tại đó hàm liên thuộc đạt giá trị cực đại (độ cao  $H$  của tập mờ  $B'$ ) tức là miền

$$G = \{ y \in Y \mid \mu_{B'}(y) = H \} \quad (1-16)$$

Xác định  $y'$  có thể chấp nhận được từ  $G$  là đoạn  $[y_1, y_2]$ . Có ba nguyên lý:

$$\text{- Nguyên lý trung bình } y' = (y_1 + y_2) / 2 \quad (1-17)$$

- Nguyên lý cận trái:  $y' = y_1$
- Nguyên lý cận phải:  $y' = y_2$  (1-18)

*Ghi chú:* Sai lệch của ba giá trị rõ, xác định theo trung bình, cận trái hay cận phải sẽ càng lớn nếu độ thoả mãn H của luật điều khiển quyết định càng nhỏ.

Trường hợp G không phải là một miền liên thông, tức là khi có nhiều luật hợp thành có cùng một đáp ứng đầu vào và cho ngưỡng giá trị quyết định khác nhau của biến ngôn ngữ đầu ra. Giá trị  $y'$  sẽ được chọn như sau: Nếu vẫn cứ áp dụng nguyên lý trung bình thì có thể giá trị rõ  $y'$  sẽ là giá trị có độ phụ thuộc nhỏ hơn H, hoặc nếu sử dụng nguyên lý cận trái hay phải thì các trường hợp còn lại là  $y_3$  và  $y_4$ . Do vậy thông thường một khoảng con liên thông  $G_1$  trong G sẽ được chọn làm khoảng liên thông có mức ưu tiên cao nhất, ví dụ là  $G_1$ , sau đó áp dụng một trong ba nguyên lý đã biết với miền  $G_1$  thay cho G.

### ***Phương pháp điểm trọng tâm***

Phương pháp điểm trọng tâm sẽ cho ra kết quả  $y'$  là hoành độ của điểm trọng tâm miền được bao bởi trục hoành và đường  $\mu_{B'}(y)$ . Công thức xác định  $y'$  theo phương pháp điểm trọng tâm như sau:

$$y' = \frac{\int_S y \mu_{B'}(y) dy}{\int_S \mu_{B'}(y) dy} \quad (1-19)$$

trong đó S là miền xác định của tập mờ  $B'$ .

Công thức (2.37) cho phép xác định giá trị  $y'$  với sự tham gia của tất cả các tập mờ đầu ra của mọi luật điều khiển một cách bình đẳng và chính xác, tuy nhiên lại không đề ý được thái độ thoả mãn của luật điều khiển quyết định và cần nhiều thời gian tính toán. Ngoài ra một trong những nhược điểm cơ bản của phương pháp điểm trọng tâm là có thể giá trị  $y'$  xác định được lại có

độ phụ thuộc nhỏ nhất, thậm chí bằng 0. Bởi vậy để tránh những trường hợp như vậy, khi định nghĩa hàm liên thuộc cho từng giá trị mờ của một biến ngôn ngữ nên để ý sao cho miền xác định của các giá trị mờ đầu ra là một miền liên thông.

Phương pháp điểm trọng tâm cho luật hợp thành Sum – Min: Giả sử có q luật điều khiển được triển khai. Vậy thì mỗi giá trị mờ B' tại đầu ra của bộ điều khiển sẽ là tổng của q giá trị mờ đầu ra của từng luật hợp thành. Ký hiệu các giá trị mờ đầu ra của luật điều khiển thứ k là  $\mu_{B_k'}(y)$  với  $k= 1, 2, \dots, p$  thì với quy tắc Sum - Min, hàm liên thuộc  $\mu_{B'}(y)$  sẽ là:

$$\mu_{B'}(y) = \sum_{k=1}^p \mu_{B_k'}(y) \quad (1-20)$$

thay (2.37) vào (2.36), sau đó đổi chỗ của tổng và tích phân cho nhau (hoàn toàn có nghĩa, vì tổng và tích phân đều hội tụ) thì công thức tính  $y'$  sẽ được đơn giản như sau:

$$y' = \frac{\int_S (y \sum_{k=1}^q \mu_{B_k'}(y)) dy}{\int_S \sum_{k=1}^q \mu_{B_k'}(y) dy} = \frac{\sum_{k=1}^q \int_S y \mu_{B_k'}(y) dy}{\sum_{k=1}^q \int_S \mu_{B_k'}(y) dy} = \frac{\sum_{k=1}^q M_k}{\sum_{k=1}^q A_k} \quad (1-21)$$

**Trong đó**  $M_k = \int_S y \mu_{B_k'}(y) dy$      $A_k = \int_S \mu_{B_k'}(y) dy$  (1-22)

### ***Phương pháp độ cao***

Giả thiết là mỗi tập  $\mu_{B_k'}(y)$  được xấp xỉ bằng một cặp giá trị  $(y_k, H_k)$  duy nhất, trong đó  $H_k$  là độ cao của  $\mu_{B_k'}(y)$  và  $y_k$  là một điểm mẫu trong miền giá trị của  $\mu_{B_k'}(y)$ . Tức là:  $\mu_{B_k'}(y_k) = H_k$

$$y' = \frac{\sum_{k=1}^q y_k H_k}{\sum H_k} \quad (1-23)$$

công thức có tên gọi là công thức xấp xỉ theo phương pháp độ cao và không chỉ áp dụng cho luật hợp thành Max – Min hay Sum - Min mà còn có thể cho

cả những luật hợp thành khác như Max - Prod hay Sum - Prod.

#### **1.1.2.4. Điều khiển mờ**

Áp dụng lí thuyết mờ trong hệ thống điều khiển ta xây dựng được một hệ điều khiển mờ, điều khiển mờ ngày nay đã được ứng dụng rộng trong cuộc sống con người. Dùng lí thuyết mờ để thiết kế bộ điều khiển tỏ rõ sự thuận lợi nhất là với các hệ thống mà đối tượng là phi tuyến khó thực hiện điều khiển theo các phương pháp thông thường. Nói chung thì một hệ thống điều khiển mờ có nguyên lý điều khiển không khác gì một hệ thống điều khiển đã biết. Các ứng dụng điều khiển mờ đã có trong thực tế là: Điều khiển quá trình lò xi măng, điều khiển rôbot, xử lý ảnh, điều khiển mô tơ, tự động lái tàu, điều chỉnh tiêu cự máy quay video, mạch vòng điều khiển servo, điều khiển máy bay, và định vị tàu vũ trụ. Và các ứng dụng công nghiệp mới đang được tiếp tục phát triển. Cốt lõi của bộ điều khiển mờ là các luật điều khiển mờ (IF ... THEN ...), là các mệnh đề hợp thành mô tả lại quan hệ vào - ra dựa trên "kinh nghiệm chuyên gia" của con người trong thực tiễn, bằng việc sử dụng các biến ngôn ngữ thay cho một mô hình toán học phức tạp. Việc sử dụng các biến ngôn ngữ, các luật điều khiển mờ, và sự suy diễn xấp xỉ sẽ cung cấp cách thức gần gũi với giao tiếp của loài người để biểu diễn lại "kinh nghiệm chuyên gia" của con người vào việc thiết kế bộ điều khiển. Thực tế đã cho thấy việc tổng hợp hệ thống bằng bộ điều khiển mờ tạo nên những ưu điểm rõ rệt sau:

- Khối lượng công việc thiết kế giảm đi nhiều do không cần sử dụng mô hình đối tượng, với các bài toán thiết kế có độ phức tạp cao, giải pháp dùng bộ điều khiển mờ cho phép giảm khối lượng tính toán và giá thành sản phẩm.
- Bộ điều khiển mờ dễ hiểu hơn so với các bộ điều khiển khác (cả về kỹ thuật) và dễ dàng thay đổi.

- Trong nhiều trường hợp bộ điều khiển mờ làm việc ổn định hơn, bền vững hơn và chất lượng điều khiển cao hơn.

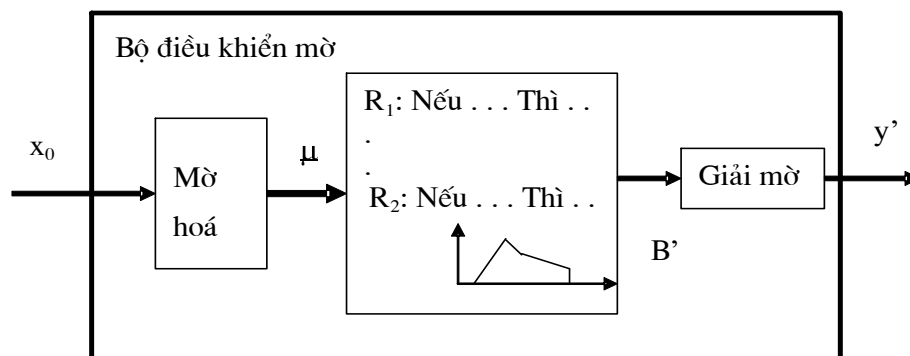
Cấu trúc chung của bộ điều khiển mờ:

Một bộ điều khiển mờ có 3 khâu cơ bản :

*Mờ hoá*: có nhiệm vụ chuyển đổi một giá trị rõ đầu vào  $x_0$  thành một vecto  $\mu$  gồm các độ phụ thuộc của giá trị vào rõ.

*Luật hợp thành và Thực hiện luật hợp thành*: Lưu giữ các mệnh đề hợp thành và xử lý vecto  $\mu$  các độ phụ thuộc của giá trị vào rõ và cho ra giá trị mờ  $B'$  của biến ngôn ngữ đầu ra.

*Giải mờ*: có nhiệm vụ chuyển đổi tập mờ  $B'$  thành một giá trị rõ  $y'$  chấp nhận được cho đối tượng (tín hiệu điều chỉnh)



**Hình 1.5: Cấu trúc bên trong của một bộ điều khiển mờ**

Ta có thể phân loại các bộ điều khiển mờ dựa trên quan hệ vào/ra toàn cục của tín hiệu vào  $x_0$  và tín hiệu ra  $y'$ . Quan hệ toàn cục đó có tên gọi là quan hệ truyền đạt. Cũng giống như việc phân loại một khâu điều khiển kinh điển, việc phân loại quan hệ truyền đạt của một bộ điều khiển mờ theo các tiêu chí sau [3]:

1. Tĩnh hay động
2. Tuyến tính hay phi tuyến
3. Tham số tập trung hay tham số rải
4. Liên tục hay rời rạc



5. Tham số tĩnh hay tham số động

6. Tiên định hay ngẫu nhiên

7. Ổn định hay không ổn định

Xét từng khâu của bộ điều khiển mờ gồm các khâu mờ hoá, thiết bị hợp thành và giải mờ trong hình 2.1.a, thì thấy rằng trong quan hệ vào/ra giá trị  $y'$  tại đầu ra chỉ phụ thuộc vào một mình giá trị  $x_0$  của đầu vào chứ không phụ thuộc vào giá trị đã qua của tín hiệu  $x(t)$ , tức là chỉ phụ thuộc vào giá trị của  $x(t)$  tại đúng thời điểm đó. Do đó bộ điều khiển mờ thực chất là một bộ điều khiển tĩnh và quan hệ truyền đạt hoàn toàn được mô tả đầy đủ bằng đường đặc tính  $y(x)$  như các đường đặc tính của khâu relay 2 hoặc 3 trạng thái quen biết trong kỹ thuật điều khiển phi tuyến kinh điển [3].

Tuy nhiên, trong nhiều trường hợp, qua thay đổi dạng hàm liên thuộc của các giá trị ngôn ngữ vào ra, hoặc nhờ việc nối thêm các khâu tích phân, vi phân vào phía trước bộ điều khiển làm vai trò tiền xử lý tín hiệu, thì bộ điều khiển chung nhận được sẽ lại có tính gần tĩnh giống như khâu relay có trễ hoặc có tính chất động như bộ điều khiển mờ PID sẽ trình bày trong phần tiếp theo.

#### **1.1.2.5. Nguyên tắc thiết kế bộ điều khiển mờ**

Khi tổng hợp các bộ điều khiển mờ cần lưu ý đến tính phi tuyến khá mạnh của các bộ điều khiển mờ. Phần lớn các đối tượng điều khiển trong thực tế có tính phi tuyến, phụ thuộc vào thời gian, có hằng số trễ lớn và tham số rải. Đối với hệ thống như vậy việc ứng dụng kỹ thuật điều khiển mờ là rất thích hợp. Tuy vậy, nền tảng cho sự ứng dụng thành công kỹ thuật điều khiển mờ là kiến thức, kinh nghiệm và sự hiểu biết của các chuyên gia về hệ thống đó. Trong quá trình thiết kế bộ điều khiển mờ không nhất thiết phải biết trước mô hình đối tượng điều khiển mà chỉ cần thể hiện những hiểu biết về đối tượng, qua các biến ngôn ngữ thể hiện động học của đối tượng. Nói chung, qui trình thiết kế bộ điều khiển mờ ngắn hơn qui trình thiết kế bộ điều khiển

kinh điển. Các bước xây dựng thiết bị điều khiển và chọn luật hợp thành khá tốn công khi thiết kế các bộ điều khiển mờ. Cho đến nay vẫn chưa có các nguyên tắc chuẩn mực cho việc thiết kế khảo sát tính ổn định, chất lượng,... đối với các bộ điều khiển mờ cũng như nguyên lý tối ưu các bộ điều khiển mờ về mặt lý thuyết. Các nguyên tắc dưới đây là cần thiết khi lựa chọn các bộ điều khiển :

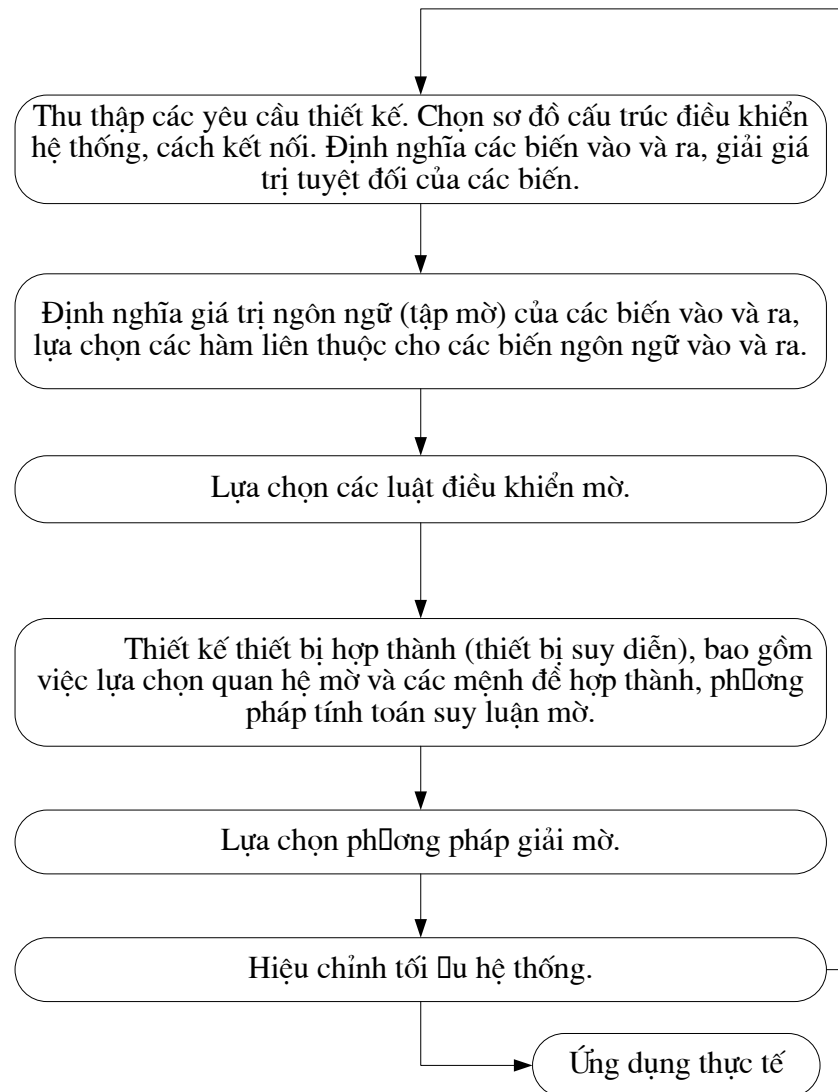
- Không bao giờ áp dụng bộ điều khiển mờ khi có thể dễ dàng giải quyết tốt vấn đề bằng bộ điều khiển kinh điển. Khi áp dụng bộ điều khiển mờ, trước hết cần đi phân tích, thu thập các kinh nghiệm và thể hiện nó trên cơ sở logic mờ. Một giải pháp rất hiệu quả là kết hợp các bộ điều khiển kinh điển với logic mờ để phát huy được ưu điểm của cả hai kiểu bộ điều khiển.
- Việc sử dụng bộ điều khiển mờ cho các hệ thống yêu cầu độ an toàn cao vẫn còn hạn chế do chỉ có thể xác định được chất lượng của hệ thống bằng thực nghiệm.
- Bộ điều khiển mờ phải được phát triển qua thực nghiệm.

### **Các bước thiết kế bộ điều khiển mờ**

Từ những phân trước, chúng ta có thể thấy rằng các phần chính của việc thiết kế một bộ điều khiển mờ thường như sau:

Hai bước thiết kế đầu tiên (1) và (2) chỉ ra rằng, trong việc thiết kế một bộ điều khiển mờ, trước tiên, từ yêu cầu thiết kế đặt ra phải nhận diện được các biến trạng thái chính của quá trình và các biến điều khiển và xác định tập các thuật ngữ (ký hiệu) để miêu tả giá trị của mỗi biến ngôn ngữ. Ví dụ, một tập các thuật ngữ như {*nhỏ, trung bình, rộng*} không thể thoả mãn trong một vài lĩnh vực và việc sử dụng một tập năm số hạng như {*rất nhỏ, nhỏ, trung bình, rộng, rất rộng*} có thể thay thế được. Do đó, số lượng các tập mờ của không gian vào-ra sẽ được chọn đủ rộng để cung cấp một sự đủ thích hợp và còn đủ nhỏ để tiết kiệm không gian bộ nhớ. Số lượng này có một ảnh hưởng

cơ bản đến chất lượng của bộ điều khiển có thể thu được. Hơn nữa, các kiểu hình dạng khác nhau của các hàm liên thuộc ví dụ như là hình tam giác, hình thang, và kiểu hình chuông,... có thể được dùng làm các giá trị của mỗi biến ngôn ngữ cũng ảnh hưởng không nhỏ đến chất lượng của bộ điều khiển mờ.



**Hình 1.6: Lưu đồ thiết kế một bộ điều khiển mờ**

Với các bước thiết kế (4) và (5), không có một phương pháp có hệ thống nào để thực hiện việc thiết kế một thiết bị hợp thành và lựa chọn một bộ giải mờ. Hầu hết các nhà thiết kế sử dụng các kinh nghiệm nghiên cứu và các kết quả thực nghiệm để đưa ra những hướng dẫn cho những sự lựa chọn này.

Bước thiết kế thứ (6) thực chất là kiểm tra và hiệu chỉnh lại bộ điều

khien trong điều kiện môi trường mô phỏng (nếu như có thể vì tạo mô hình toán học thích hợp cho một hệ thống lớn phức tạp, phi tuyến là rất khó) hay trong môi trường làm việc thực tế. Khi hiệu chỉnh ta có thể tận dụng kinh nghiệm đã có về một số tính chất của bộ điều khiển kinh điển PID cho bộ điều khiển mờ. Nếu thiết kế dù đã hiệu chỉnh mà vẫn chưa thoả mãn yêu cầu đặt ra thì ta lại phải quay trở lại bước (1) để làm lại.

Bước thiết kế thứ (3) trong việc xác định các luật điều khiển mờ phụ thuộc rất lớn vào bản chất vật lý (trạng thái tự nhiên vốn có) của các đối tượng được điều khiển. Nói chung, có bốn phương pháp để tạo nên các luật điều khiển mờ và các luật này không độc lập nhau. Một sự kết hợp của chúng có thể là cần thiết để xây dựng một phương pháp hiệu quả cho việc tạo nên các luật điều khiển mờ. Đó là bốn phương pháp:

– Kinh nghiệm chuyên gia và hiểu biết của kỹ sư điều khiển: Các luật điều khiển mờ được thiết kế nhờ việc xem xét các hoạt động của con người và/hoặc hiểu biết của các kỹ sư điều khiển. Đặc biệt hơn, chúng ta có thể hỏi một chuyên gia để biểu diễn tri thức của người này trong thuật ngữ tập mờ, đó là để biểu diễn phương pháp làm việc này trong các luật mờ Nếu - Thì. Chúng ta cũng có thể hỏi một kỹ sư điều khiển để liệt kê số các phương thức dựa trên hiểu biết của người này về quá trình được điều khiển. Cuối cùng, dùng thủ tục tìm kiếm loại trừ để tạo nên các luật điều khiển mờ.

– Mô phỏng một hành động điều khiển của người vận hành: Chúng ta có thể mô phỏng một hành động kỹ năng của người vận hành hay đặc tính điều khiển trong thuật ngữ của tập mờ sử dụng với dữ liệu vào - ra để liên kết với các hành động điều khiển của người vận hành. Sau đó chúng ta có thể nhận được "mô hình vào-ra" như là một bộ điều khiển mờ.

– Dựa trên một mô hình mờ hoặc phân tích đặc tính của một quá trình điều khiển: Trong trường hợp này, các luật điều khiển mờ được tạo ra dựa trên hoặc là mô hình mờ hoặc là phân tích đặc tính của quá trình điều khiển.

Nếu chúng ta có một mô hình mờ của quá trình hoặc nếu chúng ta biết một vài tính chất có ích của quá trình, chúng ta có thể thiết kế tạo ra một tập các luật điều khiển mờ để đạt được hoạt động tối ưu. Phương pháp này được áp dụng trong những hệ thống điều khiển thích nghi mờ.

– Dựa trên việc học (hay tự tổ chức): rất nhiều bộ điều khiển mờ đã được xây dựng để mô phỏng khả năng ra quyết định của bộ não con người. Gần đây, rất nhiều sự nghiên cứu cố gắng định hướng vào việc mô phỏng việc học của con người, chủ yếu là khả năng tạo nên các luật điều khiển mờ và sửa chữa chúng dựa trên kinh nghiệm hoạt động của hệ thống. Đây là một hướng phát triển cao hơn của điều khiển thích nghi mờ. Logic mờ kết hợp với mạng Nơron nhân tạo, ANN tạo nên một hệ thống điều khiển tự thích nghi với sự thay đổi của đối tượng điều chỉnh. Ví dụ: Một cách phối hợp đơn giản nhất là ANN quan sát nhận dạng đối tượng sau đó căn cứ trên những phiếm hàm mục tiêu điều khiển để thay đổi, tạo ra các luật của bộ điều khiển mờ.

#### **1.1.2.6. Kết luận**

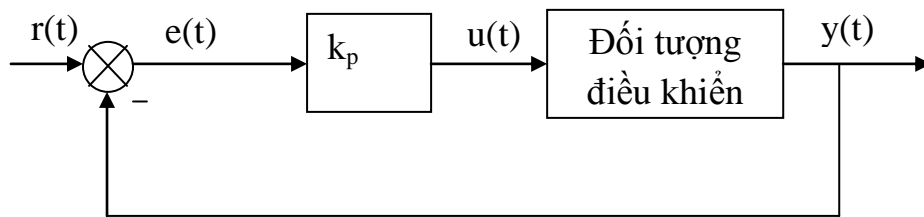
Phương pháp tổng hợp bộ điều khiển theo phương pháp mờ có nhiều ưu điểm hơn so với các phương pháp tổng hợp bộ điều khiển trước đây:

- Giảm được khối lượng công việc do không phải xác định mô hình, giảm khối lượng tính toán mà bộ điều khiển vẫn làm việc tin cậy.
- Cấu trúc đơn giản, dễ hiểu và khả năng thay đổi linh hoạt.
- Làm việc ổn định, bền vững, chất lượng cao, tăng độ tin cậy cho thiết bị và giảm giá thành sản phẩm trong nhiều trường hợp.

## **1.2. BỘ ĐIỀU KHIỂN PID SỐ**

### **1.2.1. Luật điều khiển PID.**

#### **1.2.1.1. Luật điều khiển tỷ lệ: Proportional (P)**



**Hình 1.7: Hệ kín với luật điều khiển tỷ lệ**

Với :  $r(t)$ : tín hiệu đầu vào của hệ thống.

$y(t)$ : tín hiệu đầu ra của hệ thống.

$u(t)$ : tín hiệu điều khiển tác động lên đối tượng.

$e(t)$ : tín hiệu sai lệch điều khiển.

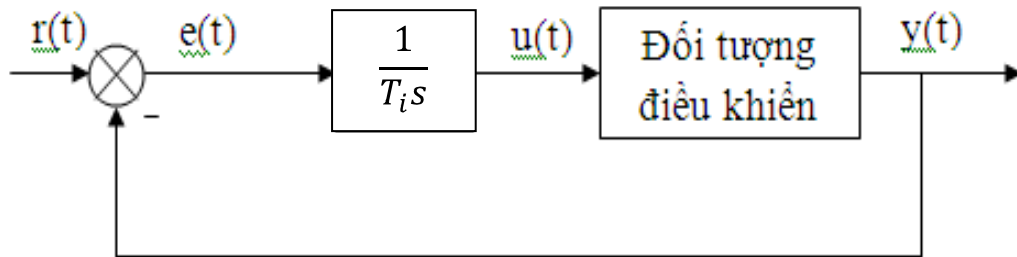
Hệ kín Hình 1.6 được làm việc với bộ điều khiển theo luật tỷ lệ. Như vậy nếu gọi sai lệch  $e(t) = r(t) - y(t)$  là đầu vào của bộ điều khiển và  $u(t)$  là đầu ra của bộ điều khiển thì quan hệ giữa đầu vào của luật điều khiển được biểu diễn theo phương trình sau đây:

$$u(t) = k_p \cdot e(t) \quad (1-24)$$

Trong đó  $k_p$  là tham số của luật điều khiển. Hầu hết các đối tượng công nghiệp đều được điều khiển theo luật này vì luật cho khả năng điều khiển rất đơn giản, nhưng lại có thể thay đổi cả chất lượng động và chất lượng tĩnh của hệ thống. Dễ dàng thấy được nhận định này là đúng đắn bởi vì khi thay đổi  $k_p$  dẫn đến sự thay đổi hệ số khuếch đại của hệ hở, điều đó dẫn đến sự thay đổi vị trí của các điểm cực và điểm không của hệ. Ngoài ra khi thay đổi hệ số  $k_p$ , ta thấy ngay chất lượng của hệ thống ở chế độ xác lập cũng thay đổi. Có thể nhận thấy ngay ở công thức xác nhận sai lệch tĩnh của hệ trong Hình 1.7:

$$\delta(t) = \lim_{t \rightarrow \infty} s \cdot E(s) = \lim_{t \rightarrow \infty} \frac{sR(s)}{1 + K_p \cdot W_{dt}(s)} \quad (1-25)$$

### 1.2.1.2. Luật điều khiển tích phân: Integral (I)



**Hình 1.8: Hệ kín với luật điều khiển tích phân**

Hệ kín trong hình 1.8 được điều khiển bằng luật điều khiển tích phân, như vậy với tín hiệu đầu vào của bộ điều khiển là sai lệch  $e(t)$  và tín hiệu đầu ra  $u(t)$  ta có thể biểu diễn luật điều khiển tích phân bằng phương trình sau:

$$u(t) = \frac{1}{T_i} \int_0^t e(\tau) d\tau \quad (1-26)$$

$$\text{hoặc } u(t) = k_i \int_0^t e(\tau) d\tau \quad (1-27)$$

Tham số của luật điều khiển là  $T_i$  được gọi là hằng số thời gian tích phân hay  $k_i$  là hệ số tích phân. Khi sử dụng luật điều khiển tích phân độ dự trữ ổn định của hệ kín sẽ giảm đi, theo tiêu chuẩn ổn định Nyquist, sai lệch tĩnh sẽ giảm vì hàm truyền hệ hở lúc đó có dạng:

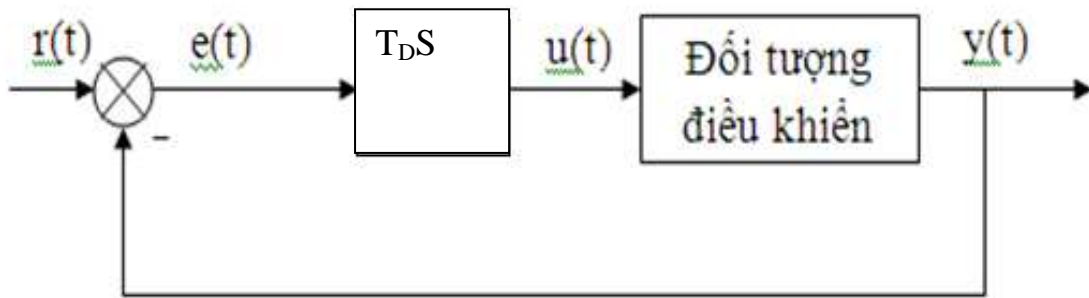
$$w_0 = w_{dk}(s) - w_{dt} = \frac{1}{T_i} w_{dt}(s)$$

và sai lệch của hệ:

$$\delta(t) = \lim_{t \rightarrow \infty} \frac{s.R(s)}{1 + \frac{1}{T_i \cdot s \cdot W_{dt}(s)}} = \lim_{t \rightarrow \infty} \frac{T_i \cdot R(s) \cdot s^2}{T_i + w_{dt}(s)} \quad (1-28)$$

Điều khiển tích phân hay còn gọi là phương pháp điều khiển theo tích lũy sai lệch, điều khiển chậm sau. Phương pháp điều khiển này có ưu điểm là ít chịu ảnh hưởng của nhiễu và làm tăng độ chính xác của hệ ở chế độ xác lập

### 1.2.1.3. Luật điều khiển vi phân: Derivative (D).



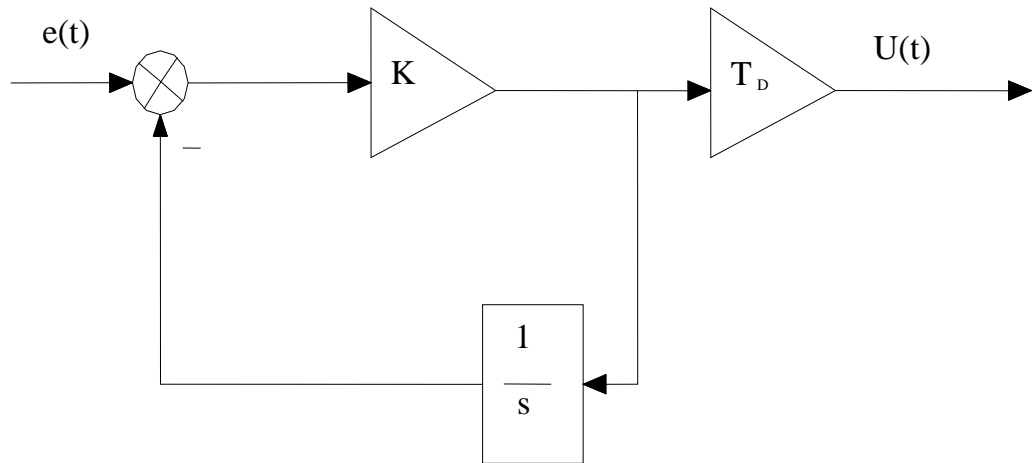
**Hình 1.9: Hệ kín với luật điều khiển vi phân**

Luật vi phân được biểu diễn theo phương trình sau đây:

$$u(t) = T_D \frac{de(t)}{dt} \quad (1-29)$$

Trong đó  $T_D$  là hằng số thời gian vi phân. Luật điều khiển vi phân có tác dụng làm cho giảm thời gian quá độ của hệ. Luật điều khiển vi phân chính vì vậy còn được gọi là luật điều khiển vượt mức. Quyết định điều khiển được đưa ra trên cơ sở thông tin về đạo hàm của sai lệch. Nhược điểm của luật chính là ở phương pháp lấy thông tin này, vì nếu đối tượng chịu ảnh hưởng của các nhiễu biến thiên thì luật sẽ ra quyết định điều khiển theo nhiễu. Chính vì vậy luật chỉ thích hợp khi nhiễu tác động vào đối tượng điều khiển là các nhiễu hằng số. Trong môi trường làm việc của đối tượng điều khiển có nhiễu biến thiên thì không nên sử dụng luật điều khiển này. Trong thực tế các khâu vi phân thường phải thiết kế gần đúng. Hình 1.9 là cơ sở thiết kế luật điều khiển vi phân hay sử dụng:





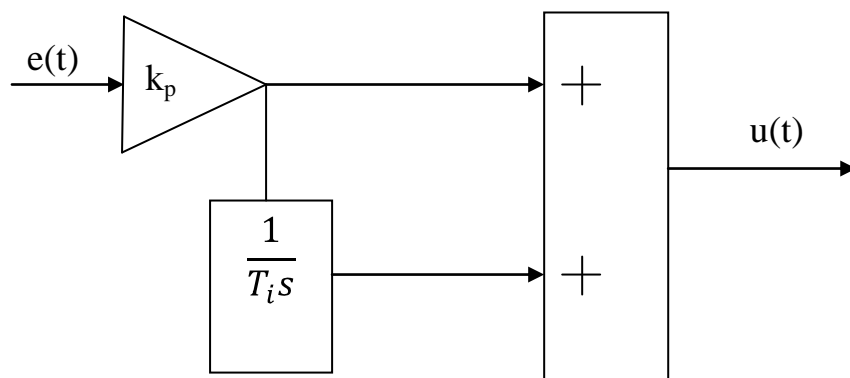
**Hình 1.10: Sơ đồ thiết kế luật vi phân thực tế**

$$\frac{U(s)}{E(s)} = T_{D1} \frac{s}{k s + 1} \approx T_{D1} s \quad (1-30)$$

Như vậy để có được  $T_{D1} s$  ta phải chọn hệ số khuếch đại  $k$  thật lớn ( $k \rightarrow \infty$ ). Thực hiện  $k$  lớn lại phụ thuộc vào khả năng của thiết bị. Đó cũng chính là những hạn chế khi thực hiện luật điều khiển vi phân.

#### 1.2.1.4. Luật điều khiển tỷ lệ - tích phân (PI).

Cấu trúc của luật điều khiển PI được biểu diễn ở hình sau:



**Hình 1.11: Cấu trúc luật PI**

$$u(t) = k_p [e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau] \quad (1-31)$$

Biểu diễn luật dưới dạng hàm truyền:

$$W_{PI}(s) = \frac{k_p(1+T_i s)}{T_i s} \quad (1-32)$$

Khi kết hợp giữa luật tỷ lệ và luật điều khiển tích phân ta có được ưu điểm của cả hai luật. Luật điều khiển PI có khả năng bù hằng số thời gian lớn nhất của đối tượng điều khiển, điều đó chỉ ra ở công thức (1-32) và do vậy nâng cao được chất lượng động của hệ thống điều khiển kín. Luật PI còn có khả năng làm giảm sai lệch tĩnh của hệ, từ công thức (2) ta có:

$$S_t = \lim_{t \rightarrow \infty} \frac{T_i \cdot R(s) \cdot s^2}{T_i \cdot s + k_p(1 + T_i \cdot s) \cdot W_{dt}(s)} \quad (1-33)$$

Phần lớn các đối tượng công nghiệp được điều khiển theo luật điều khiển này.

#### 1.2.1.5. Luật điều khiển PID

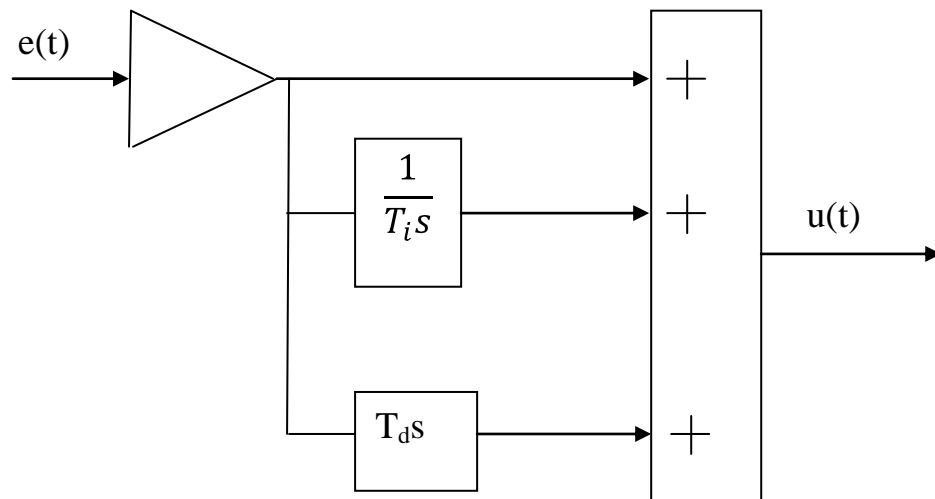
PID là chữ viết tắt của ba thành phần cơ bản có trong bộ điều khiển: Khuếch đại (P), tích phân (I) và vi phân (D). Ba thành phần này phối hợp với nhau, bổ sung những yếu điểm, hạn chế những nhược điểm của nhau, tạo thành bộ điều khiển hoàn chỉnh để điều khiển đối tượng.

Thành phần khuếch đại (P) có tốc độ xử lý tín hiệu nhanh, hệ thống sử dụng quy luật tỷ lệ sẽ có tính ổn định cao, thời gian điều khiển ngắn. Nhưng quy luật này còn có nhược điểm: hệ thống khi ở trạng thái xác lập luôn tồn tại sai lệch tĩnh.

Thành phần tích phân (I) có ưu điểm là triệt tiêu được sai lệch tĩnh nhưng tốc độ xử lý tín hiệu còn chậm.

Thành phần vi phân (D) có tác dụng làm tăng nhanh tốc độ tác dụng của tín hiệu điều khiển nhưng bên cạnh đó hệ thống khi sử dụng thành phần này còn có độ quá điều chỉnh lớn.

Sơ đồ cấu trúc của luật điều khiển PID được biểu diễn ở hình 1.12



**Hình 1.12: Cấu trúc luật PID**

Lý do bộ điều khiển PID được sử dụng rộng rãi là tính đơn giản của nó cả về cấu trúc lẫn nguyên lý làm việc. Bộ điều khiển PID có nhiệm vụ đưa sai lệch  $e(t)$  của hệ thống về 0 sao cho quá trình quá độ thoả mãn các yêu cầu cơ bản về chất lượng:

- + Nếu sai lệch  $e(t)$  càng lớn thì thông qua thành phần tỷ lệ, tín hiệu điều chỉnh  $u(t)$  càng lớn (vai trò khuếch đại),
- + Nếu sai lệch  $e(t)$  chưa bằng 0 thì thông qua thành phần tích phân, bộ điều khiển PID vẫn còn tín hiệu điều chỉnh (vai trò tích phân).
- + Nếu có sự thay đổi của sai lệch  $e(t)$  càng lớn thì thông qua thành phần vi phân, phản ứng thích hợp của tín hiệu điều chỉnh  $u(t)$  sẽ càng nhanh (vai trò của vi phân).

Luật điều khiển PID được biểu diễn bằng phương trình sau đây:

$$u(t) = k_p[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_D \frac{de(t)}{dt}] \quad (1-34)$$

Trong đó  $e(t)$  là tín hiệu đầu vào,  $u(t)$  là tín hiệu đầu ra,  $k_p$  được gọi là hệ số khuếch đại,  $T_i$  là hằng số thời gian tích phân và  $T_D$  là hằng số thời gian vi phân. Chất lượng của hệ thống phụ thuộc vào các tham số  $k_p$ ,  $T_i$ ,  $T_D$ . Muốn hệ thống có chất lượng như mong muốn thì phải phân tích đối tượng rồi trên cơ sở đó chọn các tham số trên cho phù hợp.

Bằng hàm truyền luật điều khiển PID thường được biểu diễn dưới dạng cộng hoặc dạng nhân. Hàm truyền biểu diễn dưới dạng cộng:

$$W_{PID}^+(s) = k_p[1 + \frac{1}{T_i} + T_D s] \quad (1-35)$$

Hàm truyền biểu diễn dưới dạng nhân:

$$W_{PID.MUL}(s) = \frac{k_p^* (1+T_i^* s)(1+T_D^* s)}{T_i^*} \quad (1-36)$$

Dạng (13) thuận lợi cho việc tính toán tham số của bộ điều khiển, còn luật cộng (1-35) là luật được cài đặt trong các bộ PID số do các hãng công nghiệp cung cấp cho người sử dụng, ở dạng (1-35) nhà chế tạo dễ dàng cung cấp giải pháp lựa chọn luật điều khiển khác như luật P, I, luật PI, luật PD trên nền luật PID. Quan hệ của các tham số giữa hai luật được biểu diễn bằng các phương trình dưới đây:

$$K_p = \frac{k_p^*(T_i^* + T_D^*)}{T_i^*} \quad (1-37)$$

$$T_i = T_i^* + T_D^* \quad (1-38)$$

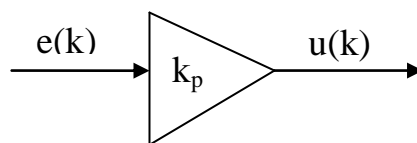
$$T_D = \frac{T_i^* T_D^*}{T_i^* + T_D^*} \quad (1-39)$$

Luật PID được sử dụng rất nhiều trong công nghiệp, theo thống kê của các nhà thiết kế hệ thống điều khiển quá trình trên thế giới cho đến nay 90% đối tượng công nghiệp được điều khiển bằng luật điều khiển này hay trên cơ sở của luật điều khiển này. Luật PID có khả năng bù các hằng số thời gian lớn của đối tượng điều khiển làm cho hệ kín chuyển trạng thái nhanh và làm giảm sai số tĩnh của hệ ở chế độ xác lập.

### 1.2.2. Bộ điều khiển PID số

Yêu cầu thiết kế được đặt ra là bộ PID số phải có tính linh hoạt cao, có nghĩa là phải có giao diện thân thiện với người sử dụng. Thông qua HMI, người sử dụng có thể chọn luật điều khiển dễ dàng. Ví dụ như có thể điều khiển các đối tượng công nghiệp theo luật P, I, PI, PID và có thể lựa chọn tham số của các luật phù hợp với đối tượng thiết kế.

#### 1.2.2.1. Luật điều khiển tỷ lệ số



**Hình 1.13: Cấu trúc luật P số.**

Đây là luật điều khiển có thể thiết kế đơn giản nhất. Dãy  $u(k)$  được tính từ dãy  $e(k)$  theo công thức:

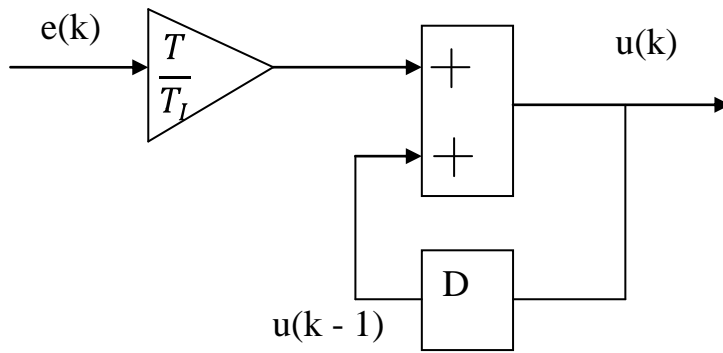
$$u(k) = k_p e(k) \quad k = 0, 1, 2, \dots \quad (1-40)$$

#### 1.2.2.2. Luật điều khiển tích phân số

Từ công thức số (3) ta có phương trình sai phân:

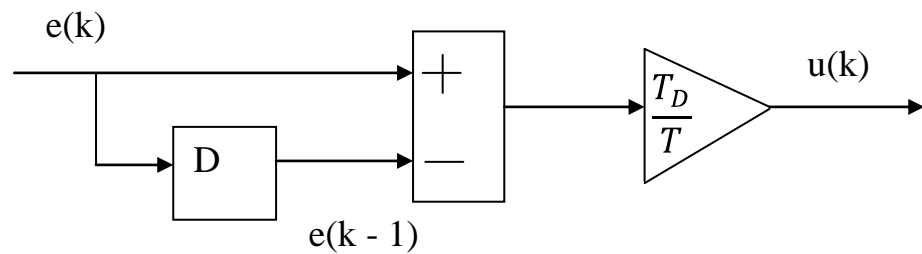
$$u(k) = \frac{T}{T_i} e(k) + u(k-1) \quad (1-41)$$

Trong đó  $T$  là thời gian trích mẫu (Sample Time)



**Hình 1.14: Cấu trúc luật I số**

### 1.2.2.3. Luật điều khiển vi phân số



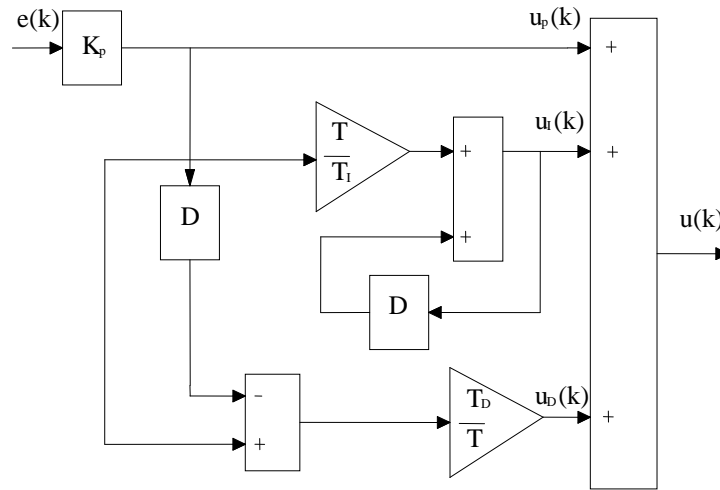
**Hình 1.15: Cấu trúc luật D số**

Thường các bộ điều khiển theo luật vi phân số được cài đặt theo các phương trình sai phân sau:

$$u(k) = \frac{T_D}{T} [e(k) - e(k - 1)] \quad (1-42)$$

Trong đó T là thời gian trích mẫu.

### 1.2.2.4. Luật điều khiển PID số



**Hình 1.16: Cấu trúc luật PID số**

Từ cấu trúc PID số trong hình 1.16, ta có:

$$u(k) = k_p \left\{ e(k) + \frac{T}{T_i} e(k) + u_i(k-1) + \frac{T_D}{T} [e(k) - e(k-1)] \right\} \quad (1-43)$$

$$u(k) = k_p \left\{ \left(1 + \frac{T_D}{T}\right) e(k) + \frac{T}{T_i} e(k) - \frac{T_D}{T} e(k-1) + u_i(k-1) \right\}$$

$$u(k) = k_p \left\{ \left(1 + \frac{T_D}{T} + \frac{T}{T_i}\right) e(k) - \frac{T_D}{T} e(k-1) + u_i(k-1) \right\}$$

Luật điều khiển PID số trong công thức trên được lựa chọn để cài đặt cho bộ điều khiển được chế tạo trên nền AVR.

## CHƯƠNG 2

### THIẾT KẾ BỘ ĐIỀU CHỈNH SỬ DỤNG CHÍP AVR

---

Với sự phát triển mạnh của rất nhiều vi điều khiển lên có rất nhiều phương án cho việc chọn lựa cho khối điều khiển trung tâm. Ở đề án này em đã chọn họ vi điều khiển AVR của hãng Atmel. Hãng Atmel đã từ lâu nổi tiếng với chip vi điều khiển họ 89Cxx phù hợp với các ứng dụng đơn giản. Chuyển sang họ AVR, Atmel đã thêm vào chip vi điều khiển này nhiều tính năng mà chip họ 8051 không có như là ADC, PWM, BUS I<sup>2</sup>C, 2 Wire v.v..., để giúp cho người sử dụng có thêm nhiều tính năng để sử dụng. AVR là vi điều khiển được thiết kế cho rất nhiều ứng dụng. Từ các ứng dụng điều khiển, đo lường...

#### 2.1. CHỈNH ĐỊNH MỜ BỘ ĐIỀU KHIỂN PID

Trong lý thuyết điều khiển tuyến tính, có nhiều phương pháp hữu hiệu để xác định tham số  $K_R$ ,  $T_i$ ,  $T_D$  cho bộ điều khiển PID. Tuy nhiên, hạn chế chung của các phương pháp này là chỉ tổng hợp được một bộ điều khiển (PID) cho một đối tượng xác định. Với một đối tượng khác cần phải tổng hợp một bộ điều khiển khác. Phương pháp chỉnh định mờ tham số bộ điều khiển PID cho phép một bộ điều khiển (PID) có thể làm việc với nhiều đối tượng khác nhau. Tư tưởng cơ bản của phương pháp này là ứng dụng lý thuyết mờ vào việc chỉnh định tham số  $K_R$ ,  $T_i$ ,  $T_D$  của bộ điều khiển PID sao cho phù hợp với đối tượng hiện tại.

Sau đây, tôi xin trình bày hai phương pháp chỉnh định mờ tham số bộ điều khiển PID:

- Phương pháp thứ nhất là phương pháp chỉnh định mờ của Zhao, Tomizuka và Isaka.
- Phương pháp thứ hai là phương pháp chỉnh định mờ tham số  $\alpha$ .



### 2.1.1. Phương pháp chỉnh định mờ của Zhao, Tomizuka và Isaka

Ta có mô hình toán học của một bộ điều khiển PID với đầu vào  $e(t)$ , đầu ra  $u(t)$ .

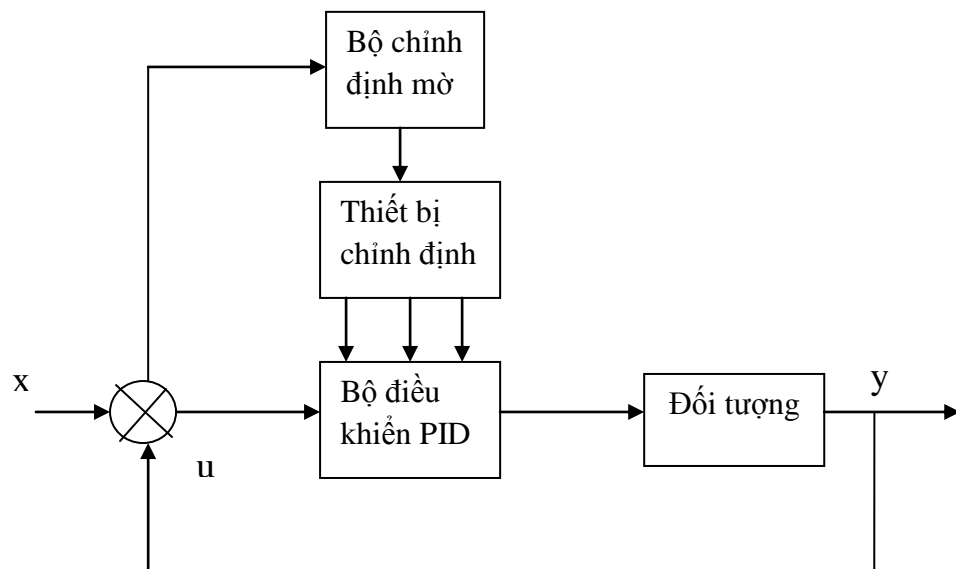
$$u(t) = k_p[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_D \frac{de(t)}{dt}] \quad (2-1)$$

Hàm truyền của bộ điều khiển:  $G_{PID}(s) = k_R[1 + \frac{1}{T_i s} + T_D s]$

Hoặc  $G_{PID}(s) = K_R + \frac{K_I}{s} + K_D s$

Trong đó:  $K_R = k_R$ ,  $K_I = \frac{k_R}{T_i}$ ,  $K_D = k_R \cdot T_D$

Các tham số  $k_R$ ,  $T_i$ ,  $T_D$  hay  $K_R$ ,  $K_I$ ,  $K_D$  của bộ điều khiển PID được chỉnh định mờ trên cơ sở phân tích tín hiệu chủ đạo và tín hiệu ra của hệ thống, chính xác hơn là sai lệch  $e(t)$  và đạo hàm của sai lệch  $\frac{de(t)}{dt}$ . Sơ đồ hệ thống sử dụng bộ điều khiển PID có các tham số được chỉnh định theo phương pháp mờ được chỉ ra ở hình sau:

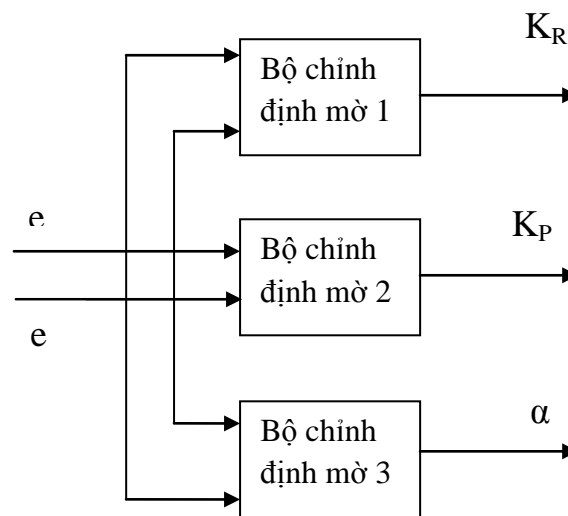


**Hình 2.2: Phương pháp chỉnh định mờ tham số bộ điều khiển PID của Zhao, Tomizuka và Isaka**

Bộ chỉnh định mờ trong sơ đồ trên có hai đầu vào là sai lệch tĩnh  $e(t)$ , đạo hàm sai lệch  $\frac{de(t)}{dt}$  và ba đầu ra là  $K_R$ ,  $K_I$  và  $\alpha$ .

$$\text{Trong đó: } \alpha = \frac{T_I}{T_D} \Rightarrow K_I = \frac{K_R^2}{\alpha K_D}$$

Do đó có thể xem bộ chỉnh định mờ gồm ba bộ chỉnh định mờ nhỏ, mỗi bộ có hai đầu vào một đầu ra.



**Hình 2.3: Cấu trúc bên trong bộ chỉnh định mờ**

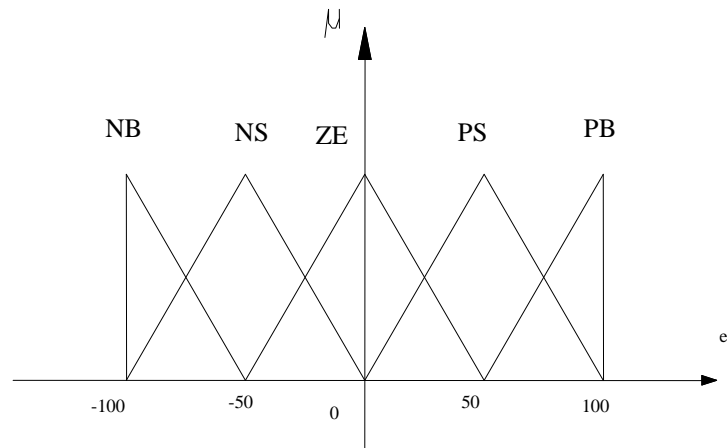
Bộ chỉnh định mờ là thành phần quan trọng trong sơ đồ Hình 2.2. Thiết kế bộ chỉnh định như sau:

- Bộ chỉnh định mờ 1 ( chỉnh định  $K_R$ )

Bộ chỉnh định mờ 1 có hai đầu vào là sai lệch  $e(t)$ , đạo hàm sai lệch  $\frac{de(t)}{dt}$ .

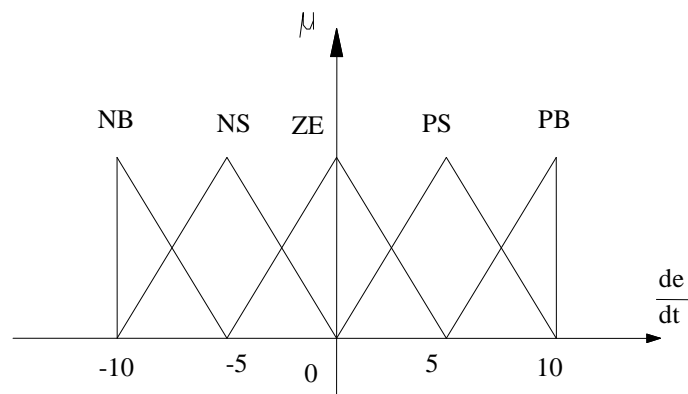
Đầu ra là giá trị chỉnh định  $K_R$

- + Đầu vào 1( sai lệch  $e(t)$ ): Chọn dải sai lệch và tập mờ như hình sau:



**Hình 2.4: Tập mờ đầu vào 1, bộ chỉnh định  $K_R$**

+ Đầu vào 2 (tốc độ sai lệch  $\frac{de(t)}{dt}$ ): Chọn dải tốc độ sai lệch và tập mờ như hình sau:



**Hình 2.5: Tập mờ đầu vào 2, bộ chỉnh định  $K_R$**

+ Luật chỉnh định  $K_R$ :

Luật điều khiển để chỉnh định các tham số của bộ điều khiển PID được xây dựng theo nguyên tắc: tín hiệu điều khiển càng mạnh nếu  $K_R$  càng lớn,  $K_D$  và  $\alpha$  càng nhỏ. Khi giá trị tuyệt đối của sai lệch càng lớn cần có tín hiệu điều khiển mạnh để đưa sai lệch nhanh về 0, theo bảng sau:

**Bảng 1.1: Luật chỉnh định  $K_R$**

		E				
		NB	NS	ZE	PS	PB
$\frac{de}{dt}$	NB	B	S	S	S	B
	NS	B	B	S	B	B
	ZE	B	B	B	B	B
	PS	B	B	S	B	B
	PB	B	S	S	S	B

+ Luật hợp thành là luật MAX – MIN, phương pháp giải mờ là phương pháp điểm trọng tâm.

- **Bộ chỉnh định mờ 2 (Chỉnh định  $K_D$ )**

Bộ chỉnh định mờ 2 cũng có hai đầu vào là sai lệch  $e(t)$ , đạo hàm sai lệch  $\frac{de(t)}{dt}$ . Đầu ra là giá trị  $K_D$  đã chỉnh định.

+ Đầu vào và đầu ra của bộ chỉnh định mờ 2 giống bộ chỉnh định I. Tức là sai lệch  $e(t)$ , đạo hàm sai lệch  $\frac{de(t)}{dt}$  và đầu ra  $K_D$  có dải giá trị và hàm thuộc như bộ chỉnh định mờ 1.

+ Luật chỉnh định  $K_D$ : Tín hiệu điều khiển càng mạnh nếu  $K_R$  càng lớn,  $K_D$  và  $\alpha$  càng nhỏ. Khi giá trị tuyệt đối của sai lệch càng lớn cần có tín hiệu điều khiển mạnh để đưa sai lệch nhanh về 0. Trên cơ sở đó, ta xây dựng luật chỉnh định  $k_D$  như bảng sau:

**Bảng 1.2: Luật chỉnh định  $K_D$**

		E				
		NB	NS	ZE	PS	PB
$\frac{de}{dt}$	NB	S	B	B	B	S
	NS	S	B	B	B	S
	ZE	S	S	B	S	S

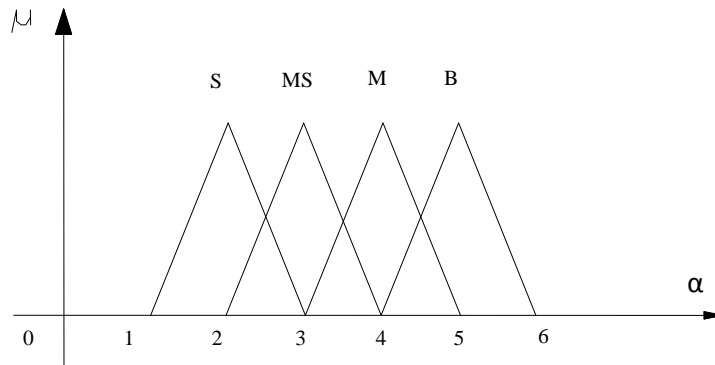
	<b>PS</b>	<b>S</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>S</b>
	<b>PB</b>	<b>S</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>S</b>

+ Luật hợp thành là luật MAX – MIN, phương pháp giải mờ là phương pháp điểm trọng tâm.

- **Bộ chỉnh định mờ 3 ( Chỉnh định  $\alpha$  )**

+ Đầu vào bộ chỉnh định mờ 3 cùng các tập mờ của chúng giống như đầu vào của bộ chỉnh định mờ 1 và 2.

+ Đầu ra cùng các hàm thuộc bộ chỉnh định mờ cho trong hình sau:



**Hình 2.6: Đầu ra bộ chỉnh định  $\alpha$**

+ Luật chỉnh định: Luật chỉnh định  $\alpha$  cho trong bảng sau.

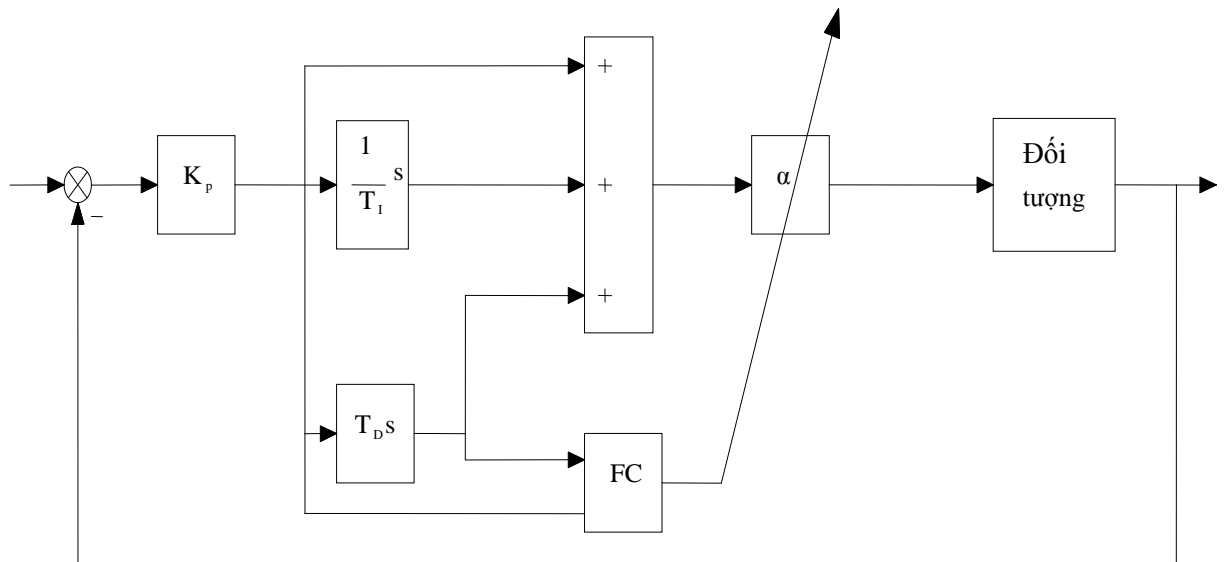
**Bảng 1.3: Luật chỉnh định  $\alpha$**

		<b>E</b>				
		<b>NB</b>	<b>NS</b>	<b>ZE</b>	<b>PS</b>	<b>PB</b>
$\frac{de}{dt}$	<b>NB</b>	S	M	B	M	S
	<b>NS</b>	S	MS	MS	MD	S
	<b>ZE</b>	S	S	MS	D	S
	<b>PS</b>	S	MS	MS	MD	S
	<b>PB</b>	S	M	B	M	S

+ Luật hợp thành là luật MAX – MIN, phương pháp giải mờ là phương pháp điểm trọng tâm.

### 2.1.2. Phương pháp chỉnh định mờ hệ số $\alpha$

- Nội dung phương pháp này thể hiện trong hình sau:



**Hình 2.7: Phương pháp chỉnh định mờ hệ số  $\alpha$**

Trong sơ đồ trên, các tham số  $k_p$ ,  $T_I$ ,  $T_D$  được đưa vào trước khi hệ thống hoạt động. Tín hiệu ra  $u$  của bộ điều khiển PID được hiệu chỉnh sao cho phù hợp với đối tượng điều khiển.

Tín hiệu đầu ra của bộ điều khiển PID trước khi chỉnh định:

$$u(t) = k_p e(t) + \frac{k_p}{T_I} \int_0^t e(\tau) d\tau + k_p T_D \frac{de(t)}{dt}$$

Tín hiệu đầu ra của bộ điều khiển PID sau khi chỉnh định:

$$\begin{aligned} \alpha u(t) &= \alpha k_p e(t) + \alpha \frac{k_p}{T_I} \int_0^t e(\tau) d\tau + \alpha k_p T_D \frac{de(t)}{dt} \\ &= \alpha k_p e(t) + \frac{k_p}{\frac{T_I}{\alpha}} \int_0^t e(\tau) d\tau + \alpha k_p T_D \frac{de(t)}{dt} \end{aligned} \quad (2-2)$$

$$\text{Đặt: } \begin{cases} T_I' = \frac{T_I}{\alpha} \\ T_D' = \alpha T_D \\ k_p' = \alpha k_p \end{cases}$$

Do đó việc chỉnh định tín hiệu  $u$  có thể coi như chỉnh định  $k_p$ ,  $T_I$ ,  $T_D$ . Với một hệ số  $\alpha$  thích hợp, sẽ có một bộ điều khiển với tham số phù hợp cho đối tượng ổn định.

Khâu FC (Fuzzy Control) trong sơ đồ trên có nhiệm vụ tạo ra tín hiệu hiệu chỉnh  $D\alpha$  để hiệu chỉnh hệ số  $\alpha$  theo nguyên tắc: Nếu sai lệch của hệ thống càng lớn thì tín hiệu chỉnh càng nhỏ và ngược lại, nếu sai lệch của hệ thống càng nhỏ thì tín hiệu hiệu chỉnh càng lớn. Khâu FC nhận 2 đầu vào lấy ra từ bộ điều khiển PID là  $k_p e$  và  $T_D \frac{de(t)}{dt}$  ( sai lệch và đạo hàm sai lệch đã nhận thêm các hệ số tương ứng). Hệ số hiệu chỉnh  $\alpha$  gồm 2 thành phần: thành phần ban đầu  $\alpha_0$  và thành phần hiệu chỉnh  $D\alpha$ :

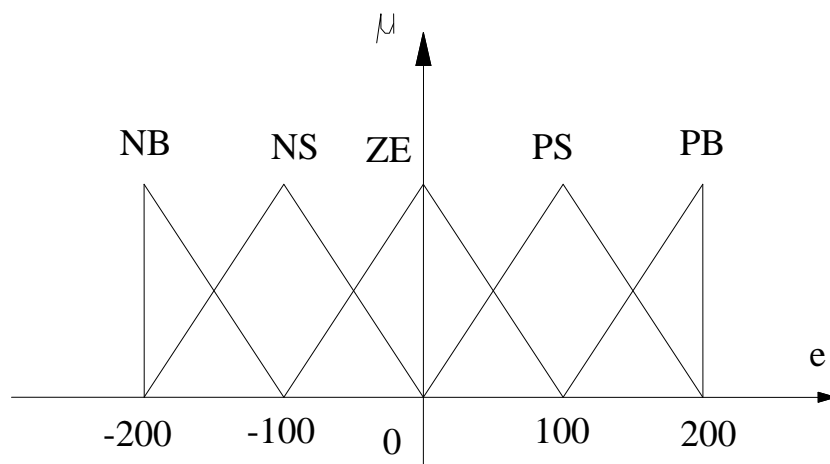
$$\alpha = \alpha_0 + D\alpha \text{ với } \alpha_0 = 1.$$

Khâu  $\alpha$  trong hình 1.20 có chức năng tạo tín hiệu điều khiển  $u_{dk} = \alpha u$ .

Tín hiệu này sẽ trực tiếp điều khiển đối tượng.

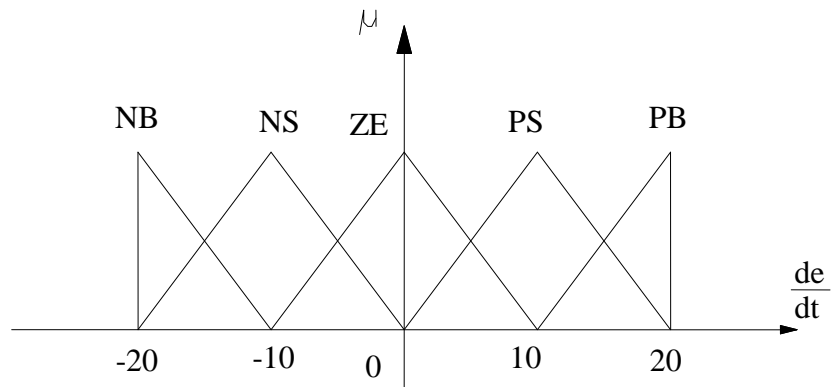
- **Thiết kế khâu FC**

+ Dải giá trị và các tập mờ của đầu vào 1



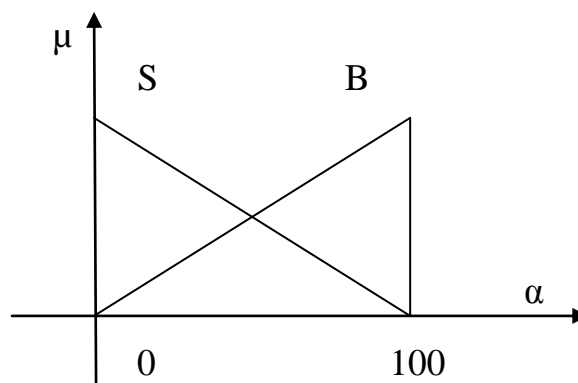
**Hình 2.8: Tập mờ đầu vào 1**

+ Dải giá trị và các tập mờ của đầu vào 2



**Hình 2.9: Tập mờ đầu vào 2**

+ Dải giá trị và các tập mờ của đầu ra



**Hình 2.10: Tập mờ đầu ra**

+ Luật chỉnh định: Luật chỉnh định làm việc theo nguyên tắc: nếu sai lệch của hệ thống càng lớn thì tín hiệu hiệu chỉnh càng nhỏ và ngược lại, nếu sai lệch của hệ thống càng nhỏ thì tín hiệu hiệu chỉnh càng lớn.

**Bảng 1.4: Luật chỉnh định hệ số  $\alpha$**

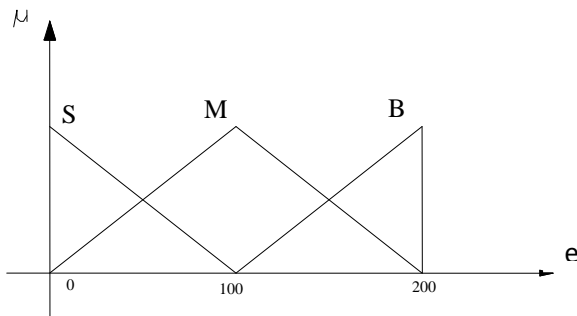
		E				
		NB	NS	ZE	PS	PB
$\frac{de}{dt}$	NB	S	B	B	B	S



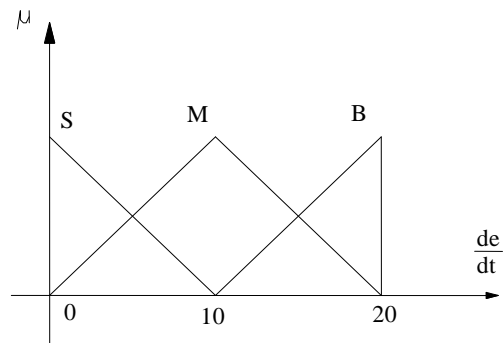
	<b>NS</b>	<b>S</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>S</b>
	<b>ZE</b>	<b>S</b>	<b>S</b>	<b>B</b>	<b>S</b>	<b>S</b>
	<b>PS</b>	<b>S</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>S</b>
	<b>PB</b>	<b>S</b>	<b>B</b>	<b>B</b>	<b>B</b>	<b>S</b>

+ Luật hợp thành là luật MAX – MIN, phương pháp giải mờ là phương pháp điểm trọng tâm.

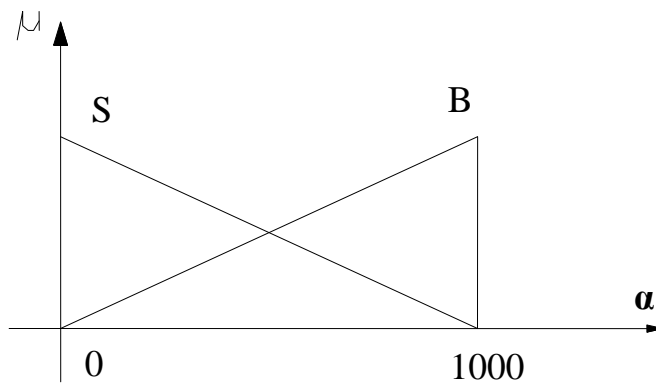
Do các tập mờ đầu vào và ra hai bên bảng luật điều chỉnh định là đối xứng nhau nên  $\alpha$  sẽ nhận giá trị như nhau với cùng một độ lớn nhưng khác dấu của sai lệch  $e(t)$ . Do đó, ta chỉnh lại các tập mờ vào ra và luật chỉnh định như sau:



**Hình 2.11: Tập mờ đầu vào 1**



**Hình 2.12: Tập mờ đầu vào 2**



**Hình 2.13: Tập mờ đầu ra**

**Bảng 1.5: Luật chỉnh định hệ số  $\alpha$  thu gọn**

		E		
		<b>S</b>	<b>M</b>	<b>B</b>
$\frac{de}{dt}$	<b>S</b>	B	B	S
	<b>M</b>	B	B	S
	<b>B</b>	B	S	S

## 2.2. Chíp AVR ATMEL16

Vi điều khiển 8 bit AVR Atmega16 là loại vi điều khiển có 16 kbyte bộ nhớ Flash để nạp chương trình điều khiển do công ty Atmel chế tạo. Atmega16 có kiến trúc hiện đại kiểu RISC nâng cao (Advanced Reducer Instruction Set Computer) và tiêu thụ ít năng lượng. Các đặc điểm cơ bản của Atmega16 như sau

### 2.2.1. Cấu trúc

- 131 lệnh, hầu hết trong đó chỉ thực hiện trong một chu kỳ đồng hồ.
- 32 thanh ghi đa năng 8 bit.
- Tốc độ xử lý dữ liệu lên đến 16 triệu lệnh 1 giây ở tần số 16 MHz.
- Có bộ thực hiện phép Nhân 2 vòng.

*Bộ nhớ, loại không tự mất dữ liệu.*

- Bộ nhớ 16K byte có thể lập trình được ngay trên bản mạch hệ thống mà không phải tháo rời chip. Số lần ghi - xóa cho phép là 10.000 lần.
- 512 byte EEPROM lập trình được ngay trong hệ thống với số lần ghi - xóa là 100.000 lần.
- 1K byte SRAM.
- Thời gian duy trì dữ liệu: 20 năm tại 85<sup>0</sup>C và 100 năm tại 25<sup>0</sup>C. Cho phép lập trình bảo mật chương trình và dữ liệu trong bộ nhớ.

*Ngoại vi:*

- 2 bộ đếm/định thời 8 bit với bộ chia tần riêng.
- 1 bộ đếm/định thời 16 bit với bộ chia tần riêng.
- Chức năng đếm thời gian thực với mạch dao động riêng.
- 8 kênh vào cho bộ chuyển đổi ADC 10 bit.
- Bộ định thời watchdog lập trình được với mạch dao động riêng có sẵn trên chip.

- Bộ so sánh tương tự trên chip.

*Giao tiếp:*

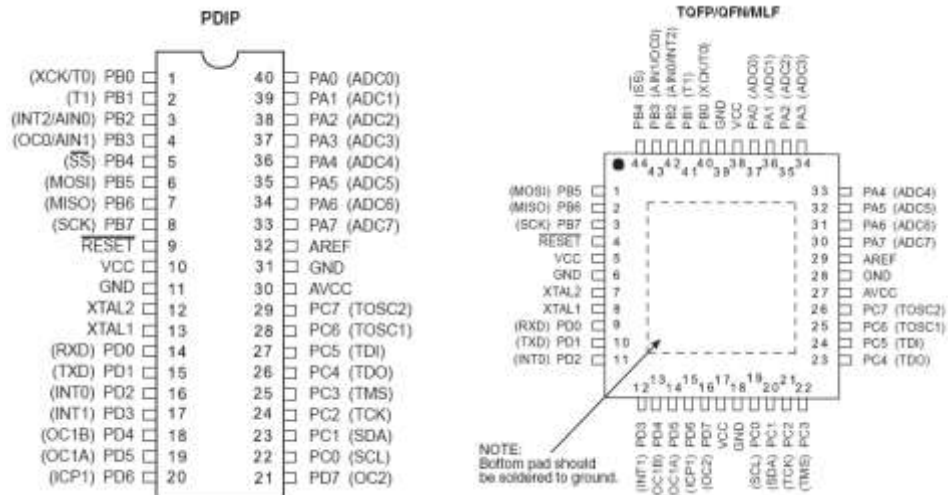
- Chức năng giao tiếp hướng theo byte kiểu 2 dây.
- Bộ USART nối tiếp lập trình được.
- Giao tiếp nối tiếp SPI chủ/tớ.
- Giao tiếp JTAG.

*Các đặc điểm khác của vi điều khiển:*

- Reset khi cấp nguồn và có thể lập trình để dò sụt áp nguồn nuôi.
- Có sẵn bộ dao động RC trên chip.
- Cho phép lập trình từ các nguồn ngắt bên trong và bên ngoài.
- Có 6 chế độ ngủ

*Vào/ ra và đóng gói:*

- 32 đường vào/ra lập trình được.
- PDIP40 có 40 chân, PLCC và TQPF 44 chân như trong hình sau:



**Hình 2.14. Sơ đồ bố trí chân và các kiểu khác nhau của ATmega16**

*Điện áp hoạt động:*

- 4.5 – 5.5 V

*Dải tốc độ:*

- 0 – 16MHz

*Công suất tiêu thụ tại 1MHz và 250C:*

- Chế độ tích cực: 1.1 mA
- Chế độ chờ: 0.35 mA
- Chế độ giảm năng lượng: < 1  $\mu$ A.

*Công nghệ chế tạo bán dẫn:*

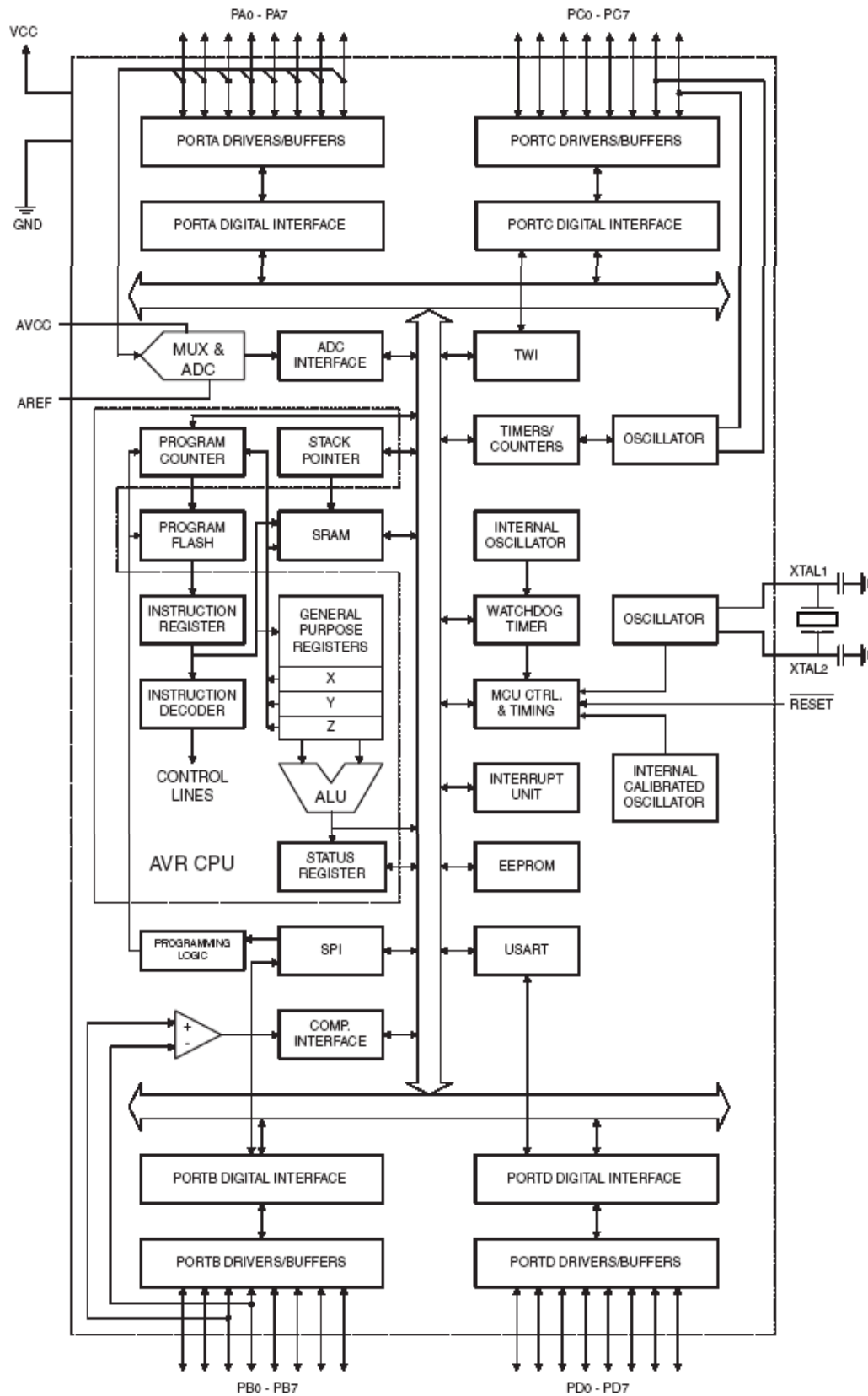
- Công nghệ CMOS.

Vi điều khiển được chế tạo với công nghệ bộ nhớ không mất dữ liệu mật độ cao của Atmel đồng thời cũng cho phép lập trình ngay trên hệ thống để nạp dữ liệu và chương trình vào bộ nhớ này. Điều này mang lại tính mềm dẻo và hiệu quả cao về thời gian và chi phí trong nhiều ứng dụng vi điều khiển. Vi điều khiển được hỗ trợ bởi một tập hợp đầy đủ các công cụ phát triển chương trình và hệ thống bao gồm các trình biên dịch ngôn ngữ lập trình C, các trình dịch hợp ngữ macro, các chương trình gỡ rối, mô phỏng chương trình, các bộ mạch mô phỏng có sẵn kèm theo các thiết bị đánh giá thử nghiệm.

Lõi của vi điều khiển AVR Atmega16 kết hợp tập lệnh phong phú với 32 thanh ghi đa năng. Tất cả 32 thanh ghi này được nối trực tiếp với đơn vị số học và lôgic ALU. Cách thức này cho phép hai thanh ghi độc lập nhau được truy nhập trong cùng một lệnh đơn, thực hiện trong một chu kỳ đồng hồ. Kết quả là tốc độ xử lý nhanh hơn gấp 10 lần kiểu vi điều khiển theo kiến trúc CISC. Vi điều khiển Atmega 16 sử dụng kiến trúc Harvard theo phân loại đối với phương thức truy cập bộ nhớ và bus riêng biệt cho chương trình và dữ liệu. Bộ nhớ chương trình được thực thi với một đường ống hai tầng. Trong khi lệnh đang được thực thi thì lệnh tiếp theo được nhận vào bộ nhớ chương trình. Giải pháp này cho phép các lệnh được thực thi trong mọi chu kỳ xung nhịp. Điều này rõ ràng đã làm tăng đáng kể tốc độ xử lý của vi điều khiển. Bộ nhớ chương trình là bộ nhớ flash lập trình được ngay trên hệ thống như đã nói đến ở trên. Các loại vi điều khiển ATmega 16 được chỉ ra trong bảng .

**Bảng 1.5 Các loại vi điều khiển ATmega16**

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
8	2.7 - 5.5V	ATmega16L-8AC	44A	Commercial (0°C to 70°C)
		ATmega16L-8PC	40P6	
		ATmega16L-8MC	44M1	
		ATmega16L-8AI	44A	Industrial (-40°C to 85°C)
		ATmega16L-8AU <sup>(1)</sup>	44A	
		ATmega16L-8PI	40P6	
		ATmega16L-8PU <sup>(1)</sup>	40P6	
ATmega16L-8MI	44M1			
ATmega16L-8MU <sup>(1)</sup>	44M1			
16	4.5 - 5.5V	ATmega16-16AC	44A	Commercial (0°C to 70°C)
		ATmega16-16PC	40P6	
		ATmega16-16MC	44M1	
		ATmega16-16AI	44A	Industrial (-40°C to 85°C)
		ATmega16-16AU <sup>(1)</sup>	44A	
		ATmega16-16PI	40P6	
		ATmega16-16PU <sup>(1)</sup>	40P6	
ATmega16-16MI	44M1			
ATmega16-16MU <sup>(1)</sup>	44M1			



**Hình 2.15: Sơ đồ cấu trúc bên trong vi điều khiển ATmega16**

### 2.2.2. Mô tả các chân

VCC: Điện áp nguồn nuôi.

GND: Đất.

Cổng A (PA7...PA0): là cổng vào/ra hai hướng 8 bit. Các chân của cổng có các điện trở nối lên nguồn dương (pull – up) có thể chọn cho từng bit. Các chân ra của cổng A có thể cho phép dòng điện 20 mA đi qua và trực tiếp điều khiển Led hiển thị. Khi các chân PA0 đến PA7 là các lối vào và được đặt xuống mức thấp từ bên ngoài, chúng sẽ là nguồn dòng nếu các điện trở bên trong nối lên nguồn dương được kích hoạt. Các chân của cổng A ở trạng thái có trở kháng cao khi tín hiệu reset ở mức tích cực hoặc ngay cả khi không có tín hiệu giữ nhịp.

Cổng B (PB7...PB0): có chức năng tương tự như cổng A. Cổng B cũng cung cấp các chức năng ứng với các tính năng đặc biệt khác của ATmega16 [9].

Cổng C (PC7...PC0): có chức năng tương tự như cổng A. Cổng C cũng cung cấp các chức năng cho giao tiếp JTAG và ứng với các tính năng đặc biệt khác của ATmega16 [9].

Cổng D (PD7...PD0): có chức năng tương tự như cổng A. Cổng B cũng cung cấp các chức năng cho giao tiếp nối tiếp dùng bộ USART và ứng với các tính năng đặc biệt khác của ATmega16 [9].

Reset: Lối vào đặt lại. Bộ vi điều khiển sẽ được đặt lại khi chân này ở mức thấp trong hơn 50 ns ngay cả khi không có tín hiệu giữ nhịp.

XTAL1, XTAL2: nối với bộ dao động thạch anh.

AVCC: Điện áp cho cổng A khi dùng với bộ ADC bên trong.

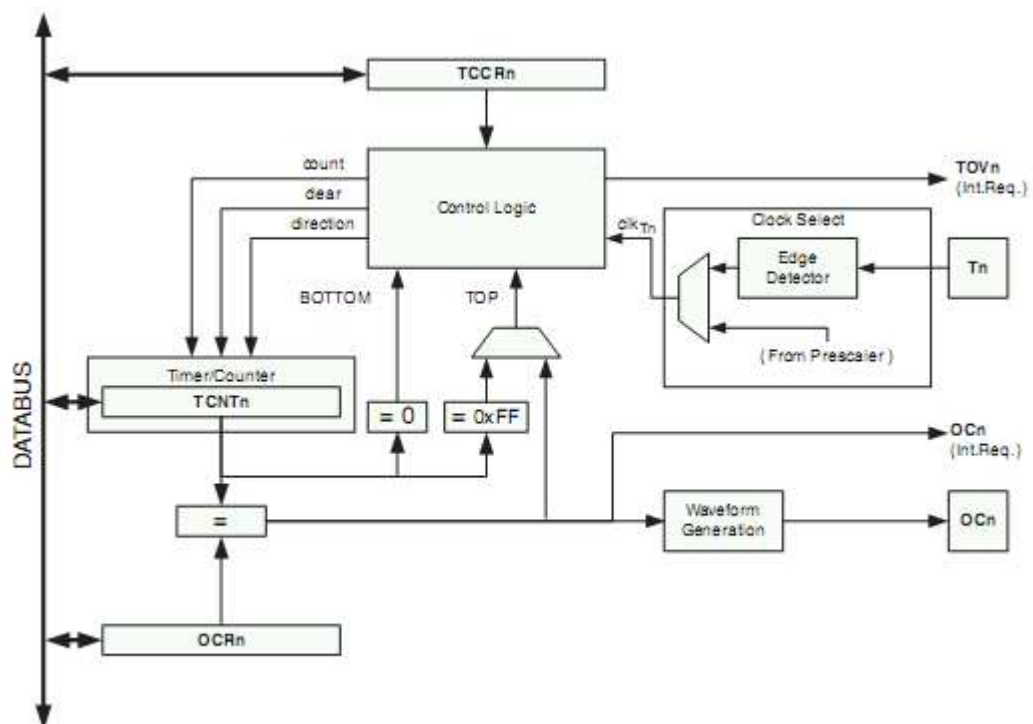
AREF: Cung cấp điện áp chuẩn cho bộ ADC.

### 2.2.3. Bộ định thời

Bộ định thời (timer/counter0) là một module định thời/đếm 8 bit, có các đặc điểm sau:

- Bộ đếm một kênh
- Xóa bộ định thời khi trong mode so sánh (tự động nạp)
- PWM
- Tạo tần số
- Bộ đếm sự kiện ngoài
- Bộ chia tần 10 bit
- Nguồn ngắt tràn bộ đếm và so sánh

Sơ đồ cấu trúc của bộ định thời:



*Hình 2.16: Sơ đồ cấu trúc bộ định thời*

### 2.2.3.1. Các thanh ghi

TCNT0 và OCR0 là các thanh ghi 8 bit. Các tín hiệu yêu cầu ngắt đều nằm trong thanh ghi TIFR. Các ngắt có thể được che bởi thanh ghi TIMSK.

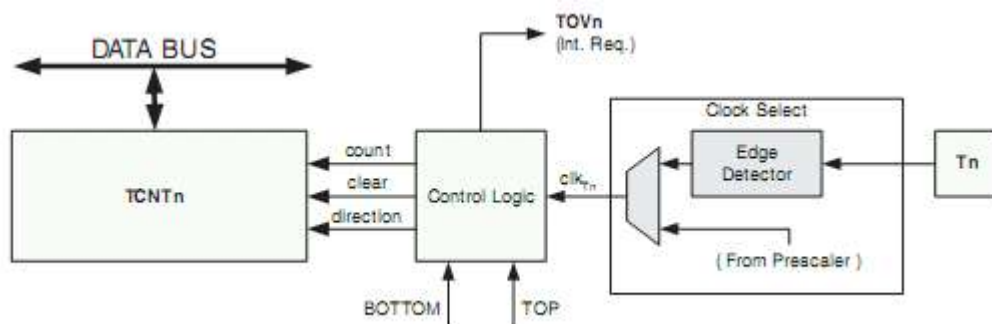
Bộ định thời có thể sử dụng xung clock nội thông qua bộ chia hoặc xung clock ngoài trên chân T0. Khối chọn xung clock điều khiển việc bộ định thời/bộ đếm sẽ dùng nguồn xung nào để tăng giá trị của nó. Ngõ ra của khối chọn xung clock được xem là xung clock của bộ định thời ( $clk_{T0}$ ).



Thanh ghi OCR0 luôn được so sánh với giá trị của bộ định thời/bộ đếm. Kết quả so sánh có thể được sử dụng để tạo ra PWM hoặc biến đổi tần số ngõ ra tại chân OC0.

### 2.2.3.2. Đơn vị đếm

Phần chính của bộ định thời 8 bit là một đơn vị đếm song hướng có thể lập trình được. Cấu trúc của nó như hình dưới đây:



**Hình 2.17: Đơn vị đếm**

count: tăng hay giảm TCNT0 1

direction: lựa chọn giữa đếm lên và đếm xuống

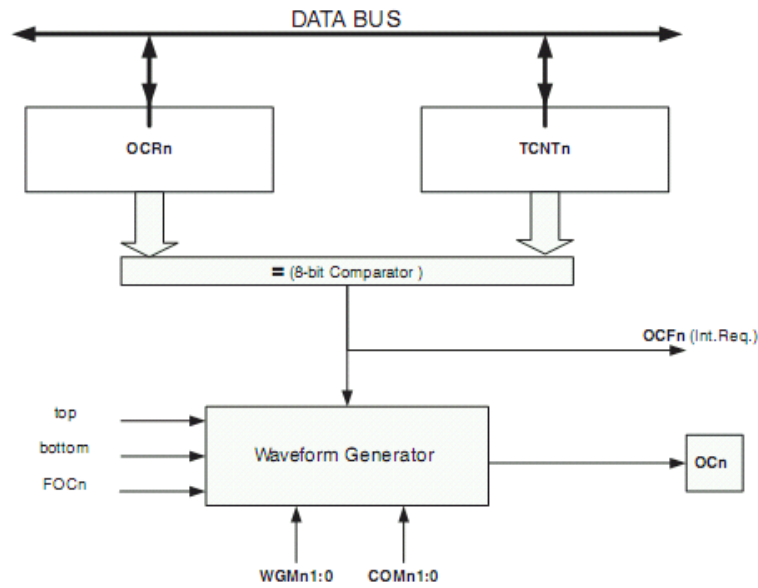
clear: xóa thanh ghi TCNT0

clk<sub>T0</sub>: xung clock của bộ định thời

TOP: báo hiệu bộ định thời đã tăng đến giá trị lớn nhất

BOTTOM: báo hiệu bộ định thời đã giảm đến giá trị nhỏ nhất (0)

### 2.2.3.3. Đơn vị so sánh ngõ ra



**Hình 2.18: Sơ đồ đơn vị so sánh ngõ ra**

Bộ so sánh 8 bit liên tục so sánh giá trị TCNT0 với giá trị trong thanh ghi so sánh ngõ ra (OCR0). Khi giá trị TCNT0 bằng với OCR0, bộ so sánh sẽ tạo một báo hiệu. Báo hiệu này sẽ đặt giá trị cờ so sánh ngõ ra (OCF0) lên 1 vào chu kỳ xung clock tiếp theo. Nếu được kích hoạt (OCIE0=1), cờ OCF0 sẽ tạo ra một ngắt so sánh ngõ ra và sẽ tự động được xóa khi ngắt được thực thi. Cờ OCF0 cũng có thể được xóa bằng phần mềm.

### 2.2.4. Mô tả các thanh ghi

#### 2.2.4.1. Thanh ghi điều khiển bộ định thời/bộ đếm TCCR0

Bit	7	6	5	4	3	2	1	0	
	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	TCCR0
Read/Write	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

\*) Bit 7-FOC0: So sánh ngõ ra bắt buộc

Bit này chỉ tích cực khi bit WGM00 chỉ định chế độ làm việc không có PWM. Khi đặt bit này lên 1, một báo hiệu so sánh bắt buộc xuất hiện tại đơn vị tạo dạng sóng.

\*) *Bit 6, 3-WGM01:0: Chế độ tạo dạng sóng*

Các bit này điều khiển đếm thứ tự của bộ đếm, nguồn cho giá trị lớn nhất của bộ đếm (TOP) và kiểu tạo dạng sóng sẽ được sử dụng.

\*) *Bit 5:4-COM01:0: Chế độ báo hiệu so sánh ngõ ra*

Các bit này điều khiển hoạt động của chân OC0. Nếu một hoặc cả hai bit COM01:0 được đặt lên 1, ngõ ra OC0 sẽ hoạt động.

\*) *Bit 2:0: CS02:0: Chọn xung đồng hồ*

Ba bit này dùng để lựa chọn nguồn xung cho bộ định thời/bộ đếm.

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$clk_{I/O}$ /(No prescaling)
0	1	0	$clk_{I/O}/8$ (From prescaler)
0	1	1	$clk_{I/O}/64$ (From prescaler)
1	0	0	$clk_{I/O}/256$ (From prescaler)
1	0	1	$clk_{I/O}/1024$ (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

#### 2.2.4.2. Thanh ghi bộ định thời/bộ đếm

Bit	7	6	5	4	3	2	1	0	
	TCNT0[7:0]								TCNT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0	0

Thanh ghi bộ định thời/bộ đếm cho phép truy cập trực tiếp (cả đọc và ghi) vào bộ đếm 8 bit.

#### 2.2.4.3. Thanh ghi so sánh ngõ ra-OCR0

Bit	7	6	5	4	3	2	1	0	
	OCR0[7:0]								OCR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0	0

Thanh ghi này chứa một giá trị 8 bit và liên tục được so sánh với giá trị của bộ đếm.

#### 2.2.4.4. Thanh ghi mặt nạ ngắt

Bit	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	TIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

*Bit 1-OCIE0: Cho phép ngắt báo hiệu so sánh*

*Bit 0-TOIE0: Cho phép ngắt tràn bộ đếm*

#### 2.2.4.5. Thanh ghi cờ ngắt bộ định thời

Bit	7	6	5	4	3	2	1	0	
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	TIFR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

*Bit 1-OCF0: Cờ so sánh ngõ ra 0*

*Bit 0-TOV0: Cờ tràn bộ đếm*

Bit TOV0 được đặt lên 1 khi bộ đếm bị tràn và được xóa bởi phần cứng khi vector ngắt tương ứng được thực hiện. Bit này cũng có thể được xóa bằng phần mềm.

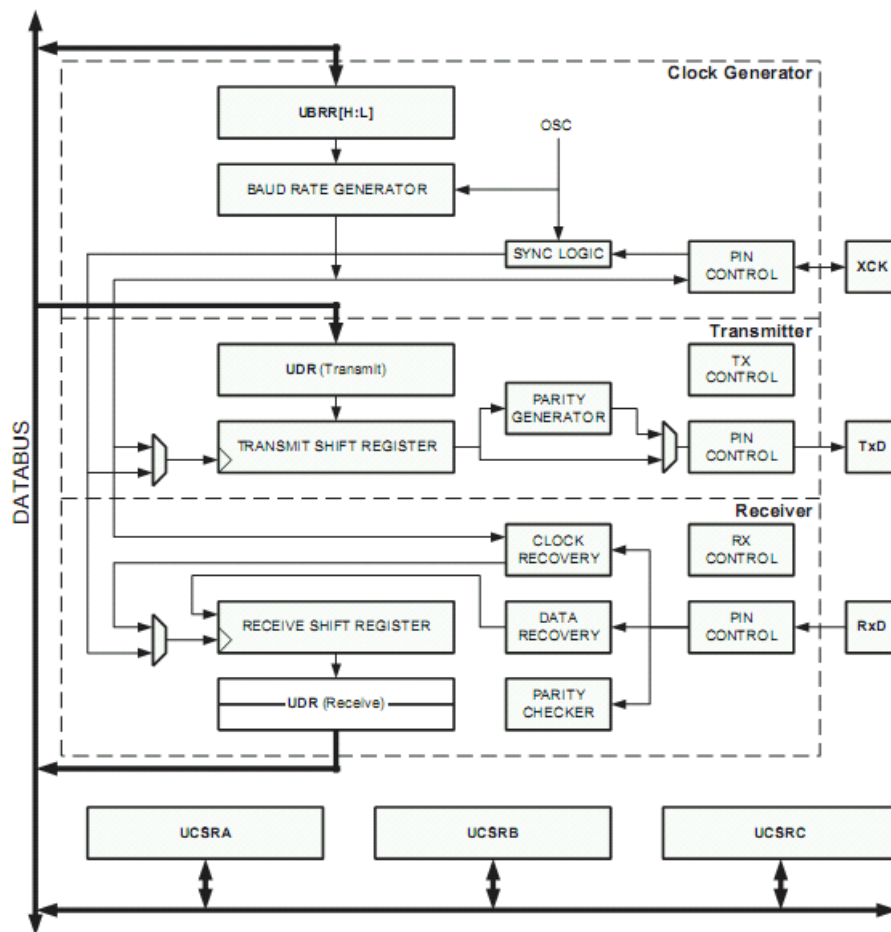
#### 2.2.5. USART

Bộ truyền nhận nối tiếp đồng bộ và bất đồng bộ là một thiết truyền thông nối tiếp có các chức năng chính như sau:

- Hoạt động song công (các thanh ghi truyền và nhận nối tiếp độc lập với nhau).
- Hoạt động đồng bộ hoặc bất đồng bộ
- Bộ tạo tốc độ baud có độ chính xác cao
- Hỗ trợ khung truyền nối tiếp với 5, 6, 7, 8, hoặc 9 bit dữ liệu và 1 hoặc 2 bit stop

- Kiểm tra chẵn lẻ
- Phát hiện tràn dữ liệu
- Phát hiện lỗi khung
- Lọc nhiễu, bao gồm phát hiện bit start lỗi và bộ lọc thông thấp số
- Ngắt khi kết thúc truyền, thanh ghi truyền hết dữ liệu và kết thúc nhận
- Chế độ truyền thông đa vi xử lý
- Chế độ truyền đồng bộ tốc độ cao

Sơ đồ khối của bộ USART như sau:

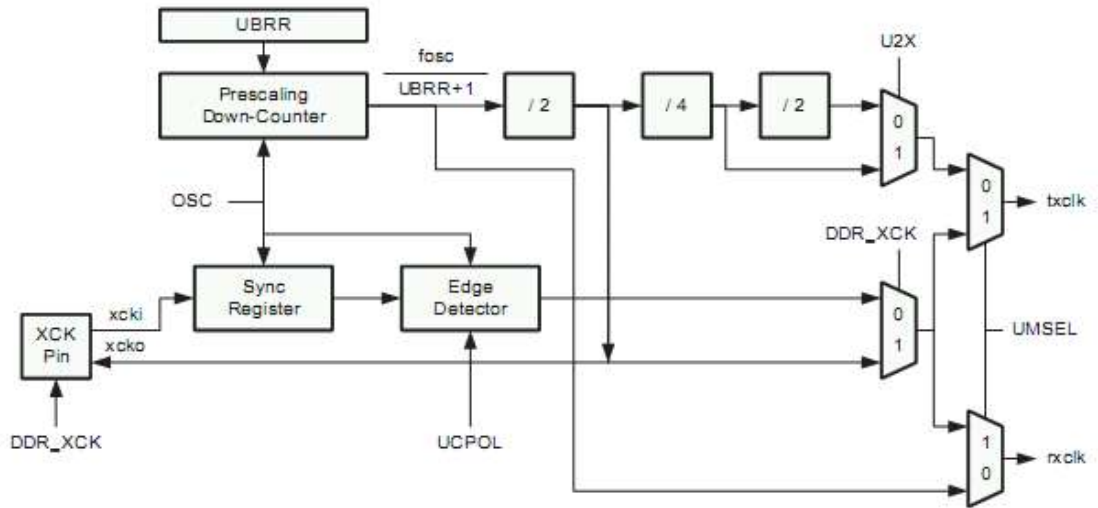


**Hình 2.19: Sơ đồ khối bộ USART**

USART bao gồm 3 phần chính: bộ tạo xung clock, bộ truyền và bộ nhận. Các thanh ghi điều khiển được sử dụng chung giữa các phần này.

### 2.2.5.1. Tạo xung clock

Bộ tạo xung clock tạo ra xung đồng hồ căn bản cho bộ truyền và bộ nhận. USART hỗ trợ 4 chế độ hoạt động xung clock: bất đồng bộ, bất đồng bộ tốc độ cao, truyền đồng bộ master và truyền đồng bộ slave. Sơ đồ khối của bộ tạo xung clock như sau:



**Hình 2.20: Đơn vị tạo xung clock**

txclk: xung đồng hồ bộ truyền

rxclk: xung đồng hồ bộ nhận

xcki: tín hiệu vào từ chân XCK, sử dụng cho hoạt động truyền đồng bộ master

xcko: tín hiệu xung clock ngõ ra tới chân XCK, sử dụng cho hoạt động truyền đồng bộ slave

fosc: tần số từ chân XTAL

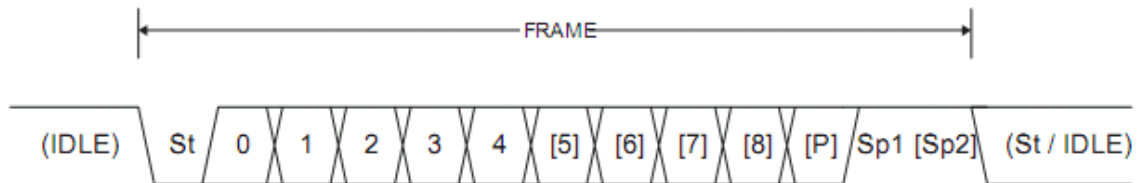
### 2.2.5.2. Định dạng khung truyền

USART chấp nhận tất cả 30 tổ hợp của các định dạng khung truyền sau đây:

- 1 bit start
- 5, 6, 7, 8, hoặc 9 bit dữ liệu
- Có hoặc không có bit chẵn lẻ

- 1 hoặc 2 bit stop

Một khung truyền bắt đầu với một bit start, theo sau đó là bit có trọng số thấp nhất (LSB) của dữ liệu (có thể lên tới 9 bit), kết thúc bằng bit có trọng số lớn nhất (MSB) và bit stop.



**Hình 2.21: Định dạng khung truyền**

St: bit start (mức thấp)

(n): bit dữ liệu (0 đến 8)

P: bit chẵn lẻ

Sp: bit stop (mức cao)

IDLE: không có dữ liệu truyền (mức cao trong suốt thời gian idle)

### 2.2.5.3. Khởi tạo USART

Quá trình khởi tạo USART bao gồm việc thiết lập tốc độ baud, thiết lập định dạng khung và kích hoạt bộ truyền và bộ nhận.

Thiết lập hoạt động truyền bất động bộ sử dụng polling (không dùng ngắt) và định dạng khung truyền là cố định. Tốc độ baud là một tham số của hàm.

### 2.2.5.4. Truyền thông dữ liệu-bộ truyền USART

Bộ truyền USART được kích hoạt bằng cách thiết lập bit TXEN trong thanh ghi UCSRB. Khi bộ truyền được kích hoạt, chân TxD hoạt động như ngõ ra của bộ truyền nối tiếp. Tốc độ baud, chế độ hoạt động và định dạng khung truyền phải được thiết lập trước khi thực hiện truyền dữ liệu.

### **a) Truyền khung 5 đến 8 bit dữ liệu**

Việc truyền dữ liệu được thiết lập bằng cách nạp dữ liệu truyền vào bộ đệm truyền. Dữ liệu trong bộ đệm sẽ được đưa vào thanh ghi dịch khi thanh ghi dịch đã sẵn sàng gửi một khung mới.

### **b) Truyền khung 9 bit dữ liệu**

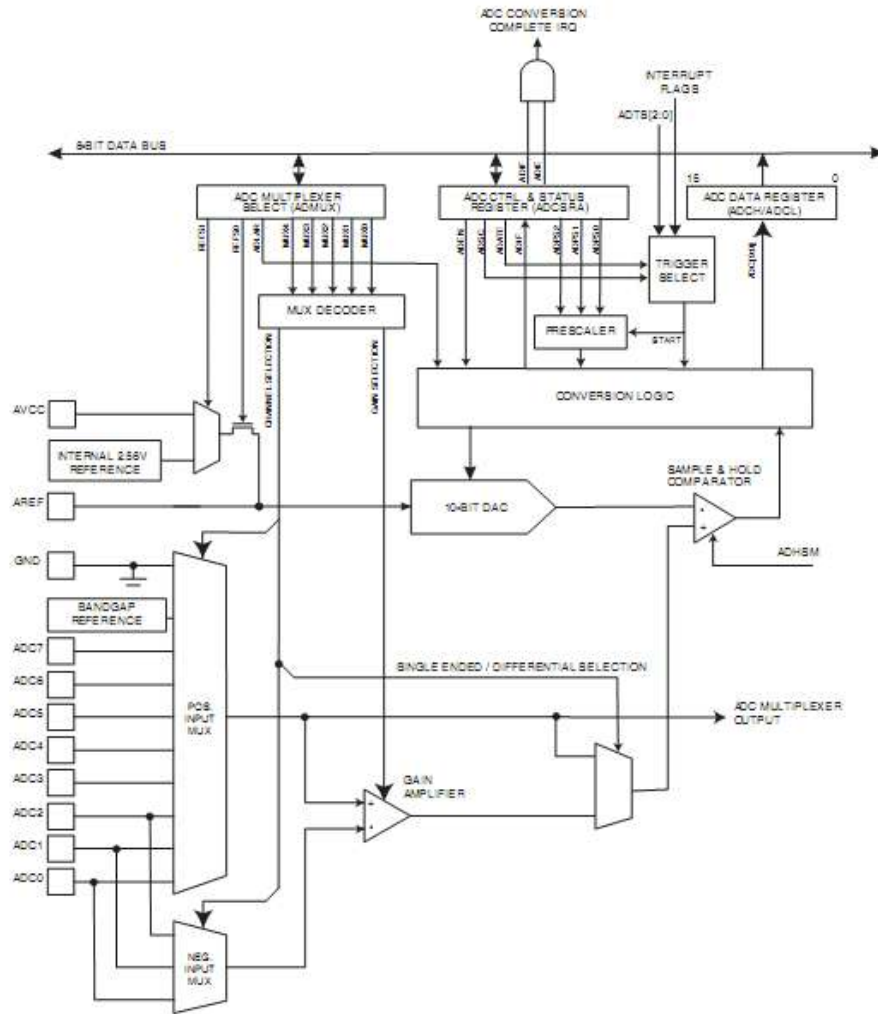
Nếu sử dụng 9 bit dữ liệu, bit thứ 9 phải được ghi vào bit TXB8 trong thanh ghi UCSRB trước khi byte còn lại được ghi vào UDR.

## **2.2.6. BỘ BIẾN ĐỔI A/D**

Vi điều khiển ATmega16 có một bộ biến đổi ADC tích hợp trong chip với các đặc điểm:

- Độ phân giải 10 bit
- Sai số tuyến tính: 0.5LSB
- Độ chính xác  $\pm 2\text{LSB}$
- Thời gian chuyển đổi: 65-260 $\mu\text{s}$
- 8 Kênh đầu vào có thể được lựa chọn
- Có hai chế độ chuyển đổi free running và single conversion
- Có nguồn báo ngắt khi hoàn thành chuyển đổi
- Loại bỏ nhiễu trong chế độ ngủ.
- Tám đầu vào của ADC là tám chân của PORTA và chúng được chọn thông qua một MUX.
- Để điều khiển hoạt động vào ra dữ liệu của ADC và CPU chúng ta có 3 thanh ghi: ADMUX là thanh ghi điều khiển lựa chọn kênh đầu vào cho ADC, ADCSRA là thanh ghi điều khiển và thanh ghi trạng thái của ADC, ADCH và ADCL là 2 thanh ghi dữ liệu.





Hình 2.22 Sơ đồ bộ biến đổi A/D

### 2.2.6.1. Thanh ghi dữ liệu ACDH và ADCL

Thanh ghi này chứa dữ liệu chuyển đổi từ tương tự sang số, được sắp xếp như hình dưới đây.

Bit	15	14	13	12	11	10	9	8	
	-	-	-	-	-	-	ADC9	ADC8	ADCH
	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

### 2.2.6.2. Nguyên tắc hoạt động và lập trình điều khiển

ADC có nhiệm vụ chuyển đổi tín hiệu điện áp tương tự thành tín hiệu số có độ phân giải 10 bit. Với giá trị nhỏ nhất của điện áp đặt ở chân AGND và giá trị cực đại của điện áp tương tự được mắc vào chân AREF. Tám kênh tương tự đầu vào được chọn lựa thông qua ADMUX và ADMUX này được điều khiển bởi thanh ghi ADMUX.

ADC này có thể hoạt động được ở hai chế độ. Đó là chuyển đổi đơn: chỉ chuyển đổi một lần khi có lệnh chuyển đổi và chế độ tự chuyển đổi (Free running mode) đây là chế độ mà ADC tự động chuyển đổi khi được hoạt động và công việc chuyển đổi có tính tuần hoàn (chỉ cần khởi động một lần).

ADC được phép hoạt động nhờ thiết lập bit ADEN. Quá trình chuyển đổi được bắt đầu bằng việc ghi vào bit ADSC mức logic 1 và trong suốt quá trình chuyển đổi bit này luôn được giữ ở mức cao. Khi quá trình chuyển đổi hoàn thành thì bit này được xóa bằng phần cứng và cờ AIDF được bật lên.

Dữ liệu sau khi chuyển đổi được đưa ra thanh ghi dữ liệu ADCL và ADCH, nhưng chú ý khi đọc dữ liệu từ hai thanh ghi này thì đọc ADCL trước rồi mới đọc ADCH. Nếu đọc ADCH trước thì dữ liệu cập nhật có thể ghi đè lên ADCL (Vi điều khiển nghĩ rằng đã đọc xong dữ liệu).

Để điều khiển vào ra dữ liệu với ADC, các bước thực hiện như sau:

Bước 1: Định nghĩa các cổng vào cho tín hiệu tương tự

Xóa bit tương ứng với chân đó trong thanh ghi DDRA. Sau đó loại bỏ điện trở treo bằng cách xóa bit tương ứng ở thanh ghi PORTA.

Bước 2: Chọn kênh tương tự vào (chọn chân vào cho ADC) thông qua thanh ghi ADMUX (có thể thay đổi trong quá trình hoạt động).

Bước 3: Thiết lập các thông số cho ADC

Tốc độ chuyển đổi thông qua xung nhịp chuyển đổi.

Chế độ chuyển đổi : đơn hoặc tự động.

Sử dụng ngắt hoặc không.

Bước 4: Bắt đầu chuyển đổi và đọc dữ liệu.

### **2.2.7. PHẦN MỀM CODEVISION AVR**

Đây là một chương trình chuyên dụng để viết chương trình ngôn ngữ C và biên dịch để nạp cho các loại vi điều khiển AVR của công ty Atmel. Phần mềm này do công ty HP Info Tech phát triển. Cửa sổ làm việc chính của chương trình hình 2.22

Chương trình được thiết kế để chạy trong môi trường hệ thống Windows 95, 98, Me, NT 4, 2000, XP và Vista 32bit. Ngôn ngữ C sử dụng trong chương trình hầu hết theo chuẩn ANSI C với một số cải tiến đặc biệt để phù hợp với cấu trúc của vi điều khiển AVR và các hệ thống nhúng.

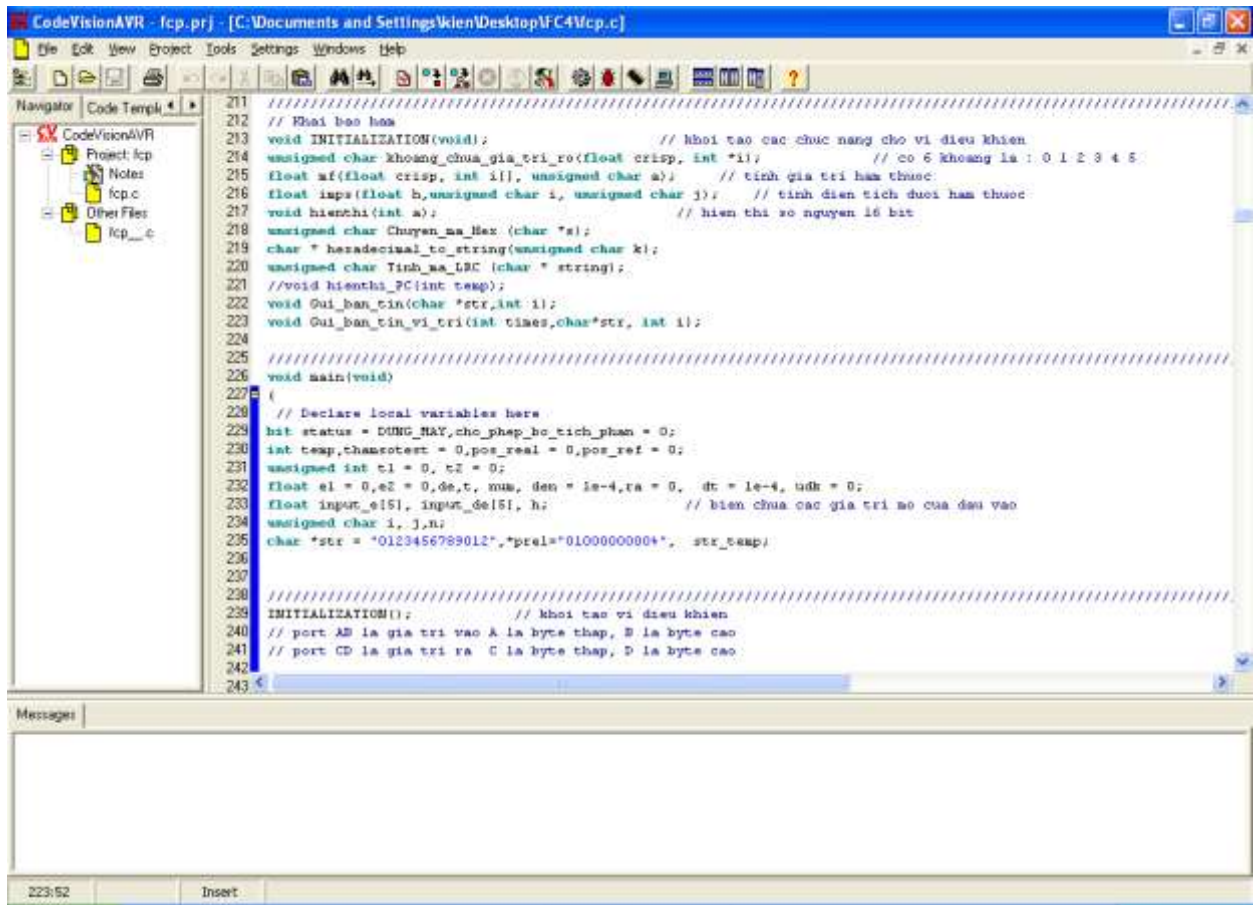
Phần mềm này là một IDE trong đó đã xây dựng sẵn nhiều công cụ lập trình cho hệ thống nhúng. Đáng chú ý nhất là công cụ cho phép lập trình trực tiếp trên hệ thống (In System Programming) khi code được viết xong. Phần mềm cho phép kết nối với nhiều thiết bị lập trình cho AVR chip. Đó là Atmel STK500, AVRISP, AVRISP MkII, AVR Dragon, JTAGICE MkII, AVRProg, Kanda Systems STK200+, STK300, Dontronics DT006, Vogel Elektronik VTEC-ISP, Futurlec JRAVR and MicroTronics ATCPU, Mega2000...

Để phân tích gỡ rối cho hệ thống dùng vi điều khiển có sử dụng truyền thông nối tiếp, trong CodeVisionAVR cũng được tích hợp công cụ gỡ rối Terminal giúp người thiết kế dễ dàng hơn rất nhiều.

Đồng thời với việc đã có các thư viện chuẩn của ngôn ngữ C, trong CodeVision cũng có các thư viện dành riêng cho:

- Màn hình hiển thị ký tự LCD
- Bus I2C của Philips
- Cảm biến nhiệt độ
- Đồng hồ thời gian thực

- Giao thức truyền thông 1 dây
- Bộ nhớ EEPROM
- Giao tiếp ngoại vi nối tiếp
- Quản lý nguồn nuôi
- Trễ
- Chuyển đổi mã Gray



**Hình 2.22: Cửa sổ làm việc chính của CodevisionAVR v1.25.7**

## CHƯƠNG 3

# XÂY DỰNG HỆ TRUYỀN ĐỘNG ĐIỆN SỬ DỤNG BỘ ĐIỀU KHIỂN VẠN NĂNG

---

Để thi công xây dựng bộ điều khiển ta phải thiết kế và thi công qua 2 bước là: 1 là thiết kế và thi công phần cứng, 2 là thiết kế thi công phần mềm

### **3.1. THIẾT KẾ PHẦN CỨNG CHO BỘ ĐIỀU KHIỂN**

#### **3.1.1. Mục đích thiết kế bộ điều khiển**

Hiện nay, trong các dây chuyền tự động, có nhiều loại đối tượng được điều khiển phục vụ cho nhiều mục đích khác nhau, mỗi đối tượng sẽ tương ứng với một dải tín hiệu, một loại tín hiệu vào/ra (tương tự, số), và được điều khiển bởi các luật khác nhau. Điều này sẽ gây ra nhiều vấn đề khi tổng hợp hệ thống cũng như khi bảo trì, sửa chữa dây chuyền sản xuất, nhất là khi phải thay thế các thiết bị.

Xuất phát từ thực tế trên, em thực hiện hệ chế tạo bộ điều khiển vạn năng với mục đích là sử dụng cho nhiều loại đối tượng trong công nghiệp. Bộ điều khiển này có thể áp dụng cho những đối tượng mà ta đã biết mô hình và cả những đối tượng mà chúng ta không có hiểu biết nhiều về mô hình đối tượng thông qua bộ điều khiển mờ (fuzzy control). Với bộ điều khiển vạn năng này em xây dựng hệ truyền động điện sử dụng bộ điều khiển vạn năng.

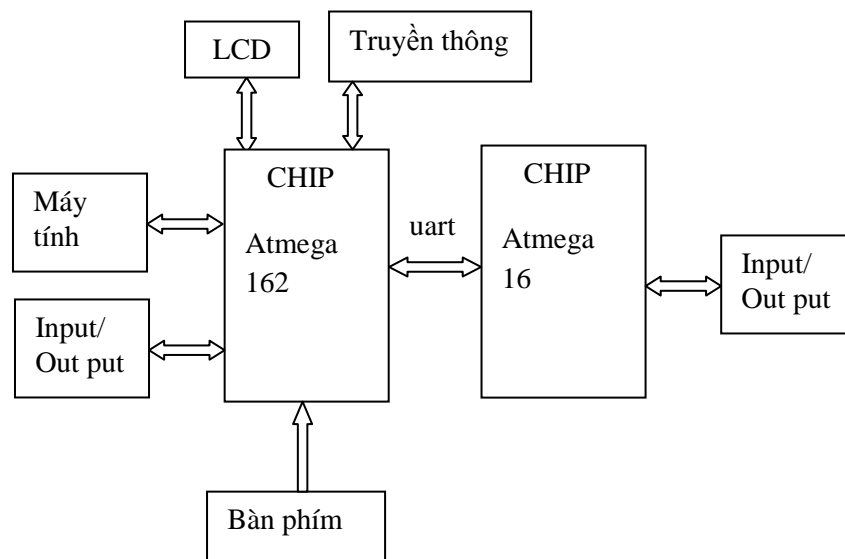
#### **3.1.2. Sơ đồ khối bộ điều khiển**

Với mục đích thiết kế trên, bộ điều khiển được thiết kế và được cài đặt luật điều khiển PID có chỉnh định mờ, để tăng tốc độ xử lý, tính toán và giảm thời gian trích mẫu trong quá trình điều khiển, ta sử dụng 2 chip AVR, trong đó, chip 1 thực hiện chỉnh định tham số cho bộ điều khiển, sau đó những tham số này được truyền đến chip 2. Lúc này chip 2 có nhiệm vụ thực hiện luật

điều khiển PID đã được cài đặt và xuất tín hiệu điều khiển đối tượng, việc truyền thông giữa hai chip được thực hiện thông qua module truyền thông USART được tích hợp sẵn trên chip.

Ngoài ra chip 2 còn có nhiệm vụ thực hiện giao tiếp với người sử dụng thông qua bàn phím và màn hình hiển thị, bên cạnh đó, những ứng dụng cần giao tiếp với máy tính để giám sát quá trình điều khiển hay phục vụ cho việc truyền thông trong mạng công nghiệp được thông qua chip 1.

Mặt khác, cả hai chip đều được thiết kế các đầu vào/ra tương tự/xung... phục vụ cho quá trình điều khiển, mỗi chip được thiết kế 8 đầu vào/ra giúp người dùng có nhiều lựa chọn tùy mục đích điều khiển.



**Hình 3.1: Sơ đồ khối của bộ điều khiển**

### 3.2. SƠ ĐỒ NGUYÊN LÝ CỦA BỘ ĐIỀU KHIỂN

#### **Thuyết minh sơ đồ:**

Do yêu cầu của đề tài, cũng như yêu cầu bài toán đặt ra, em sử dụng 2 chip AVR (Atmega 16, Atmega 162) được kết nối với nhau thông qua giao

diện truyền thông UART, việc phân bổ nhiệm vụ của 2 chip được thực hiện như sau:

Chip 1 (Atmega 162) được sử dụng để chỉnh định mờ tham số của bộ điều khiển, đồng thời chip 1 cũng được thiết kế để có thể giao tiếp với các thiết bị khác thông qua một số giao diện chuẩn như RS232.

Ngoài ra chip 1 (Atmega162) làm nhiệm vụ thực hiện giao tiếp giữa bộ điều khiển với người sử dụng thông qua bàn phím, màn hình hiển thị. Nhiệm vụ chính của chip2 là nhận tham số đã chỉnh định từ chip 1 và thực hiện luật điều khiển PID để cho ra tín hiệu điều khiển tới đối tượng

Hai chân P14, P15 của Atmega16 và P10, P11 của Atmega162 thực hiện việc truyền và nhận dữ liệu trong quá trình điều khiển.

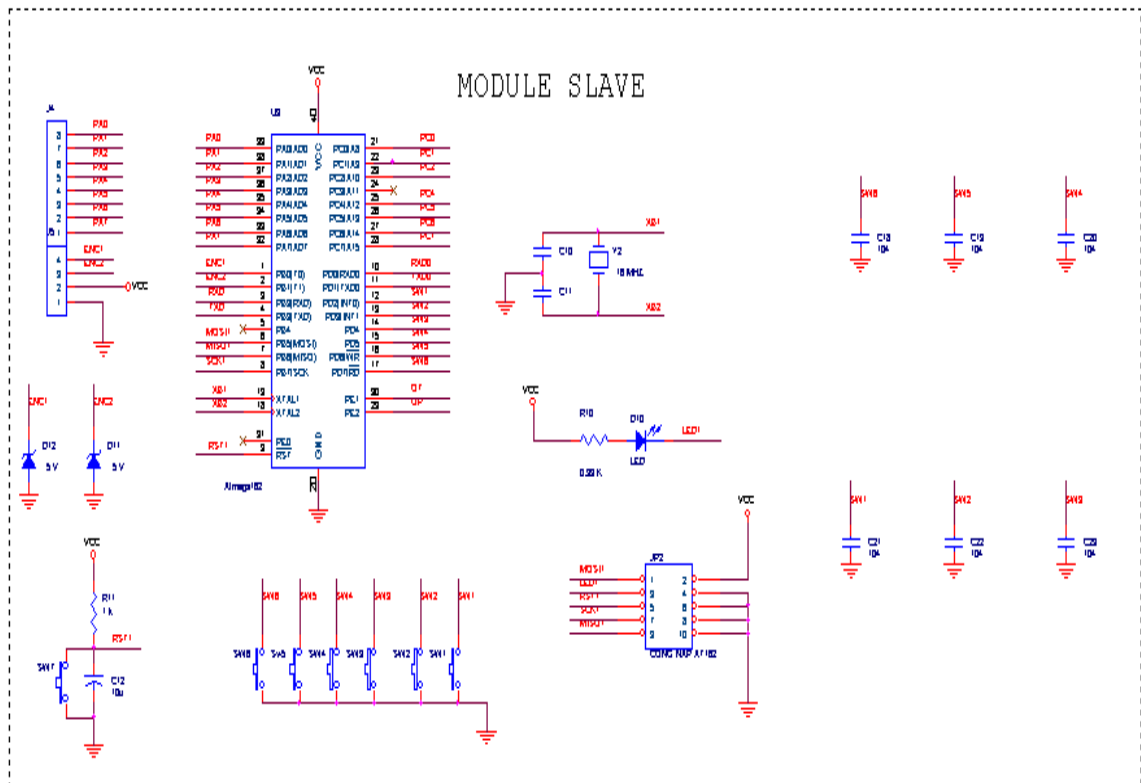
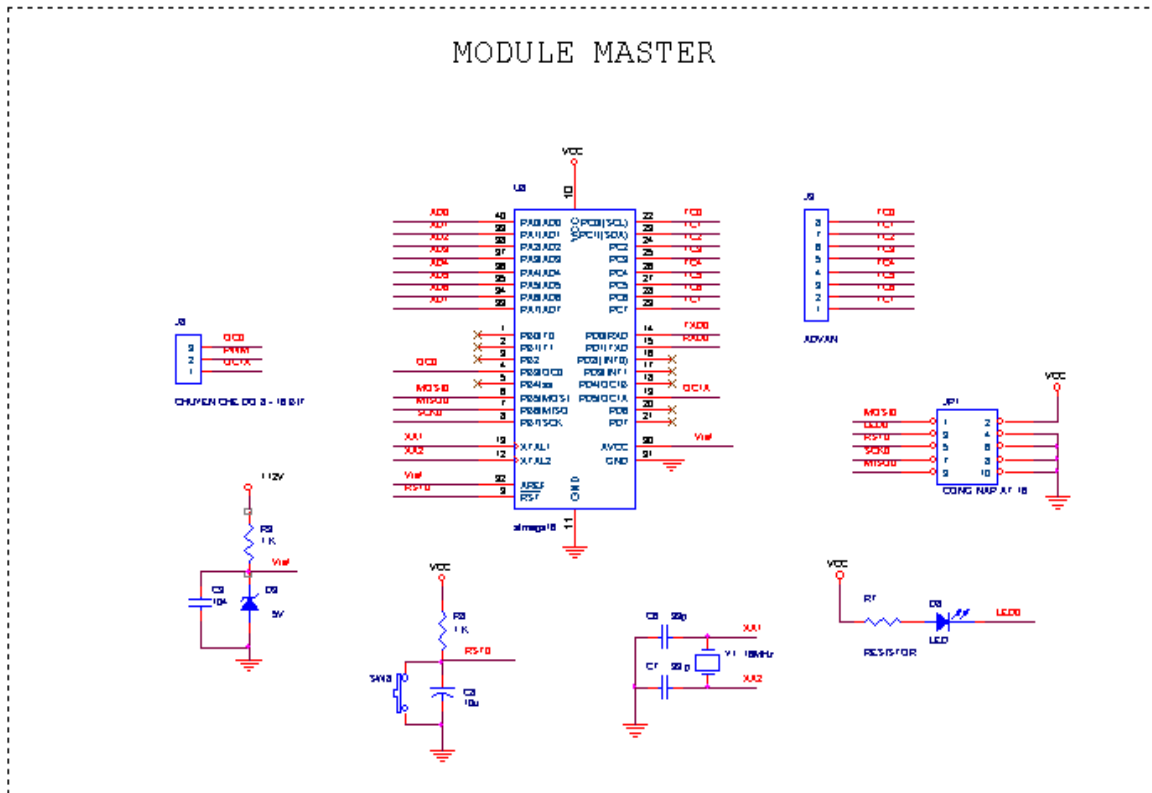
Trong sơ đồ nguyên lý trên, đối với chip 1 màn hình hiển thị LCD được kết nối với chip qua cổng PC (PC0 ÷ PC7). Trong quá trình điều khiển chân PB3 được dùng cho truyền thông RS232 thông qua giắc DB9\_Com port

Chương trình điều khiển được nạp xuống từng chip thông qua giắc JP1 và JP2, các giắc này đều được nối với chip thông qua chân PB4 ÷ PB7.

Chip được cấp nguồn thông qua 2 chân 10 và 40 của mỗi chip.

Trong sơ đồ trên bộ điều khiển được tích hợp mạch biến đổi điện áp từ 12V DC sang 5V DC để cấp nguồn cho chip và các thiết bị khác.

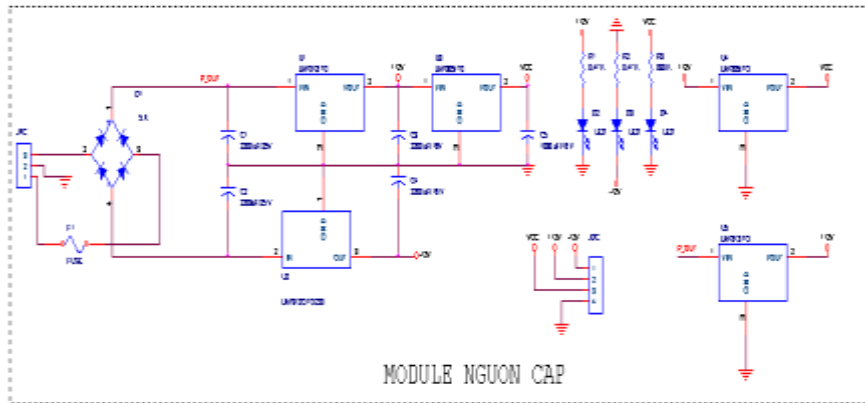
### 3.2.1. Mạch điều khiển trung tâm



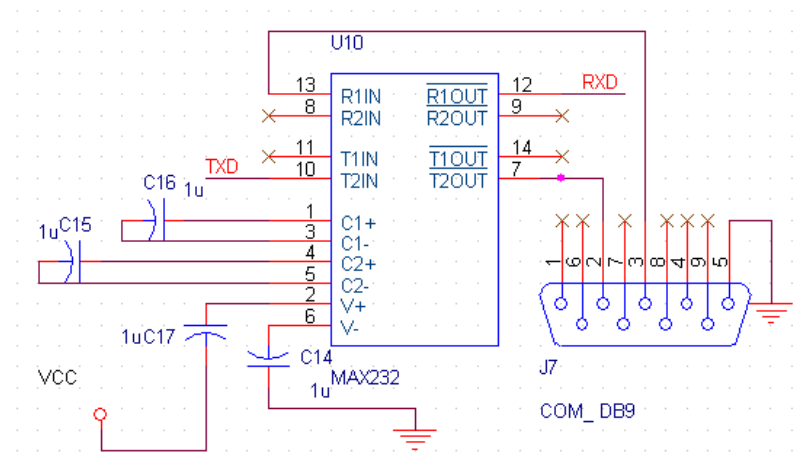
**Hình 3.2: Khối xử lý trung tâm**



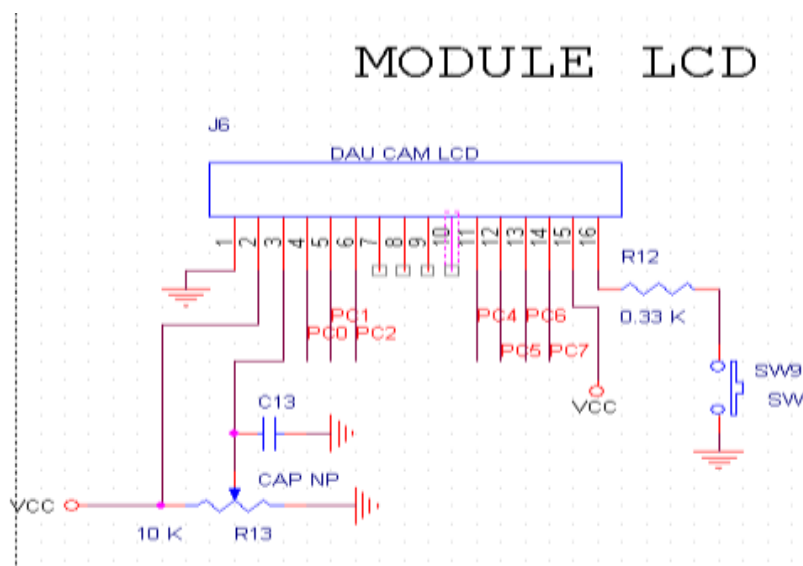
### 3.2.2. Nguồn điều khiển



### 3.2.3. Khối truyền thông

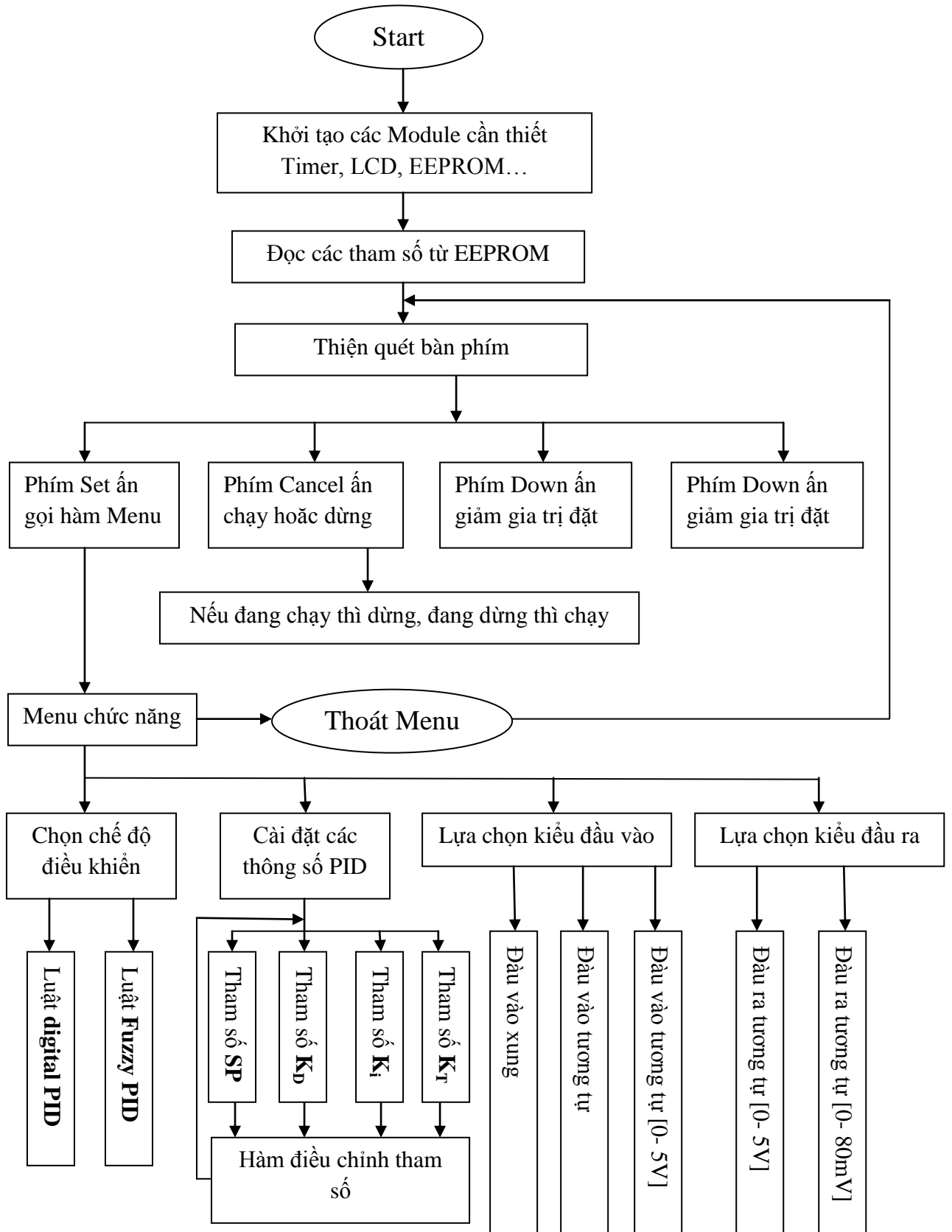


### 3.2.4. Khối hiển thị



### 3.3. THIẾT KẾ PHẦN MỀM ĐIỀU KHIỂN

#### 3.3.1. Thuật toán điều khiển



### 3.3.2. Kết quả



**Hình3.3: Mô hình mạch điều khiển**



**Hình 3.4: Mô hình điều khiển động cơ DC**

## KẾT LUẬN

Sau 12 tuần nghiên cứu tìm tòi làm đồ án tốt nghiệp, dưới sự phân công của các thầy cô trong bộ môn Điện tự động khoa điện – điện tử, được sự chỉ bảo hết sức nhiệt tình của thầy GS.TSKH Thân Ngọc Hoàn cùng với sự cố gắng, nỗ lực hết sức của bản thân em, bản đồ án tốt nghiệp đã hoàn thành.

### **Đồ án đã đạt được kết quả như sau:**

- Nghiên cứu ứng dụng lý thuyết điều khiển mới hiện đại đặc biệt là điều khiển mờ để tổng hợp lên bộ điều khiển.
- Đã thiết kế thành công hệ truyền động điện bộ điều khiển vạn năng ứng dụng điều khiển động cơ 1 chiều có khâu hồi tiếp tốc độ.
- Viết chương trình cho vi điều khiển AVR sử dụng ngôn ngữ C, để làm khâu điều khiển trung tâm. Ngoài ra còn lập giao diện trên máy tính bằng Visual basic 6.0, cho phép điều khiển tốc độ động cơ bằng tay hoặc bằng máy tính.

**Tuy nhiên hạn chế của đồ án là:** mô hình chưa được gọn gàng, phân tích cơ lý thuyết chưa được đầy đủ sâu sắc. Nhưng đó cũng là thành công của em trong suốt thời gian qua.

Em xin chân thành cảm ơn!

Hải Phòng, ngày 1 tháng 7 năm 2010

**Sinh viên**

**Nguyễn Thanh Tùng**

## TÀI LIỆU THAM KHẢO

1. PGS.TSKH Thân Ngọc Hoàn (2000), Máy Điện, NXB Xây Dựng.
2. GS.TSKH Thân Ngọc Hoàn, TS Nguyễn Tiến Ban (2007), Điều khiển tự động các hệ thống TRUYỀN ĐỘNG ĐIỆN, NXB Khoa học kỹ thuật.
3. Bùi Công Cường – Nguyễn Doãn Phước (2001), Hệ mờ mạng nơtron và ứng dụng, NXB Khoa học và Kỹ thuật.
4. Phan Xuân Minh – Nguyễn Doãn Phước (2001), Lý thuyết điều khiển mờ, NXB Khoa học và Kỹ thuật.
5. Phạm Công Ngô (1996), Lý thuyết điều khiển tự động, NXB Khoa học và Kỹ thuật.
6. Tống Văn On, Hoàng Đức Hải (1999), Họ vi điều khiển 8051, NXB Lao động – Xã hội.
7. Ngô Diên Tập (2001), Đo lường và điều khiển bằng máy tính, NXB Khoa học và Kỹ thuật.
8. Atmega16 family datasheet.