

ĐẠI HỌC THÁI NGUYÊN
TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

BÀI GIẢNG
THIẾT KẾ HỆ ĐIỀU KHIỂN SỐ

(Dùng cho sinh viên các trường kỹ thuật)

NĂM 2020

MỤC LỤC

CHƯƠNG 1. LÝ THUYẾT ĐIỀU KHIỂN SỐ.....	1
1.1. Khái niệm	1
1.2. Cấu trúc của hệ thống điều khiển số.....	4
1.3 Tín hiệu và xử lý tín hiệu số.....	6
1.3.1. Phân loại tín hiệu	6
1.3.2. Xử lý tín hiệu.....	7
1.4 Hàm truyền hệ điều khiển số.....	11
1.4.1. Định nghĩa	11
1.4.2 Biến đổi Z.....	11
1.4.3. Xác định hàm truyền hệ điều khiển số.	14
1.5 Xét ổn định hệ điều khiển số.....	15
1.5.1. Khái niệm và thông số ảnh hưởng.....	15
1.5.2. Phương pháp đổi biến (Tiêu chuẩn Routh-Hurwitz mở rộng)	16
1.5.3. Quỹ đạo nghiệm số (Root Locus) của hệ điều khiển số.....	17
1.6. Đánh giá chất lượng hệ điều khiển số	21
1.6.1. Các chỉ tiêu chất lượng.....	21
1.6.2. Đánh giá chất lượng xác lập.....	22
1.6.3. Đánh giá chất lượng quá độ.....	23
1.7 Tổng hợp và hiệu chỉnh hệ điều khiển số.....	24
CHƯƠNG 2. THIẾT KẾ HỆ ĐIỀU KHIỂN SỐ.....	25
2.1. Cơ sở thiết kế hệ điều khiển số.....	25
2.2. Phương pháp thiết kế gián tiếp.....	27
2.2.1. Thiết kế bộ điều chỉnh từ thiết bị tương tự.....	27
2.2.2. Sử dụng bộ điều chỉnh tổng quát PID	28

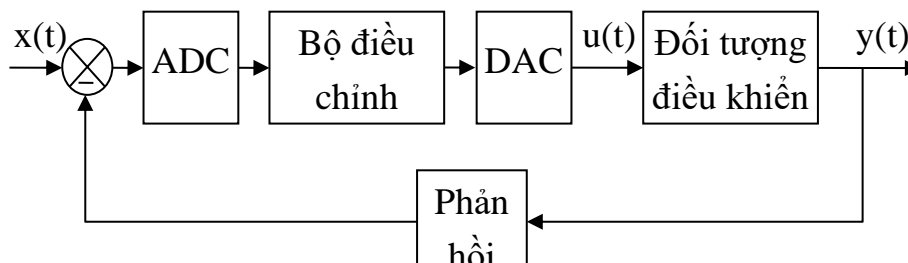
2.3. Phương pháp thiết kế dựa vào quỹ đạo nghiệm số.....	29
2.3.1. Thiết kế bộ điều khiển sớm pha	29
2.3.2. Thiết kế bộ điều khiển trễ pha	32
2.4 Phương pháp thiết kế dựa vào đáp ứng tần số.....	34
2.5 Phương pháp thiết kế bằng thực nghiệm.....	41
CHƯƠNG 3: THIẾT KẾ ĐIỀU KHIỂN SỐ CHO HỆ THỰC	42
3.1 HỆ THỐNG ĐIỀU KHIỂN SỐ SỬ DỤNG VI XỬ LÝ/VI ĐIỀU KHIỂN.....	42
3.2. MỘT SỐ VẤN ĐỀ KỸ THUẬT KHI THỰC HIỆN HỆ ĐIỀU KHIỂN SỐ VỚI VI XỬ LÝ/VI ĐIỀU KHIỂN	42
3.3. QUY TRÌNH THỰC HIỆN HỆ THỐNG ĐIỀU KHIỂN SỐ.....	42
3.4. HỆ TRUYỀN ĐỘNG ĐIỆN	43
3.4.1 Các thiết bị cơ bản trong hệ điều khiển số truyền động điện	43
3.4.2. Tổng hợp bộ điều khiển trong hệ truyền động điện	57
3.5 Hệ điều khiển quá trình	68
3.5.1 Các thiết bị cơ bản trong hệ thống điều khiển thiết bị gia nhiệt.....	68
3.5.2. Tổng hợp bộ điều khiển cho hệ điều khiển thiết bị gia nhiệt.....	73
3.5.3 Thiết kế và lập trình điều khiển.....	79
TÀI LIỆU THAM KHẢO.....	80

CHƯƠNG 1. LÝ THUYẾT ĐIỀU KHIỂN SỐ

1.1. Khái niệm

Hệ điều khiển số là hệ điều khiển mà trong hệ chỉ cần có một tín hiệu là tín hiệu xung, số hoặc chỉ cần một thiết bị làm việc theo nguyên tắc số.

Sơ đồ khối của một hệ điều khiển số như hình vẽ sau:



Hình 1.1: Sơ đồ khối của hệ điều khiển số

Trong thực tế không tồn tại hệ điều khiển số mà trong đó bao gồm toàn bộ các tín hiệu là tín hiệu số hoặc bao gồm toàn bộ thiết bị số. Nguyên nhân là do các đối tượng điều khiển trong thực tế là các thiết bị với đại lượng điều khiển là các đại lượng vật lý biến đổi liên tục theo thời gian, do vậy để điều khiển được các đại lượng này thì tín hiệu điều khiển phải là tín hiệu tương tự, mang năng lượng.

Hạn chế của điều khiển tương tự và ưu điểm của điều khiển số:

- Nhược điểm quan trọng của kỹ thuật tương tự liên quan đến sự trôi thông số do các nguyên nhân có nguồn gốc khác nhau (do nhiệt, hóa-lý, cơ học,...). Hiện tượng này làm thay đổi thông số của các linh kiện điện tử, điện dung của tụ điện, điện trở của các chiết áp. Để khử sự trôi thông số thường sử dụng các mạch bù làm tăng độ phức tạp của mạch và giá thành. Các linh kiện số chỉ có hai mức năng lượng cao và thấp (0 và 1) nên không bị ảnh hưởng bởi sự trôi thông số.

- Thiết bị tương tự thường nhạy với nhiễu. Nhiễu có thể phát sinh do bản thân linh kiện (nhiệt, sự già hóa,...) hoặc nhiễu từ các yếu tố bên ngoài do ảnh hưởng của môi trường. Các cấu trúc số có thể được bảo vệ bằng các kỹ thuật áp dụng cho kỹ thuật tương tự (màn chắn, bọc kim,...), ngoài ra còn dùng các kỹ thuật lọc số.

- Việc truyền dẫn tín hiệu tương tự cũng gặp khó khăn do sự suy giảm tín hiệu. Việc truyền dẫn tín hiệu số ít bị ảnh hưởng bởi sự suy giảm.

- Các linh kiện tương tự có tính chất khác nhau về thông số khi sản xuất hàng loạt làm cho các linh kiện tương tự kém ổn định.

- Việc thực hiện một số chức năng như nhớ hoặc trễ bằng kỹ thuật tương tự gặp nhiều trở ngại. Việc thực hiện các chức năng này bằng kỹ thuật số khá đơn giản.

- Do hệ điều khiển số luôn sử dụng thiết bị tính toán có khả năng tính toán mạnh như vi xử lý hoặc máy tính, cho phép gia công các quy luật điều khiển phức tạp, do vậy có thể thực hiện điều khiển bám sát đối tượng thực. Trong trường hợp đối tượng thực

biến động, ta có thể chủ động thay đổi cả quy luật điều khiển, do vậy điều khiển số có độ chính xác cao. Nói cách khác đối với điều khiển số với thiết bị tính toán có khả năng tính toán mạnh, ta có thể áp dụng nhiều thuật toán điều khiển hiện đại mà trước đây không thể thực hiện được.

- Việc thực hiện mạch và hiệu chỉnh mạch tương tự gặp nhiều khó khăn, phức tạp nên tốn nhiều thời gian và công sức.

Ưu điểm của thiết bị tương tự và nhược điểm của thiết bị số:

- Tác động nhanh: Các thiết bị tương tự tác động gần như tức thời trong khi các thiết bị số tác động chậm do cần thời gian biến đổi và xử lý. Thiết bị số phải thực hiện theo các bước:

- Lấy mẫu, ghi dữ liệu.
- Tính toán, xử lý theo chương trình các dữ liệu ghi được ở trên

Muốn ra kết quả thì hệ điều khiển số phải trải qua hai công đoạn trên, do vậy hệ điều khiển số có độ nhạy kém hơn hệ điều khiển tương tự (thời gian quá độ dài hơn). Để khắc phục nhược điểm này ta phải tìm cách rút ngắn thời gian ghi dữ liệu bằng cách sử dụng truyền thông song song mà không sử dụng truyền thông nối tiếp đồng thời loại bỏ những dữ liệu không cần thiết, chỉ xử lý những dữ liệu cần thiết. Ngoài ra cần phải rút ngắn thời gian tính toán bằng cách sử dụng vi xử lý có tốc độ tính toán cao đồng thời phải tối ưu hóa chương trình điều khiển.

- Tác động liên tục: cho phép sử dụng để khống chế các thông số (đại lượng) (như dòng điện, điện áp) có sự biến thiên rất nhanh.

- Đơn giản về thiết kế của điều khiển tương tự: Điều khiển tương tự trở nên nặng nề đối với các điều khiển phức tạp, tuy nhiên ở mức độ cơ cấu hợp lý thì điều khiển tương tự lại rất đơn giản về phương diện cấu trúc.

Các ưu điểm có tính chất quyết định của điều khiển số:

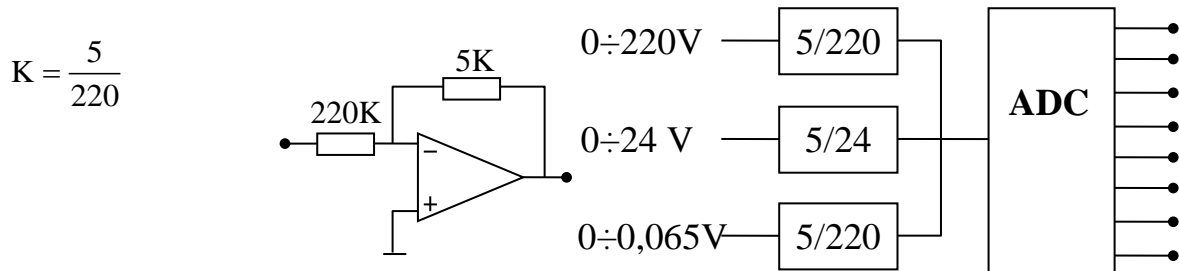
Điều khiển số cho phép tăng tỷ số giữa tính năng và giá thành. Các ưu điểm của kỹ thuật số thể hiện ở hai mặt:

- Điều khiển thông minh: các chương trình phần mềm cho phép tối ưu hóa điều khiển và thay đổi tính năng mong muốn.

- Đơn giản hóa thiết bị, tiêu chuẩn hóa và tích hợp hóa: Vì chức năng điều khiển chủ yếu được thực hiện bằng phần mềm nên với cùng một thiết bị phần cứng (một bộ vi xử lý và các giao diện) được sử dụng cho mọi ứng dụng. Điều này dẫn đến giảm các chi tiết dự phòng, do đó làm giảm giá thành.

Một số lưu ý đối với hệ điều khiển số:

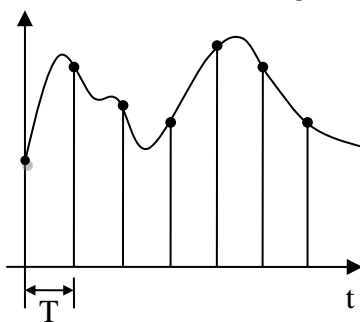
- Thiết bị số được chế tạo chuẩn hóa mà các đại lượng vật lý thực tế đo được lại biến đổi trong dải rộng. Do vậy, bắt buộc phải sử dụng thiết bị khuếch đại chuẩn hóa A/A. Đây là thiết bị tương tự nên nhiễu rất dễ xâm nhập.



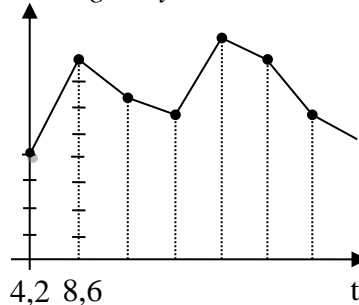
- Trong hệ điều khiển số, tín hiệu đưa vào vi xử lý phải là tín hiệu số mà đại lượng vật lý thực tế cần điều khiển là tín hiệu tương tự. Vì vậy, bắt buộc phải sử dụng các bộ chuyển đổi ADC, DAC gây ra sai số về quy luật tín hiệu và giá trị thông tin.

VD: giả thiết mức mã hóa là $\Delta = 1\text{ V}$

Rời rạc hóa theo thời gian



Tín hiệu sau khi khôi phục bằng chuyển đổi DAC



Các mức 4,2; 8,6 khi chuyển về cơ số 2 với 4 bit:

$$0100 \rightarrow 4 \times 1 = 4\text{ V}$$

$$1000 \rightarrow 8 \times 1 = 8\text{ V}$$

Như vậy xuất hiện sai số khi so sánh với 4,2 và 8,6

Muốn giảm sai số quy luật của tín hiệu thì phải giảm chu kỳ lấy mẫu T. Muốn giảm sai số về giá trị thông tin cần phải giảm mức độ mã hóa thông tin. Điều này dẫn đến số lượng mã hóa lớn nên phải tăng số bit.

Để khắc phục, phải dùng chuyển đổi ADC, DAC có số bit mã hóa cao. Khi đó phải sử dụng vi xử lý có tốc độ tính toán cao.

Hệ điều khiển số có đặc điểm phần cứng không quy định quy luật điều khiển mà được quy định bởi phần mềm. Do vậy để sử dụng được thì người vận hành và sửa chữa cần phải nắm vững cả phần cứng và phần mềm.

Do quy luật điều khiển được quy định bởi phần mềm nên có thể dễ dàng thay đổi, vì vậy gây khó khăn cho người quản lý. Do vậy, trong thực tế các chương trình phần mềm điều khiển không để trên máy tính mà được ghi trên các bộ nhớ cứng như: ROM, EPROM, EEPROM.

Xu hướng sử dụng hệ điều khiển kết hợp hệ điều khiển tương tự và hệ điều khiển số: do các đặc điểm đã nêu ở trên, trong lĩnh vực điều khiển truyền động điện người ta thường sử dụng kết hợp điều khiển số và điều khiển tương tự.

- Các chức năng đòi hỏi điều khiển tác động nhanh được thực hiện bằng điều khiển tương tự.
- Các chức năng ở mức cao, điều khiển thông minh nhưng thực hiện chậm hơn sẽ được thực hiện bằng kỹ thuật số.

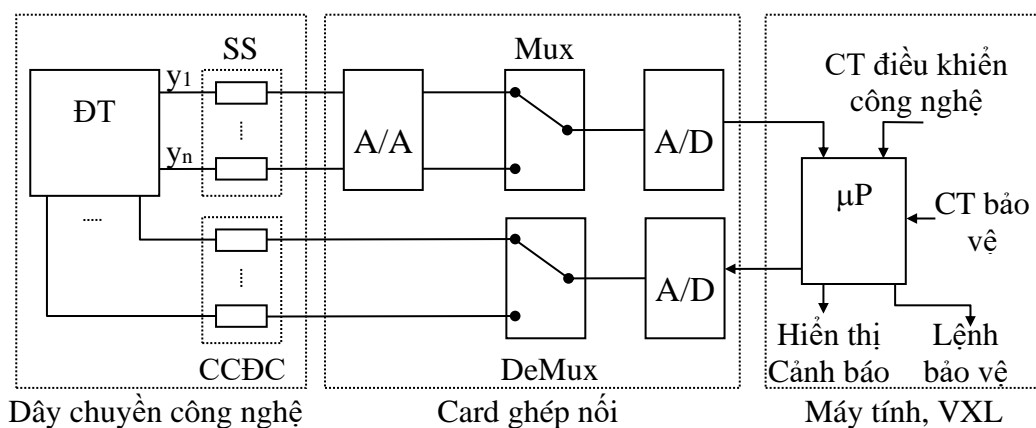
Phân loại hệ điều khiển số: hệ điều khiển số thường được phân loại theo khả năng xử lý tín hiệu

+ Hệ điều khiển số đơn kênh: là hệ chỉ điều khiển một đại lượng vật lý. Hệ này có ưu điểm là đơn giản dễ tính toán nhưng nhược điểm là việc thực hiện dây chuyền tự động rất khó khăn, cần sử dụng nhiều người vận hành.

+ Hệ điều khiển số đa kênh: là hệ sử dụng một hệ điều khiển số để điều khiển nhiều đại lượng vật lý khác nhau, các đại lượng vật lý này hoàn toàn độc lập với nhau. Hệ điều khiển số đa kênh phải sử dụng các mạch dồn kênh Mux và phân kênh Demux do vậy hệ có đặc điểm là quá trình điều khiển bị sai số do thời gian điều khiển lặp lại dài khi số đại lượng vật lý nhiều.

+ Hệ điều khiển số nhiều chiều: là hệ điều khiển nhiều đại lượng vật lý khác nhau và giữa các đại lượng vật lý có mối liên hệ với nhau. Do vậy phải thực hiện điều khiển đồng thời và thông tin, dữ liệu của các đại lượng vật lý phải được trao đổi với nhau. Vì vậy, đối với hệ điều khiển số nhiều chiều, cần phải sử dụng thuật toán ma trận. Đây là hệ điều khiển hiện đại có độ chính xác cao và được sử dụng rộng rãi trong thực tế.

1.2. Cấu trúc của hệ thống điều khiển số



Hình 1.2: Cấu trúc của hệ điều khiển số

Trong đó:

+ ĐT: đối tượng cần điều khiển, có nhiệm vụ tạo ra các đại lượng vật lý mà công nghệ yêu cầu, chẳng hạn muốn có nhiệt độ ta phải có lò gia nhiệt, muốn có tốc độ ta phải sử dụng động cơ,... Đây là các thiết bị tương tự.

+ SS: khối thiết bị đo, có nhiệm vụ đo các đại lượng vật lý tương tự thực tế (không điện) và biến đổi về đại lượng điện (thường là điện áp hoặc dòng điện). Đây là các thiết bị tương tự.

+ A/A: khối khuếch đại chuẩn hóa, có nhiệm vụ biến tín hiệu tương tự chưa chuẩn ở đầu vào thành tín hiệu tương tự chuẩn hóa ở đầu ra. Đây là thiết bị tương tự. Hiện nay thường dùng các chuẩn:

Chuẩn áp: $0 \div 5 \text{ V}$ ($0 \div \pm 5 \text{ V}$)

$0 \div 10 \text{ V}$ ($0 \div \pm 10 \text{ V}$)

$0 \div 15 \text{ V}$ ($0 \div \pm 15 \text{ V}$)

Chuẩn dòng: $0 \div 20 \text{ mA}$ ($0 \div \pm 20 \text{ mA}$)

$4 \div 20 \text{ mA}$ ($4 \div \pm 20 \text{ mA}$)

+ Mux, Demux: thiết bị dồn kênh, tách kênh. Đây là thiết bị số, có nhiệm vụ chuyển thông tin song song thành nối tiếp và ngược lại. Quá trình chuyển đổi được quét đồng bộ và được điều khiển bởi vi xử lý thông qua phần mềm điều khiển công nghệ.

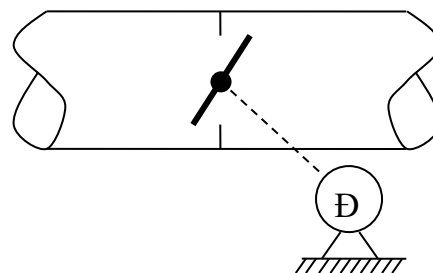
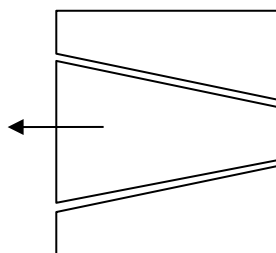
+ A/D, D/A: khối chuyển đổi tín hiệu tương tự - số, số - tương tự. Đây là thiết bị số.

+ μP : khối vi xử lý tín hiệu, có nhiệm vụ ghi chương trình điều khiển, chương trình bảo vệ, đọc chương trình điều khiển, đọc giá trị tín hiệu số của đại lượng vật lý tương ứng trong chương trình; so sánh, quyết định tín hiệu điều khiển. Muốn vậy, vi xử lý phải điều khiển trạng thái tổng trở của các cửa Mux Demux.

Ngoài ra vi xử lý đọc chương trình bảo vệ, đọc giá trị đại lượng vật lý cần bảo vệ, so sánh với ngưỡng bảo vệ trong chương trình. Khi giá trị của đại lượng vật lý vượt quá ngưỡng, μP phát lệnh bảo vệ dừng hệ thống đồng thời phát tín hiệu thông báo cho người vận hành, sửa chữa. Mặt khác, vi xử lý kiểm soát hệ thống và thông báo chế độ làm việc trong hệ thống thông qua mã lệnh, mã lỗi.

+ CCDC: cơ cấu điều chỉnh có nhiệm vụ nhận tín hiệu điều khiển và chấp hành quy luật để điều tiết đối tượng sao cho đại lượng vật lý đầu ra biến đổi theo chương trình công nghệ. Đây là thiết bị tương tự, ví dụ: các van điện tử, các van tiết lưu, động cơ servo, các bộ biến đổi, biến trở,...

Ví dụ: Để điều chỉnh lưu lượng của lưu chất, ta có thể sử dụng van tiết lưu hoặc sử dụng động cơ servo để thay đổi góc mở của van.



Lưu ý:

- Trong thực tế các khối có thể được chế tạo hợp bộ (trọn bộ), chẳng hạn khối CCĐC được chế tạo hợp bộ với đối tượng ĐT, khối SS được chế tạo hợp bộ với khối chuẩn hóa tín hiệu.

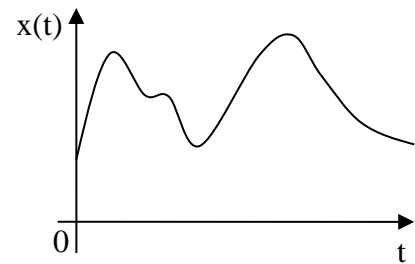
- Nếu hệ điều khiển số chỉ điều khiển một đại lượng vật lý (điều khiển số đơn kênh) thì không có các khối Mux và Demux.

1.3 Tín hiệu và xử lý tín hiệu số

1.3.1. Phân loại tín hiệu

1.3.1.1 Tín hiệu tương tự (tín hiệu liên tục)

Tín hiệu tương tự là tín hiệu mà thông tin có ở bất kỳ thời điểm nào, giá trị thông tin được mang theo qua thông số đặc trưng của đại lượng vật lý mang tin như: biên độ, tần số, góc pha đầu (thực tế thường sử dụng biên độ). Để tính toán, thường sử dụng các hàm toán học liên tục.

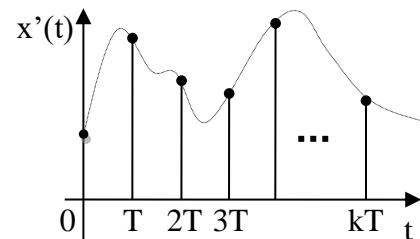


Hình 1.3: Tín hiệu tương tự

Giá trị thông tin theo hệ thập phân

1.3.1.2. Tín hiệu lấy mẫu (tín hiệu rời rạc)

Tín hiệu lấy mẫu là tín hiệu mà thông tin chỉ có ở các thời điểm cố định, ngoài các điểm lấy mẫu không có thông tin. Giá trị thông tin mang theo biên độ. Giá trị thông tin theo hệ thập phân.



Hình 1.4: Tín hiệu lấy mẫu

1.3.1.3. Tín hiệu số

Tín hiệu số $x^*(t)$ là tín hiệu mà thông tin chỉ có tại các thời điểm cố định, ngoài các thời điểm đó ra thì không có thông tin. Giá trị thông tin được mang theo mã số, được tính toán theo hệ nhị phân.

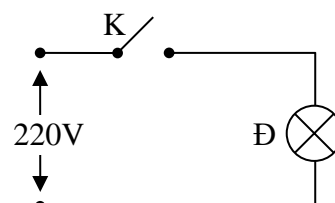
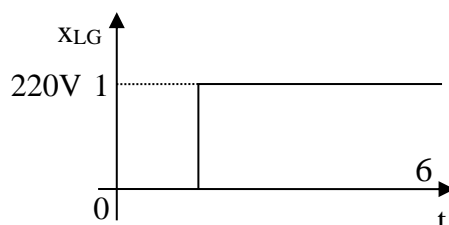
Ví dụ: rời rạc hóa tín hiệu theo mức sau đó chuyển về cơ số 2

$$x^*(t): t=0 \Rightarrow 00101101$$

$$t=1T \Rightarrow 01011100$$

1.3.1.4. Tín hiệu logic

Tín hiệu logic là tín hiệu mà thông tin có tại bất kỳ thời điểm nào, giá trị thông tin biểu diễn qua biên độ và chỉ có hai giá trị “0” hoặc “1”. Giá trị trung gian giữa hai giá trị này bị cấm (không được phép có).

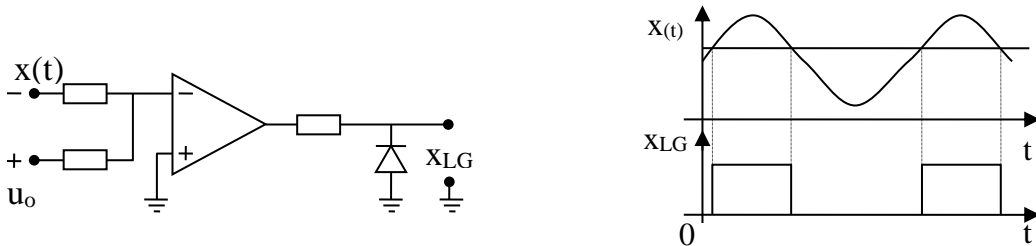


K mở: Điện áp $U = 220\text{ V}$: Đèn tắt $\Rightarrow x_{LG} = 0$

K đóng: Điện áp $U = 0\text{ V}$: Đèn sáng $\Rightarrow x_{LG} = 1$

Để chuyển đổi tín hiệu tương tự thành tín hiệu logic ta sử dụng mạch so sánh qua

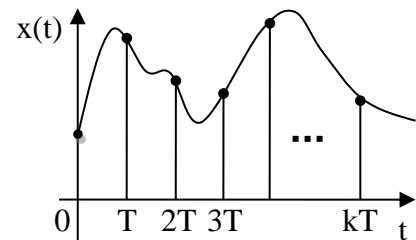
0:



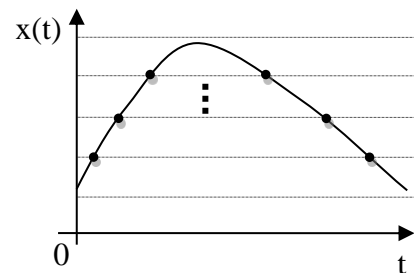
1.3.2. Xử lý tín hiệu

Như ta đã biết trong hệ điều khiển số luôn tồn tại cả tín hiệu tương tự và tín hiệu số. Do vậy, phải xảy ra quá trình chuyển đổi tín hiệu từ tín hiệu tương tự thành tín hiệu số và ngược lại.

Để chuyển đổi tín hiệu từ tín hiệu tương tự sang tín hiệu số, phải trải qua quá trình lấy mẫu (lượng tử hóa hay rời rạc hóa) tín hiệu. Có ba phương pháp lượng tử hóa tín hiệu:

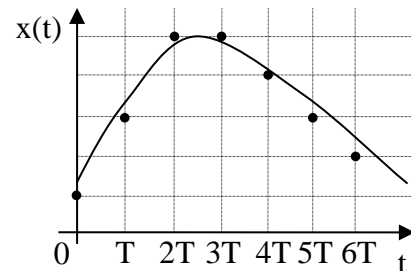


- Lượng tử hóa theo thời gian: thực hiện lấy mẫu tín hiệu tại những thời điểm định trước, cách đều nhau một khoảng thời gian T , T được gọi là chu kỳ lấy mẫu (sample time). Giá trị thu được là những giá trị (biên độ) của tín hiệu tại những thời điểm lấy mẫu.



- Lượng tử hóa theo mức: Lượng tử hóa tín hiệu khi tín hiệu đạt những giá trị định trước.

- Lượng tử hóa hỗn hợp: Lấy mẫu tín hiệu vào những thời điểm định trước, cách đều nhau một chu kỳ lấy mẫu T . Chia giá trị tín hiệu ra những mức cách đều nhau. Khoảng cách giữa các mức lân cận được gọi là một bước lượng tử. Giá trị thu được bằng mức định trước, có sai số bé nhất so với giá trị thực của tín hiệu tại thời điểm lấy mẫu.



Để thực hiện quá trình chuyển đổi tín hiệu ta sử dụng các bộ chuyển đổi tương tự - số ADC và số - tương tự DAC.

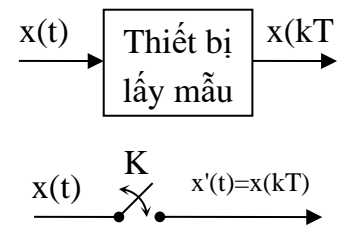
1.3.2.1. Chuyển đổi tương tự - số

Để chuyển đổi từ tín hiệu tương tự thành tín hiệu số ta phải thực hiện qua hai bước:

- Chuyển đổi tín hiệu tương tự thành tín hiệu lấy mẫu thông qua quá trình lấy mẫu tín hiệu: $x(t) \rightarrow x'(t)$

- Chuyển từ tín hiệu lấy mẫu thành tín hiệu số thông qua quá trình lượng tử hóa theo mức: $x'(t) \rightarrow x^*(t)$

Để lấy mẫu tín hiệu ta sử dụng thiết bị lấy mẫu. Thiết bị lấy mẫu sẽ tạo ra một dãy xung đơn vị từ một tín hiệu thời gian liên tục. Theo định nghĩa thiết bị lấy mẫu là một công tắc đóng cắt theo chu kỳ cố định T với tổng thời gian đóng cắt bằng 0.



Về mặt lý thuyết ta thu được ở đầu ra của thiết bị lấy mẫu tín hiệu lấy mẫu đảm bảo thông tin chính xác và đồng bộ. Tuy nhiên, về mặt thiết bị thì không tồn tại công tắc có tổng thời gian đóng cắt bằng 0. Thực tế, công tắc này là công tắc điện tử, đây là linh kiện ba trạng thái với thời gian đóng cắt cỡ μs . Vì vậy, trong khoảng thời gian này tín hiệu thực có giá trị biến đổi không đáng kể.

Giả thiết thời gian tác động của công tắc ngắn hơn nhiều khoảng thời gian giữa hai lần lấy mẫu (lý tưởng thì tổng thời gian đóng cắt bằng 0). Khi đó, giá trị của hàm lấy mẫu ở đầu ra của công tắc sẽ là giá trị tức thời của hàm liên tục $x(t)$ khi khoá K đóng.

Để có hình ảnh toán học rõ ràng về quá trình lấy mẫu ta có thể xem bộ lấy mẫu như một công cụ thực hiện phép nhân tín hiệu $x(t)$ với hàm lấy mẫu $S(t)$, tương đương với việc điều chế tín hiệu. Trong đó hàm lấy mẫu $S(t)$ đóng vai trò là sóng mang và nó được điều chế bởi tín hiệu vào $\delta(t)$.

$$x(kT) = S(t).x(t)$$

$$\delta(t) = \begin{cases} 0 & \text{khi } t \neq 0 \\ \infty & \text{khi } t = 0 \end{cases}; \quad \delta(t - \tau) = \begin{cases} 0 & \text{khi } t \neq \tau \\ \infty & \text{khi } t = \tau \end{cases}$$

Thực tế, để biểu diễn toán học thiết bị lấy mẫu người ta thường dùng hàm xung

đơn vị:
$$\delta(t) = \begin{cases} 0 & \text{khi } t \neq \tau \\ 1 & \text{khi } t = \tau \end{cases}$$

$$\text{Khi đó: } x'(t) = \sum_{k=0}^{\infty} x(t)\delta(t - kT) = \sum_{k=0}^{\infty} x(kT)\delta(t - kT) = \sum_{k=0}^{\infty} x_k \delta(t - kT)$$

$$\text{Hay: } x'(t) = x_0\delta(t) + x_1\delta(t - T) + x_2\delta(t - 2T) + \dots + x_k\delta(t - kT) + \dots$$

-Chu kỳ lấy mẫu T ảnh hưởng đến độ chính xác của quy luật tín hiệu theo hướng T càng nhỏ thì độ chính xác của tín hiệu càng cao, thiết bị chế tạo càng khó khăn. Vì vậy, theo quan điểm về tín hiệu thì T càng nhỏ thì càng tốt nhưng theo quan điểm về thiết bị thì T càng lớn càng tốt. Do vậy, trong thực tế T thường được chọn theo điều kiện

$T < T_{th}/2$ với T_{th} là chu kỳ của tín hiệu cần điều khiển, T_{th} được xác định thông qua máy phổ tần.

- Điều chế tín hiệu là quá trình biến đổi một hay nhiều thông số của một tín hiệu tuần hoàn theo sự thay đổi một tín hiệu mang thông tin cần truyền đi xa. Tín hiệu tuần hoàn gọi là sóng mang. Tín hiệu mang thông tin gọi là tín hiệu được điều chế. Ở đầu thu bộ giải điều chế sẽ dựa vào sự thay đổi thông số đó của sóng mang tái tạo lại tín hiệu mang thông tin ban đầu. Các thông số của sóng mang được dùng trong quá trình điều chế có thể là biên độ, pha, tần số.

Ví dụ: tín hiệu tiếng nói có tần số thấp, không thể truyền đi xa được. Người ta dùng một tín hiệu hình sin có tần số cao (để có thể truyền đi xa được) làm sóng mang. Biến đổi biên độ của tần số sin đó theo tín hiệu tiếng nói. Ở đầu thu người ta dựa vào sự thay đổi biên độ của tín hiệu thu được để tái tạo lại tín hiệu tiếng nói ban đầu.)

1.3.2.2. Chuyển đổi số - tương tự

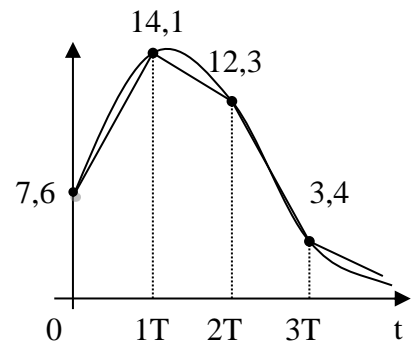
Quá trình chuyển đổi tín hiệu từ tín hiệu số thành tín hiệu tương tự diễn ra như sau:

- Giải mã tín hiệu số: $x^*(t) \rightarrow x'(t)$
- Khôi phục tín hiệu: $x'(t) \rightarrow x(t)$

Trong quá trình chuyển đổi thì nhiệm vụ khôi phục tín hiệu là vấn đề cần quan tâm, nó quyết định giá thành của thiết bị và mức độ phức tạp của kỹ thuật.

Ví dụ: mức mã hóa là 0,1 V và thực hiện chuyển đổi với DAC 8 bit

$t=0 \rightarrow$	128 64 32 16 8 4 2 1	$\rightarrow 76.0,1 = 7,6 \text{ V}$
	0 1 0 0 1 1 0 0	
$t=1T \rightarrow$	128 64 32 16 8 4 2 1	$\rightarrow 141.0,1 = 14,1 \text{ V}$
	1 0 0 0 1 1 0 1	
$t=2T \rightarrow$	128 64 32 16 8 4 2 1	$\rightarrow 123.0,1 = 12,3 \text{ V}$
	0 1 1 1 0 0 1 1	
$t=3T \rightarrow$	128 64 32 16 8 4 2 1	$\rightarrow 34.0,1 = 3,4 \text{ V}$
	0 0 1 0 0 0 1 0	



Để thực hiện khôi phục tín hiệu, ta chỉ cần khôi phục được tín hiệu trong một chu kỳ; các chu kỳ khác lặp lại với quy luật bất kỳ thông qua chuỗi Taylor:

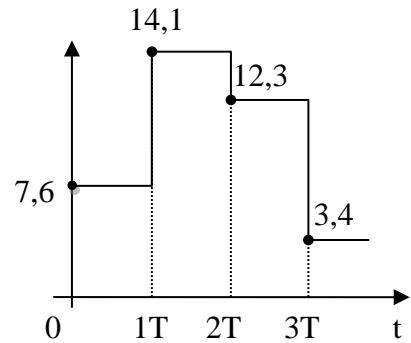
$$x(kT + \tau) = x(kT) + \frac{\tau^1}{1!} x'(kT) + \frac{\tau^2}{2!} x''(kT) + \dots + \frac{\tau^n}{n!} x^n(kT) + \dots \text{ với } 0 \leq \tau \leq T$$

Nói cách khác để khôi phục tín hiệu ta sử dụng khâu lưu giữ tín hiệu. Khâu giữ dữ liệu là khâu chuyển tín hiệu rời rạc theo thời gian thành tín hiệu liên tục theo thời gian

- + Lấy một số hạng đầu (lưu giữ bậc 0 - Zero Order Hold - ZOH):

$$x(kT + \tau) = x(kT) = \text{const}$$

Tín hiệu được phục hồi chỉ phụ thuộc vào hàm đã được lấy mẫu tại thời điểm bắt đầu của chu kỳ lấy mẫu. Lưu giữ ZOH có thể coi tương tự như máy khoá điện tử, nó duy trì mức điện áp đầu ra bằng biên độ xung đầu vào và sau đó tự lặp lại khi có xung mới đặt vào.



Phương pháp khôi phục tín hiệu sử dụng khâu lưu giữ bậc không ZOH là phương pháp đơn giản nhất, hiện nay phương pháp này thường được sử dụng trong các DAC dân dụng và DAC công nghiệp giá rẻ.

+ Lấy hai số hạng đầu (lưu giữ bậc 1 - First Order Hold - FOH):

$$x(kT + \tau) = x(kT) + \tau \dot{x}(kT) \quad \text{Trong đó:}$$

$$\dot{x}(kT) = \frac{x(kT) - x[(k-1)T]}{T} = \frac{x[(k+1)T] - x(kT)}{T}$$

$$x(kT + \tau) = x(kT) + \frac{x(kT) - x[(k-1)T]}{T} \tau = b + a\tau$$

Để thực hiện chuyển đổi D/A phải qua các bước sau:

- Giải mã tín hiệu tại thời điểm kT ta được giá trị $x(kT)=b$ và ghi vào bộ nhớ
- Đọc giá trị $x[(k-1)T]$ của chu kỳ trước đã được ghi trong bộ nhớ. Đọc giá trị $x(kT)$ vừa ghi và đọc giá trị thời gian lấy mẫu T sau đó thực hiện phép tính

$$\frac{x(kT) - x[(k-1)T]}{T} = a$$

- Điều chế điện áp đầu ra đi qua giá trị b theo quy luật đường thẳng với hệ số góc a vừa tìm được. Để thực hiện được các bước trên ta phải sử dụng vi xử lý, điều này làm tăng giá thành của chuyển đổi D/A nhưng độ chính xác chuyển đổi tín hiệu cao hơn nhiều. DAC kiểu này thường được sử dụng trong kỹ thuật quân sự.

+ Lấy ba số hạng đầu (lưu giữ bậc 2 – Second Order Hold - SOH):

$$x(kT + \tau) = x(kT) + \frac{\tau^1}{1!} \dot{x}(kT) + \frac{\tau^2}{2!} \ddot{x}(kT)$$

$$\ddot{x}(kT) = \frac{\dot{x}(kT) - \dot{x}[(k-1)T]}{T} = \frac{x(kT) - x[(k-1)T] - x[(k-1)T] + x[(k-2)T]}{2T^2}$$

$$x(kT + \tau) = x(kT) + \frac{x(kT) - x[(k-1)T]}{T} \tau + \frac{x(kT) - 2x[(k-1)T] + x[(k-2)T]}{2T^2} \tau^2 = c + b\tau + a\tau^2$$

Do khối lượng tính toán quá lớn nên các hàm giữ chậm có bậc từ 3 trở lên hiện nay chưa được sử dụng.

1.4 Hàm truyền hệ điều khiển số

1.4.1. Định nghĩa

Hàm truyền của một phần tử hoặc của hệ thống điều khiển là tỷ số giữa ảnh hưởng ra và ảnh hưởng vào của phần tử hoặc hệ thống đó theo toán tử Z với điều kiện đầu bằng không.

$$W(z) = \frac{Y(z)}{X(z)} \Big|_{\text{Điều kiện đầu}=0} = \frac{b_0 + b_1 z + \dots + b_m z^m}{a_0 + a_1 z + \dots + a_n z^n}$$

1.4.2 Biến đổi Z

1.4.2.1. Định nghĩa

Khi giải phương trình sai phân bậc cao người ta thường gặp nhiều khó khăn, vì vậy người ta thường dùng biến đổi Z để biến phương trình sai phân tuyến tính của hệ gián đoạn thành phương trình đại số. Điều này hoàn toàn tương tự như trong trường hợp hệ liên tục dùng biến đổi Laplace để biến phương trình vi tích phân thành phương trình đại số.

Trong hệ xung số (hệ rời rạc) phép biến đổi z giữ vai trò quan trọng. Nếu có hàm liên tục x(t) ta sẽ có hàm rời rạc x(kT) với chu kỳ cắt mẫu T. Khi đó:

$$x'(t) = \sum_{k=0}^{\infty} x(kT)\delta(t - kT)$$

Phép biến đổi Laplace của hàm rời rạc x'(t) gọi là chuyển đổi Laplace rời rạc và được định nghĩa như sau:

$$\begin{aligned} L[x'(t)] &= X'(s) = \int_0^{\infty} x'(t) \cdot e^{-st} dt = \int_0^{\infty} \sum_{k=0}^{\infty} x(kT)\delta(t - kT) \cdot e^{-st} dt = \sum_{k=0}^{\infty} \int_0^{\infty} x(kT)\delta(t - kT) \cdot e^{-st} dt \\ &\Leftrightarrow X'(s) = L\left[\sum_{k=0}^{\infty} x(kT)\delta(t - kT)\right] = \sum_{k=0}^{\infty} L[x(kT)\delta(t - kT)] = \\ &= \sum_{k=0}^{\infty} x(kT)L[\delta(t - kT)] = \sum_{k=0}^{\infty} x(kT) \cdot 1 \cdot e^{-kTs} \end{aligned}$$

$$\text{Đặt } z = e^{Ts} \Rightarrow X(z) = \sum_{k=0}^{\infty} x(kT)z^{-k}$$

X(z) được gọi là biến đổi Z của hàm rời rạc x'(t). Thực chất biến đổi Z là biến đổi tương đương của biến đổi Laplace rời rạc. Tính chất của hai phép biến đổi này là như nhau.

Viết dưới dạng đa thức ta có:

$$X(z) = z^0 + x(T)z^{-1} + x(2T)z^{-2} + \dots + x(kT)z^{-k} + \dots$$

1.4.2.2. Biến đổi Z thuận

Là phép biến đổi x(t), x(kT), X(s) → X(z)

- Đối với phép biến đổi x(t), x(kT) → X(z): dùng định nghĩa.

$$X(z) = \sum_{k=0}^{\infty} x(kT)z^{-k} = x_0z^{-0} + x_{(1T)}z^{-1} + x_{(2T)}z^{-2} + \dots + x_{(kT)}z^{-k} + \dots$$

Ta chỉ cần thay $t = 0T, 1T, 2T, \dots$ vào $x(t)$ hoặc $k=0,1,2,3, \dots$ vào $x(kT)$ ta sẽ có các giá trị $x_0, x_{(1T)}, \dots$ tương ứng.

- Đối với phép biến đổi $X(s) \rightarrow X(z)$: ta có thể dùng bảng ảnh – gốc (sử dụng khai triển Hevisai). Ngoài ra có thể chuyển ảnh Laplace liên tục $X(s)$ thành ảnh Laplace rời rạc $X(z)$ qua biểu thức:

$$X(z) = \sum_i \operatorname{Re} s \left(\frac{X(s)}{1 - e^{-Ts}e^{sT}} \right) \Big|_{s=s_i}$$

với $X(s) = \frac{B(s)}{A(s)}$ và s_i là nghiệm của phương trình đặc tính $A(s)=0$.

$$\text{Mà } z=e^{sT} \Rightarrow X(z) = \sum_i \operatorname{Re} s \left(\frac{X_i(s)}{1 - z^{-1}e^{sT}} \right) \Big|_{s=s_i}$$

- Nếu s_i là nghiệm đơn: $R_i = \operatorname{Re} \left(\frac{B(s)}{A'(s)} \cdot \frac{1}{1 - z^{-1}e^{sT}} \right) \Big|_{s=s_i}$; $A'(s) = \frac{dA(s)}{ds}$

- Nếu s_i là nghiệm bội:

$$R_i = \operatorname{Re} \left\{ \frac{1}{(n-1)!} \frac{d^{n-1}}{ds^{n-1}} \left[(s - s_i)^n \frac{B(s)}{A(s)} \frac{1}{1 - z^{-1}e^{sT}} \right] \right\} \Big|_{s=s_i}$$

$$X(z) = \sum_i R_i$$

1.4.2.3. Biến đổi Z ngược

Là quá trình biến đổi từ $X(z) \rightarrow x(kT) \rightarrow x(t)$. Có nhiều cách thực hiện biến đổi z ngược.

+ Dùng định nghĩa: $X(z) = \sum_{k=0}^{\infty} x(kT)z^{-k} = a_0z^{-0} + a_1z^{-1} + a_2z^{-2} + \dots + a_kz^{-k} + \dots$

$$X(z) = \sum_{k=0}^{\infty} x(kT)z^{-k} = x_{(0T)}z^{-0} + x_{(1T)}z^{-1} + x_{(2T)}z^{-2} + \dots + x_{(kT)}z^{-k} + \dots$$

$$x(0T)=a_0, x(1T)=a_1, \dots, x(kT)=a_k, \dots$$

khi đó kết quả là một bảng các giá trị cách nhau một chu kỳ T. Muốn tìm $x(t)$ ta phải qua khôi phục.

+ Dùng bảng ảnh-gốc: Xuất phát từ $X(z)$ qua khai triển Hevisai sau đó tiến hành tra bảng ảnh-gốc cho từng số hạng rồi cộng lại.

+ Dùng biểu thức tính toán:

$$\text{Theo công thức Côsi ta có: } x(kT) = \frac{1}{2\pi j} \oint_{\ell} z^{k-1} X(z) dz; \quad X(z) = \frac{B(z)}{A(z)}$$

Giải phương trình $A(z) = 0$ ta được các nghiệm z_i thì ℓ là đường cong kín bao toàn bộ các điểm nghiệm z_i .

$$R_i = \operatorname{Re} \left(\frac{B(z)}{A'(z)} z^{k-1} \right) \Bigg|_{z=z_i} ; A'(z) = \frac{dA(z)}{dz} \Rightarrow x(kT) = \sum_i R_i$$

Ví dụ: Cho $x(t)=1(t)$. Hãy tìm $X(z)$

- Theo định nghĩa: Hàm ảnh rời rạc của $1'(t)$ có dạng:

$$1'(s) = \sum_0^{\infty} e^{-kTs} \Leftrightarrow 1(z) = \sum_0^{\infty} z^{-k}$$

Đây là tổng của cấp số nhân lùi vô hạn với công bội bằng z^{-1} và số hạng đầu là $a_0=1$

$$\begin{aligned} X(z) &= \sum_{k=0}^{\infty} x(kT)z^{-k} = x_{(0T)}z^{-0} + x_{(1T)}z^{-1} + x_{(2T)}z^{-2} + \dots + x_{(kT)}z^{-k} + \dots \\ &= z^{-0} + z^{-1} + z^{-2} + \dots + z^{-k} + \dots = \frac{1}{1-z^{-1}} = \frac{z}{z-1} \end{aligned}$$

$$- x(t) = 1(t) \Rightarrow X(s) = \frac{1}{s} = \frac{B(s)}{A(s)}$$

Giải phương trình $A(s)=0$ ta được $s_1=0$:

$$X(z) = \operatorname{Re} \left(\frac{1}{1'(s)} \frac{1}{1-z^{-1}e^{Ts}} \right)_{s=0} = \frac{1}{1-z^{-1}} = \frac{z}{z-1}$$

$$X(z) = \frac{z}{z-1} = \frac{B(z)}{A(z)}$$

$$\frac{z}{z-1} = 1.z^0 + 1.z^{-1} + 1.z^{-2} + \dots + 1.z^{-k} + \dots$$

$$\Rightarrow a_0=x(0)=1; a_1=x(1T)=1; a_2=x(2T)=1; \dots a_k=x(kT)=1$$

$$\text{Theo biểu thức: } x(kT) = \frac{z}{1} z^{k-1} \Big|_{z=1} = z^k \Big|_{z=1} = 1^k$$

$$\left. \begin{aligned} k=0 &\Rightarrow x(0)=1 \\ k=1 &\Rightarrow x(1T)=1 \\ k=2 &\Rightarrow x(2T)=1 \\ &\dots \\ &\Rightarrow x(kT)=1 \end{aligned} \right\} \Rightarrow \text{Khôi phục ra hàm } 1(t)$$

1.4.2.4. Một số tính chất của biến đổi Z

+ Tính chất tuyến tính:

Giả sử ta có hàm gốc $x(t)=a.x_1(t)+b.x_2(t) \Rightarrow X(z)=a.X_1(z)+b.X_2(z)$

$$\begin{aligned} X(z) &= Z[x(t)] = Z[a.x_1(t) + b.x_2(t)] = Z[a.x_1(t)] + Z[b.x_2(t)] \\ &= a.Z[x_1(t)] + b.Z[x_2(t)] = a.X_1(z) + b.X_2(z) \end{aligned}$$

+ Tính chất hàm trễ (dịch gốc): Nếu có $x(kT)=x(k) \Rightarrow Z[x(k)]=X(z)$, với điều kiện đầu bằng 0 thì $Z[x(k-n)]=z^{-n}X(z)$ và $Z[x(k+n)]=z^nX(z)$

+ Định lý về giá trị đầu:

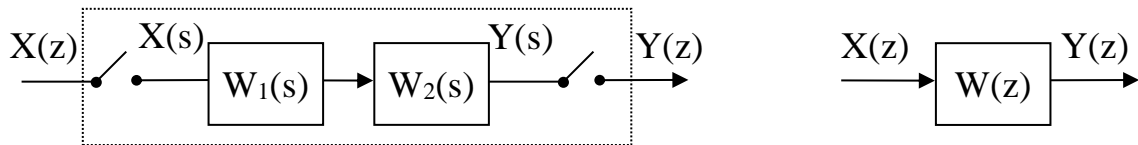
$$\lim_{k \rightarrow 0} x(kT) = \lim_{z \rightarrow \infty} X(z) = x(0)$$

+ Định lý về giá trị cuối của hàm gốc rời rạc:

$$\lim_{t \rightarrow \infty} x(t) = \lim_{z \rightarrow 1} (z-1)X(z)$$

1.4.3. Xác định hàm truyền hệ điều khiển số.

+ Thiết bị tương tự ghép nối tiếp trong hệ điều khiển số

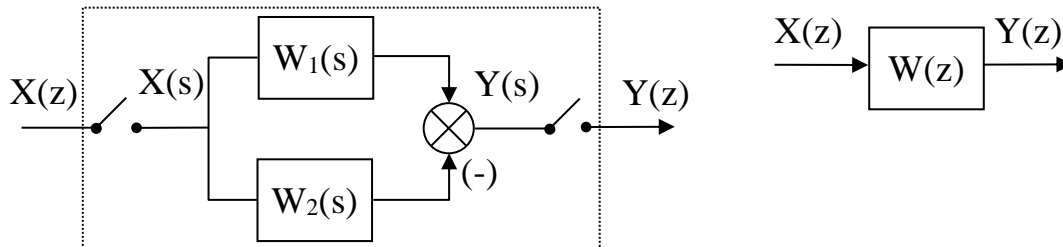


$$Y(z) = Z[Y(s)] = Z[X(s) \cdot W_1(s) \cdot W_2(s)] = Z[X(s)] \cdot Z[W_1(s) \cdot W_2(s)] \\ = X(z) \cdot Z[W_1(s) \cdot W_2(s)]$$

$$W(z) = \frac{Y(z)}{X(z)} = Z\left[\prod_{i=1}^n W_i(s)\right]$$

Lưu ý: $Z\left[\prod_{i=1}^n W_i(s)\right] \neq \prod_{i=1}^n Z[W_i(s)]$

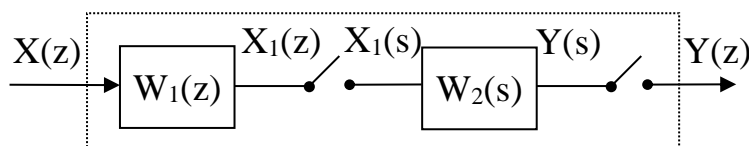
+ Thiết bị tương tự ghép song song trong hệ điều khiển số



$$Y(z) = Z[Y(s)] = Z\{X(s) \cdot [W_1(s) - W_2(s)]\} = Z[X(s)] \cdot Z[W_1(s) - W_2(s)] \\ = X(z) \cdot Z[W_1(s) - W_2(s)]$$

$$W(z) = \frac{Y(z)}{X(z)} = Z[W_1(s)] - Z[W_2(s)] = \sum_{i=1}^n Z[W_i(s)]$$

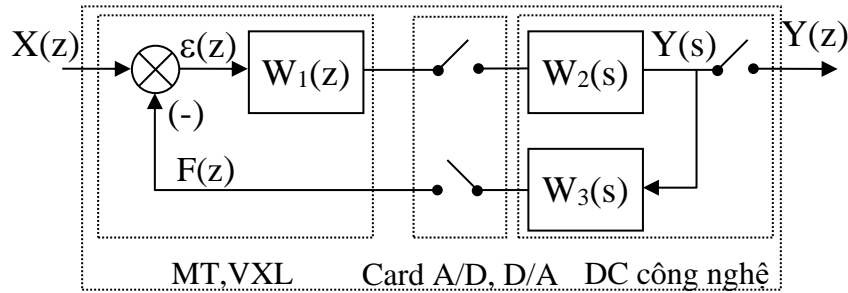
+ Ghép nối thiết bị số và tương tự



$$Y(z) = Z[Y(s)] = Z[X_1(s)W_2(s)] = Z[W_1(s)].Z[W_2(s)] \\ = X_1(z).W_1(z).Z[W_2(s)]$$

$$W(z) = \frac{Y(z)}{X(z)} = W_1(z).Z[W_2(s)]$$

+ Thiết bị ghép phản hồi



$$\varepsilon(z) = X(z) - F(z)$$

$$Y(z) = \varepsilon(z).W_1(z).Z[W_2(s)]$$

$$F(z) = \varepsilon(z).W_1(z).Z[W_2(s).W_3(s)]$$

$$W(z) = \frac{Y(z)}{X(z)} = \frac{W_1(z).Z[W_2(s)]}{1 + W_1(z).Z[W_2(s).W_3(s)]}$$

$$= \frac{\text{Hệ thống tương đương đầu nút đến đầu ra hệ}}{1 + \text{Hệ thống tương đương đầu nút đến đầu vào phản hồi}}$$

1.5 Xét ổn định hệ điều khiển số

1.5.1. Khái niệm và thông số ảnh hưởng

Khái niệm ổn định: Hệ thống điều khiển tự động được gọi là ổn định nếu hệ có khả năng tự động trở lại trạng thái ban đầu sau khi nhiễu tác động vào hệ.

Thông số ảnh hưởng: giả sử hệ điều khiển có hàm truyền liên tục:

$$W(s) = \frac{B(s)}{A(s)}$$

Tính ổn định của hệ chỉ phụ thuộc vào dấu của phần thực của các nghiệm của phương trình đặc tính $A(s)=0$: $s_i = \alpha_i \pm j\beta_i$. Xét dấu của α_i ta có thể đưa ra được tiêu chuẩn ổn định.

Sử dụng phép biến đổi Laplace:

Đặt $z = e^{Ts}$. Giả sử trong không gian Laplace phương trình đặc tính có nghiệm $s_i = \alpha_i \pm j\beta_i$ thì $z_i = e^{(\alpha_i \pm j\beta_i)T} = e^{\alpha_i T} e^{\pm j\beta_i T} = A_{z_i} e^{j\varphi_{z_i}} = |z_i| e^{j\arg z_i}$

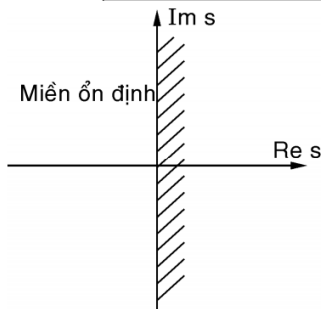
Do đó, trong không gian Laplace, nếu:

$$+ \alpha_i < 0: \text{hệ ổn định} \rightarrow |z_i| = e^{-\alpha_i T} < 1$$

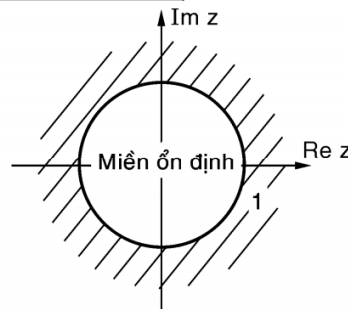
$$+ \alpha_i = 0: \text{hệ ở biên giới ổn định} \rightarrow |z_i| = e^{0T} = 1$$

+ $\alpha_i > 0$: hệ không ổn định $\rightarrow |z_i| = e^{\alpha_i T} > 1$

Hệ thống rời rạc ổn định $\Leftrightarrow |z| < 1$



Miền ổn định của hệ thống liên tục



Miền ổn định của hệ thống rời rạc

Như vậy: Tính ổn định của hệ điều khiển số chỉ phụ thuộc vào modul của các nghiệm z_i của phương trình đặc tính $A(z) = 0$ so với 1.

- Nếu tất cả các nghiệm của phương trình đặc tính $A(z) = 0$ đều có modul nhỏ hơn 1 (nằm trong vòng tròn đơn vị) thì hệ ổn định.
- Nếu có nghiệm có modul bằng 1 (nằm trên vòng tròn đơn vị) còn các nghiệm khác có modul nhỏ hơn 1 (nằm trong vòng tròn đơn vị) thì hệ ở biên giới ổn định.
- Chỉ cần có một nghiệm của phương trình đặc tính $A(z) = 0$ có modul lớn hơn 1 thì hệ không ổn định.

1.5.2. Phương pháp đổi biến (Tiêu chuẩn Routh-Hurwitz mở rộng)

Ba kết luận trên rất rõ ràng và dễ áp dụng nhưng gặp khó khăn khi giải phương trình $A(z)=0$. Chỉ nên áp dụng các kết luận trên khi $A(z)$ của hệ có dạng tích của các đa thức bậc 1, bậc 2.

Trong trường hợp tổng quát ta nên tìm cách xét modul của nghiệm z_i của phương trình đặc tính $A(z)=0$ so với 1 thông qua phép đổi biến:

$$z = \frac{w+1}{w-1} \text{ hay } w = \frac{z+1}{z-1} \quad (*) \text{ với } w \text{ là biến mới sao cho đảm bảo sự tương ứng:}$$

$$\alpha_i < 0 \Leftrightarrow |z_i| = e^{-\alpha_i T} < 1$$

$$\alpha_i = 0 \Leftrightarrow |z_i| = e^{0T} = 1$$

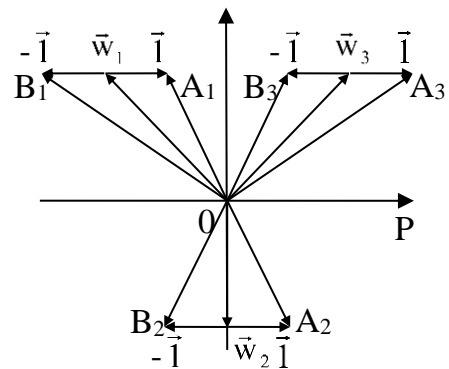
$$\alpha_i > 0 \Leftrightarrow |z_i| = e^{\alpha_i T} > 1$$

Để chứng minh ta dùng toán vector. Vector hóa hai vế biểu thức (*) ta có

$$\bar{z} = \frac{\overline{w+1}}{\overline{w-1}} = \frac{\bar{w} + \bar{1}}{\bar{w} - \bar{1}} \Rightarrow |\bar{z}| = \left| \frac{\bar{w} + \bar{1}}{\bar{w} - \bar{1}} \right| = \frac{|\bar{w} + \bar{1}|}{|\bar{w} - \bar{1}|}$$

Giả sử u_1 nằm bên trái mặt phẳng phức (phần thực âm)

$$|\bar{z}_1| = \frac{|\overline{0w_1 + \bar{1}}|}{|\overline{0w_1 - \bar{1}}|} = \frac{|0A_1|}{|0B_1|} = \frac{0A_1}{0B_1} < 1$$



Giả sử u_2 nằm trên trục ảo (phần thực bằng 0).

Tương tự ta có: $|\bar{z}_2| = \frac{\overline{OA_2}}{OB_2} = 1$

Giả sử u_3 nằm bên phải trục ảo (phần thực > 0).

Tương tự ta có: $|\bar{z}_3| = \frac{\overline{OA_3}}{OB_3} > 1$

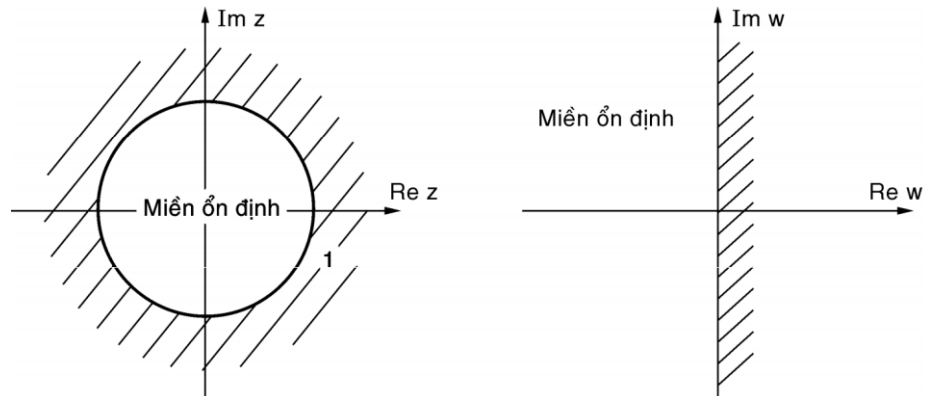
Áp dụng:

- Từ hệ điều khiển số đã có ta tìm được hàm truyền $W_{HT}(z)$.

- Thay $z = \frac{w+1}{w-1}$ vào $W_{HT}(z)$ ta được hàm truyền mới $W_{HT}(w)$. Thực hiện biến

đổi toán học và tối giản: $W_{HT}(u) = \frac{B(w)}{A(w)}$

- Lấy phương trình đặc tính $A(u) = a_0 w^n + a_1 w^{n-1} + \dots + a_n = 0$. Xét ổn định hệ thống theo tiêu chuẩn Routh hoặc Hurwitz.



1.5.3. Quỹ đạo nghiệm số (Root Locus) của hệ điều khiển số

QĐNS là tập hợp tất cả các nghiệm của phương trình đặc trưng của hệ thống khi có một thông số nào đó trong hệ thay đổi từ $0 \rightarrow \infty$.

Xét hệ thống rời rạc có phương trình đặc trưng là:

$$1 + K \frac{N(z)}{D(z)} = 0 \quad (1.5.1)$$

Đặt: $W_0(z) = K \frac{N(z)}{D(z)}$

Gọi n là số cực của $W_0(z)$, m là số zero của $W_0(z)$.

Phương trình đặc trưng (1.5.1) trở thành: $1 + W_0(z) = 0$.

$$\Leftrightarrow \begin{cases} |W_0(z)| = 1 & \text{Điều kiện biên độ} \\ \angle W_0(z) = (2l + 1)\pi & \text{Điều kiện pha} \end{cases}$$

Vì dạng phương trình đặc trưng của hệ liên tục đã khảo sát ở chương 4 và phương trình đặc trưng (1.5.1) là như nhau (chỉ thay biến s bằng biến z) nên qui tắc vẽ QĐNS là như nhau, chỉ khác ở qui tắc 8, thay vì đối với hệ liên tục ta tìm giao điểm của QĐNS

với trục ảo thì đối với hệ rời rạc ta tìm giao điểm của QĐNS với đường tròn đơn vị. Sau đây là 11 qui tắc về quỹ đạo nghiệm số của hệ thống rời rạc có phương trình đặc trưng có dạng (1.5.1).

Chú ý: Nếu phương trình đặc trưng của hệ không có dạng (1.5.1) thì ta phải biến đổi tương đương về dạng (1.5.1) trước khi áp dụng các qui tắc về QĐNS.

Qui tắc 1: Số nhánh của quỹ đạo nghiệm số = bậc của phương trình đặc trưng = số cực của $W_0(z) = n$

Qui tắc 2: Khi $K=0$: các nhánh của quỹ đạo nghiệm số xuất phát từ các cực của $W_0(z)$

Khi K tiến đến $+\infty$: nhánh của quỹ đạo nghiệm số tiến đến m zero của $W_0(z)$, $n-m$ nhánh còn lại tiến đến ∞ theo các tiệm cận xác định bởi qui tắc 5 và 6.

Qui tắc 3: Quỹ đạo nghiệm số đối xứng qua trục thực.

Qui tắc 4: Một điểm trên trục thực thuộc về quỹ đạo nghiệm số nếu tổng số cực và zero của $W_0(z)$ bên phải nó là một số lẻ.

Qui tắc 5: Góc tạo bởi các đường tiệm cận của quỹ đạo nghiệm số với trục thực xác định bởi:

$$\alpha = \frac{(2l+1)\pi}{n-m} \quad l = 0, \pm 1, \pm 2, \pm 3, \dots$$

Qui tắc 6: Giao điểm giữa các tiệm cận với trục thực là điểm A có tọa độ xác định bởi:

$$OA = \frac{\sum_1^n p_i - \sum_1^n z_i}{n-m} \quad \text{trong đó } p_i, z_i \text{ là các điểm cực, điểm không của } W_0(z)$$

Qui tắc 7: Điểm tách nhập (nếu có) của quỹ đạo nghiệm số nằm trên trục thực và là nghiệm của phương trình:

$$\frac{dK}{dz} = 0$$

Qui tắc 8: Giao điểm của quỹ đạo nghiệm số với đường tròn đơn vị có thể xác định bằng 1 trong 2 cách sau đây:

- Áp dụng tiêu chuẩn Routh – Hurwitz mở rộng hoặc tiêu chuẩn Jury.
- Thay $z = a + jb$ (điều kiện $a^2 + b^2 = 1$ do giao điểm nằm trên đường tròn đơn vị) vào phương trình đặc trưng (1.5.1), cân bằng phần thực và phần ảo sẽ tìm được giao điểm giữa QĐNS với đường tròn đơn vị và giá trị K_{gh} .

Qui tắc 9: Góc xuất phát của quỹ đạo nghiệm số tại cực phức p_i được xác định bởi:

$$\theta_j = 180^\circ + \sum_{i=1}^m \arg(p_j - z_i) - \sum_{\substack{i=1 \\ i \neq j}}^n \arg(p_j - p_i)$$

Dạng hình học của công thức trên là:

$\theta_j = 180^\circ + \sum(\text{góc từ các zero đến cực } p_i) - \sum(\text{góc từ các cực còn lại đến cực } p_i)$.

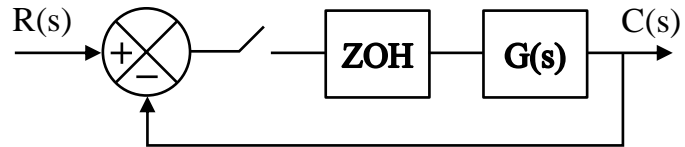
Qui tắc 10: Tổng các nghiệm là hằng số khi K thay đổi từ $0 \rightarrow +\infty$.

Qui tắc 11: Hệ số khuếch đại dọc theo quỹ nghiệm số có thể xác định từ điều kiện biên độ:

$$\left| K \frac{N(z)}{D(z)} \right| = 1$$

Ví dụ: Cho hệ điều khiển rời rạc có sơ đồ khối như sau:

$$G(s) = \frac{K}{s(s+5)}; T=0,1 \text{ (s)}$$



Vẽ QĐNS của hệ trên khi K thay đổi từ 0 đến $+\infty$. Tính K_{gh} .

Giải:

Phương trình đặc trưng của hệ là: $1+W_0(z)=0$.

$$\begin{aligned} W_0(z) &= Z[W_{ZOH}(s).W_0(s)] = Z\left[\frac{1-e^{-Ts}}{s} \frac{K}{s(s+5)}\right] = Z\left[\frac{1-z^{-1}}{s} \frac{K}{s(s+5)}\right] \\ &= \frac{K}{5} \frac{z-1}{z} Z\left[\frac{5}{s^2(s+5)}\right] = \frac{K}{5} \frac{z-1}{z} \left(\frac{Tz}{(z-1)^2} - \frac{(1-e^{-5T})z}{5(z-1)(z-e^{-5T})} \right) \\ &= \frac{K}{5} \left(\frac{T}{z-1} - \frac{1-e^{-5T}}{5(z-e^{-5T})} \right) = \frac{K}{5} \left(\frac{5T(z-e^{-5T}) - (1-e^{-5T})(z-1)}{5(z-1)(z-e^{-5T})} \right) \\ &= \frac{K}{5} \left(\frac{(0,5-1+e^{-0,5})z + 1 - e^{-0,5} - 0,5e^{-0,5}}{5(z-1)(z-e^{-0,5})} \right) = \frac{K}{5} \left(\frac{0,107z + 0,91}{5(z-1)(z-0,607)} \right) \\ &= \frac{K}{5} \frac{0,021z + 0,018}{(z-1)(z-0,607)} \end{aligned}$$

$$W_0(z) = K \frac{0,0042z + 0,0036}{(z-1)(z-0,607)}$$

$$\text{Phương trình đặc trưng của hệ là: } 1 + K \frac{0,0042z + 0,0036}{(z-1)(z-0,607)} = 0 \quad (*)$$

- $W_0(z)$ có các điểm cực: $p_1=1, p_2=0,607$ ($n=2$)
- $W_0(z)$ có zero: $z_1=-0,857$ ($m=1$)
- Góc tạo bởi tiệm cận và trục thực: $\alpha = \frac{(2l+1)\pi}{n-m} = \frac{(2.0+1)\pi}{2-1} = \pi$ ($l=0$)
- Giao điểm giữa tiệm cận với trục thực:

$$OA = \frac{\sum_1^n p_i - \sum_1^m z_i}{n-m} = \frac{1+0,607+0,857}{2-1} = 2,464$$

- Điểm tách nhập là nghiệm của phương trình: $\frac{dK}{dz} = 0$

$$K = -\frac{(z-1)(z-0,607)}{0,0042z + 0,0036} = -\frac{z^2 - 1,607z + 0,607}{0,0042z + 0,0036}$$

$$\begin{aligned}\Rightarrow \frac{dK}{dz} &= -\frac{z^2 - 1,607z + 0,607}{0,0042z + 0,0036} \\ &= -\frac{(2z - 1,607)(0,0042z + 0,0036) - 0,0042(z^2 - 1,607z + 0,607)}{(0,0042z + 0,0036)^2} \\ &= -\frac{0,0042z^2 + 0,0072z - 0,0083}{(0,0042z + 0,0036)^2} \\ \frac{dK}{dz} = 0 &\Leftrightarrow \frac{0,0042z^2 + 0,0072z - 0,0083}{(0,0042z + 0,0036)^2} = 0 \Leftrightarrow \begin{cases} z_1 = -2,506 \\ z_2 = 0,792 \end{cases}\end{aligned}$$

Cả hai nghiệm đều thuộc QĐNS nên QĐNS có 2 điểm tách nhập.

Giao điểm của QĐNS với đường tròn đơn vị:

$$(*) \Leftrightarrow (z - 1)(z - 0,607) + K(0,0042z + 0,0036) = 0$$

$$\Leftrightarrow z^2 + (0,0042K - 1,607)z + 0,0036K + 0,607 = 0 (**)$$

Cách 1: Dùng tiêu chuẩn Routh-Hurwitz mở rộng

$$\text{Thay } z = \frac{w+1}{w-1} \text{ vào } (**) \text{ ta có: } \left(\frac{w+1}{w-1}\right)^2 + (0,0042K - 1,607)\frac{w+1}{w-1} + 0,0036K + 0,607 = 0$$

$$\Leftrightarrow 0,0078Kw^2 + (0,786 - 0,0072K)w + 2,607 - 0,0006K = 0$$

Điều kiện để hệ thống ổn định là:

$$\begin{cases} K > 0 \\ K < \frac{0,786}{0,0072} = 109,17 \\ K < \frac{2,607}{0,0006} = 4345 \end{cases} \Leftrightarrow 0 < K < 109,17 \Rightarrow K_{gh} = 109,17$$

Thay $K_{gh} = 109,17$ vào phương trình (**), ta được:

$$z^2 - 1,149z + 1 = 0 \Leftrightarrow z = 0,575 \pm j0,819$$

Vậy giao điểm của QĐNS với vòng tròn đơn vị là: $z = 0,575 \pm j0,819$

Cách 2: Thay vào $z = a + jb$ phương trình (**), ta được:

$$(a + jb)^2 + (0,0042K - 1,607)(a + jb) + 0,0036K + 0,607 = 0$$

$$\Leftrightarrow a^2 - b^2 + (0,0042K - 1,607)a + 0,0036K + 0,607 + jb(2a + 0,0042K - 1,607) = 0$$

$$\Leftrightarrow \begin{cases} a^2 - b^2 + (0,0042K - 1,607)a + 0,0036K + 0,607 = 0 \\ jb(2a + 0,0042K - 1,607) = 0 \end{cases}$$

Kết hợp với điều kiện $a^2 + b^2 = 1$ ta được hệ phương trình:

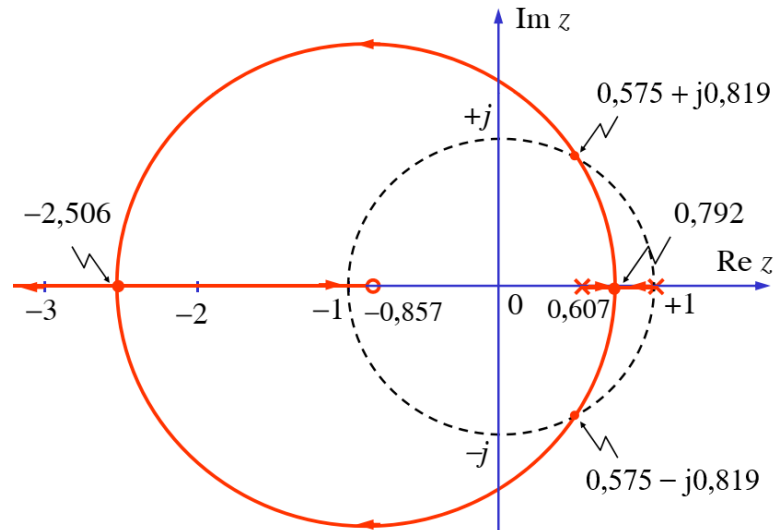
$$\begin{cases} a^2 - b^2 + (0,0042K - 1,607)a + 0,0036K + 0,607 = 0 \\ jb(2a + 0,0042K - 1,607) = 0 \\ a^2 + b^2 - 1 = 0 \end{cases}$$

Giải hệ phương trình trên, ta được bốn giao điểm là

$z = 1$, tương ứng với $K = 0$

$z = -1$, tương ứng với $K = 4345$

$z = 0,575 \pm j0,819$, tương ứng với $K = 109,17$



1.6. Đánh giá chất lượng hệ điều khiển số

1.6.1. Các chỉ tiêu chất lượng

Tương tự hệ điều khiển tương tự, hệ điều khiển số sau khi kết luận đã ổn định ta phải đánh giá chất lượng của hệ.

Tuỳ theo mô tả toán học hệ rời rạc mà ta có thể xác định được đáp ứng của hệ rời rạc bằng một trong hai cách sau đây:

Cách 1: Nếu hệ rời rạc mô tả bởi hàm truyền thì trước tiên ta tính $(Y(z))$, sau đó dùng phép biến đổi Z ngược để tìm $y(k)$.

Cách 2: Nếu hệ rời rạc mô tả bởi phương trình trạng thái thì trước tiên ta tính nghiệm $x(k)$ của phương trình trạng thái, sau đó suy ra $y(k)$.

Tương tự như hệ liên tục ta cũng có khái niệm cực quyết định cho hệ rời rạc. Đối với hệ liên tục cặp cực quyết định là cặp cực nằm gần trục ảo nhất. Do quan hệ $z=e^{Ts}$, nên đối với hệ rời rạc cặp cực quyết định là cặp cực nằm gần vòng tròn đơn vị nhất. Hệ bậc cao có nhiều hơn hai cực. Đáp ứng tương ứng với các cực nằm càng xa trục ảo suy giảm càng nhanh. Do đó có thể xấp xỉ hệ bậc cao về hệ bậc hai với cặp cực là hai cực nằm gần trục ảo nhất. Cặp cực nằm gần trục ảo nhất của hệ bậc cao gọi là cặp cực quyết định. Hệ bậc cao có thể xấp xỉ gần đúng về hệ bậc hai với 2 cực là cặp cực quyết định.

Việc đánh giá chất lượng của hệ được thực hiện thông qua các chỉ tiêu chất lượng:

+ **Chất lượng của quá trình xác lập:**

$$\text{Sai lệch tĩnh: } St\% = \frac{y_{xl} - y_{yc}}{y_{yc}} 100 = \frac{y_{yc} - y_{xl}}{y_{yc}} 100$$

St càng nhỏ thì chất lượng của hệ càng tốt và giá thành của hệ càng cao.

+ **Chất lượng của quá trình quá độ:**

Cách 1: Đánh giá chất lượng quá độ dựa vào đáp ứng của hệ thống.

Trước tiên ta phải tính được đáp ứng $y(k)$ của hệ thống

- Lượng quá điều chỉnh: $\sigma_{\max} \% = \frac{y_{\max} - y_{yc}}{y_{yc}} 100$

y_{\max} là lượng ra cực đại. Lượng quá điều chỉnh ảnh hưởng đến tuổi thọ của thiết bị do vậy, trong hệ điều khiển nói chung và hệ điều khiển số nói riêng mong muốn y_{\max} càng nhỏ càng tốt.

- Thời gian quá độ: T_{qd} là thời gian hệ chuyển từ trạng thái xác lập này sang trạng thái xác lập khác.

Gọi k_{qd} là thời điểm lấy mẫu mà từ đó trở đi đáp ứng $y(k)$ của hệ thống biến thiên không quá $St\%$ so với giá trị xác lập y_{xl} , nghĩa là:

$$|y(k) - y_{xl}| \leq \frac{St \cdot y_{xl}}{100}, \quad \forall k \geq k_{qd}$$

$$\left(1 - \frac{St}{100}\right) y_{xl} \leq y(k) \leq \left(1 + \frac{St}{100}\right) y_{xl}, \quad \forall k \geq k_{qd}$$

Thời gian quá độ được xác định bằng công thức: $T_{qd} = k_{qd} \cdot T$

- Số lần dao động n : số đỉnh nhọn hoặc số điểm cực trị.

Cách 2: Đánh giá chất lượng quá độ dựa vị trí cặp cực quyết định. Cách này chỉ cho kết quả gần đúng và chỉ áp dụng được khi chu kỳ lấy mẫu T đủ nhỏ. Khi biết cặp cực quyết định $z^* = re^{\pm j\varphi}$ của hệ rời rạc là dựa vào quan hệ $z = e^{Ts}$ để suy ra nghiệm s^* , từ đó tính được hệ số tắt ξ và tần số dao động tự nhiên ω_n , bằng các công thức:

$$\xi = \frac{-\ln r}{\sqrt{(\ln r)^2 + \varphi^2}} \quad \omega_n = \frac{1}{T} \sqrt{(\ln r)^2 + \varphi^2}$$

$$\sigma_{\max} \% = e^{-\left(\frac{\xi\pi}{\sqrt{1-\xi^2}}\right)}.100\%$$

$$T_{qd} = \frac{3}{\xi\omega_n} \text{ với } St\% = 2\%$$

$$T_{qd} = \frac{4}{\xi\omega_n} \text{ với } St\% = 5\%$$

1.6.2. Đánh giá chất lượng xác lập

a. Phương pháp thực nghiệm

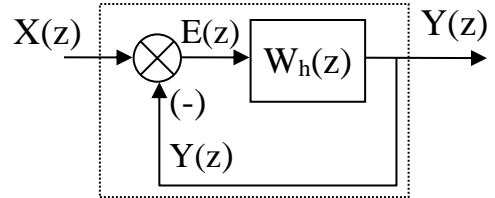
Phương pháp này được sử dụng với hệ điều khiển số đã có, đang làm việc. Khi cần hiệu chỉnh, sửa chữa định kỳ, tiến hành cho hệ làm việc ở chế độ xác lập sau đó dùng thiết bị đo đo lượng ra và đó chính là giá trị y_{xl} , tính sai lệch theo biểu thức

$$St\% = \frac{y_{xl} - y_{yc}}{y_{yc}} 100 = \frac{y_{yc} - y_{xl}}{y_{yc}} 100$$

Nếu $St\%$ lớn hơn giá trị cho phép thì phải tiến hành hiệu chỉnh, sau đó kiểm tra lại đến khi $St\%$ nhỏ hơn giá trị cho phép. Khi đó kết thúc hiệu chỉnh chuyển sang bảo dưỡng.

b. Phương pháp dùng cấu trúc hàm truyền

Phương pháp này dùng cho hệ điều khiển chưa có các thiết kế, với mục đích kiểm tra việc thiết kế. Hệ sau khi thiết kế sẽ có sơ đồ nguyên lý và thông số, từ đó ta tìm được cấu trúc và hàm truyền. Mặt khác với một hệ cần quan tâm đến sai lệch tĩnh thì hệ phải là hệ điều khiển giữ ổn định, do đó phải có phản hồi âm. Cấu trúc chuẩn của hệ như hình vẽ sau:



$$St\% = \frac{y_{xl} - y_{yc}}{y_{yc}} 100 = \frac{St}{y_{yc}} 100$$

St là sai lệch tuyệt đối ở chế độ xác lập.

Theo định lý giá trị cuối, ta có: $St = \lim_{t \rightarrow \infty} e(t) = \lim_{k \rightarrow \infty} e(kT) = \lim_{z \rightarrow 1} (1 - z^{-1})E(z)$. Đây là công thức tổng quát, có thể áp dụng cho mọi hệ rời rạc.

$$\begin{cases} E(z) = X(z) - Y(z) \\ Y(z) = E(z) \cdot W_h(z) \end{cases} \Rightarrow E(z) = \frac{X(z)}{1 + W_h(z)}$$

Từ cấu trúc ta có:

$$St = \lim_{z \rightarrow 1} \frac{(z-1)X(z)}{1 + W_h(z)}$$

$$W_h(z) = \frac{W_{HT}(z)}{1 - W_{HT}(z)} \Rightarrow St = \lim_{z \rightarrow 1} \frac{(z-1)X(z)}{1 + \frac{W_{HT}(z)}{1 - W_{HT}(z)}} = \lim_{z \rightarrow 1} (z-1)[1 - W_{HT}(z)]X(z)$$

Để xác định St% trong kỹ thuật cho phép tính gần đúng lượng ra ở chế độ xác lập:

$$\begin{aligned} y_{xl} \approx y_{yc} &= \lim_{t \rightarrow \infty} y(t) = \lim_{k \rightarrow \infty} y(kT) = \lim_{z \rightarrow 1} (z-1)Y(z) \\ &= \lim_{z \rightarrow 1} (z-1)X(z)W_{ht}(z) \end{aligned}$$

$$+ \text{Nếu } x(t) = A \cdot 1(t) \Rightarrow X(z) = \frac{Az}{z-1} \Rightarrow St = \lim_{z \rightarrow 1} z[1 - W_{HT}(z)]$$

$$\Rightarrow y_{xl} = \lim_{z \rightarrow 1} z \cdot W_{ht}(z)$$

$$St = \lim_{z \rightarrow 1} (1 - z^{-1}) \frac{1}{1 + W_h(z)} X(z)$$

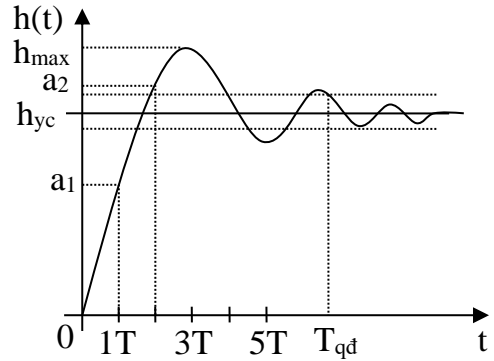
1.6.3. Đánh giá chất lượng quá độ

Ta thấy, theo khái niệm của lý thuyết điều khiển, 3 chỉ tiêu này có thể xác định dễ dàng. Nếu có đặc tính lượng ra $y(t)$ trong chế độ quá độ.

Mặt khác, để khảo sát hệ điều khiển ta sử dụng tín hiệu vào là tín hiệu có dạng bậc thang đơn vị $1(t)$. Vì vậy, lượng ra $y(t)$ sẽ là hàm quá độ $h(t)$.

Nếu tín hiệu vào $x(t)=1(t)$ thì đầu ra $y(t) = h(t)$. Do vậy các phương pháp đánh giá chất lượng hệ thống là các phương pháp đồ thị (vẽ đặc tính) $h(t)$.

$h(t)$ là đồ thị hoàn toàn vẽ được nếu biết các điểm $h(kT)$:



$$h(kT) = Z^{-1}[H(z)] = Z^{-1}\left[\frac{z}{z-1} W_{ht}(z)\right] = Z^{-1}\left[\frac{B(z)}{A(z)}\right]$$

$$\frac{B(z)}{A(z)} = a_0 z^0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_k z^{-k} + \dots$$

$$h_0(t) = a_0; h_1(t) = a_1; \dots h_k(t) = a_k \dots$$

Đưa các giá trị trên lên hệ tọa độ cách nhau chu kỳ T đã biết và nối lại ta được đặc tính $h(t)$.

$$\sigma_{\max} \% = \frac{h_{\max} - h_{yc}}{h_{yc}} 100 = \frac{\sigma_{\max}}{h_{yc}} 100$$

$$n=3 \text{ (đỉnh nhọn)}$$

1.7 Tổng hợp và hiệu chỉnh hệ điều khiển số

Với hệ điều khiển số sau khi xét ổn định, có thể hệ chưa ổn định. Sau khi đánh giá chất lượng có thể có hệ chưa đảm bảo chất lượng. Vì vậy ta phải tiến hành hiệu chỉnh để hệ chưa ổn định thành hệ ổn định, hệ chưa đảm bảo chất lượng thành hệ có chất lượng thỏa mãn các chỉ tiêu chất lượng đã đặt ra.

Đối tượng điều khiển trong thực tế thường là các thiết bị tương tự, do vậy để thành lập được hệ điều khiển số đáp ứng được các yêu cầu công nghệ đặt ra ta phải tiến hành tổng hợp hệ thống.

Để tổng hợp hệ thống ta có thể dùng các phương pháp:

- Tổng hợp hệ điều khiển số bằng phương pháp gián tiếp
- Tổng hợp hệ điều khiển số bằng phương pháp trực tiếp
- Tổng hợp hệ điều khiển số bằng thực nghiệm

Các phương pháp tổng hợp trên sẽ được đề cập ở các chương sau.

CHƯƠNG 2. THIẾT KẾ HỆ ĐIỀU KHIỂN SỐ

2.1. Cơ sở thiết kế hệ điều khiển số

Hệ điều khiển số được tổng hợp theo hai bước chủ yếu: trong miền tần số và trong không gian trạng thái. Tổng hợp trong miền tần số chủ yếu dựa vào mô tả động học của hệ tăng bằng các biến đổi Laplace và Fourier (còn gọi là phương pháp tổng hợp dùng kỹ thuật biến đổi). Hướng thứ hai là tổng hợp hệ điều khiển số trong không gian trạng thái.

Phương pháp dùng kỹ thuật biến đổi có các phương pháp gián tiếp (phương pháp tương tự) và phương pháp trực tiếp.

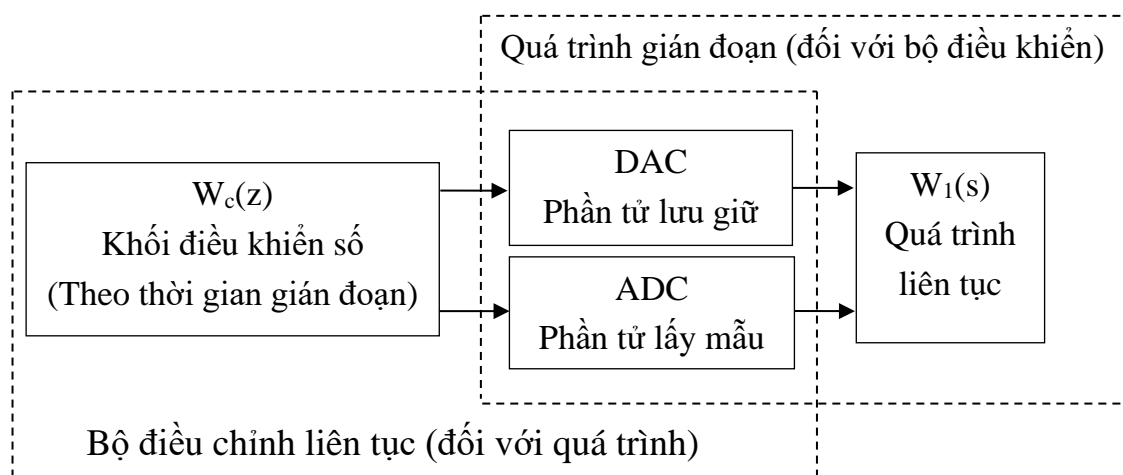
Ở phương pháp gián tiếp, một bộ điều khiển liên tục lý tưởng $W_c(s)$ được tổng hợp sau đó một tổ hợp “ADC - bộ điều khiển gián đoạn -DAC” được chọn sao cho tương ứng với $W_c(s)$ như ở hình 2.1. Phương pháp này được những người quen dùng điều khiển tương tự ưa chuộng vì chỉ cần biến đổi từ kỹ thuật tương tự sang số. Tuy nhiên việc gián đoạn hoá bộ phận điều khiển sẽ cho kết quả kém chính xác vì:

- Tín hiệu liên tục dạng bậc thang từ phần tử lưu giữ không thể tạo nên tín hiệu lý tưởng $u^*(t)$.
- Tín hiệu lý tưởng ấy phụ thuộc liên tục vào $y(t)$, còn bộ điều khiển số chỉ đo được $y(t)$ ở thời điểm lấy mẫu.

Tuy nhiên, nếu so với phổ của các tín hiệu đầu vào, đầu ra mà chọn tần số lượng tử hoá đủ lớn, có thể chọn được bộ phận điều khiển gián đoạn gần như $W_c(s)$.

Phương pháp chọn giản đơn nhất là theo hàm truyền của bộ điều chỉnh:

$$W_c(z) = W_c(s) \Big|_{s=(z-1)/T}$$



Hình 2.1. Mối quan hệ quá trình gián đoạn và liên tục

Phương pháp thứ hai là phương pháp biến đổi đơn ứng (biến đổi kép, biến đổi Tustin) duy trì được điều kiện ổn định của hàm truyền: nếu $W_c(s)$ ổn định thì $W_c(z)$ cũng ổn định do phép biến đổi đã chuyển miền bên trong đường tròn đơn vị ở mặt phẳng z sang nửa trái của mặt phẳng s . Tuy nhiên điều đó không có nghĩa là nếu $W_c(s)$ ổn định

được quá trình thì bộ điều khiển gián đoạn “ADC - $W_c(z)$ - DAC” cũng sẽ ổn định được quá trình. Do vậy sau khi chọn bộ phận điều khiển số cần đánh giá lại sai lệch và tính ổn định của hệ.

Ở phương pháp trực tiếp quá trình liên tục cùng với các phần tử lưu giữ và lấy mẫu được xem như một quá trình gián đoạn, tổng hợp trong miền z, cho phép khai thác tính năng mềm dẻo của máy tính mà phương pháp tương tự bị hạn chế.

Chọn tần số lấy mẫu

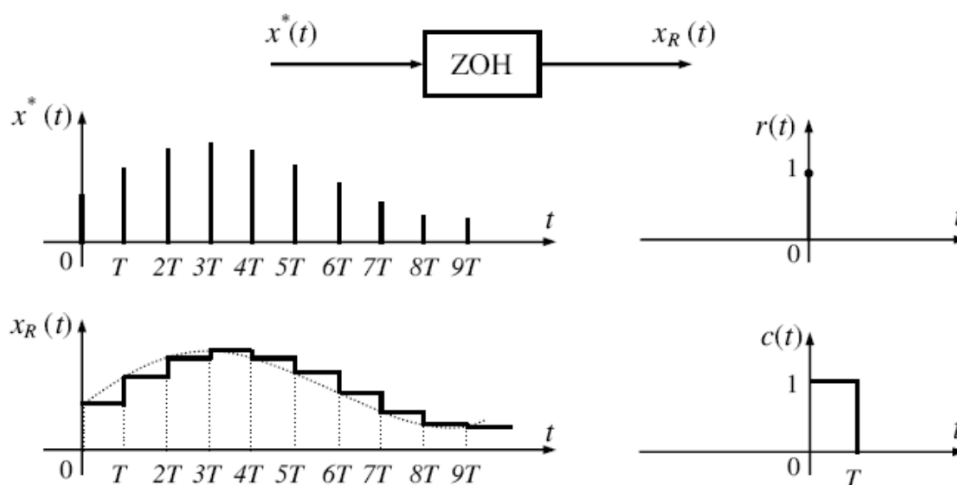
Chu kỳ lấy mẫu T ảnh hưởng rất lớn đến tính ổn định và chất lượng của hệ rời rạc. T càng lớn thì hệ càng dao động, độ quá điều chỉnh càng cao, thời gian quá độ càng lớn thậm chí mất ổn định. Vì vậy chọn chu kỳ lấy mẫu thích hợp có ý nghĩa rất lớn khi thiết kế hệ rời rạc. Định lý Shannon khẳng định tần số lấy mẫu chỉ cần lớn hơn 2 lần tần số cắt của hệ thống thì có thể phục hồi được dữ liệu mà không bị méo dạng, tuy nhiên tín hiệu chỉ không bị méo dạng nếu ta phục hồi dữ liệu bằng khâu giữ có dạng hàm $\sin(x)/x$. Trong các hệ thực thường phục hồi dữ liệu bằng khâu lưu giữ bậc 0 ZOH nên để đảm bảo chất lượng của hệ ta cần chọn tần số lấy mẫu lớn hơn 10 lần tần số cắt của hệ.

Định lý Shannon: Để có thể phục hồi dữ liệu sau khi lấy mẫu mà không bị méo dạng thì tần số lấy mẫu phải thỏa điều kiện:

$$f = \frac{1}{T} \geq 2f_c$$

trong đó là f_c tần số cắt của tín hiệu cần lấy mẫu

Trong các hệ thống điều khiển thực tế, nếu có thể bỏ qua được sai số lượng tử hóa thì các khâu chuyển đổi tương tự số A/D chính là các khâu lấy mẫu. Để chuyển đổi từ tín hiệu rời rạc thành tín hiệu liên tục, người ta sử dụng các khâu lưu giữ tín hiệu. Khâu giữ dữ liệu có nhiều dạng khác nhau, đơn giản nhất và được sử dụng nhiều nhất trong các hệ thống điều khiển rời rạc là khâu lưu giữ bậc 0 (Zero-Order Hold – ZOH).



Hình 2.2. Khâu lưu giữ bậc 0 ZOH

Nếu tín hiệu vào $r(t)$ của khâu ZOH là xung dirac thì tín hiệu ra $c(t)$ là xung vuông có độ rộng bằng T .

$$r(t) = \delta(t) \Rightarrow R(s) = 1$$

$$C(s) = L\{c(t)\} = L\{u(t) - u(t-T)\} = \frac{1}{s} - \frac{1}{s} e^{-Ts} = \frac{1 - e^{-Ts}}{s}$$

$$\text{Vậy hàm truyền của khâu ZOH là: } G_{ZOH}(s) = \frac{C(s)}{R(s)} = \frac{1 - e^{-Ts}}{s}$$

Trong các hệ thống điều khiển thực tế, nếu có thể bỏ qua được sai số lượng tử hóa thì các khâu chuyển đổi số/tương tự (D/A hoặc DAC) chính là các khâu lưu giữ bậc 0 (ZOH).

Thiết kế bộ điều khiển số là xác định hàm truyền $G_c(z)$ hoặc để hệ thống thỏa mãn yêu cầu về độ ổn định, chất lượng quá độ, sai số xác lập. Thực tế trong đa số trường hợp bộ điều khiển số là các thuật toán phần mềm chạy trên máy tính PC hoặc vi xử lý. Từ hàm truyền $G_c(z)$ của bộ điều khiển số hoặc hệ số phản hồi trạng thái K ta suy ra được phương trình sai phân mô tả quan hệ giữa vào ra của bộ điều khiển. Quan hệ này được sử dụng để lập trình phần mềm điều khiển chạy trên máy tính hoặc vi xử lý. Có nhiều phương pháp được sử dụng để thiết kế bộ điều khiển số, trong nội dung bài giảng này chỉ đề cập phương pháp thiết kế dùng quỹ đạo nghiệm số, phương pháp thiết kế bộ điều khiển sử dụng bộ điều khiển PID.

Lưu ý: Ở chương 1 của giáo trình này chúng ta sử dụng ký hiệu W để biểu thị hàm truyền của một phần tử hoặc hệ thống. Kể từ đây chúng ta sẽ sử dụng thêm ký hiệu G để biểu thị hàm truyền. Như vậy, G và W sẽ được sử dụng song song để biểu thị hàm truyền của một phần tử (khâu) hoặc hệ thống.

2.2. Phương pháp thiết kế gián tiếp

2.2.1. Thiết kế bộ điều chỉnh từ thiết bị tương tự

Để thực hiện tổng hợp bộ điều chỉnh ta xuất phát từ đối tượng và công nghệ đã biết, áp dụng các phương pháp tuyến tính, hiệu chỉnh tương tự:

- Phương pháp dùng đặc tính tần số logarit $L(\omega)$.
- Dùng phương pháp module tối ưu
- Dùng phương pháp module đối xứng

Từ đó tìm ra quy luật (cấu trúc và tham số) của bộ điều chỉnh. Sau đó tìm cách chuyển về không gian Z . Thực hiện biến đổi toán học ta tìm được quan hệ giữa tín hiệu ra và tín hiệu vào của bộ điều chỉnh theo toán tử z . Áp dụng tính chất của biến đổi Z ta tìm được phương trình sai phân mô tả quan hệ vào ra của bộ điều chỉnh. Đây chính là thuật toán để lập trình, sau đó dùng ngôn ngữ bất kỳ để thực hiện.

2.2.2. Sử dụng bộ điều chỉnh tổng quát PID

Với đối tượng cụ thể và công nghệ cụ thể bất kỳ ta dùng luật điều khiển tổng quát PID sẽ đáp ứng được thông qua việc điều chỉnh hàm lượng 3 thành phần thể hiện ở 3 tham số K_P , K_D , K_I . Để xác định 3 thông số này, hiện nay chưa có lý thuyết để xác định chính xác mà chúng thường được xác định bằng thực nghiệm. Do đó, ta sử dụng module PID số đã chế tạo sẵn đưa vào hệ và tiến hành hiệu chỉnh 3 tham số sau khi chạy thử. Nếu thỏa mãn yêu cầu công nghệ, ta ghi lại các thông số này

Nếu dùng vi xử lý hoặc máy tính, nhiệm vụ của ta là xác định thuật toán lập trình PID, sau đó lập chương trình nạp vào cho hệ.

$$u_{ra} = u_{dk} = K_P u_v(t) + K_I \int u_v(t) dt + K_D \frac{du_v(t)}{dt}$$

$$\Rightarrow U_{dk}(z) = Z[u_{dk}(t)] = U_v(z) W_{PID}(z)$$

$$= Z[K_P u_v(t)] + Z[K_I \int u_v(t) dt] + Z[K_D \frac{du_v(t)}{dt}]$$

$$Z[K_P u_v(t)] = K_P Z[u_v(t)] = K_P U_v(z)$$

$$y(t) = \int u_v(t) dt \Rightarrow y[kT] = y[(k-1)T] + u_v[kT].T$$

$$y[kT] - y[(k-1)T] = u_v[kT].T$$

$$Y(z)(1 - z^{-1}) = U_v(z).T \text{ hay } Y(z) = \frac{Tz}{z-1} U_v(z)$$

$$y(t) = \frac{du_v(t)}{dt} \rightarrow y[kT] = \frac{u_v[kT] - u_v[(k-1)T]}{T}$$

$$Y(z) = U_v(z) \frac{1 - z^{-1}}{T} = U_v(z) \frac{z-1}{zT}$$

$$Z[K_D \frac{du_v(t)}{dt}] = K_D Z[\frac{du_v(t)}{dt}] = \frac{K_D(z-1)}{Tz} U_v(z)$$

$$\Rightarrow U_{dk}(z) = [K_P + K_I \frac{z}{z-1} + K_D \frac{z-1}{Tz}] U_v(z)$$

$$= \frac{TK_P z(z-1) + K_I T z^2 + K_D (z-1)^2}{Tz(z-1)} U_v(z)$$

$$U_{dk}(z) = \frac{(TK_P + K_I T + K_D)z^2 - (2K_D + TK_P)z + K_D}{Tz^2 - Tz} U_v(z) = \frac{az^2 + bz + c}{Tz^2 - Tz} U_v(z)$$

$$\Rightarrow Tz^2 U_{dk}(z) - Tz U_{dk}(z) = az^2 U_v(z) + bz U_v(z) + c U_v(z)$$

$$U_{dk}(z) = z^{-1} U_{dk}(z) + \frac{a}{T} U_v(z) + \frac{b}{T} z^{-1} U_v(z) + \frac{c}{T} z^{-2} U_v(z)$$

Áp dụng tính chất của biến đổi z ta có:

$$u_{dk}(k) = u_{dk}(k-1) + \frac{a}{T}u_v(k) + \frac{b}{T}u_v(k-1) + \frac{c}{T}u_v(k-2)$$

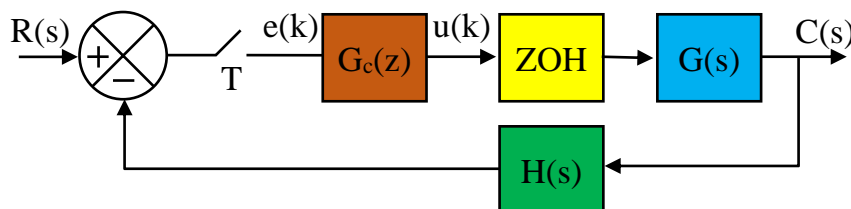
Đây chính là phương trình sai phân mô tả thuật toán lập trình.

2.3. Phương pháp thiết kế dựa vào quỹ đạo nghiệm số

Tương tự như hệ liên tục, đối với hệ rời rạc chúng ta cũng có khái niệm quỹ đạo nghiệm số (QĐNS). QĐNS là tập hợp tất cả các nghiệm của phương trình đặc trưng của hệ thống khi có một thông số nào đó trong hệ thay đổi từ $0 \rightarrow \infty$.

Vì dạng phương trình đặc trưng của hệ liên tục đã khảo sát ở chương 4 và phương trình đặc trưng là như nhau (chỉ thay biến s bằng biến z) nên qui tắc vẽ QĐNS là như nhau, chỉ khác ở qui tắc 8, thay vì đối với hệ liên tục ta tìm giao điểm của QĐNS với trục ảo thì đối với hệ rời rạc ta tìm giao điểm của QĐNS với đường tròn đơn vị.

Xét hệ điều khiển số có sơ đồ cấu trúc như hình vẽ sau:



Hình 2.3. Sơ đồ cấu trúc của hệ điều khiển số

Phương trình đặc trưng của hệ trước khi hiệu chỉnh: $1 + GH(z) = 0$

Phương trình đặc trưng của hệ sau khi hiệu chỉnh: $1 + G_c(z)GH(z) = 0$

Khâu hiệu chỉnh có dạng: $G_c(z) = K_c \frac{z + z_c}{z + p_c}$

Nếu $z_c < p_c$ thì khâu hiệu chỉnh là hiệu chỉnh sớm pha

Nếu $z_c > p_c$ thì khâu hiệu chỉnh là hiệu chỉnh trễ pha

Bài toán thiết kế đặt ra là chọn các giá trị K_c, z_c, p_c sao cho đáp ứng của hệ thống thoả mãn yêu cầu về chất lượng quá độ (chất lượng quá độ thể hiện qua vị trí của cặp cực quyết định) hoặc yêu cầu về chất lượng xác lập. Nguyên tắc thiết kế tương tự như thiết kế hệ liên tục, nghĩa là sử dụng khâu hiệu chỉnh $G_c(z)$ để sửa dạng QĐNS của hệ thống sao cho QĐNS đi qua cặp cực mong muốn.

2.3.1. Thiết kế bộ điều chỉnh sớm pha

Bộ điều chỉnh sớm pha có dạng: $G_c(z) = K_c \frac{z + z_c}{z + p_c}$ với $z_c < p_c$. Bài toán thiết kế đặt

ra là chọn các giá trị K_c, z_c, p_c sao cho đáp ứng của hệ thống thoả mãn yêu cầu về chất lượng quá độ (chất lượng quá độ thể hiện qua vị trí của cặp cực quyết định). Nguyên tắc thiết kế tương tự như thiết kế hệ liên tục, nghĩa là sử dụng khâu hiệu chỉnh $G_c(z)$ để sửa dạng QĐNS của hệ thống sao cho QĐNS đi qua cặp cực mong muốn. Sau đây là trình tự thiết kế:

Bước 1: Xác định cặp cực quyết định từ yêu cầu thiết kế về chất lượng của hệ thống trong quá trình quá độ:

Độ quá điều chỉnh, thời gian quá độ,... $\Rightarrow \begin{cases} \xi \\ \omega_n \end{cases} \Rightarrow s_{1,2}^* = -\xi\omega_n \pm j\omega_n\sqrt{1-\xi^2} \Rightarrow z_{1,2}^* = e^{Ts^*}$

$r = |z^*| = e^{-T\xi\omega_n}$; $\varphi = \angle z^* = T\omega_n\sqrt{1-\xi^2}$

Bước 2: Xác định góc pha cần bù để cặp cực quyết định nằm trên QĐNS của hệ thống sau khi hiệu chỉnh bằng công thức:

$$\varphi^* = -180^\circ + \sum_{i=1}^n \arg(z^* - p_i) - \sum_{i=1}^m \arg(z^* - z_i)$$

p_i, z_i là các cực và zero của $G(z)$ trước khi hiệu chỉnh

Dạng hình học của công thức trên là:

$$\varphi^* = -180^\circ + \sum \text{Góc từ các cực của } GH(z) \text{ đến cực } z^* - \sum \text{Góc từ các zero của } GH(z) \text{ đến cực } z^*$$

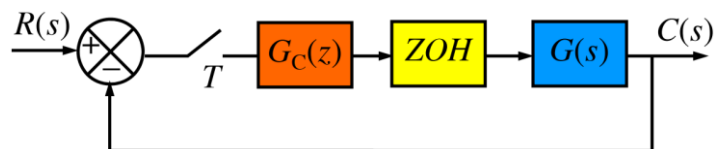
Bước 3: Xác định vị trí cực và zero của khâu hiệu chỉnh

Vẽ 2 nửa đường thẳng bất kỳ xuất phát từ cực quyết định z^* sao cho 2 nửa đường thẳng này tạo với nhau một góc bằng φ^* . Giao điểm của hai nửa đường thẳng này với trục thực là vị trí cực và zero của khâu hiệu chỉnh.

Đối với hệ rời rạc, người ta thường áp dụng phương pháp triệt tiêu nghiệm cực của hệ thống để chọn cực và zero của khâu hiệu chỉnh.

Bước 4: Tính K_C bằng cách áp dụng công thức: $|G_c(z)GH(z)|_{z=z^*} = 1$

Ví dụ: Cho hệ thống điều khiển số có sơ đồ cấu trúc



$$G(s) = \frac{10}{s(s+5)}; T = 0,1 \text{ (s)}$$

Hãy thiết kế khâu hiệu chỉnh sớm pha sao cho hệ thống sao khi hiệu chỉnh có cặp cực quyết định với $\xi = 0,707, \omega_n = 10 \text{ (rad/s)}$

Giải:

Phương trình đặc trưng của hệ trước khi hiệu chỉnh: $1 + G(z) = 0$

$$G(z) = Z\{G_{ZOH}(s)G(s)\} = Z\left\{\frac{1-e^{-Ts}}{s} \frac{K}{s(s+5)}\right\} = K(1-z^{-1})Z\left\{\frac{K}{s^2(s+5)}\right\}$$

$$= K\left(\frac{z-1}{z}\right)\left(\frac{Tz}{(z-1)^2} - \frac{(1-e^{-5T})z}{5(z-1)(z-e^{-5T})}\right) = K\left(\frac{T}{(z-1)} - \frac{1-e^{-5T}}{5(z-e^{-5T})}\right)$$

$$G(z) = \frac{0,21z + 0,18}{(z-1)(z-0,607)}$$

Cặp cực quyết định mong muốn: $z_{1,2}^* = re^{\pm j\varphi}$

$$r = e^{-T\xi\omega_n} = e^{-0,1 \times 0,707 \times 10} = 0,493;$$

$$\varphi = T\omega_n \sqrt{1 - \xi^2} = 0,1 \times 10 \sqrt{1 - 0,707^2} = 0,707$$

$$\Rightarrow z_{1,2}^* = 0,493 e^{\pm j0,707} = 0,375 \pm j0,32$$

Góc pha cần bù là:

$$\varphi^* = -180^\circ + \beta_1 + \beta_2 - \beta_3$$

$$= -180^\circ + \arg(0,375 + j0,32 - 1) + \arg(0,375 + j0,32 - 0,607) - \arg(0,375 + j0,32 + 0,857)$$

$$\Rightarrow \beta_1 = 152,9^\circ$$

$$\beta_2 = 125,9^\circ$$

$$\beta_3 = 14,6^\circ$$

$$\Rightarrow \varphi^* = -180^\circ + 152,9^\circ + 125,9^\circ - 14,6^\circ = 84^\circ$$

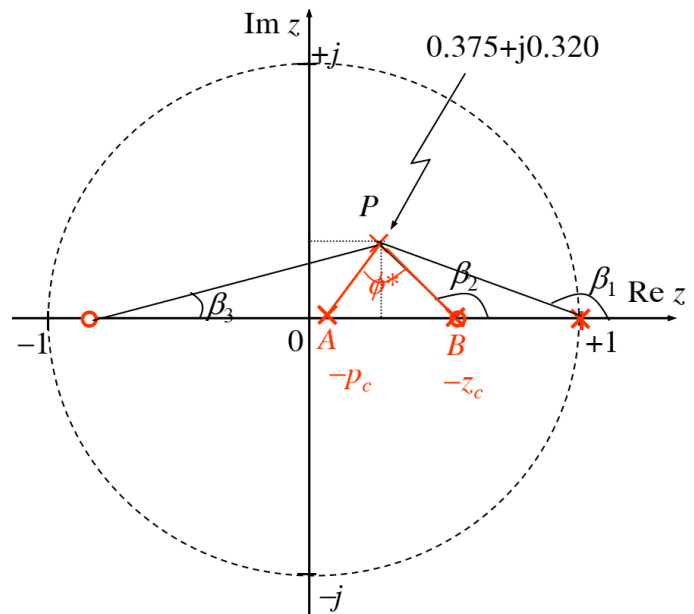
Chọn cực và zero của khâu hiệu chỉnh bằng phương pháp triệt tiêu nghiệm.

$$\text{Chọn } -z_c = 0,607$$

$$\Rightarrow z_c = -0,607$$

Tính cực của khâu hiệu chỉnh:

$$AB = PB \frac{\sin \varphi^*}{\sin PAB}$$



$$PB = \sqrt{(0,607 - 0,375)^2 + 0,32^2} = 0,388$$

$$PAB = \beta_2 - \varphi^* = 125,9^\circ - 84^\circ = 41,9^\circ$$

$$AB = 0,388 \frac{\sin(84^\circ)}{\sin(41,9^\circ)} = 0,578$$

$$\Rightarrow -p_c = OA = OB - AB = 0,029 \Rightarrow p_c = -0,029$$

$$\Rightarrow G_c(z) = K_c \frac{z - 0,607}{z - 0,029}$$

Tính K_C từ điều kiện: $|G_c(z)G(z)|_{z=z^*} = 1$

$$\Leftrightarrow \left| K_c \frac{z - 0,607}{z - 0,029} \frac{0,21z + 0,18}{(z - 1)(z - 0,607)} \right|_{z=0,375 + j0,32} = 1$$

$$\Leftrightarrow \left| K_c \frac{0,21(0,375 + j0,32) + 0,18}{(0,375 + j0,32 - 0,029)(0,375 + j0,32 - 1)} \right| = 1$$

$$K_c \frac{0,267}{0,471 \times 0,702} = 1 \Rightarrow K_c = \frac{0,471 \times 0,702}{0,267} = 1,24$$

Vậy $G_c(z) = 1,24 \frac{z - 0,607}{z - 0,029}$

Quỹ đạo nghiệm số của hệ thống trước khi hiệu chỉnh không qua điểm z^* , do đó hệ thống sẽ không bao giờ đạt được chất lượng đáp ứng quá độ như yêu cầu dù có thay đổi hệ số khuếch đại của hệ thống.

Bằng cách sử dụng khâu hiệu chỉnh sớm pha, quỹ đạo nghiệm số của hệ thống bị sửa dạng và qua điểm z^* , do đó bằng cách chọn hệ số khuếch đại thích hợp (bước 4) hệ thống sẽ có cặp cực quyết định như mong muốn do đó đáp ứng quá độ đạt yêu cầu thiết kế.

2.3.2. Thiết kế bộ điều khiển trễ pha

Bộ điều chỉnh trễ pha có dạng: $G_c(z) = K_c \frac{z + z_c}{z + p_c}$ với $z_c > p_c$. Bài toán thiết kế đặt ra là chọn các giá trị K_c, z_c, p_c sao cho suy giảm sai lệch tĩnh của hệ thống mà không ảnh hưởng đáng kể đến chất lượng quá độ của hệ.

Đặt $\beta = \frac{1 + p_c}{1 + z_c}$

Nguyên tắc thiết kế tương tự như thiết kế hệ liên tục, nghĩa là sử dụng khâu hiệu chỉnh $G_c(z)$ để sửa dạng QĐNS của hệ thống sao cho QĐNS đi qua cặp cực mong muốn. Sau đây là trình tự thiết kế:

Bước 1: Xác định yêu cầu thiết kế về sai số xác lập của hệ thống:

+ Nếu yêu cầu về sai số xác lập cho dưới dạng hệ số vị trí K_p^* thì $\beta = \frac{K_p}{K_p^*}$

K_p, K_p^* là hệ số vị trí trước và sau khi hiệu chỉnh (hệ số mong muốn)

+ Nếu yêu cầu về sai số xác lập cho dưới dạng hệ số tốc độ K_v^* thì $\beta = \frac{K_v}{K_v^*}$

K_v, K_v^* là hệ số tốc độ trước và sau khi hiệu chỉnh (hệ số mong muốn)

Bước 2: Chọn zero của khâu hiệu chỉnh rất gần điểm $+1$ để không làm ảnh hưởng đáng kể đến dạng QĐNS, suy ra:

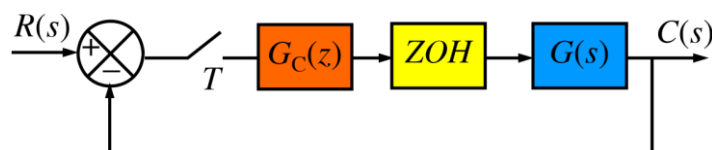
$$-z_c \approx 1 \Rightarrow z_c \approx -1$$

Chú ý điều kiện $|z_c| < 1$

Bước 3: Tính cực của khâu hiệu chỉnh: $p_c = -1 + \beta(1 + z_c)$

Bước 4: Tính K_c bằng cách áp dụng công thức: $|G_c(z)GH(z)|_{z=z^*} = 1$

Ví dụ: Cho hệ thống điều khiển số có sơ đồ cấu trúc



$$G(s) = \frac{10}{s(s+5)}; T = 0,1 \text{ (s)}$$

Hãy thiết kế khâu hiệu chỉnh trễ pha sao cho hệ thống sao khi hiệu chỉnh có hệ số vận tốc là $K_v^* = 100$.

Giải:

Phương trình đặc trưng của hệ trước khi hiệu chỉnh: $1 + G(z) = 0$

$$\begin{aligned} G(z) &= Z\{G_{ZOH}(s)G(s)\} = Z\left\{\frac{1-e^{-Ts}}{s} \frac{K}{s(s+5)}\right\} = K(1-z^{-1})Z\left\{\frac{K}{s^2(s+5)}\right\} \\ &= K\left(\frac{z-1}{z}\right)\left(\frac{Tz}{(z-1)^2} - \frac{(1-e^{-5T})z}{5(z-1)(z-e^{-5T})}\right) = K\left(\frac{T}{(z-1)} - \frac{1-e^{-5T}}{5(z-e^{-5T})}\right) \end{aligned}$$

$$G(z) = \frac{0,21z + 0,18}{(z-1)(z-0,607)}$$

Cặp cực quyết định của hệ thống trước khi hiệu chỉnh là nghiệm của phương trình: $1 + G(z) = 0$

$$\begin{aligned} 1 + \frac{0,21z + 0,18}{(z-1)(z-0,607)} &= 0 \\ \Leftrightarrow z^2 - 1,607z + 0,787 &= 0 \Leftrightarrow z_{1,2} = 0,699 \pm j0,547 \end{aligned}$$

Hệ số vận tốc của hệ trước khi hiệu chỉnh là:

$$K_v = \frac{1}{T} \lim_{z \rightarrow 1} (1-z^{-1})GH(z) = 10 \lim_{z \rightarrow 1} \left(\frac{z-1}{z} \frac{0,21z + 0,18}{(z-1)(z-0,607)}\right) = 9,9$$

$$\text{Do đó } \beta = \frac{K_v}{K_v^*} = \frac{9,9}{100} = 0,099$$

Chọn zero của khâu hiệu chỉnh rất gần điểm +1: $-z_c = 0,99 \Rightarrow z_c = -0,99$

Tính cực của khâu hiệu chỉnh:

$$\begin{aligned} p_c &= -1 + \beta(1+z_c) = -1 + 0,099(1-0,99) \\ \Rightarrow p_c &= 0,999 \end{aligned}$$

$$\Rightarrow G_c(z) = K_c \frac{z-0,99}{z-0,999}$$

Tính K_c từ điều kiện: $|G_c(z)G(z)|_{z=z^*} = 1$

$$\Leftrightarrow \left| K_c \frac{z-0,99}{z-0,999} \frac{0,21z + 0,18}{(z-1)(z-0,607)} \right|_{z=0,699+j0,547} = 1$$

$$\Leftrightarrow \left| K_c \frac{0,699 + j0,547 - 0,99}{0,699 + j0,547 - 0,999} \right| = 1$$

$$K_c = \frac{0,624}{0,619} = 1,008 \approx 1$$

$$\text{Vậy } G_c(z) = \frac{z-0,99}{z-0,999}$$

QĐNS của hệ thống trước và sau khi hiệu chỉnh gần giống nhau nên đáp ứng quá độ gần như không thay đổi.

2.4 Phương pháp thiết kế dựa vào đáp ứng tần số

Các phương pháp thiết kế theo đáp ứng tần số, đặc biệt là dựa trên đồ thị Bode được sử dụng rộng rãi khi thiết kế các hệ thống điều khiển tương tự. Chúng khai thác thực tế rằng đồ thị Bode của các hệ số khuếch đại mạch vòng hữu tỉ (chẳng hạn, những mạch vòng không có trễ) có thể được xấp xỉ bằng các đường thẳng. Hơn nữa, nếu hàm truyền là góc pha cực tiểu (minimum phase), góc pha có thể được xác định từ đồ thị biên độ, điều này cho phép đơn giản hóa việc thiết kế khâu bù để cung cấp các đặc tính ổn định cụ thể. Các thông số này thường được cho dưới dạng độ dự trữ góc pha và độ dự trữ hệ số khuếch đại. Tương tự các phương pháp thiết kế khác, phương pháp thiết kế này cũng đơn giản hơn nhiều do sự sẵn có của các gói CAD (Computer aided design).

Thật không may, hàm truyền rời rạc không phải là hàm truyền hữu tỉ theo $j\omega$ vì tần số được đưa vào thông qua phép thế $z = e^{j\omega T}$ vào hàm truyền theo z . Do đó, mất đi sự đơn giản của phương pháp dựa trên đồ thị Bode của hệ tương tự. Một giải pháp cho vấn đề này là chuyển đổi song tuyến tính (bilinear) mặt phẳng z thành một mặt phẳng mới gọi là mặt phẳng w , trong đó hàm truyền tương ứng là hàm hữu tỉ và phép xấp xỉ Bode là có giá trị. Vì mục đích này, phép biến đổi song tuyến tính được sử dụng

$$w = c \frac{z-1}{z+1} \quad (1)$$

Trong đó $c=2/T$

Phép biến đổi (1) ánh xạ các điểm trong mặt phẳng bên trái LHP (Left-hand plan) vào các điểm bên trong vòng tròn đơn vị. Để chuyển đổi bên trong của vòng tròn đơn vị thành LHP, ta sử dụng phép biến đổi song tuyến tính ngược

$$z = \frac{1 + w \frac{T}{2}}{1 - w \frac{T}{2}} \quad (2)$$

Phép biến đổi ngược chuyển đổi hàm truyền của hệ từ mặt phẳng z thành mặt phẳng w . Mặt phẳng w là một mặt phẳng phức có phần ảo được biểu thị bằng v . Để biểu diễn quan hệ giữa tần số ω trong mặt phẳng s và tần số v trong mặt phẳng w , đặt $s=j\omega$ và $z = e^{j\omega T}$. Thay vào (1) ta có:

$$w = jvc = j \frac{2}{T} \operatorname{tg} \frac{\omega T}{2} \quad (3)$$

Từ (3) ta thấy vì ω thay đổi trong khoảng $[0, \pi/T]$, z sẽ di chuyển trên vòng tròn đơn vị, v đi từ 0 đến ∞ . Điều này ngụ ý rằng có độ méo hoặc sai lệch về tỷ lệ tần số giữa w và ω . Độ méo này là đáng kể, đặc biệt là ở tần số cao. Tuy nhiên, nếu $\omega \ll \omega_s/2 = \pi/T$, từ (3) ta có $\omega=v$ và méo tần số có thể bỏ qua.

Ngoài vấn đề méo tần, hàm truyền sau chuyển đổi $G(w)$ có hai đặc tính có thể làm cho việc thiết kế trở nên phức tạp:

+ Hàm truyền sẽ luôn thiếu điểm không (ví dụ, số điểm cực bằng số điểm không)
 + Phép chuyển đổi song tuyến tính (1) có thể đưa ra các điểm không ở RHP và dẫn đến một hệ thống pha không cực tiểu

Các hệ thống pha không cực tiểu hạn chế đặc tính có thể đạt được. Ví dụ, vì các nhánh quỹ đạo nghiệm bắt đầu ở điểm cực của mạch vòng hở và kết thúc ở các điểm không, sự xuất hiện của các điểm không ở mặt phẳng bên phải sẽ hạn chế giải ổn định của hệ số khuếch đại K. Vì vậy, nỗ lực làm giảm sai lệch tĩnh và tăng tốc độ đáp ứng bằng cách tăng hệ số khuếch đại của hệ thống sẽ dẫn đến mất ổn định. Những hạn chế này tạo ra các trở ngại cho việc thiết kế mặt phẳng w.

Thủ tục 1. Các bước thiết kế bộ điều khiển trong mặt phẳng w

1. Chọn chu kỳ cắt mẫu và có được hàm truyền $G_{ZAS}(z)$ của quá trình rời rạc hóa
2. Chuyển đổi $G_{ZAS}(z)$ thành $G(w)$ sử dụng (2)
3. Vẽ đồ thị Bode của $G(j\omega)$ và sử dụng phương pháp đáp ứng tần số tương tự để thiết kế bộ điều khiển $C(w)$ thỏa mãn các thông số của miền tần số.
4. Chuyển đổi bộ điều khiển trở lại mặt phẳng z bằng (1), vì vậy xác định được $C(z)$
5. Kiểm tra các đặc tính nhận được là thỏa mãn.

Trong bài toán thiết kế bộ điều khiển, các thông số kỹ thuật thường được cho dưới dạng đáp ứng quá độ của hệ thống kín như thời gian quá độ và độ quá điều chỉnh. Độ quá điều chỉnh tỷ số tắt dần ξ được sử dụng cùng với thời gian quá độ để có được tần số tự nhiên ω_n . Vì vậy, ta cần nhận được các thông số của đáp ứng tần số từ ξ và ω_n theo thủ tục 1.

Quan hệ giữa các chỉ tiêu của miền thời gian và miền tần số nói chung là khá phức tạp. Tuy nhiên, quan hệ này sẽ đơn giản hơn nhiều nếu hệ thống kín có thể được làm gần đúng về dạng hàm truyền bậc hai

$$T(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (4)$$

Hệ số khuếch đại tương ứng với phản hồi đơn vị được cho bởi

$$L(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s} \quad (5)$$

Thay $s=j\omega$ nhằm có được đáp ứng tần số tương ứng và cân bằng bình phương biên độ với 1 ta có:

$$|L(j\omega)|^2 = \frac{1}{\left(\frac{\omega}{\omega_n}\right)^4 + 4\xi^2\left(\frac{\omega}{\omega_n}\right)^2} = 1 \quad (6)$$

Biên độ của hệ số khuếch đại mạch vòng hở cũng như bình phương của nó là đồng nhất ở tần số cắt. Đối với trường hợp hệ bậc hai không dao động, ta có quan hệ

$$\omega_{gc} = \omega_n [\sqrt{4\xi^4 + 1 - 2\xi^2}]^{1/2} \quad (7)$$

Tiếp theo ta xét độ dự trữ góc pha và nhận được

$$PM = 180^\circ + \angle G(j\omega_{gc}) = \text{tg}^{-1} \left(\frac{2\xi}{[\sqrt{4\xi^4 + 1 - 2\xi^2}]^{1/2}} \right)$$

Biểu thức trên có thể xấp xỉ bởi $PM \approx 100\xi$ (8)

Các phương trình (6) và (7) cung cấp các phép chuyển đổi ta cần để có được các thông số kỹ thuật trong miền tần số từ các thông số của đáp ứng quá độ. Cùng với thủ tục 1, các phương trình này cho phép ta thiết kế hệ điều khiển số sử dụng phương pháp mặt phẳng w .

Ví dụ 1: Cho quá trình tương tự có hàm truyền $G(s) = \frac{1}{s+1}$. Hãy chuyển đổi $G_{ZAS}(z)$ tương ứng thành mặt phẳng w cho trường hợp $T=0.1$ và $T=0.01$. Đánh giá vai trò của chu kỳ lấy mẫu bằng cách phân tích đồ thị Bode tương ứng.

Khi $T=0.1$ ta có:

$$G_{ZAS}(z) = \frac{0.09516}{z - 0.9048}$$

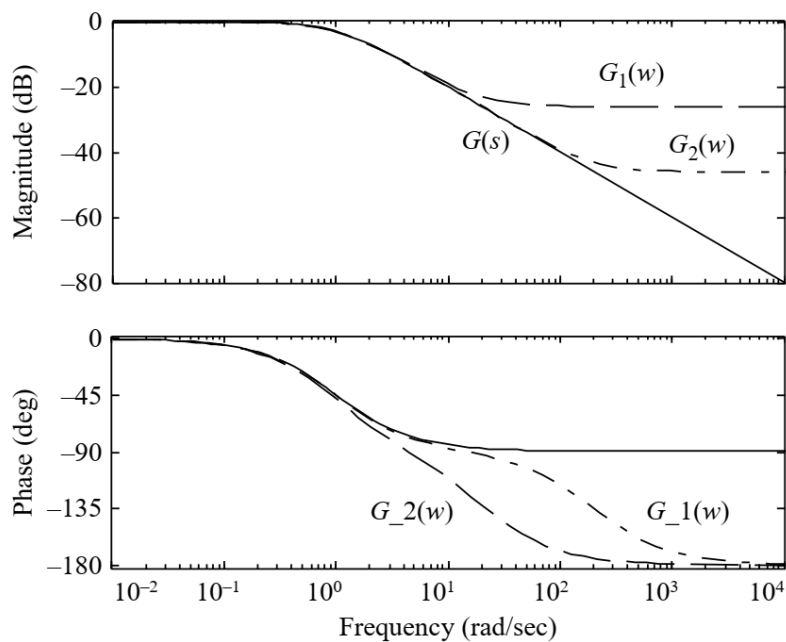
Áp dụng $w = c \frac{z-1}{z+1}$ với $c=2/T$ ta được $G_1(w) = \frac{-0.05w+1}{w+1}$

Khi $T=0.01$ ta có: $G_{ZAS}(z) = \frac{0.00995}{z - 0.99}$

Áp dụng lại $w = c \frac{z-1}{z+1}$ với $c=2/T$ ta được $G_2(w) = \frac{-0.005w+1}{w+1}$

Lưu ý: có thể nhận được $G(w)$ bằng cách sử dụng lệnh Matlab:

`>>Gw=d2c(Gzas,'tustin')`



Hình 2.4. Đồ thị Bode của ví dụ 1

Các đồ thị Bode của $G(s)$, $G_1(w)$, $G_2(w)$ được biểu diễn trên hình 1. Với cả hai chu kỳ lấy mẫu, điểm cực trong mặt phẳng w có vị trí giống như điểm cực trong mặt

phẳng s . Tuy nhiên, cả $G_1(w)$ và $G_2(w)$ đều có một điểm không trong khi $G(s)$ thì không. Điều này dẫn đến sự khác biệt rất lớn giữa đáp ứng tần số của hệ thống tương tự và đáp ứng tần số của hệ thống số ở vùng tần số cao. Tuy nhiên, ảnh hưởng của điểm không đến động học của hệ thống càng đáng kể khi chu kỳ lấy mẫu càng nhỏ. Lưu ý rằng cả chu kỳ lấy mẫu và độ méo tần đều được bỏ qua ở vùng tần số thấp. Đối với cả hai hệ thống, hệ số khuếch đại khi w tiến đến 0 là đồng nhất, giống như hệ số khuếch đại một chiều của hệ tương tự. Điều này đúng cho việc chọn $c=2/T$ trong biểu thức (1). Các lựa chọn có thể khác không được xét vì chúng không mang lại hệ số khuếch đại một chiều giống nhau.

Ví dụ 2: Xét hệ thống điều tốc động cơ một chiều trong đó đối tượng tương tự có hàm truyền

$$G(s) = \frac{1}{(s+1)(s+10)}$$

Thiết kế bộ điều khiển số bằng cách sử dụng phương pháp đáp ứng tần số để sai lệch tĩnh bằng không khi tín hiệu vào dạng bậc thang đơn vị, quá điều chỉnh nhỏ hơn 10%, thời gian quá độ khoảng 1s.

Từ những thông số đã cho, ta thấy rằng bộ điều khiển trong mặt phẳng w phải có một điểm cực ở góc tọa độ. Đối với quá điều chỉnh 10%, ta tính toán hệ số tắt dần như sau

$$\xi = \frac{|\ln(0.1)|}{\sqrt{|\ln(0.1)|^2 + \pi^2}} \approx 0.6$$

Sử dụng biểu thức xấp xỉ (8), dự trữ góc pha bằng khoảng 100 lần hệ số tắt dần của hệ thống kín, và dự trữ góc pha yêu cầu là khoảng 60° . Đối với thời gian quá độ khoảng 1s, ta tính toán tần số dao động tự nhiên

$$\omega_n = \frac{4}{\xi T_s} \approx 6.7 \text{ rad/s}$$

Sử dụng biểu thức (7) ta được tần số cắt $\omega_{gc} = 4.8 \text{ rad/s}$.

Chu kỳ lấy mẫu thích hợp cho đặc tính động học đã chọn là $T=0.02\text{s}$ (Xem thêm Ví dụ 6.9 sách Digital Control Engineering analysis and design). Quá trình rời rạc hóa được xác định như sau

$$G_{ZAS}(z) = (1-z^{-1})Z\left\{\frac{G(s)}{s}\right\} = 1.8604 \times 10^{-4} \frac{z+0.9293}{(z-0.8187)(z-0.9802)}$$

Sử dụng biểu thức (2) ta nhận được hàm truyền trong mặt phẳng w

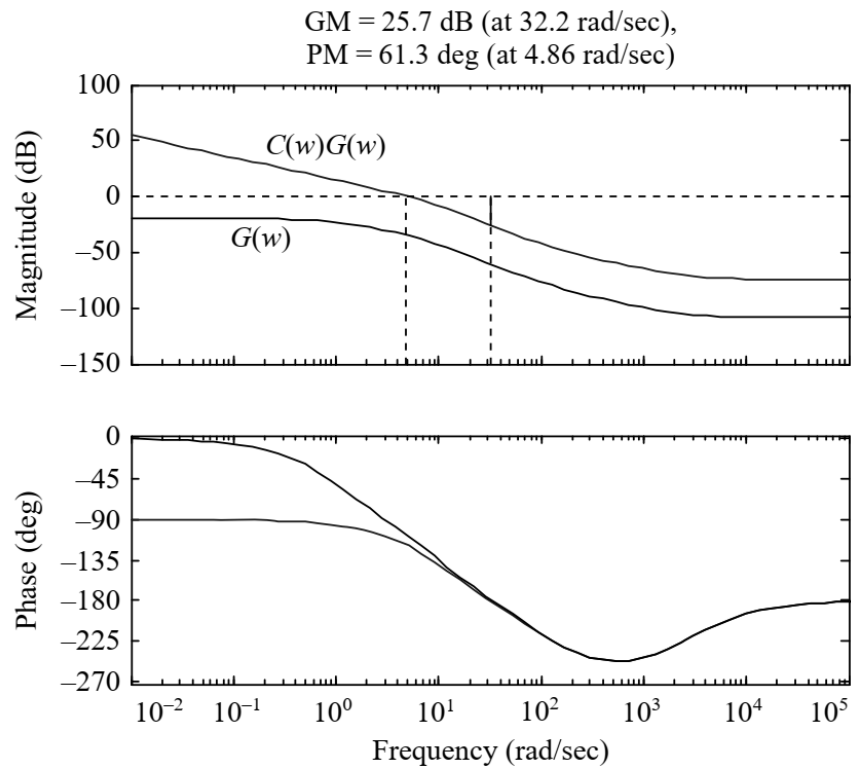
$$G(w) = \frac{-3.65 \cdot 10^{-6} (w+2729)(w-100)}{(w+9.967)(w+1)}$$

Lưu ý rằng hai điểm không bổ sung thêm (đối với $G(s)$) không ảnh hưởng đáng kể đến động học của hệ thống trong dải tần quan tâm cho thiết kế. Hai điểm cực gần như có cùng vị trí với các điểm cực của $G(s)$. Thiết kế đơn giản nhất đáp ứng được các

Chỉ tiêu kỹ thuật mong muốn là chèn thêm một điểm cực vào hệ ban đầu, để loại bỏ điểm cực chi phối (chiếm ưu thế) ở -1 và để tăng hệ số khuếch đại đến khi đạt được tần số cắt yêu cầu. Do vậy, hàm truyền của bộ điều khiển là:

$$C(w) = 54 \frac{w+1}{w}$$

Đồ thị Bode của hàm truyền hở $C(w)G(w)$ cùng với đồ thị Bode của $G(w)$ được biểu diễn trên hình 2. Hình 2 cũng cho thấy độ dự trữ biên độ và góc pha.



Hình 2.5. Đồ thị Bode của $C(w)G(w)$ và $G(w)$

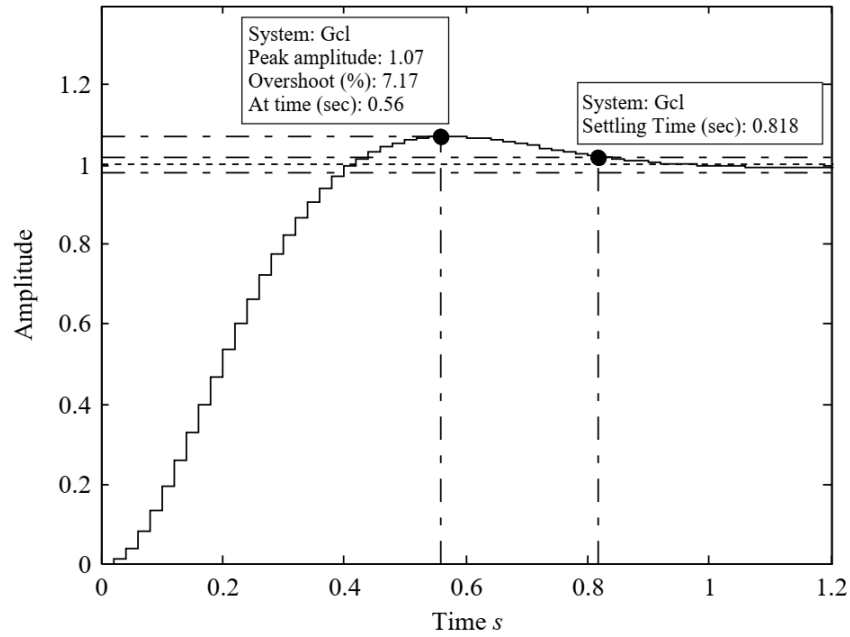
Bằng cách chuyển đổi bộ điều khiển trở lại mặt phẳng z sử dụng biểu thức (1), ta được:

$$C(z) = \frac{54.54z - 54.46}{z - 1}$$

Hàm truyền $C(z)$ cũng có thể đạt được bằng cách sử dụng lệnh Matlab sau:

```
>>Cz=c2d(Cw,0.02,'matched')
```

Đáp ứng quá độ của hệ thống kín đã rời rạc được biểu diễn ở hình 3 và rõ ràng đáp ứng được các thông số kỹ thuật thiết kế.



Hình 2.6. Đáp ứng quá độ của hệ kín

Ví dụ 3. Xét hệ điều khiển tốc độ động cơ một chiều với hàm truyền

$$G(s) = \frac{1}{(s+1)(s+3)}$$

Thiết kế một bộ điều khiển số sử dụng phương pháp đáp ứng tần số nhằm đạt được sai lệch tĩnh bằng không khi tín hiệu vào dạng bậc thang đơn vị, quá điều chỉnh nhỏ hơn 10%, thời gian quá độ khoảng 1s. Chu kỳ lấy mẫu $T=0.2s$.

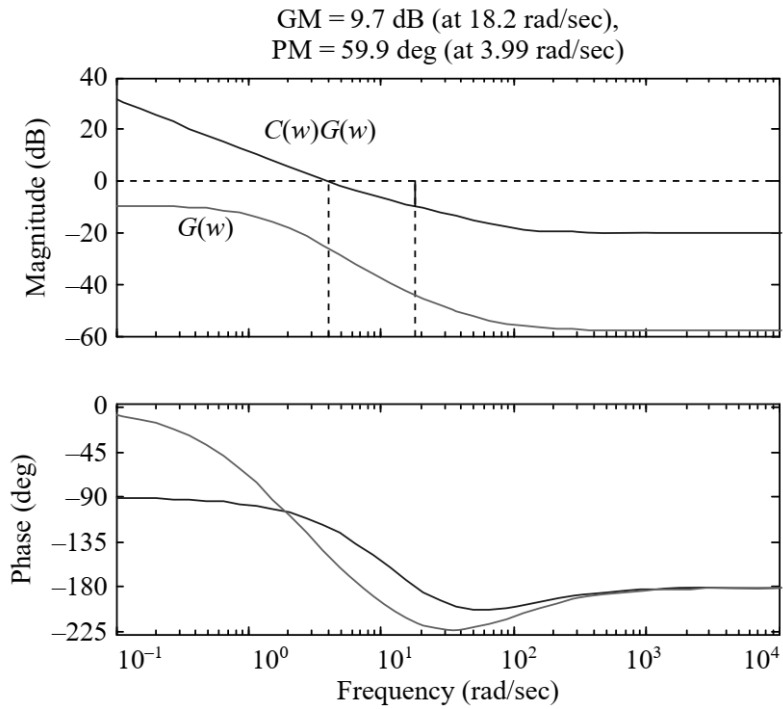
Như ở ví dụ 2, chỉ tiêu về sai lệch tĩnh đã cho yêu cầu bộ điều khiển với một cực ở gốc tọa độ, chỉ tiêu về quá điều chỉnh yêu cầu dự trữ góc pha vào khoảng 60° và thời gian quá độ mang lại tần số cắt vào khoảng 5 rad/s. Hàm truyền của hệ thống với DAC và ADC:

$$G_{ZAS}(z) = (1-z^{-1})Z\left\{\frac{G(s)}{s}\right\} = 0.015437 \frac{z+0.7661}{(z-0.8187)(z-0.5488)}$$

Chuyển đổi đến mặt phẳng w sử dụng (2) ta được

$$G(w) = \frac{-12.819 \cdot 10^{-4} (w+75.5)(w-10)}{(w+2.913)(w+0.9967)}$$

Lại lưu ý rằng cả hai điểm cực gần như là có cùng vị trí như là các cực của $G(s)$. Ở đây điểm không ở mặt phẳng bên phải $w=10$ phải được xem xét trong thiết kế vì tần số cắt yêu cầu là khoảng 5 rad/s. Nhằm đạt được dự trữ góc pha yêu cầu cả hai điểm cực phải được loại bỏ bằng hai điểm không của bộ điều khiển. Đối với bộ điều khiển có thể thực hiện ta cần ít nhất hai điểm cực của bộ điều khiển. Bổ sung thêm điểm cực vào gốc tọa độ, ta chọn một điểm cực của bộ điều khiển ở vùng cao tần vì vậy không ảnh hưởng đến đáp ứng tần số ở lân cận của tần số cắt. Tiếp theo, ta điều chỉnh hệ số khuếch đại của hệ thống nhằm đạt được các chỉ tiêu kỹ thuật mong muốn.



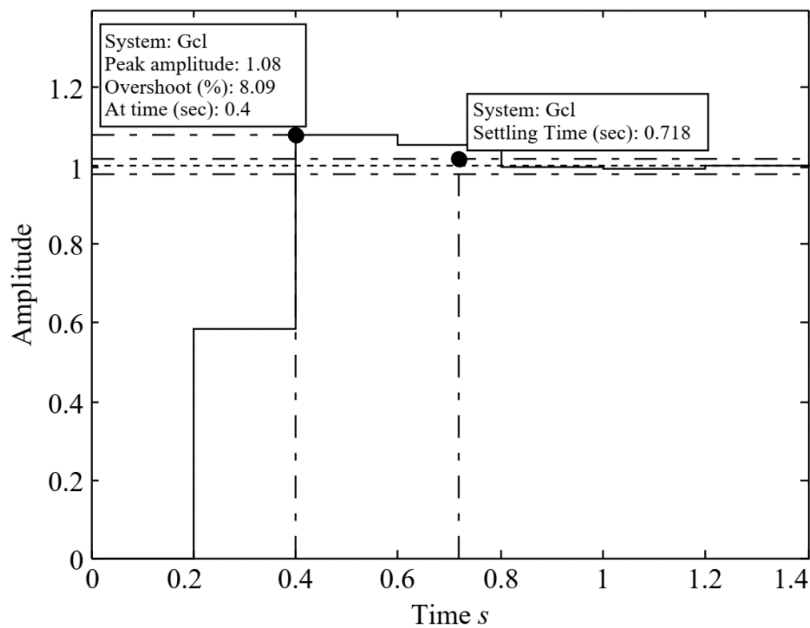
Hình 2.7. Đồ thị Bode của $C(w)G(w)$ và $G(w)$

Như là một nỗ lực ban đầu, ta chọn điểm cực cao tần ở $w = -20$ và tăng hệ số khuếch đại tới 78 đối với tần số cắt 4 rad/s với độ dự trữ góc pha 60° . Vì vậy, bộ điều khiển là

$$C(w) = 78 \frac{(w + 2.913)(w + 0.9967)}{w(w + 20)}$$

Đồ thị Bode của hệ hở tương ứng được biểu diễn trên hình 4 cùng với đồ thị Bode của $G(w)$. Chuyển đổi bộ điều khiển trở lại mặt phẳng z sử dụng (1), ta được

$$C(z) = C(w) \Big|_{w = \frac{2}{T} \frac{z-1}{z+1}} = \frac{36.9201z^2 - 50.4902z + 16.5897}{(z-1)(z-0.3333)}$$



Hình 2.8. Đáp ứng quá độ của hệ kín

2.5 Phương pháp thiết kế bằng thực nghiệm

Phương pháp thiết kế bằng thực nghiệm được sử dụng để chỉnh định thông số bộ điều khiển khi lắp đặt hệ thống điều khiển số nhằm tìm ra bộ thông số phù hợp nhất với đối tượng điều khiển của hệ thống thực. Để thực hiện Phương pháp thiết kế bằng thực nghiệm cần có các thiết bị:

- + Bộ điều khiển vạn năng PID số
- + Thiết bị đo ghi đại lượng vật lý đầu ra tốc độ cao

Quy trình thực hiện chỉnh định thông số:

- + Cho hệ thống chạy ở chế độ không tải.
- Đặt $K_I=0$, $K_D=0$
- Tăng dần K_p và quan sát trên thiết bị đo đến khi có hiện tượng dao động nhiều (gần mất ổn định) thì giảm K_p một lượng.
- Nếu sai lệch tĩnh chưa thỏa mãn công nghệ thì đưa K_I vào và tăng dần cho đến khi sai lệch tĩnh thỏa mãn yêu cầu công nghệ.
- Nếu thời gian quá độ chưa thỏa mãn yêu cầu công nghệ thì đưa thêm K_D và tăng dần với lượng nhỏ (do K_D dễ làm hệ mất ổn định) đến khi đặc tính thỏa mãn công nghệ thì chuyển sang chạy ở chế độ có tải 50% tải định mức.

+ Cho hệ thống chạy ở chế độ có tải với mức tải 50% tải định mức.

Lặp lại quy trình giống như chế độ không tải cho đến khi đạt được yêu cầu công nghệ thì chuyển sang chạy ở chế độ tải 100%.

+ Cho hệ thống chạy ở chế độ có tải với mức tải 100% tải định mức.

- Điều chỉnh lượng nhỏ các thông số cho đến khi thỏa mãn yêu cầu thì dừng lại và ghi lại các thông số và đặc tính tương ứng. Tiến hành trên 10 lần
- Thực hiện so sánh các đặc tính và lấy bộ thông số thỏa mãn yêu cầu công nghệ và có chất lượng tốt nhất.

Chốt thông số và đặt password.

CHƯƠNG 3: THIẾT KẾ ĐIỀU KHIỂN SỐ CHO HỆ THỰC

3.1 HỆ THỐNG ĐIỀU KHIỂN SỐ SỬ DỤNG VI XỬ LÝ/VI ĐIỀU KHIỂN

Ngày nay vi xử lý/vi điều khiển ($\mu P/\mu C$) được sử dụng rộng rãi trong các ứng dụng thực tiễn bởi tính linh hoạt, dễ triển khai và giá thành rẻ của nó. Nắm bắt được vai trò và xu hướng sử dụng $\mu P/\mu C$ trong tương lai, các công ty đã cho ra đời hàng loạt các mẫu $\mu P/\mu C$ tiên tiến có cấu hình mạnh: tốc độ cao, bộ nhớ lớn, tích hợp nhiều chức năng trên phiến. Đi kèm với đó là các công cụ, trình biên dịch ngày càng đơn giản thuận tiện cho người sử dụng. Chính những yếu tố trên đã góp phần phổ cập hóa các hệ thống điều khiển số từ đơn giản đến phức tạp trong các hệ thống công nghiệp.

3.2. MỘT SỐ VẤN ĐỀ KỸ THUẬT KHI THỰC HIỆN HỆ ĐIỀU KHIỂN SỐ VỚI VI XỬ LÝ/VI ĐIỀU KHIỂN

Đối với hệ thống điều khiển số khi thực hiện trên $\mu P/\mu C$, chúng ta cần chú ý một số điểm sau:

- Ảnh hưởng của giá trị chu kỳ lấy mẫu đến đặc tính hệ thống.
- Hệ thống thực khác với hệ thống lý thuyết ở điểm tín hiệu và năng lượng của hệ thống bị giới hạn về biên độ. Do vậy cần phải có khâu giới hạn tín hiệu đặt vào các điểm phù hợp trong sơ đồ cấu trúc hệ thống.
- Ảnh hưởng của các khâu A/D, D/A lên hệ thống bao gồm các vấn đề: lượng tử hóa và sai số do lượng tử hóa, đặc tính giới hạn của các khâu biến đổi.
- Đặc tính vật lý của đối tượng, bản chất vật lý của các dòng tín hiệu tại các điểm trong cấu trúc hệ thống để từ đó thực hiện phép lượng tử hóa trên sơ đồ mô phỏng cũng hệ thực một cách phù hợp.
- Độ chính xác của cảm biến.
- Các phép tính trong luật điều khiển số thường được quy về kiểu tính số nguyên. Đây là cách tính toán phổ biến trong các hệ thống nhúng có tài nguyên hữu hạn. Các phép tính số nguyên sẽ tạo ra sai số tính toán và xuất hiện hiện tượng tràn số liệu trong trường hợp nhất định.

3.3. QUY TRÌNH THỰC HIỆN HỆ THỐNG ĐIỀU KHIỂN SỐ

Để thực hiện một hệ thống điều khiển số trên $\mu P/\mu C$, người thiết kế thường phải thực hiện một số bước quy trình thiết kế như sau:

Bước 1: Dựa vào yêu cầu của bài toán thiết kế điều khiển để xây dựng và phác thảo sơ đồ cấu trúc và các thành phần, đối tượng, thiết bị cấu thành hệ thống.

Bước 2: Xây dựng các mô hình toán học từng phần tử trong hệ thống từ đó xác định được mô hình toán học của toàn hệ thống dưới dạng phương trình trạng thái hoặc hàm truyền đạt.

Bước 3: Lựa chọn chu kỳ lấy mẫu và rời rạc hóa hệ thống (nếu thiết kế theo phương pháp rời rạc).

Bước 4: Sử dụng lý thuyết điều khiển để xác định bộ điều khiển một cách phù hợp với yêu cầu điều khiển.

Bước 5: Thực hiện mô phỏng hệ thống trên phần mềm chuyên dụng để hiệu chỉnh tham số bộ điều khiển cho phù hợp. Sau khi thực hiện mô phỏng,

Bước 6: Phỏng thực hệ thống với phần mềm và phần cứng để hiệu chỉnh.

Bước 7: Thiết kế phần cứng, phần mềm hệ thống trên đơn vị tính toán số (vi xử lý, vi điều khiển, DSP, PLC...).

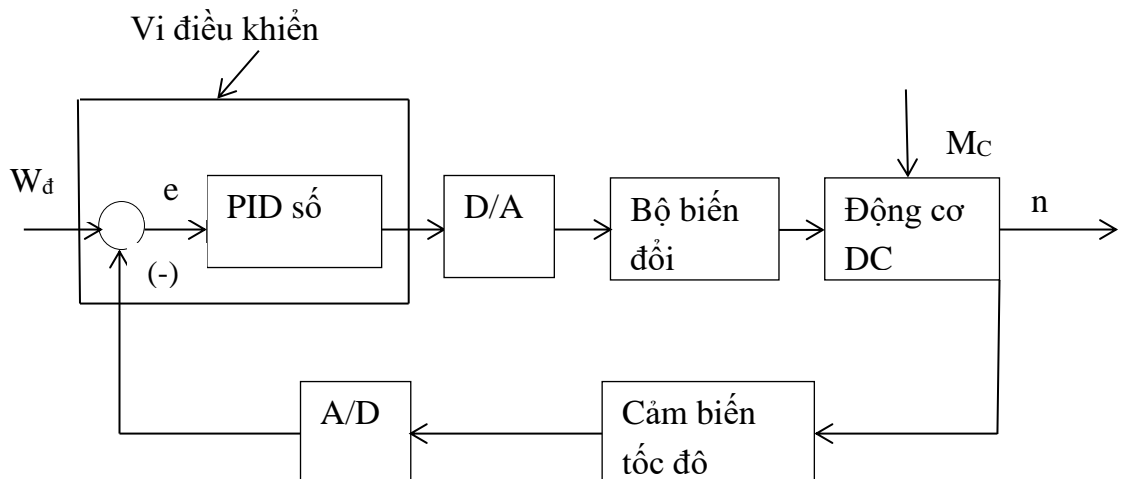
Bước 8: Chạy thử nghiệm và hiệu chỉnh hệ thống thực.

3.4. HỆ TRUYỀN ĐỘNG ĐIỆN

3.4.1 Các thiết bị cơ bản trong hệ điều khiển số truyền động điện

Sơ đồ khối chức năng hệ điều khiển số động cơ một chiều dùng vi xử lý/ vi điều khiển như hình 3.1.

Hệ thống điều khiển lấy giá trị đặt W_d do người sử dụng thiết lập từ bàn phím. Giá trị thực tế của động cơ đo được là tốc độ thông qua cảm biến tốc độ. Sau đó tín hiệu tương tự này sẽ được chuyển bộ chuyển đổi A/D trước khi đưa vào bộ điều khiển.



Hình 3.1: Sơ đồ khối hệ điều khiển số động cơ một chiều

Bộ điều khiển sẽ tính sai lệch e giữa giá trị đặt và giá trị đo rồi tính toán đầu ra của bộ điều khiển theo luật PID để xuất tín hiệu điều khiển đối tượng. Hệ thống với

thuật toán luật điều khiển PID số sẽ có xu hướng luôn đưa sai lệch e về giá trị 0, tức là giá trị đạt được sau một thời gian sẽ bằng giá trị đặt. Do đó, tốc độ của động cơ luôn ổn định.

***Cơ sở nguyên lý để thực hiện điều khiển động cơ:**

-Thực hiện điều khiển động cơ bằng bộ điều khiển PID số thiết kế và được cài đặt trên vi điều khiển.

- Sử dụng cảm biến tốc độ để phản hồi tốc độ động cơ.
- Giao tiếp thông qua các modul trên mạch điều khiển để hiển thị đặc tính.
- Trên mạch có các khối hiển thị để hiển thị các chế độ hoạt động của động cơ cùng các thông số như tốc độ, điện áp...

Do vậy, để xây dựng một mô hình hệ thống điều khiển số động cơ một chiều ta cần thiết kế các phần mạch sau:

3.4.1.1 Động cơ điện một chiều

Cho đến nay các động cơ điện một chiều vẫn được sử dụng rộng rãi trong các hệ truyền động có điều chỉnh tốc độ chất lượng cao với dải công suất từ rất nhỏ cho đến rất lớn (hàng ngàn kW). Một trong những ưu điểm của động cơ điện một chiều là có khả năng điều khiển hiệu quả trong phạm vi tốc độ rộng, giá thành của hệ thống điều khiển thấp, đặc biệt là ở dải công suất lớn. Việc sử dụng vi xử lý chuyên dùng trong các hệ truyền động cho phép áp dụng các thuật toán điều khiển hiện đại để điều chỉnh điện áp phản ứng của động cơ cho đến khi tốc độ động cơ đạt đến giá trị lớn nhất ứng với điện áp tối đa cho phép của động cơ (tốc độ cơ bản). Đối với các ứng dụng đòi hỏi điều chỉnh tốc độ động cơ lớn hơn tốc độ cơ bản thì hệ thống điều khiển kích từ tích hợp trong thiết bị sẽ phối hợp để giảm dòng kích từ theo một thuật toán được lập trình sẵn và giữ cho công suất của hệ thống truyền động là không thay đổi.



Hình 3.2: Động cơ điện một chiều

3.4.1.2 Cảm biến tốc độ

*** Máy phát tốc:**

Máy phát tốc là thiết bị dùng để đo tốc độ động cơ.

Về cấu tạo máy phát tốc gồm 2 phần chính là phần cảm và phần ứng.

Nguyên lý hoạt động:

- Máy phát tốc có thể tính tốc độ bằng việc đo thời gian giữa các xung nhận được.
- Máy phát tốc có thể tính tốc độ bằng việc đo tần số của các xung nhận được.
- Máy phát tốc có thể là một máy phát điện gắn đồng trục với trục đối tượng quay, phát ra điện áp tỉ lệ thuận với tốc độ quay.

** Encoder:*

Encoder là thiết bị dùng để đọc tốc độ và vị trí của động cơ, nhờ các xung vuông có tần số thay đổi phụ thuộc vào tốc độ của động cơ.

Cấu tạo, nguyên lý hoạt động:

Encoder là một đĩa tròn xoay, quay quanh trục. Trên đĩa có các lỗ (rãnh). Người ta dùng một đèn Led để chiếu lên mặt đĩa. Khi đĩa quay, chỗ không có lỗ (rãnh), đèn Led không chiếu xuyên qua được, chỗ có lỗ (rãnh), đèn Led sẽ chiếu xuyên qua. Khi đó, phía mặt bên kia của đĩa người ta đặt một mắt thu. Với các tín hiệu có, hoặc không có ánh sáng chiếu qua, người ta ghi nhận được đèn Led có chiếu qua lỗ hay không. Số xung đếm được và tăng lên nó tính bằng số lần ánh sáng bị cắt. Như vậy là encoder sẽ tạo ra các tín hiệu xung vuông và các tín hiệu xung vuông được cắt từ ánh sáng xuyên qua lỗ. Do vậy, tần số của xung đầu ra sẽ phụ thuộc vào tốc độ quay của tấm tròn đó.

3.4.1.3 Điện tử công suất

Nhiệm vụ của phần mạch điện tử công suất là xử lý và điều khiển dòng năng lượng điện bằng cách cung cấp điện áp và dòng điện ở dạng thích hợp cho động cơ. Động cơ sẽ quyết định các thông số về điện áp, dòng điện, tần số, và số pha tại ngõ ra của bộ biến đổi. Thông thường, một bộ điều khiển có hồi tiếp sẽ theo dõi ngõ ra của bộ biến đổi và cực tiểu hóa sai lệch giữa giá trị thực của ngõ ra và giá trị mong muốn (hay giá trị đặt).

Đối tượng nghiên cứu cơ bản của điện tử công suất là các bộ biến đổi bán dẫn.

**Bộ biến đổi:*

Để biến đổi một nguồn điện áp xoay chiều thành điện áp một chiều và điều chỉnh được giá trị của điện áp một chiều đầu ra người ta có thể thực hiện bằng nhiều cách khác nhau. Nhưng phổ biến nhất và có hiệu suất cao nhất là sử dụng các sơ đồ chỉnh lưu bằng các dụng cụ bán dẫn. Các sơ đồ chỉnh lưu (các bộ biến đổi xoay chiều-một chiều) là các bộ biến đổi ứng dụng tính chất dẫn dòng một chiều của các dụng cụ điện tử hoặc bán dẫn để biến đổi điện áp xoay chiều thành điện áp một chiều một cách trực tiếp. Hiện nay, dụng cụ sử dụng chủ yếu trong các sơ đồ chỉnh lưu là các thyristor và các diode bán dẫn. Các sơ đồ chỉnh lưu được sử dụng từ công suất rất nhỏ đến công suất rất lớn.

Trong hệ truyền động này, có thể sử dụng sơ đồ chỉnh lưu: hình cầu 1pha, cầu 3 pha, tia 3 pha, tia 3 pha có D_0 dùng thyristor...; bộ biến đổi một chiều-một chiều...

3.4.1.4 Phần cứng

a. Khối vi xử lý/vi điều khiển

Khối $\mu P/\mu C$ được thiết kế thành một Module riêng nhằm tạo ra khả năng linh hoạt trong việc thay đổi cấu hình điều khiển (chẳng hạn khi cần thay bằng loại vi xử lý mới). Khối này được thiết kế dựa trên vi điều khiển chuyên dụng.

Vi điều khiển là một máy tính được tích hợp trên một chip, nó thường được sử dụng để điều khiển các thiết bị điện tử. Vi điều khiển thực chất, là một hệ thống bao gồm một vi xử lý có hiệu suất đủ dùng và giá thành thấp (khác với các bộ vi xử lý đa năng dùng trong máy tính) kết hợp với các khối ngoại vi như bộ nhớ, các module vào/ra, các module biến đổi số sang tương tự và tương tự sang số... Ở máy tính thì các module thường được xây dựng bởi các chip và mạch ngoài.

Vi điều khiển có nhiều bit khác nhau giống như vi xử lý (hiện nay có các loại vi điều khiển 4bit, 8bit, 16bit, 32bit, 64bit và 128 bit)

Bên trong vi điều khiển chứa đầy đủ tất cả các tính năng cần thiết cho một hệ thống máy tính và các chức năng như một máy tính mà không cần thêm các bộ phận kỹ thuật số bên ngoài.

Hầu hết các chân trong chip vi điều khiển có thể được lập trình bởi người dùng.

Vi điều khiển có khả năng xử lý các hàm logic.

Tốc độ và hiệu suất cao.

Cấu trúc on-chip ROM trong vi điều khiển giúp bảo mật firmware tốt hơn.

Đễ thiết kế với chi phí thấp và kích thước nhỏ.

Để lập trình cho chip, người sử dụng được cung cấp một phần mềm lập trình riêng cho mỗi chip vi điều khiển. Ngoài ra, để cài đặt được chương trình điều khiển vào chip có thể sử dụng trình gỡ rối. Phần mềm phát triển được xây dựng trên cơ sở hướng đối tượng với cấu trúc module hóa các khối chức năng. Việc lập trình cho chip tùy thuộc vào người sử dụng thông qua một số thư viện chuẩn. Người sử dụng thiết lập cấu hình trên chip chỉ đơn giản bằng cách muốn chip có chức năng gì thì chọn chức năng đó và đặt vào khối tài nguyên số hoặc tương tự, hoặc cả hai tùy theo từng chức năng. Việc thiết lập ngắt, thiết lập chế độ hoạt động các chân vào ra tùy thuộc vào người sử dụng khi thiết kế và lập trình cho chip.

Các loại vi điều khiển được sử dụng trên thị trường nước ta hiện nay: Vi điều khiển 8051, Vi điều khiển AVR, Vi điều khiển PIC, Vi điều khiển ARM...

Trong phần thiết kế này, tập trung nghiên cứu về Arduino.

***Arduino:**

- Arduino Uno là một board mạch vi điều khiển được phát triển bởi Arduino.cc, một nền tảng điện tử mã nguồn mở chủ yếu dựa trên vi điều khiển AVR Atmega328P. Với Arduino chúng ta có thể xây dựng các ứng dụng điện tử tương tác với nhau thông qua phần mềm và phần cứng hỗ trợ.

Hiện nay Arduino được sử dụng rất rộng rãi để lập trình tương tác với các thiết bị phần cứng như cảm biến, động cơ, đèn hoặc các thiết bị khác chủ yếu dựa trên vi điều khiển AVR. Đặc điểm nổi bật của Arduino là môi trường phát triển ứng dụng cực kỳ dễ sử dụng, với một ngôn ngữ lập trình có thể học một cách nhanh chóng ngay cả với người ít am hiểu về điện tử và lập trình. Và điều làm nên hiện tượng Arduino chính là mức giá rất thấp và tính chất nguồn mở từ phần cứng tới phần mềm.

Trên thị trường hiện nay có rất nhiều phiên bản Arduino như Arduino Uno R3, Arduino Uno R3 CH340, Arduino Mega2560, Arduino Nano, Arduino Pro Mino, Arduino Lenadro, Arduino Industrial...

- Khi thiết kế có thể chọn một trong các Arduino thông dụng sau:

+*Arduino Uno R3 chip cấm, Arduino chip dán:*

Arduino Uno R3 là một board mạch vi điều khiển được phát triển bởi Arduino.cc, một nền tảng điện tử mã nguồn mở chủ yếu dựa trên vi điều khiển AVR Atmega328P.

Phiên bản hiện tại của Arduino Uno R3 đi kèm với giao diện USB, 6 chân đầu vào analog, 14 cổng kỹ thuật số I / O được sử dụng để kết nối với các mạch điện tử, thiết bị bên ngoài. Trong đó có 14 cổng I/O, 6 chân đầu ra xung [PWM](#) cho phép các nhà thiết kế kiểm soát và điều khiển các thiết bị mạch điện tử ngoại vi một cách trực quan.

Arduino Uno R3 được kết nối trực tiếp với máy tính thông qua USB để giao tiếp với phần mềm lập trình IDE, tương thích với Windows, MAC hoặc Linux Systems, tuy nhiên, Windows thích hợp hơn để sử dụng. Các ngôn ngữ lập trình như C và C ++ được sử dụng trong IDE.

Ngoài USB, người dùng có thể dùng nguồn điện ngoài để cấp nguồn cho bo mạch.

Các bo mạch Arduino Uno khá giống với các bo mạch khác trong các loại Arduino về mặt sử dụng và chức năng, tuy nhiên các bo mạch Uno không đi kèm với chip điều khiển FTDI USB to Serial.

Có rất nhiều phiên bản bo mạch Uno. Tuy nhiên, [Arduino Nano V3](#) và Arduino Uno là những phiên bản chính thức nhất đi kèm với vi điều khiển Atmega328 8 bit AVR Atmel trong đó bộ nhớ RAM là 32KB.

Khi tính chất và chức năng của nhiệm vụ trở nên phức tạp, thẻ nhớ SD Mirco có thể được kết nối thêm vào Arduino để lưu trữ được nhiều thông tin hơn.



Hình 3.3: Arduino Uno R3 chip cắm



Hình 3.4: Arduino Uno R3 chip dán CH340

Các tính năng Arduino trên Board:

Arduino Uno đi kèm với giao diện USB tức là cổng USB được thêm vào bo mạch Arduino để phát triển giao tiếp nối tiếp với máy tính.

Bộ vi điều khiển Atmega328 sử dụng trên bo mạch đi kèm với một số tính năng như hẹn giờ, bộ đếm, ngắt, chân PWM, CPU, chân I / O và dựa trên xung nhịp 16 MHz giúp tạo ra nhiều tần số và số lệnh hơn trong mỗi chu kỳ.

Đây là một nền tảng mã nguồn mở, nơi mọi người có thể sửa đổi và tối ưu hóa bảng dựa trên số lượng hướng dẫn và nhiệm vụ muốn đạt được.

Arduino đi kèm với một tính năng điều chỉnh tích hợp giúp giữ điện áp trong tầm kiểm soát khi thiết bị được kết nối với thiết bị bên ngoài.

Chân reset trên Arduino để thiết lập lại toàn bộ và đưa chương trình đang chạy trở về ban đầu. Chân reset này hữu ích khi Arduino bị treo khi đang chạy chương trình

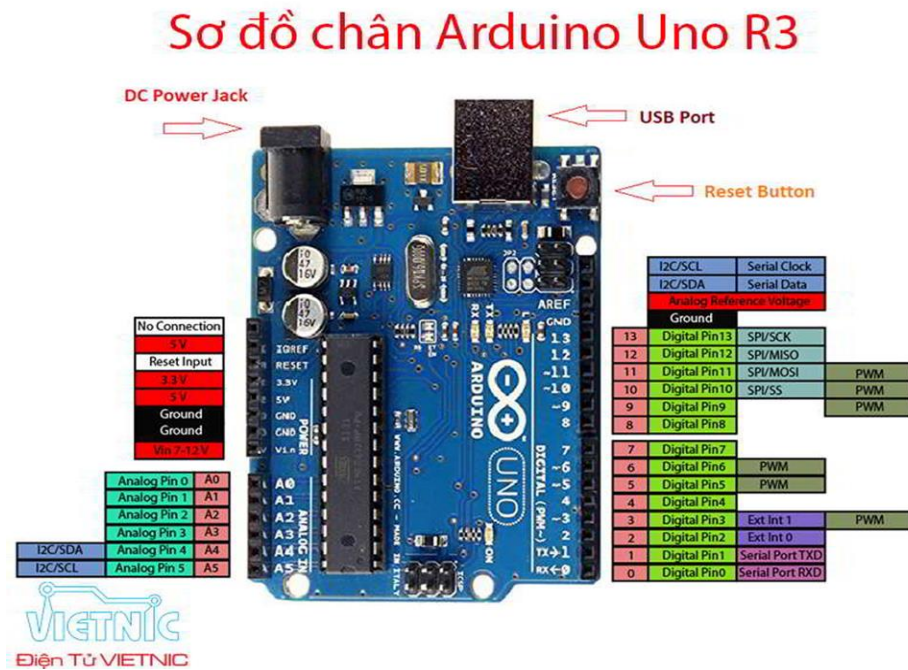
Có 14 chân I / O digital và 6 chân analog được tích hợp trên Arduino cho phép kết nối bên ngoài với bất kỳ mạch nào với Arduino. Các chân này cung cấp sự linh hoạt và dễ sử dụng cho các thiết bị bên ngoài có thể được kết nối thông qua các chân này.

6 chân analog được đánh dấu là A0 đến A5 và có độ phân giải 10 bit. Các chân này đo từ 0 đến 5V, tuy nhiên, chúng có thể được cấu hình ở phạm vi cao bằng cách sử dụng chức năng analogReference () và chân ISF.

Bộ nhớ flash 13KB được sử dụng để lưu trữ số lượng hướng dẫn dưới dạng mã.

Chỉ cần nguồn 5V để sử dụng với Arduino, hoặc lấy nguồn trực tiếp từ cổng USB. Arduino có thể hỗ trợ nguồn điện bên ngoài lên đến 12 V có thể được điều chỉnh và giới hạn ở mức 5 V hoặc 3,3 V dựa trên yêu cầu của projects.

Các chân input - output của Arduino Uno R3:



Hình 3.5: Sơ đồ chân Arduino Uno R3

Có một số chân I/O Digital và analog được đặt trên bo mạch hoạt động ở mức logic 5V với dòng từ khoảng từ 20mA đến 40mA

Thông số kỹ thuật:

Chip ATMEGA328P-PU

Nguồn Cấp : 7-12V

Dòng Max chân 5V : 500mA

Dòng Max 3.3V : 50mA

Dòng Max Chân I/O : 30mA

14 Chân Digital I/O (6 chân PWM)

6 Chân Analog Inputs

32k Flash Memory

16Mhz Clock Speed

SRAM 2 KB

EEPROM 1 KB

Đèn LED: Arduino Uno đi kèm với đèn LED tích hợp được kết nối thông qua chân 13. Cung cấp mức logic HIGH tương ứng ON và LOW tương ứng tắt.

Vin: Đây là điện áp đầu vào được cung cấp cho board mạch Arduino. Khác với 5V được cung cấp qua cổng USB. Pin này được sử dụng để cung cấp điện áp toàn mạch thông qua jack nguồn, thông thường khoảng 7-12VDC

5V: Chân 5V được sử dụng để cung cấp điện áp đầu ra. Arduino được cấp nguồn bằng ba cách đó là USB, chân Vin của bo mạch hoặc giắc nguồn DC.

USB: Hỗ trợ điện áp khoảng 5V trong khi Vin và Power Jack hỗ trợ dải điện áp trong khoảng từ 7V đến 20V.

GND: Chân mass chung cho toàn mạch Arduino

Reset: Chân reset để thiết lập lại về ban đầu

IOREF: Chân này rất hữu ích để cung cấp tham chiếu điện áp cho Arduino

PWM: PWM được cung cấp bởi các chân 3,5,6,9,10,11. Các chân này được cấu hình để cung cấp PWM đầu ra 8 bit.

SPI: Chân này được gọi là giao diện ngoại vi nối tiếp. Các chân 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK) cung cấp liên lạc SPI với sự trợ giúp của thư viện SPI.

AREF: Chân này được gọi là tham chiếu tương tự, được sử dụng để cung cấp điện áp tham chiếu cho các đầu vào tương tự.

TWI: Chân Giao tiếp TWI được truy cập thông qua thư viện dây. Chân A4 và A5 được sử dụng cho mục đích này.

Serial Communication :Giao tiếp nối tiếp được thực hiện thông qua hai chân 0 (Rx) và 1 (Tx).

Rx: Chân này được sử dụng để nhận dữ liệu trong khi chân Tx được sử dụng để truyền dữ liệu.

External Interrupts (ngắt ngoài): Chân 2 và 3 được sử dụng để cung cấp các ngắt ngoài.

+Arduino Nano:

Arduino Nano USB Mini là board mạch sử dụng vi điều khiển ATmega328P hoặc ATmega168 tích hợp các chân I/O đơn giản nhỏ gọn dựa trên mã nguồn mở được phát triển bởi Arduino.cc, có lợi thế lớn về kích thước so với phiên bản Arduino Uno và Arduino Mega. Arduino Nano có thể hoạt động độc lập và tương tác hiệu quả với các thiết bị điện tử, cũng có thể giúp những người mới tìm hiểu về Arduino có thể kết nối với PC, phối hợp với Flash, Xử lý, Max / Msp, PD, và các phần mềm khác một cách dễ dàng. Điều này giúp Arduino Nano là sự lựa chọn ưa thích khi muốn thực hiện một projects mà yêu cầu kết nối với các thiết bị ngoại vi ít và đơn giản.

Các chức năng rất giống giống với phiên bản Arduino Uno nhưng kích thước nhỏ gọn hơn.

Arduino Nano hoạt động với điện áp 5V. Tuy nhiên nguồn cấp điện áp đầu vào có thể thay đổi từ 7 đến 12V.

Arduino Nano bo gồm tất cả 14 chân Digital, 8 chân Analog, 2 chân Reset và 6 chân nguồn.

Mỗi chân Digital và Analog có thể thực hiện với nhiều chức năng khác nhau nhưng chức năng chính vẫn là được mặc định cấu hình làm đầu vào hoặc đầu ra. Khi

giao tiếp với cảm biến các chân Digital / Analog đóng vai trò chân Input, và khi sử dụng để điều khiển động cơ, tạo xung, kích dẫn relay, thiết bị chuyển mạch thì các chân Digital / Anologs đóng vai trò Output.

Các hàm cơ bản như pinMode() và digitalWrite() được sử dụng để điều khiển hoạt động của các chân Digital trong khi analog Read() được sử dụng để điều khiển các chân Analog.

Các chân Analog có độ phân giải 10 bit, giá trị thay đổi từ 0V đến 5V.

Arduino Nano tích hợp thạch anh dao động với tần số 16 MHz. Nếu sử dụng Arduino Nano để thực hiện một projects liên quan đến đồng hồ số thì độ chính xác trong dao động tần số rất cao.

Có một hạn chế khi sử dụng Arduino Nano đó là không đi kèm giắc nguồn DC, có nghĩa là không thể cung cấp nguồn điện bên ngoài thông qua các chân Arduino Nano như các phiên bản Arduino Uno và Arduino Mega. Phải cắm trực tiếp thông qua cổng USB Mini.

Board mạch Arduino Nano không sử dụng cổng USB chuẩn thường dùng để kết nối với máy tính, thay vào đó Arduino Nano chỉ hỗ trợ cổng USB Mini để kết nối.

Với kích thước nhỏ gọn và các tính năng giống với phiên bản Arduino Uno nên Arduino Nano trở thành một lựa chọn lý tưởng cho hầu hết các ứng dụng thực hiện các projects trong đó kích thước của các thành phần điện tử được tối ưu.

Bộ nhớ flash là 16KB hoặc 32KB, phụ thuộc vào Vi điều khiển Atmega, tức là Atmega168 đi có bộ nhớ flash 16KB trong khi Atmega328 đi có bộ nhớ flash là 32KB. Bộ nhớ flash được sử dụng để lưu trữ mã hóa dữ liệu. Bộ nhớ 2KB trong tổng số bộ nhớ flash được sử dụng cho chương trình khởi động gọi là Bootloader.

Bộ nhớ SRAM có thể thay đổi từ 1KB hoặc 2KB và EEPROM tương ứng là 512 byte hoặc 1KB đối với Atmega168 và Atmega328.

Sử dụng phần mềm IDE để lập trình với cáp USB Mini giao tiếp với máy tính

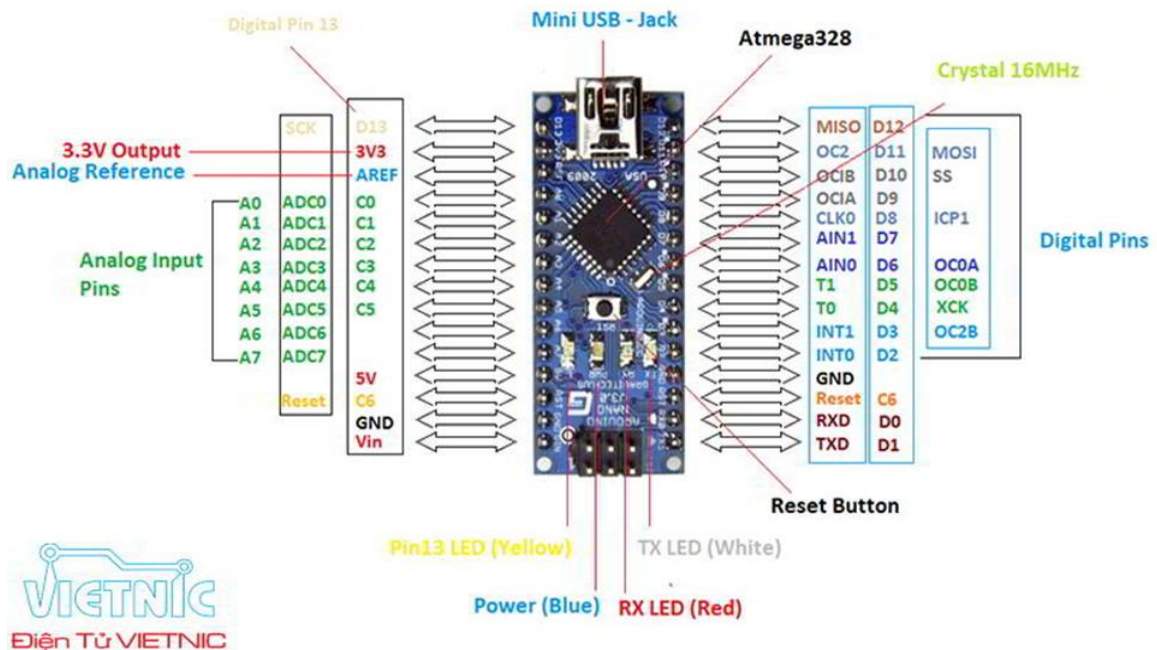


Hình 3.6: Arduino nano

Sơ đồ chân và thông số kỹ thuật Arduino Nano sử dụng Vi điều khiển ATmega328P :

Chip : ATmega328P
 Điện áp logic : 5V
 Điện áp hoạt động : 7-12V
 Các chân I/O : 14 chân (Bao gồm 6 chân PWM)
 Dòng điện I/O : 40mA
 Bộ nhớ Flash : 32Kb
 SRAM : 2Kb
 EEPROM : 1Kb
 Tần số dao động : 16Mhz

Sơ đồ chân Arduino Nano



Hình 3.7: Sơ đồ chân Arduino Nano

Mỗi chân trên mạch Arduino Nano có chức năng cụ thể trên các chân đó. Chẳng hạn như các chân Analog có thể sử dụng như một bộ chuyển đổi Analog sang Digital trong đó các chân A4 và A5 cũng có thể được sử dụng cho vấn đề giao tiếp I2C. Tương tự, có 14 chân Digital, trong đó có 6 chân được sử dụng để tạo ra xung PWM.

Chân Vin: Đây là chân cung cấp điện áp đầu vào cho mạch Arduino nano khi sử dụng nguồn ngoài từ 7VDC đến 12 VDC.

Chân 5V: Là mức điện áp cung cấp quy định của Arduino được sử dụng để cấp nguồn nuôi cho vi điều khiển và các bộ phận linh kiện khác trên board Arduino.

Chân 3,3V: Đây là một mức điện áp tối thiểu được tạo ra bởi bộ điều chỉnh điện áp trên board (sử dụng Lm1117 - 3.3V).

Chân GND: Chân mass cho Arduino, có nhiều chân GND trên board Arduino cho mục đích dễ dàng kết nối với thiết bị ngoại vi sử dụng dây testboard.

Chân Reset: Khi tác động nút nhấn reset, Arduino được trả về lại chương trình ban đầu. Rất hữu ích khi chạy chương trình phức tạp và bị treo Vi điều khiển ATmega. Mức tích cực LOW được thiết lập sẽ reset lại Arduino Nano.

Các chân Analogs: Có 8 chân Analog trên board mạch Arduino Nano được ký hiệu là A0 đến A7. Được sử dụng để đo điện áp tương tự trong khoảng từ 0V đến 5V.

Chân Rx, Tx: Được sử dụng cho giao tiếp nối tiếp trong đó Tx là truyền dữ liệu và Rx là nhận dữ liệu.

Chân 13: Để thực hiện bật tắt LED trên board Arduino Nano, sử dụng để quan sát, kiểm tra chương trình cần thiết.

Chân AREF: Chân này được sử dụng lấy điện áp tham chiếu cho điện áp đầu vào.

Chân xung PWM: Bao gồm 6 chân là chân 3,5,6,9,10,11 được sử dụng để cung cấp đầu ra 8-bit xung PWM.

Giao tiếp SPI: Chân 10(SS), chân 11(MOSI), chân 12(MISO), chân 13(SCK) được sử dụng cho SPI Giao diện ngoại vi nối tiếp. SPI được sử dụng chủ yếu để truyền dữ liệu giữa các bộ vi điều khiển và các thiết bị ngoại vi khác như cảm biến, thanh ghi và thẻ nhớ SD.

Ngắt ngoài (External Interrupts) : Chân 2 và 3 sử dụng làm ngắt ngoài được thiết lập trong trường hợp khẩn cấp khi chúng ta cần dừng chương trình chính và tác động các cảnh báo hướng dẫn tại thời điểm đó. Chương trình chính sẽ tiếp tục lại sau khi lệnh ngắt được loại bỏ.

Giao tiếp I2C : Giao tiếp I2C sử dụng các chân A4 (SDA) và A5 (SCL)

+Arduino Mega 2560:

Arduino Mega 2560 là board mạch vi điều khiển dựa trên chip xử lý Atmega 2560 được mở rộng thêm bộ nhớ và các chân I / O so với các bo mạch khác có sẵn trên thị trường.

Có 54 chân I/O digital và 16 chân analog được tích hợp trên bo mạch giúp thiết bị này trở nên riêng biệt và nổi bật so với các thiết bị khác.

Có tất cả 54 chân ra I/O digital, trong đó có 15 chân được sử dụng cho xuất xung PWM.

Một bộ dao động thạch anh có tần số 16 MHz được tích hợp trên board Arduino Mega2560.

Arduino Mega 2560 có cổng USB được sử dụng để kết nối và chuyển mã từ máy tính đến mạch Arduino Mega dựa trên phần mềm IDE.

Tích hợp jack nguồn DC để cấp nguồn cho bo mạch. Một số phiên bản Arduino khác thiếu tính năng này như Arduino Pro Mini không đi kèm jack cắm nguồn DC.

Đầu jack kết nối ICSP (Header đôi gần nút reset) được bổ sung đáng chú ý cho Arduino Mega, sử dụng để lập trình Arduino và tải file lên từ máy tính qua phần mềm IDE.

Arduino Mega 2560 có hai mức điện áp là 5V và 3.3V cung cấp sự linh hoạt để điều chỉnh điện áp theo yêu cầu so với Arduino Pro Mini chỉ đi kèm với một bộ điều chỉnh điện áp.

So sánh với Arduino Uno thì Arduino Mega không có nhiều sự khác biệt giữa Arduino Uno và Arduino Mega ngoại trừ Arduino Mega được mở rộng bộ nhớ và các chân I/O nhiều hơn, kích thước lớn hơn.

Về phần mềm lập trình, tất cả các loại Arduino đều dùng chung phần mềm IDE.

Tính khả dụng của Atmega16 trên bo mạch Arduino Mega làm cho nó khác với Arduino Pro Mini chỉ sử dụng USB để chuyển đổi nối tiếp để lập trình.

Tích hợp nút reset trên board mạch và 4 cổng nối tiếp phần cứng được gọi là USART, tạo ra tốc độ tối đa để giao tiếp.



Hình 3.8: Arduino Mega2560

Sơ đồ chân và thông số kỹ thuật của Arduino Mega 2560:

Chip ATMEGA 2560

Điện Áp hoạt động : 5VDC

Nguồn cấp : 7-12V (Giới Hạn 6-20V)

Dòng Max chân 5V : 500mA

Dòng Max 3.3V : 50mA

Dòng Max chân I/O : 40mA

54 Chân Digital I/O (15 Chân PWM)

16 Chân Analog Inputs

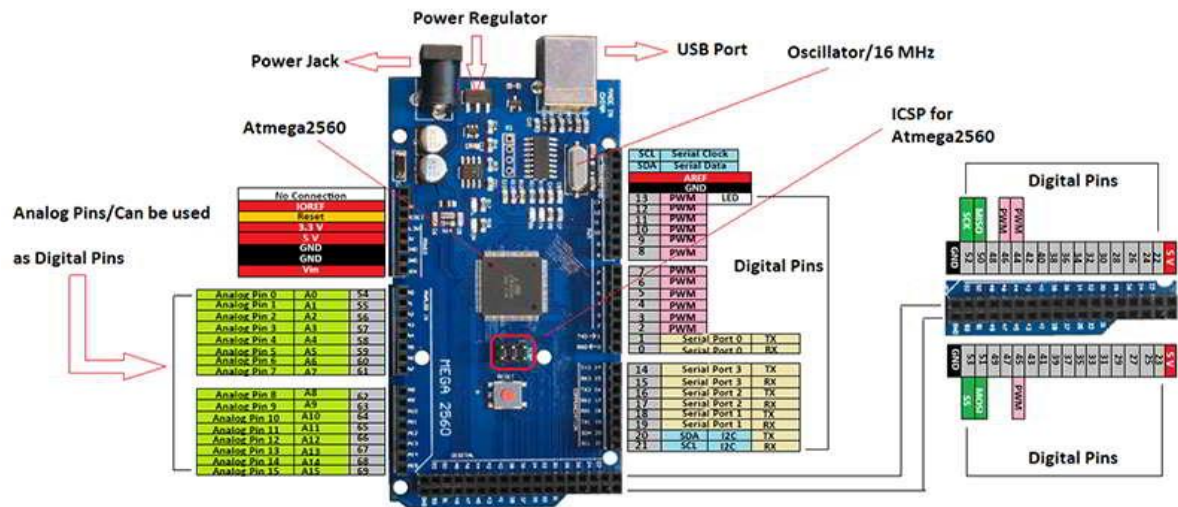
Bộ nhớ Flash : 256K

16Mhz Clock Speed

SRAM 8 KB

EEPROM 4 KB

Sơ đồ chân Arduino Mega2560



Hình 3.9: Sơ đồ chân Arduino Mega 2560

Chi tiết về các chân trên board mạch Arduino Mega 2560:

Chân 5V & 3.3V: Chân này được sử dụng để cung cấp điện áp đầu ra khoảng 5V.

Chân GND: Có 5 chân nối mass có sẵn trên board Arduino Mega, giúp dễ dàng kết nối nếu thực hiện dự án với nhiều kết nối thiết bị ngoại vi.

Chân reset: Được sử dụng để thiết lập lại board mạch về lại ban đầu. Mức tích cực LOW được thiết lập sẽ reset lại board mạch.

Chân Vin: Là chân điện áp đầu vào cung cấp cho mạch Arduino Mega, điện áp từ 7V đến 20V. Mặt khác điện áp được cấp bởi jack nguồn DC có thể được lấy thông qua chân này. Tuy nhiên, điện áp đầu ra thông qua chân này đến mạch Arduino sẽ được tự động thiết lập là 5V.

Chân truyền thông nối tiếp (Serial Communication): RXD và TXD là các chân nối tiếp được sử dụng để truyền và nhận dữ liệu nối tiếp, chân Rx đại diện cho việc truyền dữ liệu còn Tx được sử dụng để nhận dữ liệu. Có tất cả 4 kết hợp các chân nối tiếp này được sử dụng trong đó Serial 0 là chân RX(0) và TX(1), Serial 1 là chân TX(18) và RX(19), Serial 2 là chân TX(16) và RX(17), và Serial 3 là chân TX(14) và RX(15).

Chân ngắt ngoài (External Interrupts): 6 chân được sử dụng để tạo các ngắt ngoài đó là ngắt 0 (chân 0), ngắt 1 (chân 3), ngắt 2 (chân 21), ngắt 3 (chân 20), ngắt 4 (chân 19), ngắt 5 (chân 18). Các chân này tạo ra các ngắt bằng một số cách tức là cung cấp giá trị LOW, tăng hoặc giảm hoặc thay đổi giá trị cho các chân ngắt.

Đèn LED: Arduino Mega 2560 tích hợp đèn LED trên board mạch kết nối với chân 13. Giá trị HIGH đèn LED được bật và LOW đèn LED tắt. Giúp người lập trình quan sát trực quan khi test, kiểm tra chương trình trên board Arduino.

Chân AREF: Chân tạo điện áp tham chiếu cho đầu vào analogs.

Các chân tương tự (Analog): Có 16 chân analog được tích hợp trên board Arduino có ký hiệu là A0 đến A15. Điều quan trọng cần lưu ý là tất cả các chân analog này có thể được sử dụng làm chân I / O Digital. Mỗi chân analog đi kèm với độ phân giải 10 bit. Các chân này có thể có điện áp thay đổi từ 0V đến 5V. Tuy nhiên, giá trị trên có thể được thay đổi bằng cách sử dụng hàm ISF và analogReference ().

Giao tiếp I2C: Hai chân 20 và 21 hỗ trợ giao tiếp I2C trong đó 20 đại diện cho SDA (Dòng dữ liệu nối tiếp chủ yếu được sử dụng để giữ dữ liệu) và 21 đại diện cho SCL (Dòng đồng hồ nối tiếp chủ yếu được sử dụng để cung cấp đồng bộ hóa dữ liệu giữa các thiết bị).

Truyền thông SPI: Được sử dụng để truyền dữ liệu giữa Arduino và các thiết bị ngoại vi khác. Chân 50 (MISO), Chân 51 (MOSI), Chân 52 (SCK), Chân 53 (SS) được sử dụng để liên lạc SPI.

b. Mạch điều khiển trung tâm

Khối mạch điều khiển trung tâm có nhiệm vụ quản lý, giám sát, điều khiển hoạt động của toàn bộ các modul sử dụng trong mạch có nghĩa là mọi hoạt động của hệ thống trên cơ sở vi điều khiển.

Khối mạch điều khiển trung tâm kết nối khối vi xử lý với các thành phần khác trong hệ thống (mạch đồng bộ hóa, các đầu vào điều khiển và phản hồi...)

Mạch điều khiển trung tâm bao gồm các khối sau:

- Khối vi xử lý.
- Khối đồng bộ hóa.
- Khối đầu vào tương tự dùng để đo các đại lượng: Điện áp và dòng điện phản ứng, điện áp và dòng điện kích từ, tốc độ động cơ, ...
- Khối đầu ra tương tự bao gồm cả nguồn áp và nguồn dòng.
- Khối đầu vào số dùng kết hợp với mạch không chế.
- Khối đầu ra số bao gồm cả mạch điều khiển rơ le.
- Khối thời gian thực.
- Khối nhớ ngoài
- Khối giao tiếp bàn phím và hiển thị LED.
- Khối nguồn.

...

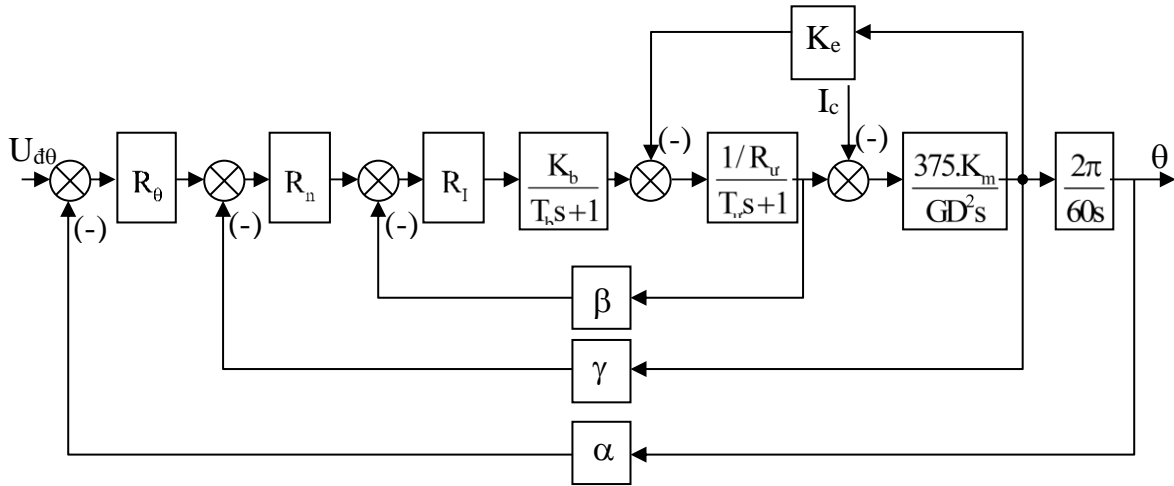
c. Mạch công suất xung

Mạch công suất xung và được thiết kế theo từng kênh điều khiển riêng biệt.

3.4.2. Tổng hợp bộ điều khiển trong hệ truyền động điện

3.4.2.1. Sơ đồ cấu trúc tổng quát

Cấu trúc tổng quát của hệ truyền động động cơ điện một chiều có dạng như hình vẽ



Hình 3.2: Sơ đồ cấu trúc hệ truyền động động cơ một chiều

Trong đó:

Động cơ và bộ biến đổi đã biết, vì vậy hàm truyền được xác định với:

$$+ K_e = \frac{U_{dm} - I_{dm} R_u}{n_{dm}} \text{ là hệ số sức điện động}$$

$$+ K_m = 9,55 K_e \text{ là hệ số mô men}$$

$$+ T_u = \frac{L_u}{R_u} \text{ là hằng số thời gian điện từ}$$

$$+ GD^2 [\text{kgm}^2] \text{ là mô men vô lăng}$$

$$+ K_b = \frac{\Delta U_d}{\Delta U_{dk}} \text{ là hệ số khuếch đại của bộ biến đổi}$$

$$+ T_b = \frac{1}{2mf} \text{ là hằng số thời gian của bộ biến đổi}$$

$$+ U_{dk} = \frac{\alpha}{\pi} U_{rcmax} \text{ là quan hệ giữa góc mở của van và điện áp điều khiển}$$

$$+ U_d = U_{do} \cdot \cos \alpha \text{ là quan hệ giữa góc mở của van và điện áp ra của chỉnh lưu.}$$

$$+ U_d = f(u_{dk}) \text{ là quan hệ giữa điện áp điều khiển và điện áp ra của chỉnh lưu}$$

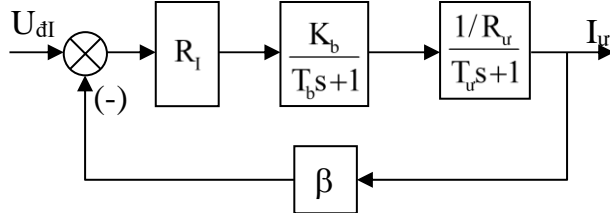
+ β, γ, α là các hệ số phản hồi dòng điện, tốc độ, vị trí. Để đơn giản khi tổng hợp thường chọn trước các hệ số phản hồi sau đó chỉnh.

R_θ, R_n, R_I là các bộ điều chỉnh trong các mạch vòng vị trí, tốc độ, dòng điện.

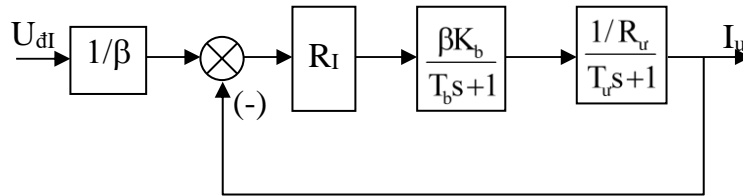
3.4.2.2. Tính toán lập trình bộ điều chỉnh dòng điện:

a. Bỏ qua ảnh hưởng của sức điện động E_d :

Ta biết, tốc độ biến đổi chậm hơn nhiều so với dòng điện, vậy trong chu kỳ tính toán dòng điện gần đúng coi tốc độ là không biến đổi. Vì vậy, $E_d = K_e \cdot n$ không đổi nên có thể coi như E_d không tham gia điều khiển mạch vòng dòng điện. Từ cấu trúc tổng quát, ta có cấu trúc mạch vòng dòng điện như hình vẽ sau:



Để xác định R_I ta dùng phương pháp mô đun tối ưu. Khi đó, ta cấu trúc chuẩn của hệ như sau:



$$W_{oi}(s) = \frac{\beta K_b}{R_u (T_b s + 1)(T_u s + 1)}$$

Hàm truyền kín của mạch vòng dòng điện là:

$$W_{kl}(s) = \frac{R_I(s) W_{oi}(s)}{1 + R_I(s) W_{oi}(s)} \Rightarrow R_I(s) = \frac{W_{kl}(s)}{[1 - W_{kl}(s)] W_{oi}(s)}$$

Áp dụng tiêu chuẩn tối ưu modul với hàm truyền chuẩn có dạng:

$$W_c = \frac{1}{2\tau^2 s^2 + 2\tau s + 1}$$

$$\Rightarrow R_I(s) = \frac{W_c(s)}{[1 - W_c(s)] W_{oi}(s)} = \frac{1}{2\tau s(1 + \tau s) W_{oi}(s)} = \frac{R_u (T_b s + 1)(T_u s + 1)}{\beta K_b 2\tau s(1 + \tau s)}$$

- Giả sử $T_u < T_b < t_p/6$

Chọn $\tau = T_u$

$$\Rightarrow R_I(s) = \frac{R_u (T_b s + 1)}{2\beta K_b T_u s} = K_I \frac{(1 + T_b s)}{s}$$

$$= \frac{K_I}{s} + K_I T_b = \frac{K_I}{s} + K_p$$

Trong đó $K_I = \frac{R_u}{2\beta T_u K_b}$; $K_p = K_I \cdot T_b$

$$R_1(z) = Z[R_1(s)] = Z\left[\frac{K_I}{s} + K_P\right] = \left[K_I \frac{z}{z-1} + K_P\right] = \frac{U_{dk}(z)}{U_v(z)}$$

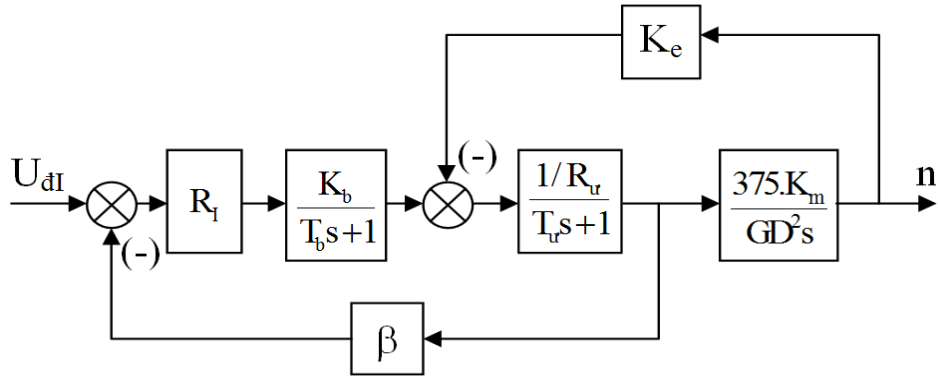
$$U_{dk}(z) = z^{-1} U_{dk}(z) + (K_I + K_P)U_v(z) - K_P z^{-1} U_v(z)$$

$$u_{dk}[k] = u_{dk}[k-1] + (K_I + K_P)u_v[k] - K_P u_v[k-1]$$

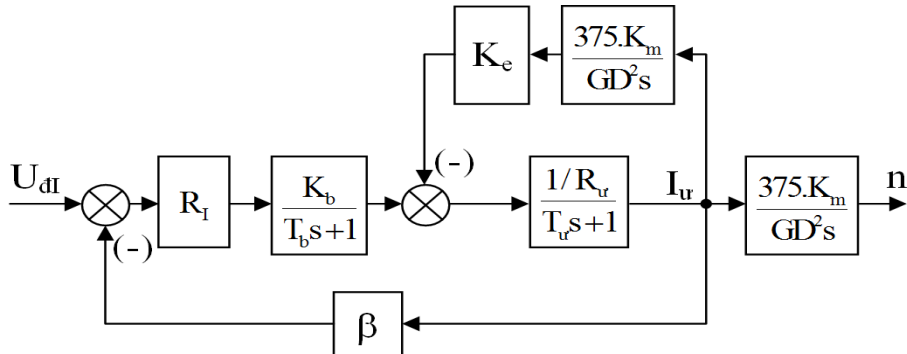
Đây là phương trình sai phân biểu diễn quy luật của bộ điều chỉnh dòng điện.

b. Có tính đến ảnh hưởng của E_d

Bỏ qua nhiễu phụ tải ($M_c = 0$). Ta có sơ đồ cấu trúc của mạch vòng dòng điện như sau:



Biến đổi sơ đồ ta có:



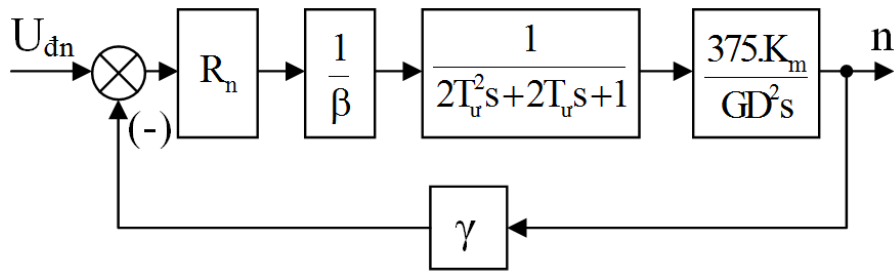
Làm tương tự như khi không kể đến ảnh hưởng của sức điện động E_d ta cũng sẽ tìm được thuật toán lập trình cho bộ điều chỉnh dòng.

3.4.2.3. Tính toán lập trình cho bộ điều chỉnh tốc độ

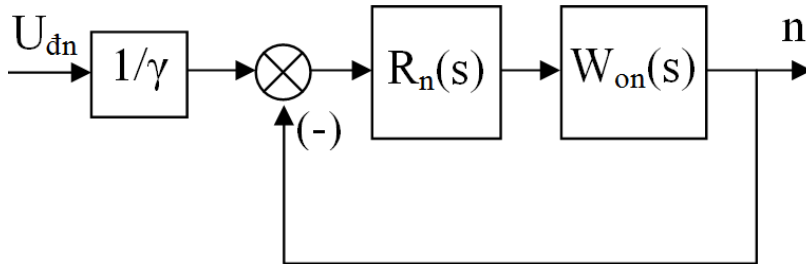
Sau khi tính toán lập trình cho bộ điều chỉnh dòng điện ta được hàm W_{tr} như sau:

$$W_{tr} = \frac{1}{2T_r^2 s^2 + 2T_r s + 1}$$

Từ đó ta có cấu trúc của mạch vòng điều chỉnh tốc độ như sau:



Biến đổi sơ đồ cấu trúc ta được:



$$W_{on}(s) = \frac{375K_m}{\beta(2T_u^2 s^2 + 2T_u s + 1)GD^2 s}$$

Với:

Hàm truyền kín của mạch vòng tốc độ là:

$$W_{kn}(s) = \frac{R_n(s)W_{on}(s)}{1 + R_n(s)W_{on}(s)} \Rightarrow R_n(s) = \frac{W_{kn}(s)}{[1 - W_{kn}(s)]W_{on}(s)}$$

Áp dụng tiêu chuẩn tối ưu modul với hàm truyền chuẩn có dạng:

$$W_c = \frac{1}{2\tau^2 s^2 + 2\tau s + 1}$$

Do đó:

$$R_n(s) = \frac{W_c(s)}{[1 - W_c(s)]W_{on}(s)} = \frac{1}{2\tau s(1 + \tau s)W_{on}(s)} = \frac{\beta GD^2 (2T_u^2 s^2 + 2T_u s + 1)s}{\gamma 375K_m 2\tau s(1 + \tau s)}$$

+ Trường hợp: $2T_u^2 < 0,001$ thì ta có thể bỏ qua, khi đó:

$$\Rightarrow R_n(s) = \frac{\beta GD^2 (2T_u s + 1)s}{\gamma 375K_m 2\tau s(1 + \tau s)}$$

$$\Rightarrow R_n(s) = \frac{\beta GD^2}{\gamma 375K_m 4T_u} = K_p$$

+ Nếu $2T_u < t_p/6$ thì ta đặt $\tau = 2T_u$:

$$R_n(z) = Z[R_n(s)] = Z[K_p] = \frac{U_{dk}(z)}{U_v(z)}$$

$$U_{dk}(z) = K_p U_v(z)$$

$$U_{dk}[k] = K_p U_v[k]$$

Đây là phương trình sai phân biểu diễn quy luật của bộ điều chỉnh tốc độ.

$$R_n(s) = \frac{\beta G D^2 (2T_{ur}p + 1)}{\gamma 375 K_m 2a(1 + as)}$$

+ Nếu $2T_{ur} > t_p/6$ thì ta đặt $\tau = a$, từ đó ta có:

Làm tương tự như trường hợp trên ta cũng tìm ra $R_n(z)$ và tìm ra phương trình sai phân là thuật toán lập trình cho bộ điều chỉnh tốc độ

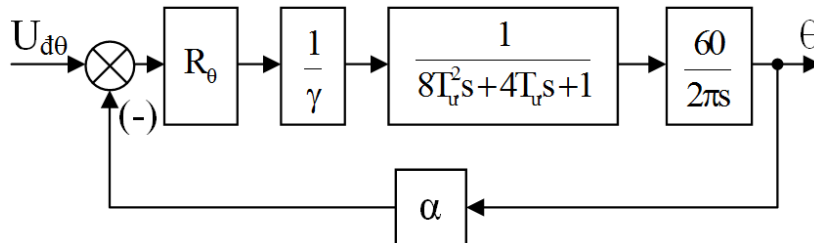
Luật điều khiển PID số trong các công thức trên được lựa chọn để cài đặt cho bộ điều khiển được chế tạo trên chip vi điều khiển.

3.4.2.4. Tính toán lập trình cho bộ điều chỉnh vị trí

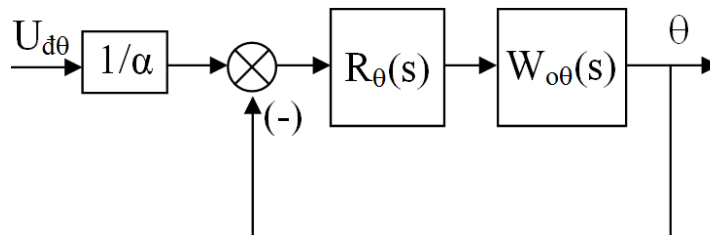
Sau khi tổng hợp cho mạch vòng tốc độ thì ta được hàm tối ưu như sau:

$$W_{tur} = \frac{1}{8T_{ur}^2 s^2 + 4T_{ur}s + 1}$$

Từ đó ta có cấu trúc của mạch vòng vị trí như sau:



Biến đổi sơ đồ cấu trúc ta được:



Ta có hàm truyền hở của mạch vòng điều chỉnh vị trí:

$$W_{oθ}(s) = \frac{2\pi\alpha}{60\gamma(8T_{ur}^2 s^2 + 4T_{ur}s + 1)s}$$

Hàm truyền kín của mạch vòng điều chỉnh vị trí là:

$$W_{kθ}(s) = \frac{R_θ(s)W_{oθ}(s)}{1 + R_θ(s)W_{oθ}(s)} \Rightarrow R_θ(s) = \frac{W_{kθ}(s)}{[1 - W_{kθ}(s)]W_{oθ}(s)}$$

Áp dụng tiêu chuẩn tối ưu modul với hàm truyền chuẩn có dạng:

$$W_c = \frac{1}{2\tau^2 s^2 + 2\tau s + 1}$$

$$R_θ(s) = \frac{W_c(s)}{[1 - W_c(s)]W_{oθ}(s)} = \frac{1}{2\tau s(1 + \tau s)W_{oθ}(s)} = \frac{60\gamma(8T_{ur}^2 s^2 + 4T_{ur}s + 1)s}{4\pi\alpha(1 + \tau s)\tau s}$$

Giả sử $8T_{ur}^2 < 0,001$ nên ta bỏ qua, khi đó

$$R_{\theta}(s) = \frac{60\gamma(4T_u s + 1)s}{4\pi\alpha(1 + \tau s)\tau s}$$

+ Giả sử $4T_u < t_p/6$ nên chọn $\tau = 4T_u$

$$\Rightarrow R_{\theta}(s) = \frac{15\gamma}{\pi\alpha T_u} = K_{p0}$$

+ Giả sử $4T_u > t_p/6$ nên chọn $\tau = t_p/6 = a$:

$$\Rightarrow R_{\theta}(s) = \frac{15\gamma(4T_u s + 1)}{\alpha\pi a(1 + as)}$$

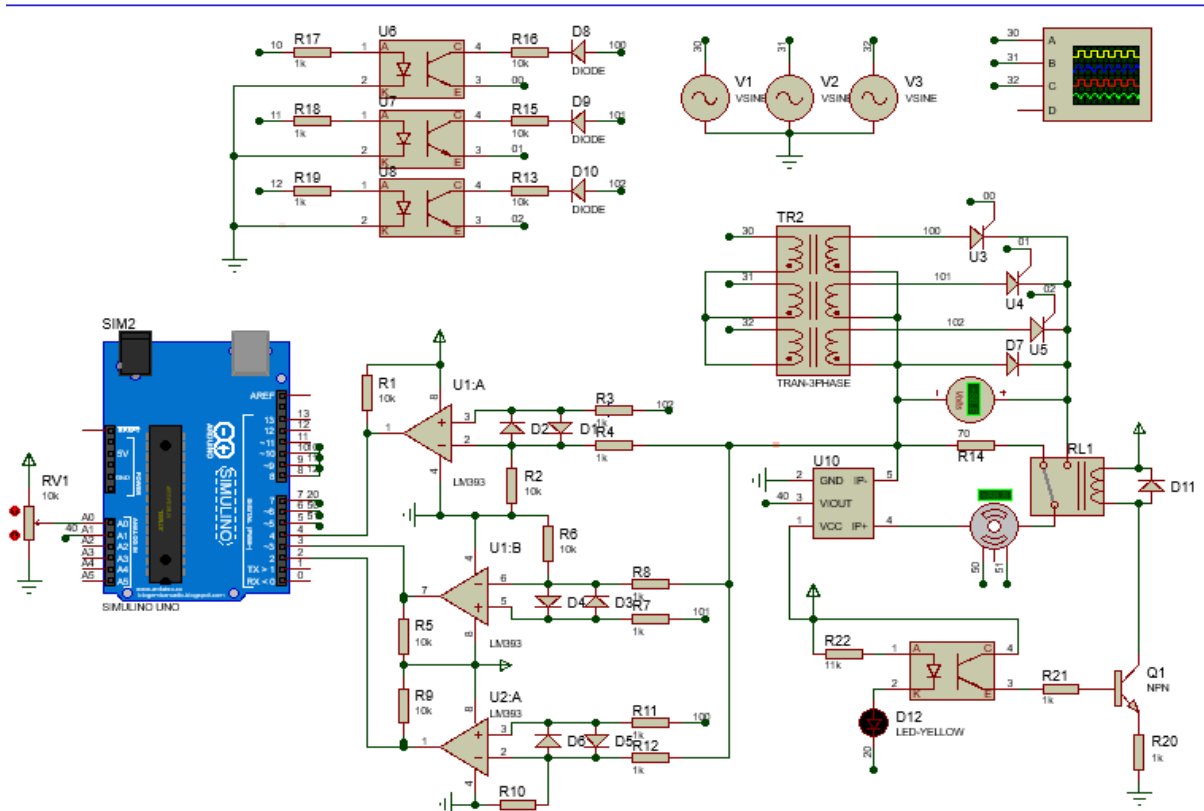
Làm tương tự như trường hợp trên ta cũng tìm ra $R_{\theta}(z)$ và tìm ra phương trình sai phân là thuật toán lập trình cho bộ điều chỉnh vị trí.

3.4.3 Thiết kế và lập trình điều khiển

Thiết kế bộ điều khiển sử dụng phần mềm mô phỏng Proteus:

Các linh kiện sử dụng trong hệ thống:

- + ARDUINO UNO R3
- + DIODE
- + LED YELLOW
- + LM393
- + MOTOR- ENCODER
- + NPN
- + PC817
- + POT
- + RES
- + RT114012F
- + THYRISTOR
- + TRAN 3PHASE
- + VSINE
- + OSCILLOSCOPE
- + 5V
- + GND



Cấu trúc AVR board Arduino uno R3 sử dụng chip Atmega328:

- Tham khảo schematic trên mã nguồn mở :

<https://www.arduino.cc/>

- Tham khảo dạng ngôn ngữ lập trình C++ và lệnh qua mã nguồn:

<https://www.arduino.cc/reference/en/>

- Tham khảo thư viện cho thiết bị sử dụng qua mã nguồn mở :

<https://github.com/>

*Lập trình điều khiển:

Sử dụng phần mềm Arduino IDE để lập trình điều khiển.

Dưới đây là chương trình điều khiển để điều khiển tốc độ qua thay đổi góc mở Thyristor trong bộ biến đổi tia 3 pha có D_0 :

```
#include "PinChangeInt.h" //
#include <PID_v1.h>
#define T11 8
#define T12 8
#define T21 9
#define T22 9
#define T31 10
#define T32 10
#define relay 7
#define pot A0
```

```

#define HALLpin    A1
#define encodPinA1    5
#define encodPinB1    6
double kp = 5 , ki = 1 , kd = 0.01 ,input = 0, output = 0, setpoint = 0;
volatile long encoderPos = 0;
long temp;
PID myPID(&input, &output, &setpoint, kp, ki, kd, DIRECT);
byte ZC = 0;
uint16_t alpha;
void setup() {
    Serial.begin(9600);
    pinMode(8, OUTPUT);
    pinMode(9, OUTPUT);
    pinMode(10, OUTPUT);
    pinMode(relay, OUTPUT);
    digitalWrite(relay, HIGH);
    digitalWrite(8, LOW);
    digitalWrite(9, LOW);
    digitalWrite(10, LOW);
    attachInterrupt(0, ZC_detect1, FALLING);
    attachInterrupt(0, ZC_detect2, FALLING);
    attachInterrupt(0, ZC_detect3, FALLING);
    pinMode(encodPinA1, INPUT_PULLUP);
    pinMode(encodPinB1, INPUT_PULLUP);
    attachInterrupt(0, encoder, FALLING);
    TCCR1B = TCCR1B & 0b11111000 | 1;
    myPID.SetMode(AUTOMATIC);
    myPID.SetSampleTime(1);
    myPID.SetOutputLimits(-255, 255);
}
void loop() {
    temp += (analogRead(pot) * 4);
    if (temp < 0) {
        encoderPos = 0;
        temp = 0;
    }
    setpoint = temp / 500;
    input = encoderPos ;
}

```

```

myPID.Compute();
pwmOut(output);
int value = analogRead(HALLpin); //giá trị số nguyên có giá trị trong khoảng
từ 0 đến 1023
int volt = value / 5 * 30; //giá trị cảm biến dùng cho 30A
Serial.println(volt);
if ( analogRead(A0) < 0) {
  digitalWrite(relay, LOW);
}
if (ZC == 1) {
  delayMicroseconds(alpha);
  digitalWrite(T11, HIGH);
  delayMicroseconds(2000);
  digitalWrite(T11, LOW);
  alpha = volt * temp ;
  ZC = 0;
}
if (ZC == 2) {
  delayMicroseconds(alpha);
  digitalWrite(T12, HIGH);
  delayMicroseconds(2000);
  digitalWrite(T12, LOW);
  alpha = volt * temp ;
  ZC = 0;
}
if (ZC == 3) {
  delayMicroseconds(alpha);
  digitalWrite(T21, HIGH);
  delayMicroseconds(2000);
  digitalWrite(T21, LOW);
  alpha = volt * temp ;
  ZC = 0;
}
if (ZC == 4) {
  delayMicroseconds(alpha);
  digitalWrite(T22, HIGH);
  delayMicroseconds(2000);
  digitalWrite(T22, LOW);
}

```

```

    alpha = volt * temp ;
    ZC = 0;
}
if (ZC == 5) {
    delayMicroseconds(alpha);
    digitalWrite(T31, HIGH);
    delayMicroseconds(2000);
    digitalWrite(T31, LOW);
    alpha = volt * temp ;
    ZC = 0;
}
if (ZC == 6) {
    delayMicroseconds(alpha);
    digitalWrite(T32, HIGH);
    delayMicroseconds(2000);
    digitalWrite(T32, LOW);
    alpha = volt * temp ;
    ZC = 0;
}
}
void pwmOut(int out) {
    if (out > 0) {
        float encoder = out;
    }
    else {
        float encoder = abs(out);
    }
}
void encoder() {

    if (PINB & 0b00000001)
        encoderPos++;
    // if (digitalRead(encodPinB1)==HIGH) count ++;
    else
        encoderPos--;
    // if (digitalRead(encodPinB1)==LOW) count --;
}
void ZC_detect1() {

```

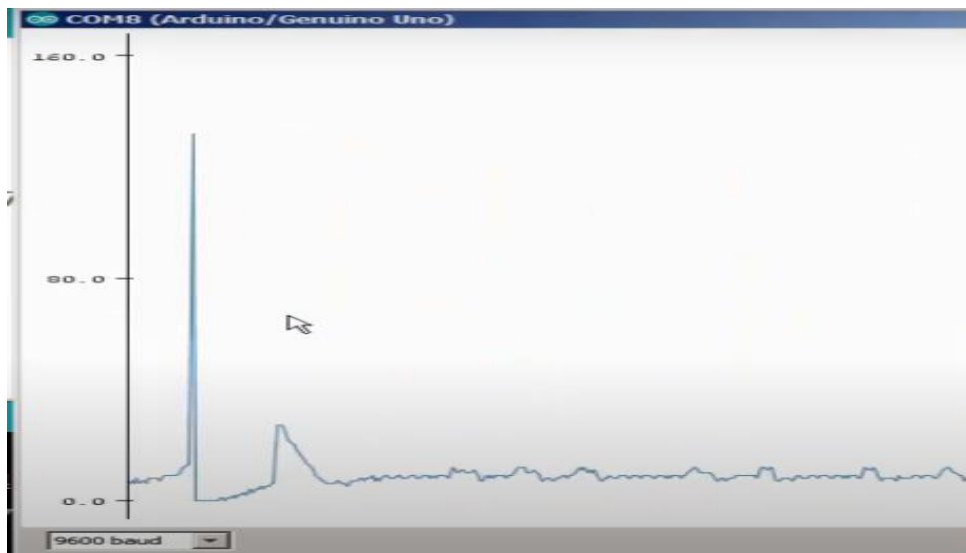
```

if (digitalRead(2))
  ZC = 1;
else
  ZC = 2;
}
void ZC_detect2() {
  if (digitalRead(3))
    ZC = 3;
  else
    ZC = 4;
}
void ZC_detect3() {
  if (digitalRead(4))
    ZC = 5;
  else
    ZC = 6;
}

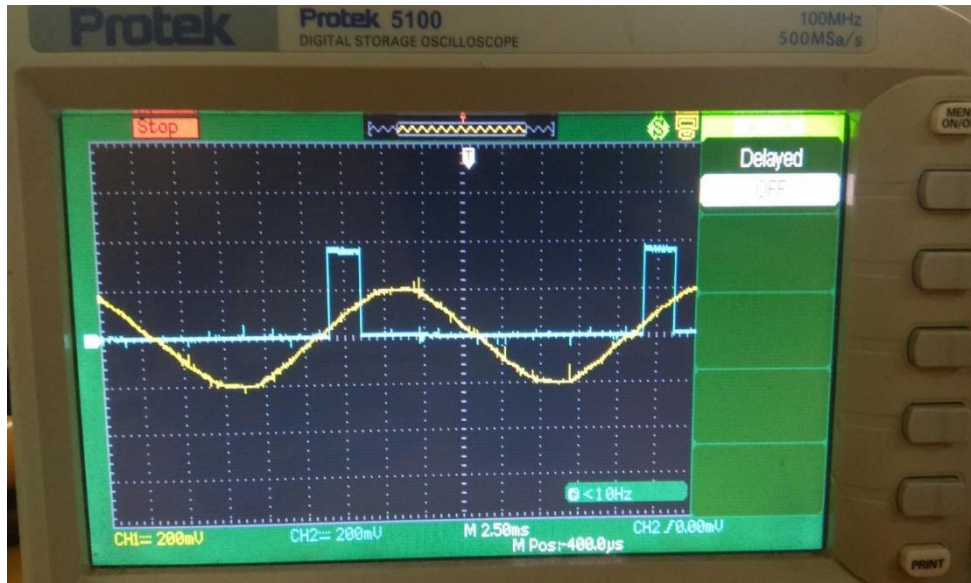
```

***Các kết quả thực nghiệm:**

Sau khi lập trình, kiểm chứng mạch thực và kết hợp quá trình mô phỏng ta thu được:



Hình 3.11: Đặc tính dòng điện



Hình 3.12: Xung đầu ra của một kênh điều khiển

CH1 (vàng): xung đồng bộ hình sin.

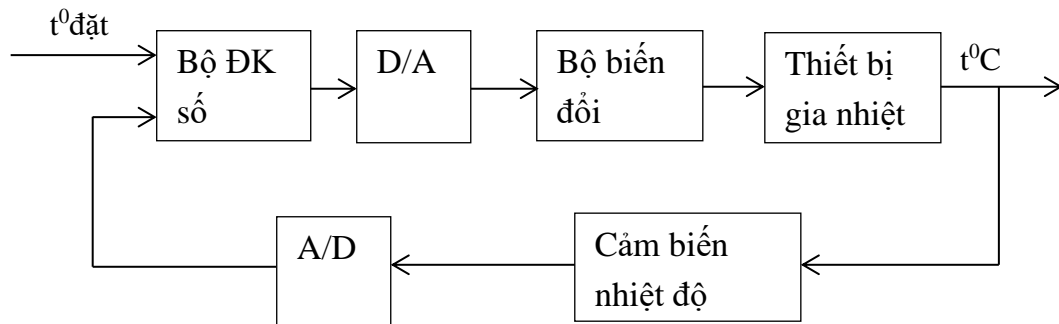
CH2(xanh): xung điều khiển số cho thyristor pha A (pha B,C tương tự)

Tương tự, sinh viên có thể lập trình điều khiển tốc độ qua thay đổi góc mở Thyristor trong các trường hợp sử dụng bộ biến đổi cầu 1pha, cầu 3 pha...

3.5 Hệ điều khiển quá trình

3.5.1 Các thiết bị cơ bản trong hệ thống điều khiển thiết bị gia nhiệt

Sơ đồ khối chức năng hệ thống điều khiển thiết bị gia nhiệt như sau:



Hình 3.3: Sơ đồ khối hệ thống điều khiển thiết bị gia nhiệt

Hệ thống điều khiển thiết bị gia nhiệt nhận nhiệt độ từ đối tượng bằng cảm biến nhiệt độ. Bộ cảm biến này sẽ chuyển nhiệt độ này thành mức điện áp tương ứng mức điện áp này ở dạng tín hiệu tương tự. Sau đó tín hiệu tương tự này sẽ được chuyển về dạng số bằng bộ chuyển đổi tương tự sang số (A/D) trước khi đưa vào bộ điều khiển. Bộ điều khiển này nhận được nhiệt độ đo, kiểm tra nhiệt độ xem đã đạt hay chưa, nếu nhiệt độ chưa đủ thì điều khiển tăng hoặc ngược lại, nhiệt độ cao hơn thì điều khiển giảm. Quá trình điều khiển bằng điện áp xuất ra, qua bộ chuyển đổi D/A được điện áp tương tự, đưa đến mạch điều khiển thyristor. Mạch điều khiển thyristor sẽ tạo ra xung để mở thyristor phù hợp với yêu cầu tăng hoặc giảm nhiệt độ.

***Cơ sở nguyên lý để thực hiện điều khiển thiết bị gia nhiệt:**

-Thực hiện điều khiển thiết bị gia nhiệt bằng bộ điều khiển PID số thiết kế và được cài đặt trên vi điều khiển.

- Sử dụng cảm biến nhiệt độ để phản hồi nhiệt độ.

- Giao tiếp thông qua các modul trên mạch điều khiển để hiển thị đặc tính.

- Trên mạch có các khối hiển thị để hiển thị các chế độ hoạt động của hệ thống cùng các thông số như nhiệt độ, điện áp...

Do vậy, để xây dựng một mô hình hệ thống điều khiển thiết bị gia nhiệt ta cần thiết kế các phần mạch sau:

3.5.1.1 Thiết bị gia nhiệt

Trong đời sống cũng như sản xuất, yêu cầu về sử dụng nhiệt năng rất lớn. Trong các ngành công nghiệp khác nhau, nhiệt năng dùng để nung, sấy, nhiệt luyện, nấu chảy... Nguồn nhiệt năng này được chuyển từ điện năng qua các lò điện là phổ biến vì nó rất thuận tiện, dễ tự động hoá điều chỉnh nhiệt độ trong lò. Trong sinh hoạt đời sống, nhiệt năng chủ yếu để đun, nấu, nướng, sưởi... Nguồn nhiệt năng cũng được chuyển từ điện năng qua các thiết bị điện như bàn là điện, bếp điện, nồi cơm điện, bình nóng lạnh... Đây là nguồn năng lượng sạch, không gây nên khói, bụi, không ảnh hưởng tới môi trường, sử dụng thuận tiện, dễ dàng.

Việc biến đổi điện năng thành nhiệt năng có nhiều cách: nhờ hiệu ứng Joule (lò điện trở, bếp điện), nhờ phóng điện hồ quang (lò hồ quang, hàn điện), nhờ tác dụng nhiệt của dòng điện xoáy Foucault thông qua hiện tượng cảm ứng điện từ (bếp từ)... Các thiết bị gia nhiệt dùng trong sinh hoạt trừ lò vi sóng và bếp từ, còn hầu hết dùng dây điện trở như bàn là, bếp điện, nồi cơm điện, siêu điện, bình nóng lạnh...

Những dây điện trở sử dụng thường là hợp kim Nikel-Crôm có điện trở suất $\rho = 1,1 \Omega\text{mm}^2/\text{m}$, nhiệt độ làm việc đến 1100°C . Các dây điện trở dùng để chế tạo các dụng cụ sinh hoạt thường được đặt trong ống kín, trong ống chèn chặt bằng chất chịu nhiệt, dẫn nhiệt và cách điện với vỏ ống. Việc đặt dây điện trở trong ống kín sẽ tránh hơi ẩm và ôxy lọt vào, giảm được sự ôxy hoá, tăng độ bền và tuổi thọ cho thiết bị gia nhiệt.

Trong phần thiết kế này, tập trung nghiên cứu, thiết kế về thiết bị gia nhiệt bằng lò điện trở.

a. Lò điện trở

Lò điện trở là thiết bị biến đổi điện năng thành nhiệt năng, dùng trong công nghệ nung nóng, nấu chảy vật liệu. Lò điện trở được dùng rất phổ biến trong nhiều ngành công nghiệp.

b. Nguyên lý làm việc

Phương pháp gia nhiệt bằng điện trở dựa trên định luật Joule –Lence. Khi cho dòng điện chạy qua dây dẫn, thì trên dây dẫn toả ra một nhiệt lượng. Nhiệt lượng này được tính theo biểu thức:

$$Q=I^2Rt$$

Trong đó:

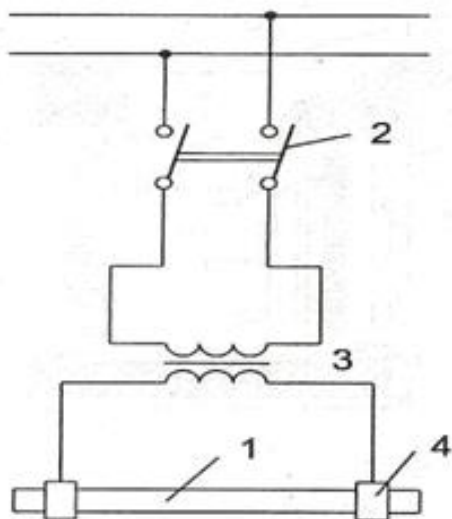
Q: Nhiệt lượng (J)

I: Cường độ dòng điện (A)

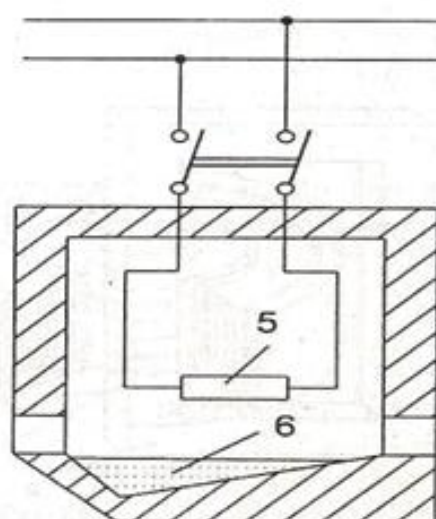
R: Điện trở (Ω)

t: Thời gian (s)

Theo phương pháp tỏa nhiệt, lò điện trở được chia thành 2 loại: Lò điện trở đốt nóng trực tiếp và lò điện trở đốt nóng gián tiếp. Sơ đồ nguyên lý làm việc của lò điện trở thể hiện trên hình 3.4.a,b.



Hìn 3.4a - Đốt nóng trực tiếp



Hìn 3.4a - Đốt nóng gián tiếp

3.5.1.2 Bộ biến đổi

Nhiệt độ của thiết bị gia nhiệt được điều chỉnh bằng cách thay đổi điện áp cấp cho dây đốt qua bộ biến đổi xoay chiều – xoay chiều

Trong phần thiết kế này, sử dụng bộ biến đổi xoay chiều- xoay chiều 3 pha hoặc 1 pha dùng tiristor.

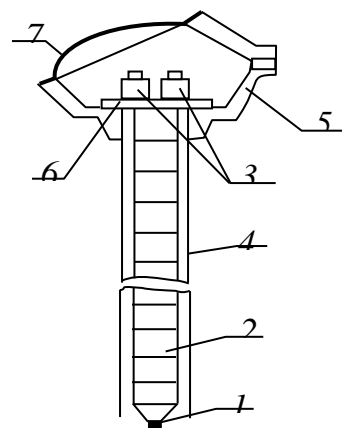
3.5.1.3 Cảm biến nhiệt độ

a. Cặp nhiệt điện:

Là cảm biến đo nhiệt độ, chuyển tín hiệu nhiệt độ sang tín hiệu điện dựa trên hiện tượng nhiệt điện. Khi có sự chênh lệch nhiệt điện ở hai đầu nối của hai dây dẫn bằng kim loại khác nhau làm xuất hiện một sức điện động. Nhiệt độ tăng làm sức điện động (điện áp) ra trên cặp kim loại cấu tạo nên nó. Để thuận tiện cho người sử dụng, các cặp nhiệt được chế tạo sẵn dưới dạng các can nhiệt điện. Miền đo của can nhiệt điện phụ

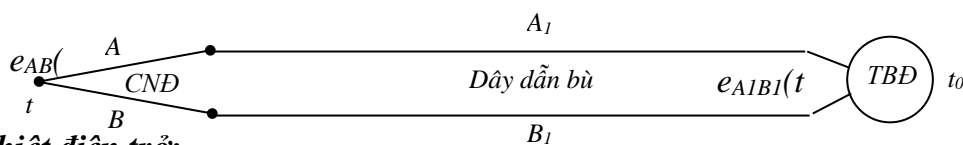
thuộc vào vật chế tạo. Đối với can nhiệt đồng/vàng-cooban có thể đo được từ -270°C đến 2700°C .

Can nhiệt điện có sơ đồ cấu trúc được mô tả trong hình 3.5. Đầu vào làm việc 1 của hai dây điện cực nhiệt được hàn chặt vào nhau. Các dây điện cực được lồng vào trong ống cách điện 2. Hai đầu tự do của hai dây điện cực nhiệt được gắn các cốt nối 3 thuận tiện cho việc ghép nối với bên ngoài. Vỏ bảo vệ 4 ngăn cản sự xâm thực của môi trường đo lên các dây điện cực nhiệt. Vỏ bảo vệ 4 được gắn chặt lên đầu nối 5 của can nhiệt điện. Hệ thống hai dây điện cực, ống cách điện 2 và cốt nối 3 cũng được gắn chặt lên đầu nối 5 qua tấm lót cách điện 6. Tấm lót cách điện 6 còn đóng vai trò ngăn cản nước xâm nhập vào trong lòng can nhiệt điện. Bộ phận 7 là nắp đậy. Can nhiệt điện được chế tạo với nhiều loại khác nhau. Chiều dài của can nhiệt điện cũng rất đa dạng đáp ứng nhu cầu sử dụng. Can nhiệt điện có chiều dài lớn nhất là 2m. Đường kính dây điện cực nhiệt lớn nhất là 3mm.



Hình 3.5: Can nhiệt điện

Để đo được nhiệt độ thì đầu tự do của cặp nhiệt điện phải có nhiệt độ ổn định, cách xa bề mặt được đốt nóng. Thông thường, đầu tự do của cặp nhiệt điện phải đưa về phòng điều khiển trung tâm để ghép nối với thiết bị đo qua dây bù. Sơ đồ hệ thống nối cặp nhiệt điện với thiết bị đo trong công nghiệp như sau:



b. Nhiệt điện trở:

Là loại cảm biến nhiệt độ mà khi nhiệt độ tăng thì điện trở lại giảm. Nhiệt điện trở có độ phân giải cao hơn độ phân giải của điện trở kim loại khoảng 10 lần. Các nhiệt điện trở thông thường được chế tạo từ các ôxit bán dẫn đa tinh thể. Miền đo phụ thuộc vào loại nhiệt điện trở, có thể từ -273°C đến 300°C .

3.5.1.4 Phần cứng

a. Khối vi xử lý

Khối vi xử lý được thiết kế thành một Module riêng nhằm tạo ra khả năng linh hoạt trong việc thay đổi cấu hình điều khiển (chẳng hạn khi cần thay bằng loại vi xử lý mới). Khối này được thiết kế dựa trên vi điều khiển chuyên dụng.

Vi điều khiển là một máy tính được tích hợp trên một chip, nó thường được sử dụng để điều khiển các thiết bị điện tử. Vi điều khiển, thực chất, là một hệ thống bao gồm một vi xử lý có hiệu suất đủ dùng và giá thành thấp (khác với các bộ vi xử lý đa năng dùng trong máy tính) kết hợp với các khối ngoại vi như bộ nhớ, các module vào/ra,

các module biến đổi số sang tương tự và tương tự sang số... Ở máy tính thì các module thường được xây dựng bởi các chip và mạch ngoài.

Vi điều khiển có nhiều bit khác nhau giống như vi xử lý (hiện nay có các loại vi điều khiển 4bit, 8bit, 16bit, 32bit, 64bit và 128 bit)

Bên trong vi điều khiển chứa đầy đủ tất cả các tính năng cần thiết cho một hệ thống máy tính và các chức năng như một máy tính mà không cần thêm các bộ phận kỹ thuật số bên ngoài.

Hầu hết các chân trong chip vi điều khiển có thể được lập trình bởi người dùng.

Vi điều khiển có khả năng xử lý các hàm logic.

Tốc độ và hiệu suất cao.

Cấu trúc on-chip ROM trong vi điều khiển giúp bảo mật firmware tốt hơn.

Dễ thiết kế với chi phí thấp và kích thước nhỏ.

Để lập trình cho chip vi điều khiển, người sử dụng được cung cấp một phần mềm lập trình riêng cho mỗi chip vi điều khiển. Ngoài ra, để cài đặt được chương trình điều khiển vào chip có thể sử dụng trình gỡ rối. Phần mềm phát triển được xây dựng trên cơ sở hướng đối tượng với cấu trúc module hóa các khối chức năng. Việc lập trình cho chip tùy thuộc vào người sử dụng thông qua một số thư viện chuẩn. Người sử dụng thiết lập cấu hình trên chip chỉ đơn giản bằng cách muốn chip có chức năng gì thì chọn chức năng đó và đặt vào khối tài nguyên số hoặc tương tự, hoặc cả hai tùy theo từng chức năng. Việc thiết lập ngắt, thiết lập chế độ hoạt động các chân vào ra tùy thuộc vào người sử dụng khi thiết kế và lập trình cho chip.

Các loại vi điều khiển được sử dụng trên thị trường nước ta hiện nay: Vi điều khiển 8051, Vi điều khiển AVR, Vi điều khiển PIC, Vi điều khiển ARM...

Trong phần thiết kế này, tập trung nghiên cứu về Arduino.

Arduino là một bo mạch vi xử lý được dùng để lập trình tương tác với các thiết bị phân cứng như cảm biến, thiết bị gia nhiệt, đèn hoặc các thiết bị khác chủ yếu dựa trên vi điều khiển AVR. Đặc điểm nổi bật của Arduino là môi trường phát triển ứng dụng cực kỳ dễ sử dụng, với một ngôn ngữ lập trình có thể học một cách nhanh chóng ngay cả với người ít am hiểu về điện tử và lập trình. Và điều làm nên hiện tượng Arduino chính là mức giá rất thấp và tính chất nguồn mở từ phần cứng tới phần mềm.

b. Mạch điều khiển trung tâm

Khối mạch trung tâm có nhiệm vụ quản lý, giám sát, điều khiển hoạt động của toàn bộ các modul sử dụng trong mạch có nghĩa là mọi hoạt động của hệ thống trên cơ sở vi điều khiển.

Khối mạch điều khiển trung tâm kết nối khối vi xử lý với các thành phần khác trong hệ thống (mạch đồng bộ hóa, các đầu vào điều khiển và phản hồi...)

Mạch điều khiển trung tâm bao gồm các khối sau:

- Khối vi xử lý.

- Khối đồng bộ hóa.
- Khối đầu vào tương tự dùng để đo các đại lượng: nhiệt độ, điện áp...
- Khối đầu ra tương tự bao gồm cả nguồn áp và nguồn dòng.
- Khối đầu vào số dùng kết hợp với mạch không chế.
- Khối đầu ra số bao gồm cả mạch điều khiển rơ le.
- Khối thời gian thực.
- Khối nhớ ngoài.
- Khối giao tiếp bàn phím và hiển thị LED.
- Khối nguồn.

...

c. Mạch công suất xung

Mạch công suất xung và được thiết kế theo từng kênh điều khiển riêng biệt.

3.5.2. Tổng hợp bộ điều khiển cho hệ điều khiển thiết bị gia nhiệt

$$\text{Hàm truyền của thiết bị gia nhiệt: } W(s) = \frac{T(s)}{U(s)} = \frac{K}{1+Ts}$$

$$\text{Hàm truyền của bộ biến đổi: } W_b(s) = \frac{U(s)}{U_{dk}(s)} = \frac{K_b}{1+\tau_b s}$$

Hàm truyền của hệ hở gồm hai khâu mắc nối tiếp (Đối tượng điều khiển):

$$G(s) = W(s)W_b(s)$$

***Tổng hợp bộ điều khiển cho thiết bị gia nhiệt trong phòng thí nghiệm:**

Mô hình toán học của thiết bị gia nhiệt trong phòng thí nghiệm được biểu diễn bằng mô hình hàm truyền:

$$W(s) = \frac{T(s)}{U(s)} = \frac{K}{1+Ts} = \frac{4,69}{1+272,51s}$$

Nhiệt độ của thiết bị gia nhiệt được điều chỉnh bằng cách thay đổi điện áp cấp cho dây đốt qua bộ biến đổi xoay chiều -xoay chiều một pha từ nguồn xoay chiều 36V, điện áp điều khiển bộ biến đổi này có điện áp từ 0÷10V. Từ đó ta xác định được hàm truyền bộ biến đổi:

$$W_b(s) = \frac{U(s)}{U_{dk}(s)} = \frac{K_b}{1+\tau_b s}$$

$$K_b : \text{Hệ số khuếch đại bộ biến đổi; } K_b = \frac{36}{10} = 3,6$$

$$\tau_b : \text{Hằng số thời gian } \tau_b = \frac{1}{2mf} = \frac{1}{2.2.50} = 0,005 \text{ (s)}$$

m: số lần đập mạch trong 1 chu kì điện áp nguồn, m=2

f: tần số điện áp nguồn, f=50

$$\text{Thay số ta có: } W_b(s) = \frac{U(s)}{U_{dk}(s)} = \frac{3,6}{1+0,005s}$$

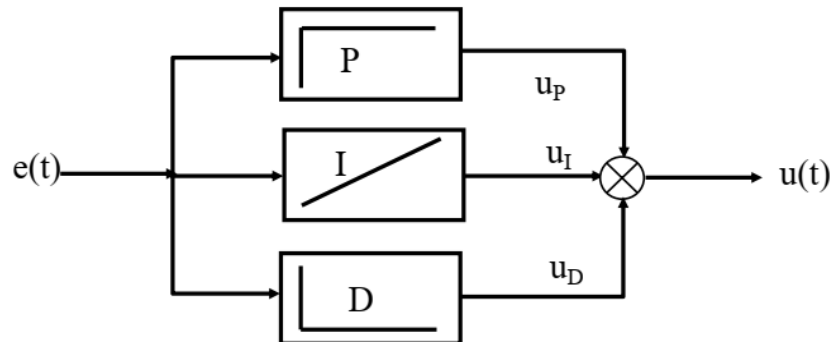
Hàm truyền của hệ hở gồm hai khâu mắc nối tiếp:

$$G(s) = W(s)W_b(s) = \frac{4,69}{1 + 272,51s} \frac{3,6}{1 + 0,005s} = \frac{16,88}{(1 + 272,51s)(1 + 0,005s)}$$

Như vậy đối tượng điều khiển (bao gồm bộ biến đổi và thiết bị gia nhiệt) là lớp đối tượng 02 khâu quán tính bậc 1 mắc nối tiếp. Một số phương pháp tổng hợp bộ điều khiển PID cho thiết bị gia nhiệt sẽ được trình bày dưới đây.

3.5.2.1 Tổng quan về bộ điều khiển PID

Bộ điều khiển PID được viết tắt từ 3 thành phần cơ bản trong bộ điều khiển: khuếch đại tỷ lệ (P), tích phân (I) và vi phân (D).



Hình 3.6: Cấu trúc bộ điều khiển PID

Biến đổi tương đương:

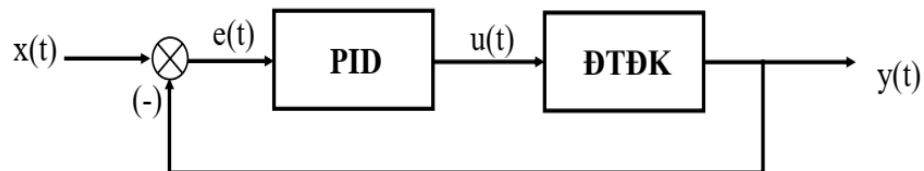


Với $u(t) = u_P + u_I + u_D$

Bộ điều khiển PID đảm bảo bổ sung hoàn hảo 3 trạng thái, 3 tính cách khác nhau:

- Phục tùng và làm việc chính xác (P)
- Làm việc có tích lũy kinh nghiệm (I)
- Có khả năng phản ứng nhanh nhạy và sáng tạo (D)

Bộ điều khiển PID được ứng dụng rất rộng rãi đối với các đối tượng SISO theo nguyên lý phản hồi như hình vẽ:



Hình 3.7: Sơ đồ cấu trúc bộ điều khiển PID

Bộ điều khiển PID được mô tả: $u(t) = K_p \left[e(t) + \frac{1}{T_i} \int e(t) dt + T_D \frac{de(t)}{dt} \right]$

$$W_{PID}(s) = \frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} + T_D s \right)$$

Việc xác định các tham số K_P , T_I , T_D quyết định chất lượng hệ thống điều khiển. Các phương pháp thường gặp để xác định tham số bộ điều khiển:

- Phương pháp thực nghiệm dựa trên hàm $h(t)$
- Phương pháp thiết kế trên miền tần số.

3.5.2.2. Phương pháp thực nghiệm dựa trên hàm $h(t)$

a. Phương pháp hằng số thời gian tổng nhỏ nhất của Kuhn

Phương pháp thiết kế của Kuhn là phương pháp thực nghiệm rất dễ dàng thực hiện mà không cần biết chính xác mô hình đối tượng. Thay vào đó, ta phải xác định được các tham số đặc trưng của đối tượng điều khiển bằng phương pháp thực nghiệm. Các tham số cần xác định từ thực nghiệm là: Hệ số khuếch đại của đối tượng K , hằng số thời gian tổng nhỏ nhất T_s . Nếu có mô hình hàm truyền của đối tượng, ta có thể xác định các tham số này từ mô hình với:

$$T_s = \frac{\sum_{j=1}^m T_{Dj}}{\sum_{i=1}^n T_i + \tau}$$

Trong đó:

T_{Dj} : Hằng số thời gian vi phân

T_i : Hằng số thời gian quán tính

τ : Hằng số thời gian trễ

Ta có hằng số thời gian nhỏ nhất của hệ thống gia nhiệt là:

$$T_s = 272.51 + 0.005 = 272.515$$

Hệ số khuếch đại của đối tượng : $K=16,88$

Sau khi có được các tham số của đối tượng từ thực nghiệm ta có thể tra bảng thiết kế của phương pháp để xác định tham số bộ điều khiển cần tìm.

Bảng 1: Tổng hợp bộ điều khiển theo Kuhn

Luật điều khiển $W_{đk}(s)$	Hệ số tỉ lệ (K_P)	Hằng số thời gian tích phân (T_I)	Hằng số thời gian vi phân (T_D)
PI: $K_p (1+1/T_I.s)$	0,5/ K	0,5 T_s	-
PID: $K_p (1+1/T_I.s+T_D.s)$	1/ K	0,8 T_s	0,194 T_s

Tra bảng 1 ta có bộ điều khiển PID tương tự cho thiết bị gia nhiệt:

$$K_p = \frac{1}{K} = \frac{1}{16,88} = 0,59$$

$$K_I = 0,8T_s = 0,8.272,52 = 218,01$$

$$K_D = 0,19T_s = 0,19.272,52 = 52,87$$

$$W_{PID}(s) = K_p + \frac{K_I}{s} + K_D s = 0,59 + \frac{218,01}{s} + 52,87s$$

Bộ điều khiển PID số:

Hàm truyền liên tục của bộ điều khiển PID có thể được viết dưới dạng sau:

$$W_{PID}(s) = K_p + \frac{K_I}{s} + K_D s$$

Để chuyển từ bộ PID liên tục sang bộ PID số ta có thể sử dụng phương pháp chuyển gần đúng từng thành phần của bộ PID từ liên tục sang dạng rời rạc như sau:

- Thành phần tỉ lệ được giữ nguyên.
- Thành phần tích phân được lấy gần đúng theo Tustin: $\frac{K_I}{s} = \frac{K_I T(z+1)}{2(z-1)}$
- Thành phần vi phân được lấy gần đúng: $K_d = \frac{2K_D(z+1)}{Tz}$

Như vậy, hàm truyền rời rạc của bộ PID số là:

$$\begin{aligned} W_{PID}(z) &= K_p + \frac{K_I T(z+1)}{2(z-1)} + \frac{2K_D(z+1)}{Tz} \\ &= \frac{2TK_p z(z-1) + K_I T^2 z(z+1) + 2K_D(z-1)^2}{2Tz(z-1)} \\ &= \frac{\left(K_p + 0,5K_I T + \frac{K_D}{T}\right)z^2 - \left(K_p - 0,5K_I T + 2\frac{K_D}{T}\right)z + \frac{K_D}{T}}{z(z-1)} \\ &= \frac{\left(0,59 + 0,5 \cdot 218,01T + \frac{52,87}{T}\right)z^2 - \left(0,59 - 0,5 \cdot 218,01T + 2\frac{52,87}{T}\right)z + \frac{52,87}{T}}{z(z-1)} \end{aligned}$$

Với T là chu kỳ trích mẫu.

b. Phương pháp Ziegler- Nichols 1

Ziegler và Nichols đã đưa ra hai phương pháp thực nghiệm để xác định tham số bộ điều khiển động. Mô hình thiết bị gia nhiệt thích hợp với phương pháp Ziegler-Nichols 1. Từ bảng tra theo Ziegler -Nichols 1 ta xác định được các tham số bộ điều khiển. Bảng thiết kế theo phương pháp Ziegler-Nichols được biểu diễn trong bảng sau:

Bảng 2: Tổng hợp bộ điều khiển theo Ziegler- Nichols

Luật điều khiển $W_{dk}(s)$	Hệ số tỉ lệ (K_P)	Hằng số thời gian tích phân (T_I)	Hằng số thời gian vi phân (T_D)
P: K_p	$T/K \cdot \tau$	-	-
PI: $K_p (1+1/T_I \cdot s)$	$0,9T/K \cdot \tau$	$0,33\tau$	-
PID: $K_p (1+1/T_I \cdot s+T_D \cdot s)$	$1,2T/K \cdot \tau$	$2,5\tau$	$0,5\tau$

Theo bảng 2 xác định được hàm truyền liên tục của bộ điều khiển PID cho hệ thống thiết bị gia nhiệt như sau:

$$W_{PID}(s) = 3936,98 + \frac{314958,05}{s} + 9,84s$$

Bộ điều khiển PID số:

Hàm truyền rời rạc của bộ PID số là:

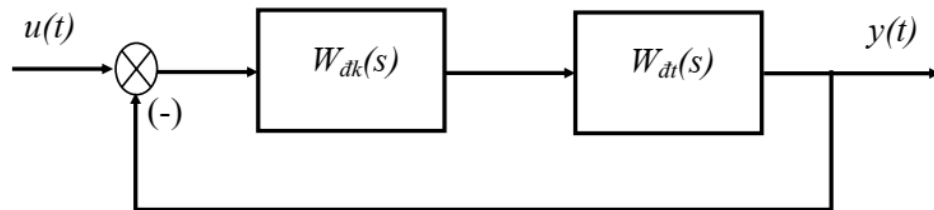
$$\begin{aligned}
W_{PID}(z) &= K_p + \frac{K_I T(z+1)}{2(z-1)} + \frac{2K_D(z+1)}{Tz} \\
&= \frac{2TK_p z(z-1) + K_I T^2 z(z+1) + 2K_D(z-1)^2}{2Tz(z-1)} \\
&= \frac{\left(K_p + 0,5K_I T + \frac{K_D}{T}\right)z^2 - \left(K_p - 0,5K_I T + 2\frac{K_D}{T}\right)z + \frac{K_D}{T}}{z(z-1)} \\
&= \frac{\left(3936,98 + 157479,02T + \frac{9,84}{T}\right)z^2 - \left(3936,98 - 157479,02T + 2\frac{9,84}{T}\right)z + \frac{9,84}{T}}{z(z-1)}
\end{aligned}$$

Với T là chu kì trích mẫu.

3.5.2.3 Phương pháp thiết kế điều khiển ở miền tần số:

a. Nguyên tắc thiết kế:

Một hệ thống điều khiển được mô tả:



Hình 3.8: Sơ đồ hệ thống điều khiển

Bài toán đặt ra thiết kế bộ điều khiển sao cho tín hiệu ra $y(t)$ phải bám được tín hiệu vào $u(t)$ trong khoảng thời gian ngắn nhất. Một cách lý tưởng thì hàm truyền hệ kín:

$$W_k(s) = \frac{W_{dk}(s)W_{dt}(s)}{1 + W_{dk}(s)W_{dt}(s)}$$

Hay $|W_k(j\omega)| = 1$

Vậy ta cần phải xác định cấu trúc và tham số bộ điều khiển với mọi $\omega \in \Omega_{\max}$ để có $|W_k(j\omega)| = 1$ với ω càng lớn càng tốt.

Thiết kế điều khiển ở miền tần số có thể sử dụng phương pháp tối ưu modul hoặc phương pháp tối ưu đối xứng. Tuy nhiên, phương pháp tối ưu đối xứng chỉ phù hợp với lớp đối tượng có khâu tích phân. Với đối tượng là khâu quán tính bậc nhất như thiết bị gia nhiệt, ta sử dụng phương pháp modul tối ưu để thiết kế bộ điều khiển cho đối tượng.

b. Phương pháp modul tối ưu:

Phương pháp tối ưu modul được thực hiện theo ý tưởng: Chọn cấu trúc và tham số của bộ điều khiển sao cho module vector đặc tính tần số của hệ kín $|W_k(j\omega)| = 1$ với mọi ω và được gọi là thiết kế bộ điều khiển sao cho vector đặc tính tần số của hệ kín là tối ưu.

Giả sử hệ thống có hàm truyền hệ hở là $W_H(s)$ (Hình 3.9). Ta phải tìm khâu hiệu chỉnh $W_{HC}(s)$ sao cho hàm truyền hệ thống kín với phản hồi đơn vị (-1).

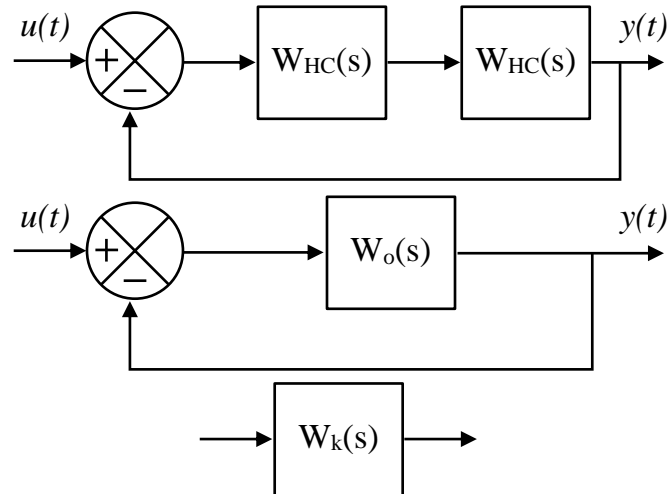
Thoả mãn điều kiện chuẩn Modul tối ưu sau: $W_k(s) = \frac{1}{2\tau^2 s^2 + 2\tau s + 1}$

Trong đó:

$$W_k(s) = \frac{W_o(s)}{1 + W_o(s)}$$

$$W_o(s) = W_{HC}(s)W_H(s)$$

Thay vào ta tìm được: $W_{HC}(s) = \frac{1}{2\tau s(\tau s + 1)W_H(s)}$



Hình 3.9: Tổng hợp bộ điều khiển bằng phương pháp modul tối ưu

Để thiết bị hiệu chỉnh đơn giản ta chọn τ thỏa mãn điều kiện về thời gian quá độ và trùng với hằng số thời gian nhỏ nào đó của W_H (bù được khâu có hằng số thời gian lớn).

Áp dụng cho hệ thống điều khiển thiết bị gia nhiệt ta có:

$$W_{PI}(s) = \frac{1}{2\tau s(2\tau s + 1)G(s)} = \frac{1}{2\tau s(2\tau s + 1) \frac{16,88}{(1 + 272,51s)(1 + 0,005s)}}$$

Ta chọn $\tau = 0,005$ (s), bộ điều khiển là:

$$W_{PI}(s) = \frac{1 + 272,51s}{2.16,88.0,005s} = 1614,36 + \frac{1}{0,17s} = 1614,36 + \frac{5,92}{s}$$

Bộ điều khiển PI số:

Hàm truyền liên tục của bộ điều khiển PI có thể được viết dưới dạng sau:

$$W_{PI}(s) = K_p + \frac{K_I}{s}$$

Để chuyển từ bộ PI liên tục sang bộ PI số ta có thể sử dụng phương pháp chuyển gần đúng từng thành phần của bộ PI từ liên tục sang dạng rời rạc như sau:

- Thành phần tỉ lệ được giữ nguyên.

- Thành phần tích phân được lấy gần đúng theo Tustin: $\frac{K_I}{s} = \frac{K_I T(z+1)}{2(z-1)}$

Như vậy, hàm truyền rời rạc của bộ PI số là:

$$W_{PID}(z) = K_p + \frac{K_I T(z+1)}{2(z-1)} = \frac{(2K_p +)z + (K_I T - 2K_p)}{2(z-1)}$$

$$= \frac{(3228,72 + 5,92T)z + (5,92T - 3228,72)}{2(z-1)}$$

Với T: là chu kỳ trích mẫu

3.5.3 Thiết kế và lập trình điều khiển

Thiết kế bộ điều khiển sử dụng phần mềm mô phỏng Proteus:

Các linh kiện sử dụng trong hệ thống:

- + ARDUINO
- + DIODE
- + LED YELLOW
- + LM393
- + HEATING EQUIPMENT
- + NPN
- + PC817
- + POT
- + RES
- + RT114012F
- + THYRISTOR
- + TRAN 3PHASE
- + VSINE
- + OSCILLOSCOPE
- + 5V
- + GND

Có thể sử dụng 1 trong các arduino thông dụng hiện nay: Arduino Uno R3; Arduino Nano; Arduino Mega 2560

*Lập trình điều khiển:

Sử dụng phần mềm Arduino IDE để lập trình điều khiển.

Tương tự, sinh viên có thể viết chương trình điều khiển thay đổi góc mở Thyristor trong bộ biến đổi xoay chiều-xoay chiều 3 pha để khống chế nhiệt độ của thiết bị gia nhiệt.

- Tham khảo schematic trên mã nguồn mở :

<https://www.arduino.cc/>

- Tham khảo dạng ngôn ngữ lập trình C++ và lệnh qua mã nguồn:

<https://www.arduino.cc/reference/en/>

- Tham khảo thư viện cho thiết bị sử dụng qua mã nguồn mở :

<https://github.com/>

TÀI LIỆU THAM KHẢO

1. Giáo trình Thiết kế hệ điều khiển số của bộ môn Tự động hóa – Khoa Điện
2. Giáo trình Lý thuyết điều khiển tự động của bộ môn Tự động hoá - Khoa Điện
3. Nguyễn Thương Ngô; Lý thuyết tự động thông thường và hiện đại - Quyển 2 hệ xung-số; NXB Khoa học và kỹ thuật, Hà Nội, 2003.
4. Nguyễn Thị Phương Hà, Huỳnh Thái Hoàng; Lý thuyết điều khiển tự động, NXB ĐH Quốc gia TP. Hồ Chí Minh, 2002
5. Lê Văn Doanh - Nguyễn Thế Công - Nguyễn Trung Sơn – Cao Văn Thành; Điều khiển số máy điện; NXB Khoa học và kỹ thuật, Hà Nội, 1999.
6. Nguyễn Như Hiền - Lại Khắc Lãi; Điều khiển số; NXB Khoa học và kỹ thuật, Hà Nội, 2007.
7. M. Sami Fadali, Antonio Visioli; Digital Control Engineering Analysis and Design, Elsevier, 2013.