

ĐẠI HỌC BÁCH KHOA THÀNH PHỐ HỒ CHÍ MINH
KHOA KỸ THUẬT HÓA HỌC
BỘ MÔN QUÁ TRÌNH – THIẾT BỊ



BÁO CÁO
THÍ NGHIỆM CƠ SỞ ĐIỀU KHIỂN QUÁ TRÌNH
MSMH 3342

BÁO CÁO THÍ NGHIỆM

Sinh viên: Nguyễn Phúc Vinh - 21115299

Trần Minh Thiện - 2112358

Lê Tú Hiếu - 2113351

GVHD: TS. Bùi Ngọc Pha

Lớp: L01

Thành phố Hồ Chí Minh, ngày 13 tháng 05 năm 2024

MỤC LỤC

BÀI THÍ NGHIỆM 2: THIẾT BỊ ĐIỀU KHIỂN VÀ LẬP TRÌNH VI ĐIỀU KHIỂN	1
1. Mục đích	1
2. Thiết bị thí nghiệm	1
2.1 Bo mạch Arduino.....	2
2.2 Arduino IDE.....	3
3. Thực hiện thí nghiệm	5
3.1 Thí nghiệm 1: Thực hiện mô phỏng động cơ khuấy	5
3.2 Thí nghiệm 2: Thực hiện mô phỏng điều khiển động cơ khuấy.....	7
3.3 Thí nghiệm 3: Mô phỏng vận hành điều khiển hệ thống chung cát	8
3.4 Thí nghiệm 4: Lập trình điều chỉnh tự động nhiệt độ nhập liệu vào tháp chung cát sử dụng bộ điều khiển ON-OFF	12
3.5 Thí nghiệm 5: Lập trình điều chỉnh tự động mức chất lỏng đáy tháp chung cát sử dụng bộ điều khiển PID.....	12
4. Kết luận và kiến nghị.....	14
5. Tài liệu tham khảo	14
BÀI THÍ NGHIỆM 3: KHẢO SÁT CÁC BỘ ĐIỀU KHIỂN ON-OFF VÀ PID .	15
1. Mục đích thí nghiệm.....	15
2. Cơ sở lý thuyết	15
2.1. Bộ điều khiển ON-OFF	15
2.2. Bộ điều khiển liên tục PID	16
3. Mô tả thiết bị thí nghiệm.....	19
3.1. Lưu đồ P&ID	19
3.2. Sơ đồ khái niệm hệ thống điều khiển quá trình	20

4. Quy trình thực hiện	20
4.1. Thí nghiệm 1: Khảo sát bộ điều khiển ON/OFF.....	20
4.2. Thí nghiệm 2: Khảo sát các bộ điều khiển P, PI và PID	21
5. Kết quả và bàn luận.....	22
5.1. Thí nghiệm 1	22
5.2. Thí nghiệm 2	23
6. Kết luận và kiến nghị.....	26
7. Tài liệu tham khảo	27
BÀI THÍ NGHIỆM 4: THIẾT KẾ HỆ THỐNG ĐIỀU CHỈNH CÁC ĐẠI LƯỢNG CƠ BẢN.....	28
1. Tóm tắt.....	28
2. Cơ sở lý thuyết	28
2.1. Ảnh hưởng các thông số bộ điều khiển	28
2.2. Thiết kế bộ điều khiển	30
3. Mô tả thiết bị thí nghiệm.....	32
4. Quy trình thực hiện	33
5. Kết quả và bàn luận.....	33
5.1. Thí nghiệm 1: Khảo sát sự ảnh hưởng các thông số bộ điều khiển PID ..	33
5.2. Bàn luận	34
6. Kết luận và kiến nghị.....	34
7. Trả lời câu hỏi	35
8. Tài liệu tham khảo	37

BÀI THÍ NGHIỆM 2: THIẾT BỊ ĐIỀU KHIỂN VÀ LẬP TRÌNH VI ĐIỀU KHIỂN

Tóm tắt: Thí nghiệm này nhằm giúp chúng ta hiểu sâu hơn về chức năng của bộ vi điều khiển và cách lập trình các thuật toán điều khiển. Bằng cách thiết lập sơ đồ kết nối các thành phần trong hệ thống điều khiển và thiết kế hệ thống tự động để điều khiển quá trình (khuấy, chưng cất, điều khiển mực chất lỏng, nhiệt độ nhập liệu), chúng ta có thể áp dụng kiến thức của mình vào các ứng dụng thực tế. Mục tiêu cuối cùng là xây dựng một hệ thống tự động hoạt động hiệu quả, từ đó giúp tăng cường kiến thức và kỹ năng trong lĩnh vực điều khiển quá trình này.

1. Mục đích

- Có kiến thức về chức năng của bộ vi điều khiển.
- Có khả năng lập trình các thuật toán điều khiển cho bộ điều khiển.
- Xác định sơ đồ kết nối giữa các thành phần trong hệ thống điều khiển.
- Thiết kế hệ thống điều khiển tự động để điều chỉnh quá trình khởi động và dừng trong quá trình chưng cất.

2. Thiết bị thí nghiệm

Trong lĩnh vực công nghiệp hóa chất và dầu khí, hệ thống điều khiển đóng vai trò quan trọng trong việc điều chỉnh, giám sát và tự động hóa các quy trình sản xuất.

Các chức năng của hệ thống điều khiển bao gồm điều chỉnh, điều khiển trình tự, điều khiển rò rỉ rạc và giám sát vận hành.

Trong bài thí nghiệm này, chúng ta sử dụng bộ điều khiển Arduino, một bo mạch vi điều khiển chuẩn hóa, được sử dụng rộng rãi và được hỗ trợ bởi cộng đồng người dùng ở Việt Nam và toàn cầu.

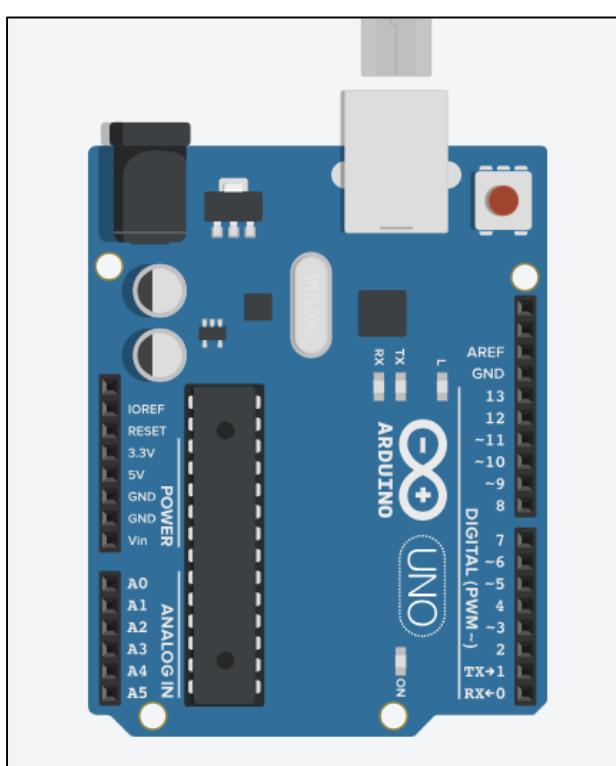
Phần mềm Arduino IDE cùng với công cụ mô phỏng trực tuyến www.tinkercad.com đóng vai trò quan trọng trong việc lập trình và mô phỏng các thuật toán điều khiển.

2.1 Bo mạch Arduino

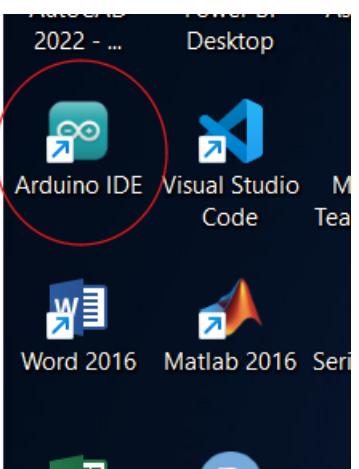
Các mạch Arduino của hãng Arduino có nhiều phiên bản khác nhau như Mega, Uno, Micro, Nano v.v. Mặc dù có những điểm khác biệt về kích thước và tính năng, nhưng cấu trúc chung của chúng vẫn tương tự nhau. Dưới đây là mô tả về mạch Nano được giới thiệu trên trang web của Arduino, nhưng đồng thời cũng áp dụng cho các phiên bản khác:

- **Khối Nguồn (POWER):** Mạch Arduino có thể được cấp nguồn từ bên ngoài với điện áp trong khoảng từ 5V đến 9V.
- **Khối Chức Năng Nhận và Xuất Tín Hiệu (INPUT và OUTPUT):** Mạch có các chân tín hiệu (PIN) để nhận và xuất tín hiệu. Các chân này có thể được sử dụng để kết nối với các cảm biến, bộ điều khiển và các thiết bị khác.
- **Khối Chức Năng Truyền Dữ Liệu (Serial):** Mạch hỗ trợ giao tiếp dữ liệu thông qua giao tiếp Serial, cho phép nó kết nối với các thiết bị ngoại vi khác như máy tính, cảm biến, hoặc các mạch điều khiển khác.

Các chân (PIN) trên mạch Arduino có khả năng nhận và xuất tín hiệu. Các chân kỹ thuật số (Digital) có thể hoạt động với các giá trị điện thế từ 0V đến 5V, tương ứng với các giá trị LOW và HIGH. Các chân Analog (A) có thể nhận giá trị điện thế từ 0V đến 5V. Ngoài ra, một số chân như ~D3, ~D5, ~D6 v.v. cũng có khả năng tạo ra các xung điện.



2.2 Arduino IDE



Integrated Development Environment (IDE) là môi trường phát triển tích hợp được sử dụng để lập trình cho các bo mạch Arduino. IDE này có thể được tải về miễn phí từ trang web: www.arduino.cc và cài đặt trên máy tính.

Giao diện của IDE thường bao gồm ba phần chính:

Vùng Các Nút Lệnh (1): Đây là nơi chứa các nút lệnh và các tính năng hỗ trợ việc lập trình như nút lưu, nút biên dịch, và nút tải chương trình lên bo mạch.

Vùng Viết Chương Trình (2): Đây là nơi bạn viết mã nguồn cho chương trình của mình bằng ngôn ngữ lập trình Arduino. Các chức năng như tô màu cú pháp và gợi ý từ khóa thường được hỗ trợ ở đây.

Vùng Thông Báo Trạng Thái (3): Phần này thường hiển thị các thông báo từ IDE như lỗi biên dịch, thông báo thành công khi tải chương trình lên bo mạch, và các thông tin khác về quá trình lập trình và thử nghiệm.

Giao diện này giúp người dùng dễ dàng lập trình và thử nghiệm chương trình trên các bo mạch Arduino một cách thuận tiện.

The image shows a screenshot of the Arduino IDE version 2.3.2. The interface is divided into three main sections:

- Section 1 (Top):** Contains the toolbar with icons for file operations, a dropdown menu for boards (set to "Arduino Nano"), and a status bar showing the sketch name "Bai2_TN3_TNCSDKQT.ino".
- Section 2 (Middle):** The code editor area where the following C++ code is written:

```
1 int congtacon = 2;
2 int pump3 = 3, pump2 = 4, pump1 = 5;
3 int van2 = 6, van1 = 7, van0 = 8;
4 int re2 = 9, re1 = 10, re0 = 11;
5 int i = 0;
6
7 void chutrinhON() {
8     digitalWrite(pump1, 1); digitalWrite(van0, 1); delay(3000);
9     digitalWrite(van0, 0); digitalWrite(re0, 1); digitalWrite(van2, 1); delay(6000);
10    digitalWrite(van1, 1); digitalWrite(re1, 1); delay(1000);
11    digitalWrite(pump2, 1); digitalWrite(re2, 1); delay(1000);
12    digitalWrite(pump3, 1);
13 }
14
15 void chutrinhOFF() {
16     digitalWrite(re0, 0); digitalWrite(re1, 0); digitalWrite(re2, 0); delay(6000);
17     digitalWrite(pump1, 0); digitalWrite(pump2, 0); digitalWrite(pump3, 0);
18     digitalWrite(van1, 0);
19     delay(6000);
20     digitalWrite(van2, 0);
21 }
```

- Section 3 (Bottom):** The status bar at the bottom displays the following messages:
 - "Sketch uses 1428 bytes (4%) of program storage space. Maximum is 30720 bytes."
 - "Global variables use 11 bytes (0%) of dynamic memory, leaving 2037 bytes for local variables. Maximum is 2048 bytes."
 - "avrdude: ser_open(): can't open device "\.\COM7": The system cannot find the file specified."
 - "Failed uploading: uploading error: exit status 1"

Một chương trình cơ bản trên Arduino thường bao gồm hai phần chính: **setup()** và **loop()**. Dưới đây là mô tả và ví dụ về một số lệnh cơ bản được sử dụng trong lập trình Arduino:

- **setup()**: Phần này được thực thi một lần duy nhất sau khi cấp nguồn cho Arduino. Nó thường được sử dụng để khai báo các giá trị ban đầu, thiết đặt thông số và khởi tạo các biến. Ví dụ:

```
void setup() {  
    pinMode(13, OUTPUT); // Thiết lập chân 13 là OUTPUT  
}
```

- **loop()**: Phần này bắt đầu chạy sau khi setup() hoàn thành. Các lệnh trong phần này được lặp đi lặp lại cho đến khi nguồn cấp được ngừng. Ví dụ:

```
void loop() {  
    digitalWrite(13, HIGH); // Đặt chân 13 lên mức HIGH (5V)  
    delay(1000); // Dừng 1 giây  
    digitalWrite(13, LOW); // Đặt chân 13 xuống mức LOW (0V)  
    delay(1000); // Dừng 1 giây  
}
```

- Các lệnh cơ bản:

- **pinMode(pin, mode)**: Thiết lập chế độ của một chân (OUTPUT hoặc INPUT).
- **digitalWrite(pin, value)**: Xuất tín hiệu điện từ một chân (HIGH hoặc LOW).
- **digitalRead(pin)**: Đọc giá trị điện áp từ một chân (HIGH hoặc LOW).
- **analogRead(pin)**: Đọc giá trị analog từ một chân Analog (giá trị từ 0 đến 1023).
- **analogWrite(pin, value)**: Xuất tín hiệu analog (phát xung PWM) từ một chân.
- **if (condition) { //code } else { //code }**: Câu lệnh điều kiện.
- **for(initialization; condition; increment/decrement) { //code }**: Vòng lặp for.
- **while(condition) { //code }**: Vòng lặp while.

Ví dụ:

```

1 int sensorValue;
2
3 void setup() {
4     pinMode(A0, INPUT); // Thiết lập chân A0 là INPUT
5 }
6
7 void loop() {
8     sensorValue = analogRead(A0); // Đọc giá trị từ chân A0
9     if (sensorValue > 500) {
10         digitalWrite(13, HIGH); // Nếu giá trị lớn hơn 500, bật LED trên chân 13
11     } else {
12         digitalWrite(13, LOW); // Nếu giá trị nhỏ hơn hoặc bằng 500, tắt LED trên chân 13
13     }
14 }
15

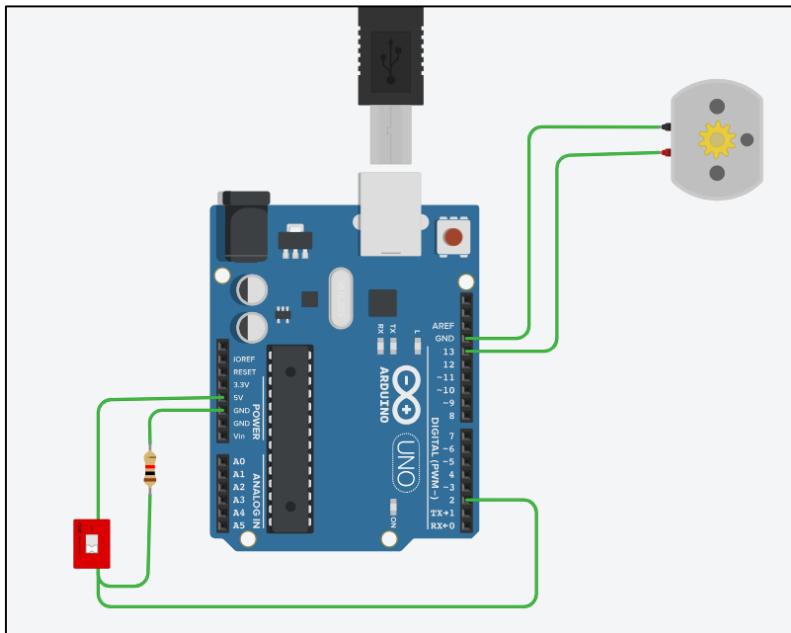
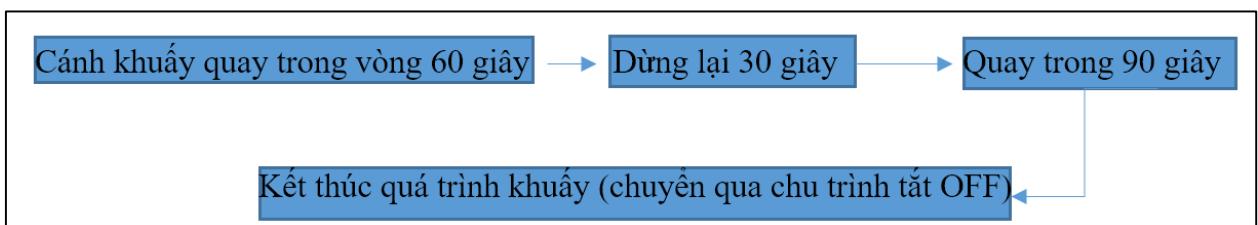
```

3. Thực hiện thí nghiệm

3.1 Thí nghiệm 1: Thực hiện mô phỏng động cơ khuấy

Yêu cầu

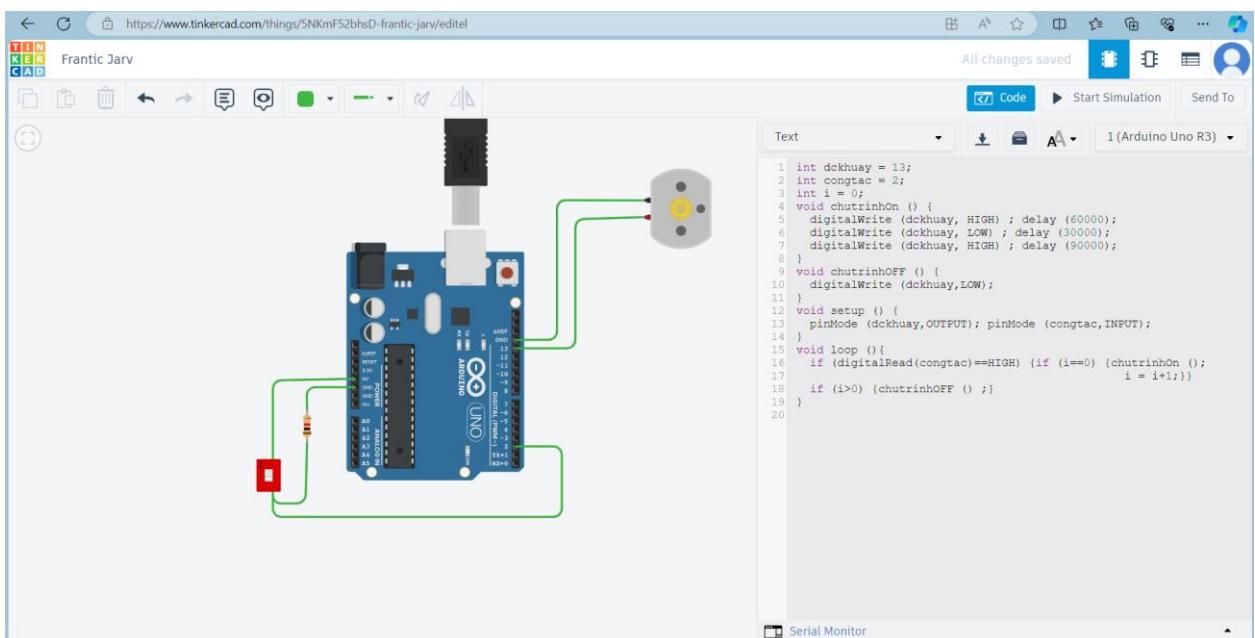
Thông qua 1 công tắc điều khiển khi nhấn nút bật (ON) máy khuấy thì hoạt động theo chu trình ON:



Thiết kế chương trình:

- Lắp động cơ khuấy vào chân **D13**

- Công tắc vào chân **D2** sử dụng điện trở kéo về **GND**
- Sử dụng **pinMode OUTPUT** để thiết lập chân số 13 là chân nhận tín hiệu đầu ra.
- Tín hiệu chân ra digital có 2 giá trị HIGH hoặc LOW sử dụng lệnh **digitalWrite()**.
- Lệnh **delay()** để giữ nguyên trạng thái hoạt động trong khoảng thời gian mong muốn.
- Việc đọc tín hiệu phải để trong vòng **void loop** để chương trình liên tục quét nhằm phát hiện khi nào người dùng bật công tắc.
- Do chu trình ON chỉ chạy 1 lần nên có sử dụng biến **i** để xác định xem chương trình ON đã thực hiện hay chưa.



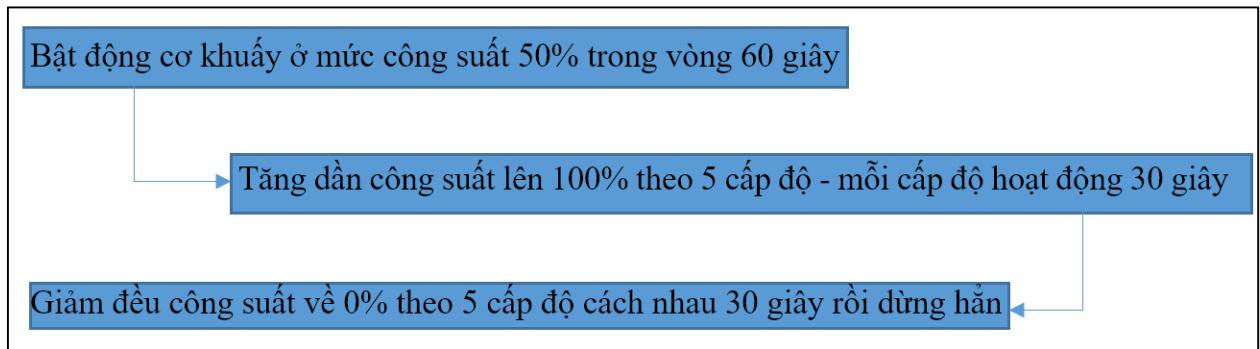
Kết quả và bàn luận

- Kết quả:** Với các bước thực hiện thiết kế chương trình trên, cùng với đoạn code hoàn chỉnh và chạy thực tế trên mô hình tại phòng thí nghiệm. Ta thấy đoạn code đưa ra kết quả hiệu quả và có kết quả chính xác đáp ứng theo ý đồ điều khiển của chúng ta với các chu trình ON, OFF định sẵn. Khi bật công tắc, cánh khuấy quay 60 vòng trên giây, sau đó dừng 30 giây rồi quay trong 90 giây trước khi kết thúc quá trình khuấy và chuyển qua quá trình OFF. Đồng thời, với đoạn code này, ta có thể phát triển lên thêm cho quá trình khuấy hoạt động theo 3 chu kỳ ON liên tục như trên, mỗi chu trình cách nhau 30 giây

- Một số hạn chế: Đoạn chương trình thiết kế ra vẫn chưa đảm bảo được an toàn, có lặp công tắc vì khi ở trạng thái mở sẵn, khi vừa nạp chương trình vào mạch thì động cơ hoạt động ngay lập tức. Điều này sẽ không đảm bảo an toàn khi vận hành ở thực tế. Ngoài ra khi hoàn thành một chu kì gồm một chu trình ON và một chu trình OFF, ta không thể dùng công tắc để khởi động lại quá trình một lần nữa, điều này có nghĩa chương trình chỉ dùng cho một lần. Ta nên cải tiến thêm đoạn code để khắc phục một số yếu tố này

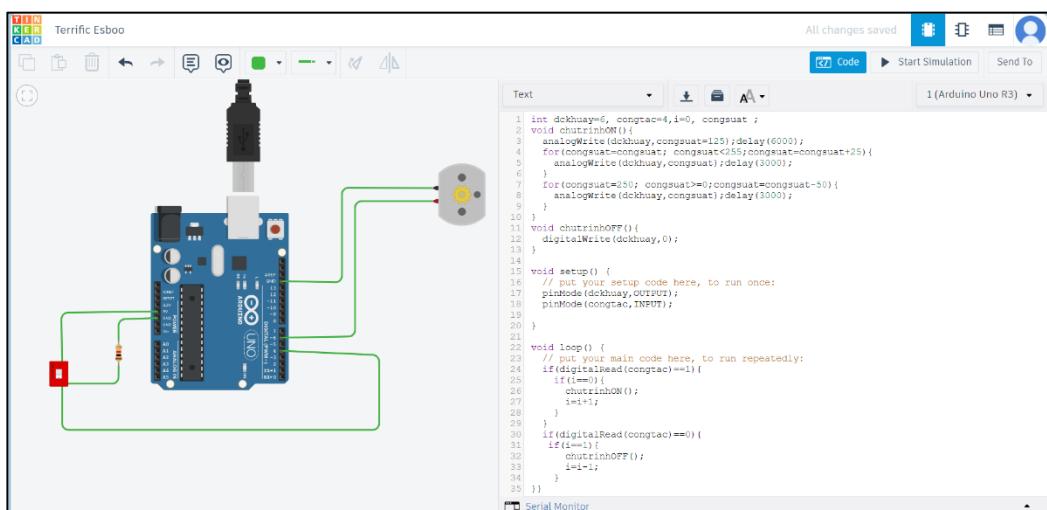
3.2 Thí nghiệm 2: Thực hiện mô phỏng điều khiển động cơ khuấy

Yêu cầu



Thiết kế chương trình

- Lắp động cơ khuấy từ chân **D13 sang chân ~D11** là chân PWM
- Sử dụng **pinMode OUTPUT** để thiết lập chân số 6 là chân nhận tín hiệu đầu ra.
- Tín hiệu chân ra **analog (PWM)** thiết lập bằng **125 (khoảng 50%)** trong 60 giây.
- Tăng giá trị chân **PWM** từ **125 lên 250** và **giảm xuống 0** theo bước nhảy như đề bài.



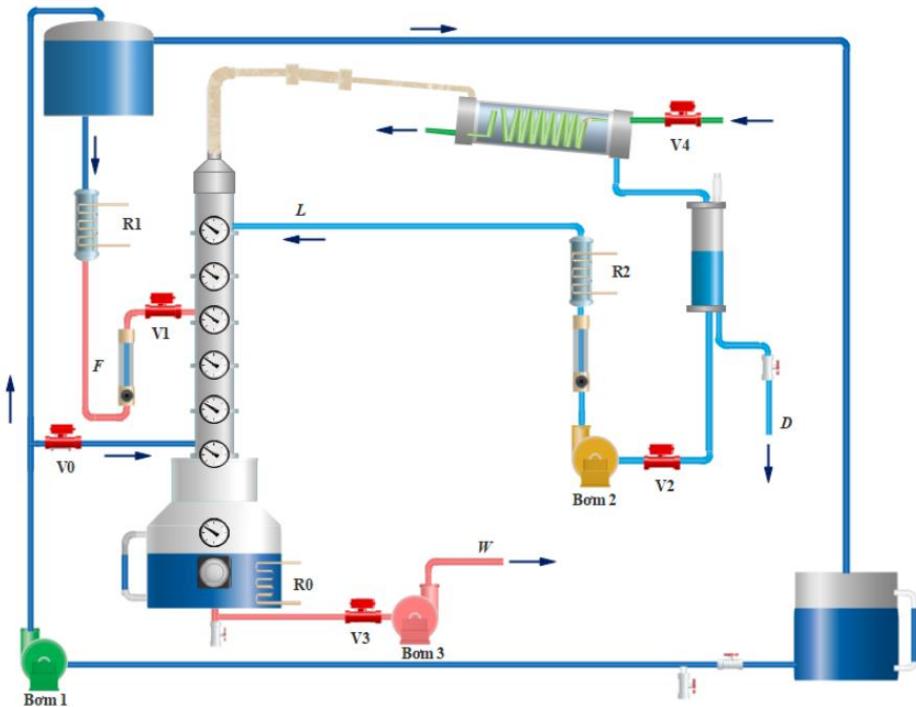
Kết quả và bàn luận

- **Kết quả:** Với các bước thực hiện thiết kế chương trình trên, cùng với đoạn code hoàn chỉnh và chạy thực tế trên mô hình tại phòng thí nghiệm. Ta thấy đoạn code đưa ra kết quả hiệu quả và có kết quả chính xác đáp ứng theo ý đồ điều khiển của chúng ta. bật động cơ khuấy ở mức công suất 50% trong vòng 60 giây, tăng dần công suất lên 100% theo 5 cấp độ - mỗi cấp độ hoạt động 30 giây, sau đó giảm đều công suất về 0% theo 5 cấp độ cách nhau 30 giây rồi dừng hẳn.

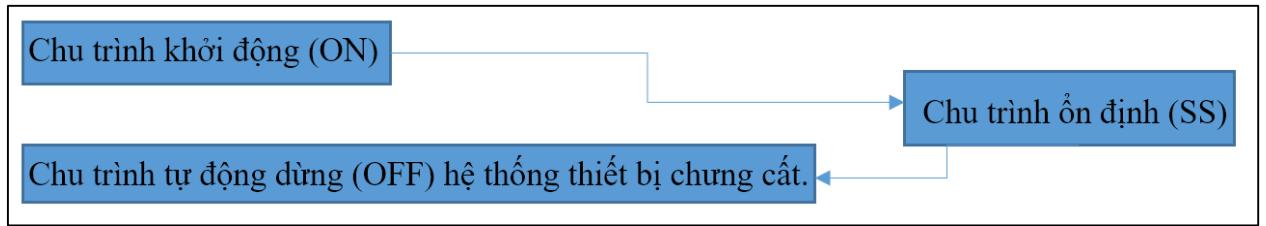
- **Một số hạn chế:** Khi nạp code vào mạch và chạy chương trình, ta không thể thay đổi các thông số công suất động cơ trong quá trình chạy cũng như là ta không thể tắt động cơ qua công tắc. Điều này ở thực tế sẽ nguy hiểm khi không có một cơ chế tắt khẩn cấp và an toàn để dừng động cơ. Do đó, cần cải tiến thêm đoạn code để có thể đáp ứng việc dừng khẩn cấp cũng như là tính linh động của động cơ trong quá trình điều khiển.

3.3 Thí nghiệm 3: Mô phỏng vận hành điều khiển hệ thống chung cất

Hệ thống chung cất trong bài thí nghiệm:



Điều khiển quá hệ thống chung cát



Điều khiển cụ thể như sau:

Sử dụng **công tắc ON/OFF** viết chương trình **điều khiển tự động 3 chu trình** cho hệ thống chung cát:

- **Bật nút ON** hệ thống khởi động theo quy trình định sẵn, sau đó hệ chuyển sang trạng thái làm việc, giữ trạng thái đến khi dừng hệ thống.
- **Bật nút OFF** hệ thống chuyển trạng thái dừng theo quy trình

* Chu trình khởi động ON

Các bước thực hiện	Sự kiện	Thời gian chờ (giây)
B1 – Khởi động	Nhấn nút ON	--
B2 – Nhận dung dịch vào nồi đun	Bật bơm 1 Mở van 0	30
B3 – Gia nhiệt nồi đun và cấp nước giải nhiệt cho thiết bị ngưng tụ	Tắt van 0 Mở điện trở R0 Mở van V2	60
B4 – Khởi động dòng nhập liệu	Mở van V1 Bật điện trở R1	10
B5 – Khởi động dòng hoàn lưu	Bật bơm 2 Bật điện trở R1	10
B6 – Tháo sản phẩm đáy	Mở Van V3 Bật bơm 3	--

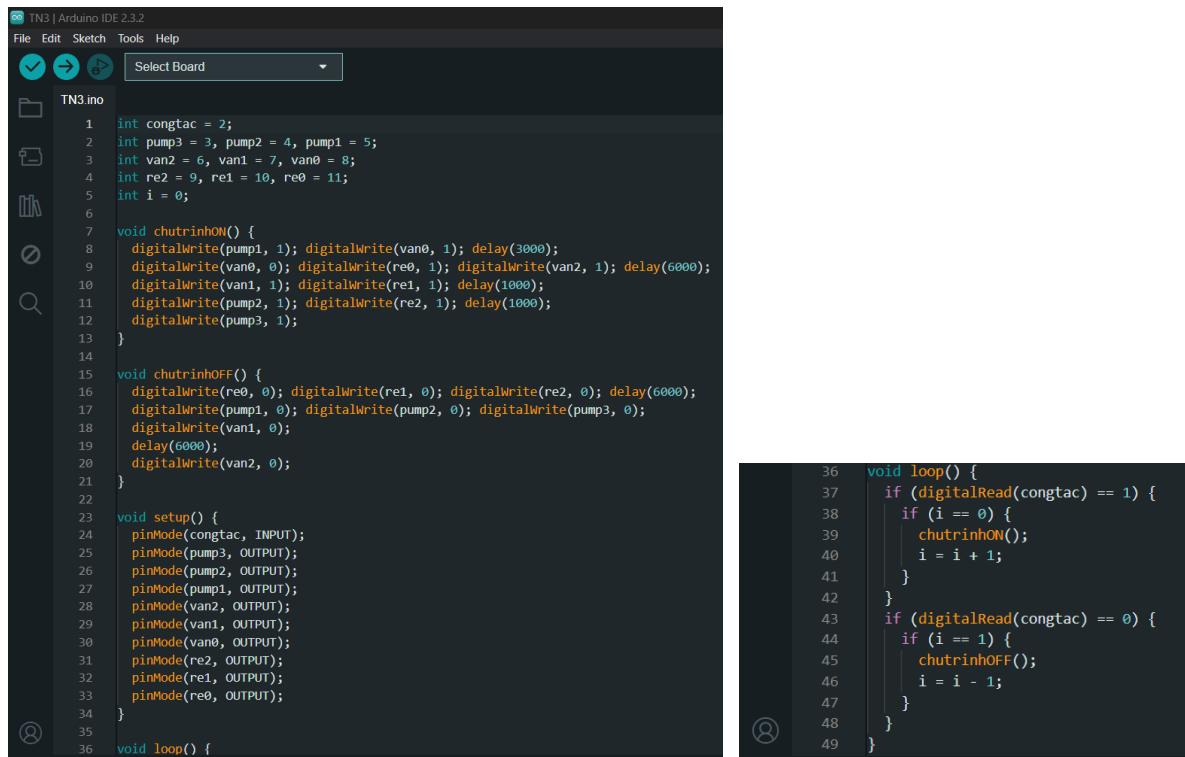
*Trạng thái làm việc (SS)

Tất cả thiết bị ở trạng thái làm việc (ON), nhưng van V0 đóng

*Chu trình dừng (OFF)

Các bước thực hiện	Sự kiện	Thời gian chờ (giây)
B1 – Dừng	Nhấn nút OFF	--
B2 – tắt tất cả nguồn nhiệt cấp cho tháp	Tắt điện trở R0, R1 và R2	60
B3 – Ngưng dòng nhập liệu và hồi lưu	Tắt các bơm 1, 2, 3 Đóng van 1	60
B4 – Ngưng dòng nước lạnh	Đóng van 2	--

* Đoạn code cho chương trình

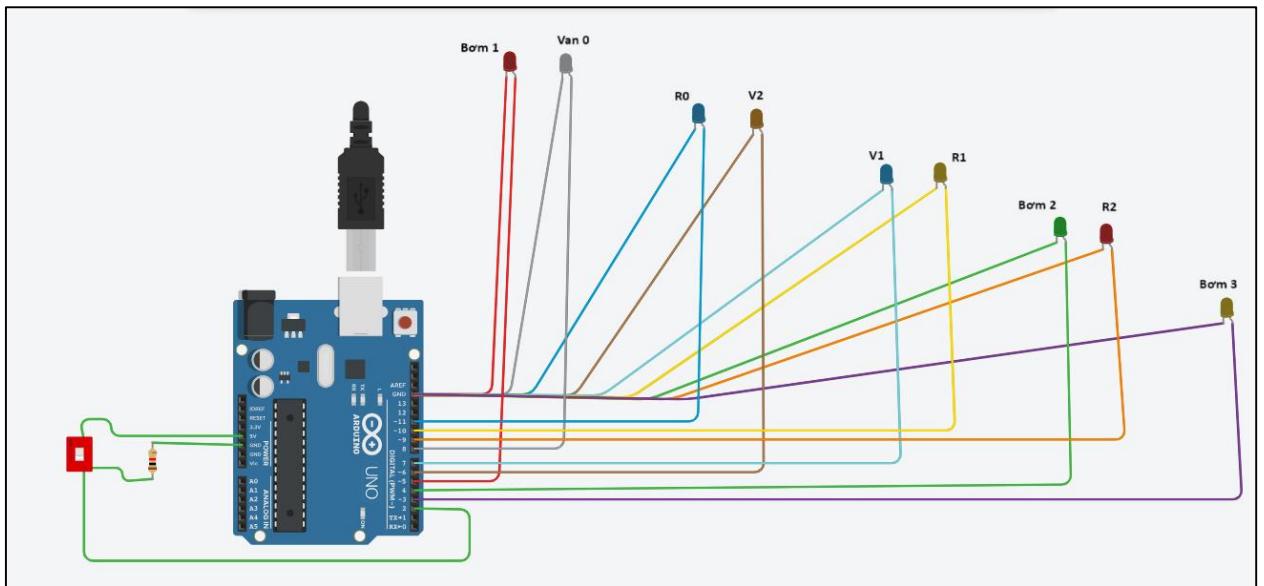


```

TN3 | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Select Board
TN3.ino
1 int congta = 2;
2 int pump3 = 3, pump2 = 4, pump1 = 5;
3 int van2 = 6, van1 = 7, van0 = 8;
4 int re2 = 9, re1 = 10, re0 = 11;
5 int i = 0;
6
7 void chutrinhhON() {
8     digitalWrite(pump1, 1); digitalWrite(van0, 1); delay(3000);
9     digitalWrite(van0, 0); digitalWrite(re0, 1); digitalWrite(van2, 1); delay(6000);
10    digitalWrite(van1, 1); digitalWrite(re1, 1); delay(1000);
11    digitalWrite(pump2, 1); digitalWrite(re2, 1); delay(1000);
12    digitalWrite(pump3, 1);
13 }
14
15 void chutrinhhOFF() {
16     digitalWrite(re0, 0); digitalWrite(re1, 0); digitalWrite(re2, 0); delay(6000);
17     digitalWrite(pump1, 0); digitalWrite(pump2, 0); digitalWrite(pump3, 0);
18     digitalWrite(van1, 0);
19     delay(6000);
20     digitalWrite(van2, 0);
21 }
22
23 void setup() {
24     pinMode(congta, INPUT);
25     pinMode(pump3, OUTPUT);
26     pinMode(pump2, OUTPUT);
27     pinMode(pump1, OUTPUT);
28     pinMode(van2, OUTPUT);
29     pinMode(van1, OUTPUT);
30     pinMode(van0, OUTPUT);
31     pinMode(re2, OUTPUT);
32     pinMode(re1, OUTPUT);
33     pinMode(re0, OUTPUT);
34 }
35
36 void loop() {
37     if (digitalRead(congta) == 1) {
38         if (i == 0) {
39             chutrinhhON();
40             i = i + 1;
41         }
42     }
43     if (digitalRead(congta) == 0) {
44         if (i == 1) {
45             chutrinhhOFF();
46             i = i - 1;
47         }
48     }
49 }

```

Mô phỏng với các đèn tín hiệu ứng với từng thiết bị trong hệ thống



Kết quả và bàn luận

- Kết quả: Khi nạp code vào mạch, sử dụng công tắc ON/OFF viết chương trình điều khiển tự động 3 chu trình cho hệ thống chung cất. Bật nút ON hệ thống khởi động theo quy trình định sẵn, sau đó hệ chuyển sang trạng thái làm việc, giữ trạng thái đến khi dừng hệ thống. Bật nút OFF hệ thống chuyển trạng thái dừng theo quy trình. Đoạn code thực hiện đảm bảo được các bước bật tắt thiết bị theo trình tự đảm bảo yêu cầu điều khiển

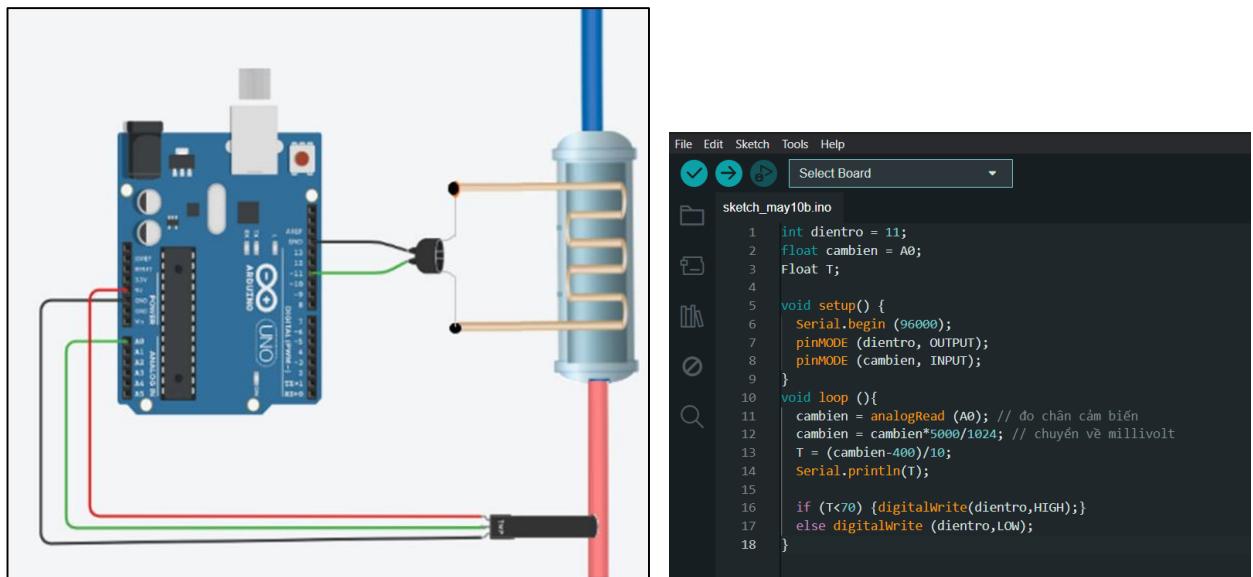
- Bàn luận: Khi thiết kế đoạn code, nhóm gặp phải vấn đề khi V0 tắt đúng trình tự nhưng đến khi kết thúc và muốn thực hiện lại chu trình thì không thể thực hiện được. Nhóm đã tìm được nguyên nhân và cách khắc phục lỗi này, đó là ở câu lệnh điều kiện cuối, khi gán trạng thái công tắc với một biến i để đánh dấu thời điểm hoàn thành chu trình, ta cần trả lại giá trị i khi thay đổi trạng thái công tắc. Ví dụ khi bật công tắc, với giá trị ban đầu $i=0$, chu trình sẽ hoạt động, sau đó $i=i+1$. Đến khi công tắc đóng, với $i=1$ thì chu trình sẽ tắt, lúc này cần trả I lại giá trị 0 thì sau khi bật lại công tắc, chu trình mới có thể tiếp tục. Đây là vấn đề mà tất cả các nhóm điều mắc phải trong buổi thí nghiệm và cần chú ý.

3.4 Thí nghiệm 4: Lập trình điều chỉnh tự động nhiệt độ nhập liệu vào tháp chưng cất sử dụng bộ điều khiển ON-OFF

Sử dụng cảm biến nhiệt độ TMP, giá trị nhiệt độ được cảm biến chuyển thành tín hiệu điện áp tương ứng. Cứ 10mV tương ứng với 1°C, khoảng đo của cảm biến từ -40°C đến 125 °C.

Cảm biến được gắn vào chân analog A0 của Arduino có giá trị từ 0-1023 tương ứng với điện áp từ 0-5000mV.

Như vậy giá trị đọc được từ chân A0 (gọi là sensor) được chuyển về dạng điện áp theo công thức: voltage = sensor*5000/1024. Sau đó giá trị này được chuyển về đơn vị nhiệt độ với độ bù sai sêch điện áp (trong trường hợp này lấy 400mV), ta có nhiệt độ mà cảm biến đo được là : $T = (\text{voltage}-400)/10$



Kết quả và bàn luận

Ta thấy được đoạn chương trình đáp ứng tốt yêu cầu điều khiển của hệ

3.5 Thí nghiệm 5: Lập trình điều chỉnh tự động mức chất lỏng đáy tháp chưng cất sử dụng bộ điều khiển PID

Sử dụng cảm biến siêu âm HC-SR04 đo mức chất lỏng:

Sử dụng sóng siêu âm phát từ chân Trig (sử dụng lệnh digitalWrite(trig,HIGH);) và nhận tín hiệu sóng phản xạ tại chân Echo. Đo thời gian từ lúc phát sóng đến lúc nhận sóng phản xạ ta có được khoảng cách từ cảm biến đến bề mặt vật thể.

Thời gian được arduino đo bằng lệnh: time = pulseIn(echo, HIGH);

Vận tốc âm thanh ở 30°C: 349 m/s = 0.0349 cm/μs

$$\text{Khoảng cách đến vật thể: } \text{distance} = \frac{\text{time}}{2} \times 0,0349 \text{ cm}$$

The screenshot shows the Arduino IDE interface with the following components:

- Sketch:** sketch_may10b.ino
- Code Preview:**

```

1 int trig = 8;
2 int echo = 9;
3 int pump = 11;
4
5 float khoangcach_cm, mucchatlong;
6 float H=60;
7 unsigned int thoigian;
8
9 float Kp = 10.0, Ki = 0.5, Kd = 15.0;
10 float P, I, D, PID;
11 float error, pre_error;
12 float Setpoint = 20;
13 float delta_t = 1;
14
15 void setup() {
16     Serial.begin (8600)
17     pinMode(trig, OUTPUT);
18     pinMode (echo, INPUT);
19     pinMode (pump, OUTPUT);
20 }
21
22 void loop () {
23     thoigian = 0;
24     digitalWrite (trig,0);
25     delayMicroseconds (10);
26     digitalWrite (trig,1);
27     digitalWrite (trig,0);
28     thoigian = pulseIn (echo, HIGH);
29     khoangcach_cm = thoigian*0.0349/2.0;
30     mucchatlong = H - khoangcach_cm;
31
32     error = Setpoint - mucchatlong;
33     P = Kp*error;
34     I = Ki*error*delta_t;
35     D = Kd*(error - pre_error)/delta_t;
36     PID = P + I + D;
37     int U = constrain (PID, 50, 255);
38     analogWrite (pump, U);
39     pre_error = error;
40     delay(1000);
41 }
42
43 }
```
- Circuit Diagram:** A schematic diagram of a water tank level control system. It features a blue Arduino Uno board at the top connected to a breadboard. On the breadboard, there is a green ultrasonic sensor module with two yellow pins (Trig and Echo) connected to digital pins 8 and 9 respectively. A red pump motor is connected to digital pin 11. A blue cylindrical water tank is shown below the breadboard. A vertical grey pipe extends from the tank's side. A red horizontal tube connects the tank to a grey cylindrical component labeled "R0". A red valve is located at the bottom of the tank. A green wire connects the "R0" component to the ground rail of the breadboard.

Kết quả và bàn luận

- Kết quả:** Đoạn chương trình đáp ứng tương đối yêu cầu điều khiển của hệ được đề ra

- Bàn luận: Đoạn chương trình vẫn chưa đáp ứng tốt khi có nhiễu (rung động bình chử lồng). Khi này sai sót khá lớn và hệ mất thời lâu để ổn định lại. Có thể cảm biến siêu âm chưa thật sự tối ưu cho việc xác định mức chất lớn vì bề mặt thoáng của chất lỏng có thể xảy ra dao động thường xuyên và không ổn định trong quá trình hoạt động

4. Kết luận và kiến nghị

Qua quá trình tìm hiểu và thực hiện các thí nghiệm, nhóm rút ra được nhiều nội dung ý nghĩa:

Hiểu được chức năng bộ vi điều khiển: Thí nghiệm giúp người tham gia hiểu rõ về chức năng và vai trò của bộ vi điều khiển trong hệ thống điều khiển tự động. Bằng cách thực hành lập trình và kết nối các thành phần, thể trực tiếp tương tác với bộ vi điều khiển và hiểu được cách nó hoạt động.

Biết lập trình các thuật toán điều khiển cho bộ điều khiển: Thí nghiệm cho phép học cách lập trình các thuật toán điều khiển, như viết code để điều khiển các thiết bị hoạt động theo các chu trình nhất định. Điều này giúp phát triển kỹ năng lập trình và áp dụng chúng vào các ứng dụng thực tế.

Thiết lập sơ đồ kết nối các thành phần của hệ thống điều khiển: Thí nghiệm yêu cầu thiết lập sơ đồ kết nối các thành phần trong hệ thống điều khiển. Qua đó, nhóm học được cách kết nối và giao tiếp giữa các thành phần, điều này là rất quan trọng trong việc xây dựng và vận hành các hệ thống tự động.

Thiết kế hệ thống điều khiển tự động để khởi động, dừng cho quá trình: Mục tiêu cuối cùng của thí nghiệm là thiết kế và triển khai một hệ thống điều khiển tự động. Thí nghiệm giúp hiểu và áp dụng các nguyên lý điều khiển để tạo ra một hệ thống hoạt động một cách tự động và hiệu quả, như trong trường hợp này, để điều khiển quá trình vận hành hệ thống chung cất.

5. Tài liệu tham khảo

- [1]. Điều khiển Quá trình Công nghệ Hóa học - Cơ sở điều khiển Quá trình – Quyển 1.
- [2]. Điều khiển Quá trình Công nghệ Hóa học – Hướng dẫn thí nghiệm, Thực hành cơ sở Điều khiển – Quyển 2.

BÀI THÍ NGHIỆM 3: KHẢO SÁT CÁC BỘ ĐIỀU KHIỂN ON-OFF VÀ PID

1. Mục đích thí nghiệm

Dựa trên các Hệ thống điều khiển các đại lượng cơ bản trong phòng thí nghiệm:

- Tìm hiểu các quy luật điều khiển ON-OFF và PID
- Đánh giá chất lượng điều khiển khi sử dụng bộ điều khiển ON-OFF
- Đánh giá chất lượng điều khiển khi sử dụng bộ điều khiển P, PI, PID

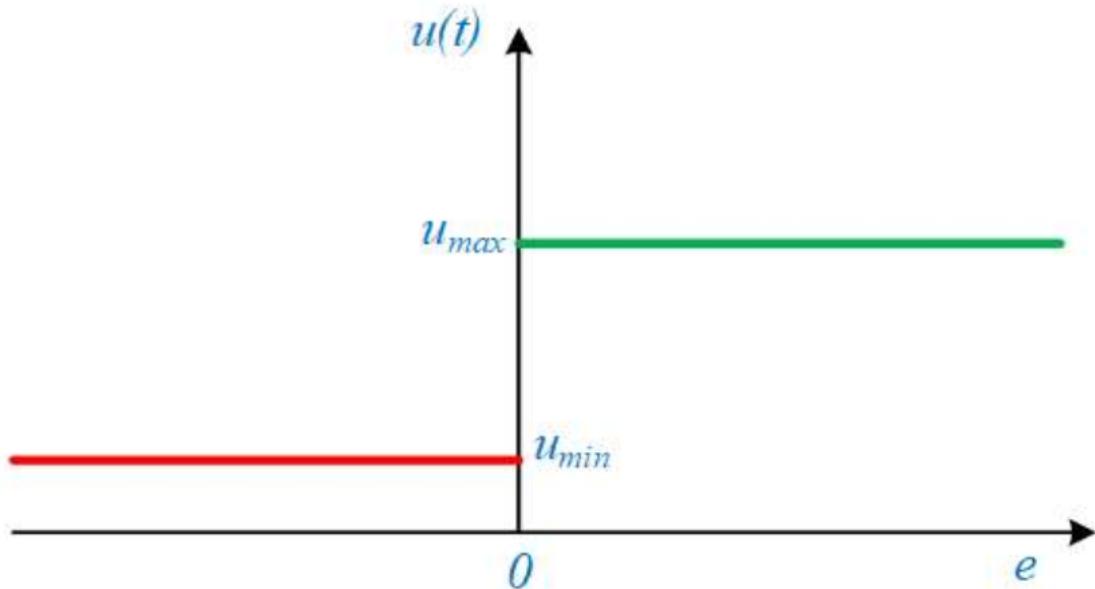
2. Cơ sở lý thuyết

2.1. Bộ điều khiển ON-OFF

Bộ điều khiển ON-OFF là bộ điều khiển chỉ cho tín hiệu ra ở hai chế độ (u_{max} và u_{min}) hoặc hai trạng thái (bật và tắt) tùy thuộc vào sai lệch điều khiển e mà nó nhận được.

Quy luật điều khiển là thuật toán ON-OFF:

$$u(t) = \begin{cases} u_{max} & \text{khi } e \geq 0 \\ u_{min} & \text{khi } e < 0 \end{cases}$$



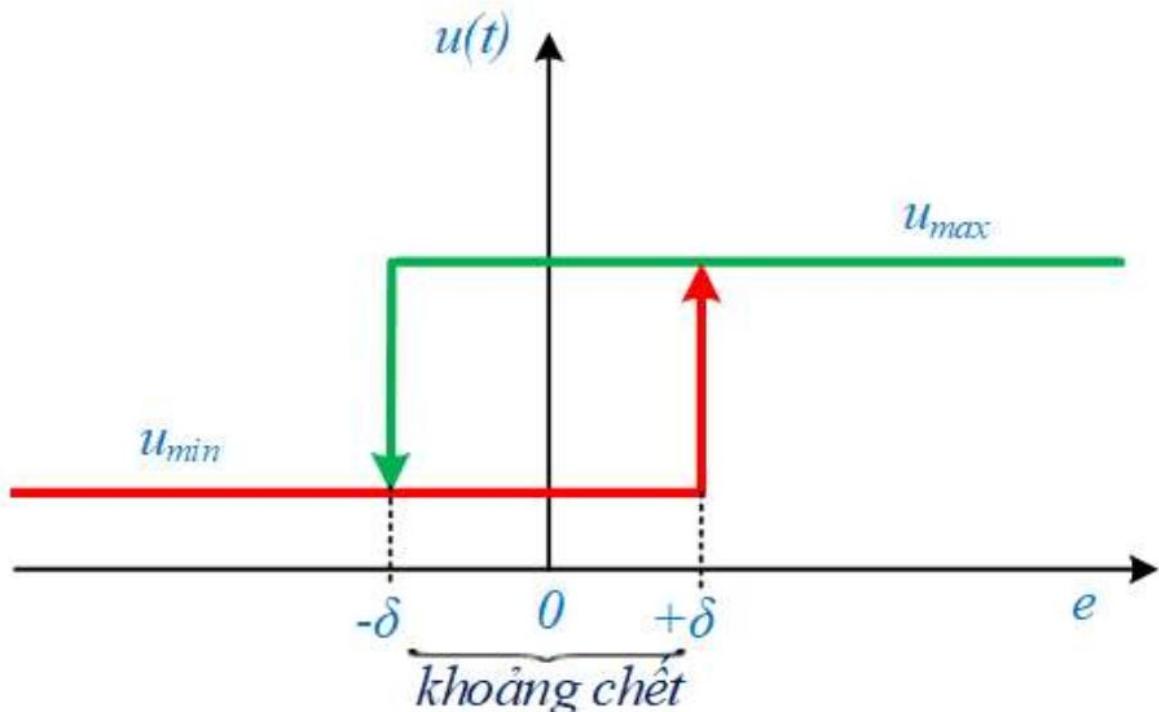
Hình 1. Thuật toán ON-OFF lí tưởng

Loại điều khiển này không thực sự giữ biến được điều khiển chính xác tại điểm đặt, mà biến được điều khiển luôn có một khoảng cách so với điểm đặt

Thuật toán ON-OFF thực tế: nhược điểm của bộ điều khiển hai vị trí dạng lý tưởng là chất lượng điều khiển không cao, tín hiệu điều khiển trong một giới hạn cho phép nào

đó bằng cách sử dụng *khoảng chêt* – kiểu tác động Rôle (Relay). Việc chuyển đổi giữa hai mức tín hiệu được thực hiện qua một giá trị dung sai $\pm\delta$ theo quy luật điều khiển sau:

$$u(t) = \begin{cases} u_{max} & \text{khi } e > 0 \\ u(t) & \text{khi } -\delta \leq e \leq \delta \\ u_{min} & \text{khi } e < 0 \end{cases}$$



Hình 2. Thuật toán ON-OFF thực tế

2.2. Bộ điều khiển liên tục PID

Bộ điều khiển PID là bộ điều khiển đưa ra tác động điều khiển dựa trên các tác động thành phần bao gồm P, I và D

- Thành phần **P** là thành phần có tác động tỉ lệ với độ lớn sai lệch điều khiển $e(t)$
- Thành phần **I** là thành phần có tác động tỉ lệ với tích phân sai lệch điều khiển $e(t)$
- Thành phần **D** là thành phần có tác động tỉ lệ với vi phân sai lệch điều khiển $e(t)$

Trong khi thành phần tỉ lệ **P** phản ứng với độ lớn sai lệch điều khiển ở thời điểm hiện tại, thành phần tích phân **I** phản ứng với khoảng thời gian tồn tại của sai lệch điều khiển đến thời điểm hiện tại, thì thành phần vi phân **D** phản ứng với tốc độ thay đổi của sai

lệch để dự báo xu hướng thay đổi của nó trong tương lai và đưa ra các tác động điều khiển chống lại sự thay đổi đó.

Thuộc họ PID có những bộ điều khiển thông dụng như P, PI hay PID. Hai dạng mô tả bộ điều khiển PID thường được sử dụng: dạng truyền thống và dạng chuẩn

PID dạng truyền thống – hay còn gọi là dạng song song:

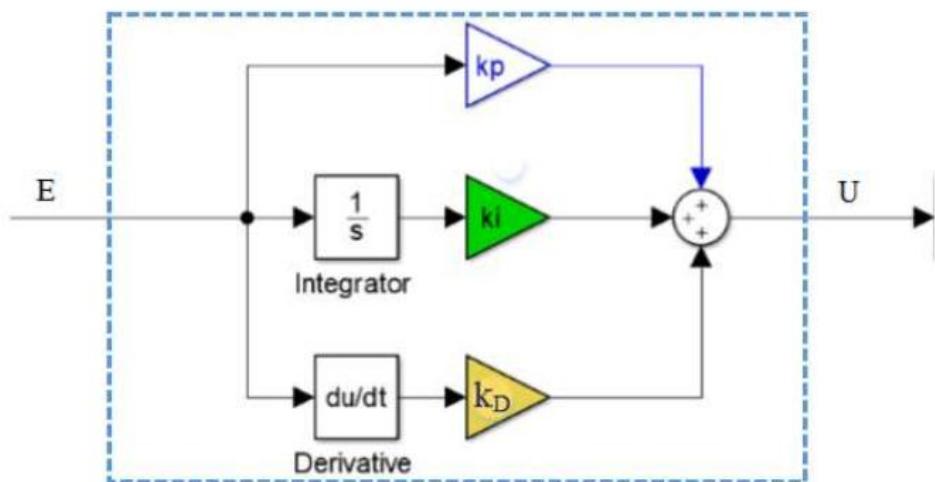
- Thuật toán PID truyền thống được biểu diễn bởi công thức:

$$u = \bar{u} + k_P e + k_I \int_0^t e dt + k_D \frac{de}{dt}$$

- Tương ứng với hàm truyền:

$$G_c = k_P + \frac{k_I}{s} + k_D \cdot s$$

- Biểu diễn trên sơ đồ khối:



Trong đó các thông số được gọi như sau:

k_P – hệ số khuếch đại tỉ lệ (proportional gain)

k_I – hệ số khuếch đại tích phân (integral gain)

k_D – hệ số khuếch đại vi phân (derivative gain)

PID dạng chuẩn:

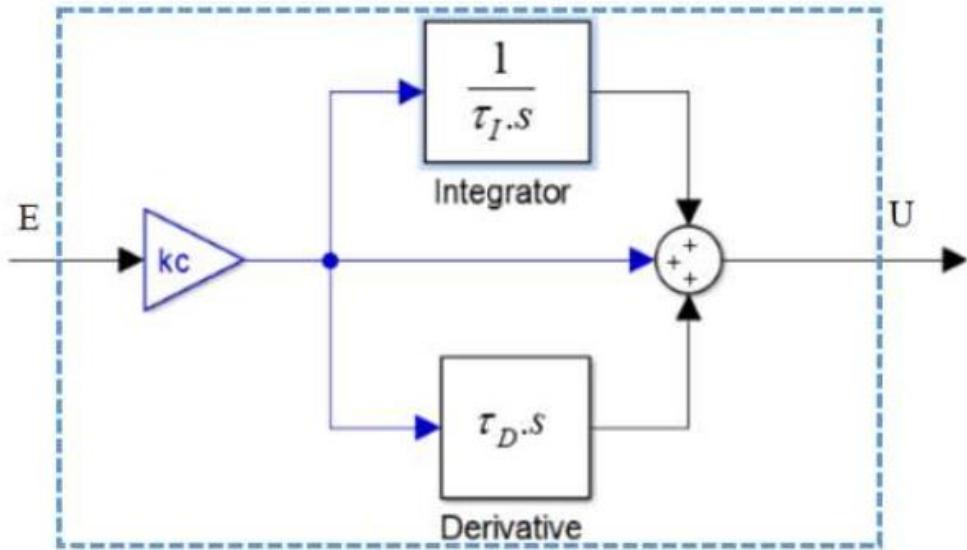
- Thuật toán PID dạng chuẩn được biểu diễn bởi công thức:

$$u = \bar{u} + k_c \left(e + \frac{1}{\tau_I} \int_0^t e dt + \tau_D \frac{de}{dt} \right)$$

- Tương ứng với hàm truyền:

$$G_c = k_c \left(1 + \frac{1}{\tau_I \cdot s} + \tau_D \cdot s \right)$$

- Biểu diễn sơ đồ khối:



Trong đó các thông số điều khiển:

$k_c = k_P$ là hệ số khuếch đại của bộ điều khiển

$\tau_I = \frac{k_c}{k_I}$ là thời gian tích phân

$\tau_D = \frac{k_D}{k_c}$ là thời gian vi phân

Các thành phần P, I và D trong bộ điều khiển PID (Proportional-Integral-Derivative) đều có ảnh hưởng đặc biệt đến chất lượng của hệ thống điều khiển. Dưới đây là nhận xét về ảnh hưởng của mỗi thành phần:

Thành phần Proportional (P):

Ưu điểm: Thành phần P giúp cải thiện độ chính xác và thời gian đáp ứng của hệ thống điều khiển. Nó tạo ra tín hiệu điều khiển tỷ lệ với sự chênh lệch giữa giá trị đo và giá trị đặt.

Nhược điểm: Tuy thành phần P giúp cải thiện độ chính xác, nhưng nó có thể dẫn đến hiện tượng dư lượng (overshoot) và dao động, đặc biệt khi giá trị tỷ lệ (gain) được thiết lập quá cao.

Thành phần Integral (I):

Ưu điểm: Thành phần I giúp giảm thiểu sai số tĩnh và đảm bảo rằng giá trị đo tiến về giá trị đặt theo thời gian. Nó tích lũy và điều chỉnh lỗi tích phân theo thời gian.

Nhược điểm: Nếu thiết lập quá cao, thành phần I có thể dẫn đến hiện tượng dư lượng và dao động tăng lên, cũng như tăng nguy cơ hiện tượng chết điều khiển (integral windup).

Thành phần Derivative (D):

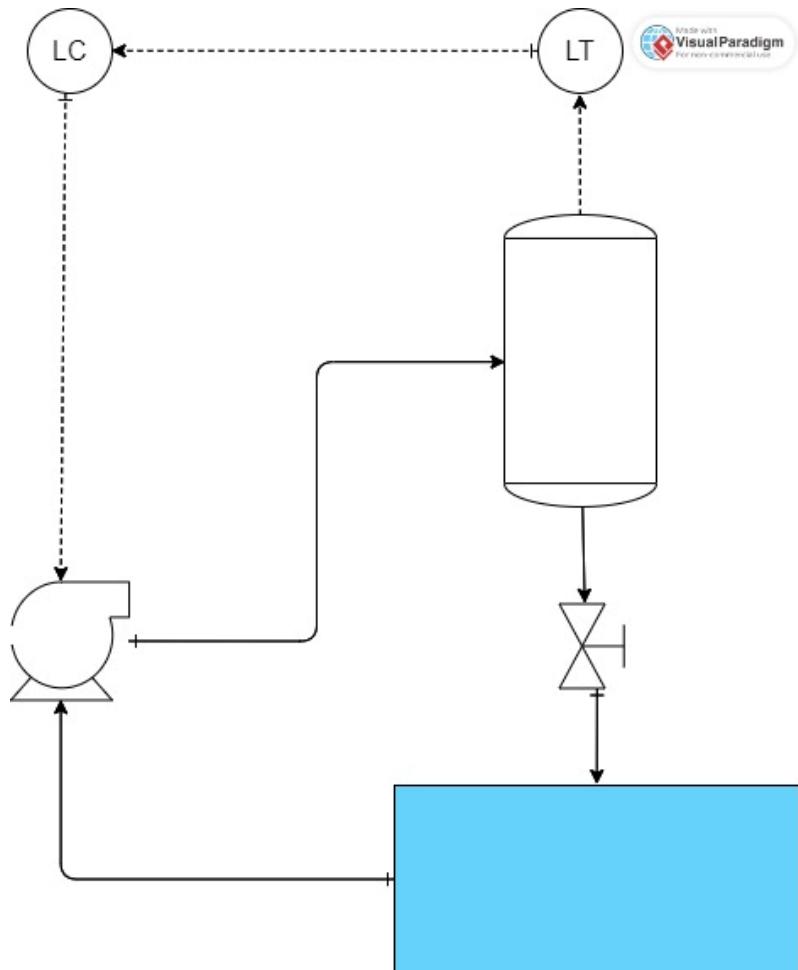
Ưu điểm: Thành phần D giúp ổn định hệ thống và giảm thiểu dao động bằng cách dự đoán xu hướng tương lai của lỗi và giảm tốc độ phản ứng khi giá trị đo tiến gần đến giá trị đặt.

Nhược điểm: Nếu thiết lập quá cao, thành phần D có thể tăng đáng kể độ nhạy của hệ thống và gây ra dao động không mong muốn, đặc biệt là ở điều kiện nhiễu.

Tóm lại, sự cân bằng giữa các thành phần P, I và D là quan trọng để đạt được chất lượng điều khiển tốt nhất. Việc điều chỉnh các tham số của từng thành phần để đáp ứng yêu cầu cụ thể của hệ thống là một phần quan trọng trong việc thiết lập và tối ưu hóa điều khiển PID.

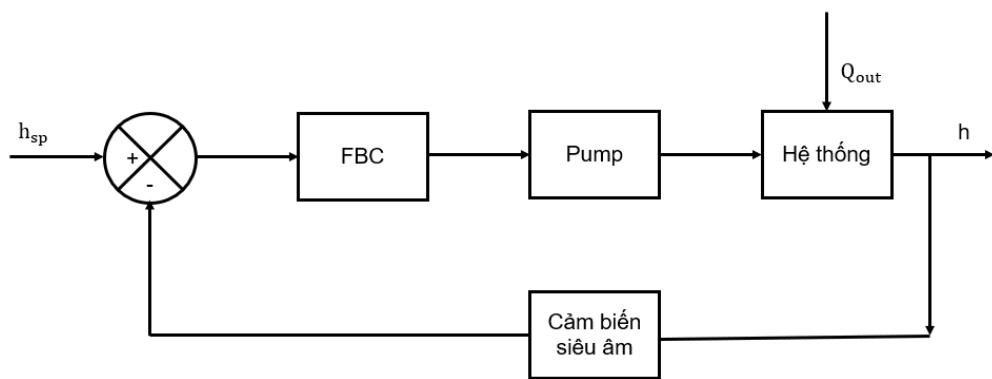
3. Mô tả thiết bị thí nghiệm

3.1. Lưu đồ P&ID



Hình 3. Lưu đồ P&iD cho hệ thống điều khiển

3.2. Sơ đồ khối hệ thống điều khiển quá trình



Hình 4. Sơ đồ khối hệ thống điều khiển theo phương thức phản hồi

4. Quy trình thực hiện

4.1. Thí nghiệm 1: Khảo sát bộ điều khiển ON/OFF

- Nhập giá trị biến cần điều khiển mong muốn y_{sp}

- Chọn bộ điều khiển ON/OFF trên dao điện điều khiển
- Thực hiện khảo sát bằng cách nhấn vào nút Auto trên giao diện điều khiển
- Ghi nhận chỉ tiêu chất lượng vào bảng
- Khi hệ thống ổn định, đưa nhiều bậc thang vào hệ thống, ghi nhận đáp ứng vào bảng ghi kết quả

Bộ điều khiển ON-OFF	y_{sp}
Thời gian xác lập	
Sai lệch xác lập	
Độ quá điều chỉnh	

4.2. Thí nghiệm 2: Khảo sát các bộ điều khiển P, PI và PID

- Nhập giá trị biến cần điều khiển mong muốn y_{sp}
- Chọn lần lượt bộ điều khiển P, PI và PID trên giao diện điều khiển
- Thực hiện khảo sát bằng cách nhấn vào nút Auto trên giao diện điều khiển
- Ghi nhận chỉ tiêu chất lượng vào bảng ghi kết quả
- Khi hệ thống ổn định, đưa nhiều bậc thang vào hệ thống, ghi nhận đáp ứng vào bảng ghi kết quả.

Bộ điều khiển P	y_{sp}
Thời gian xác lập	
Sai lệch xác lập	
Độ quá điều chỉnh	

Bộ điều khiển PI	y_{sp}
Thời gian xác lập	
Sai lệch xác lập	
Độ quá điều chỉnh	

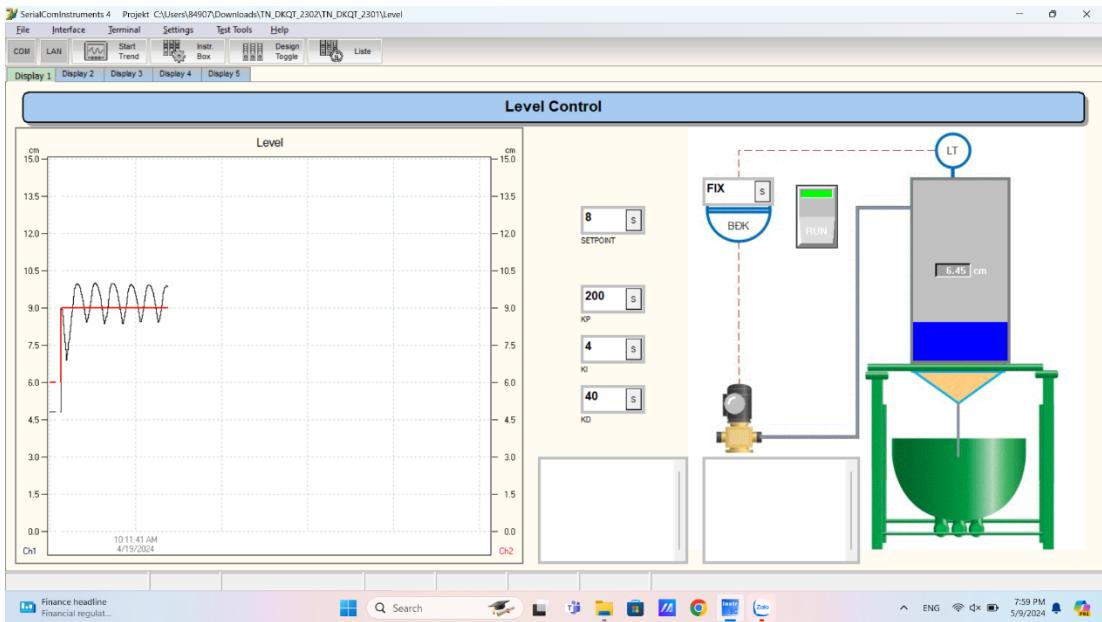
Bộ điều khiển PID	y_{sp}
Thời gian xác lập	
Sai lệch xác lập	
Độ quá điều chỉnh	

5. Kết quả và bàn luận

5.1. Thí nghiệm 1

- Nhập giá trị biến cần điều khiển mong muốn y_{sp}
- Chọn bộ điều khiển ON/OFF trên dao diện điều khiển
- Thực hiện khảo sát bằng cách nhấn vào nút Auto trên giao diện điều khiển
- Ghi nhận chỉ tiêu chất lượng vào bảng ghi kết quả
- Khi hệ thống ổn định, đưa nhiều bậc thang vào hệ thống, ghi nhận đáp ứng vào bảng ghi kết quả.

Bộ điều khiển ON-OFF	y_{sp}
Thời gian xác lập	32s
Chu kì	50s
Độ quá điều chỉnh	11.1%



Hình 5. Đồ thị mức chất lỏng theo thời gian khi sử dụng phương thức phản hồi với bộ điều khiển ON/OFF

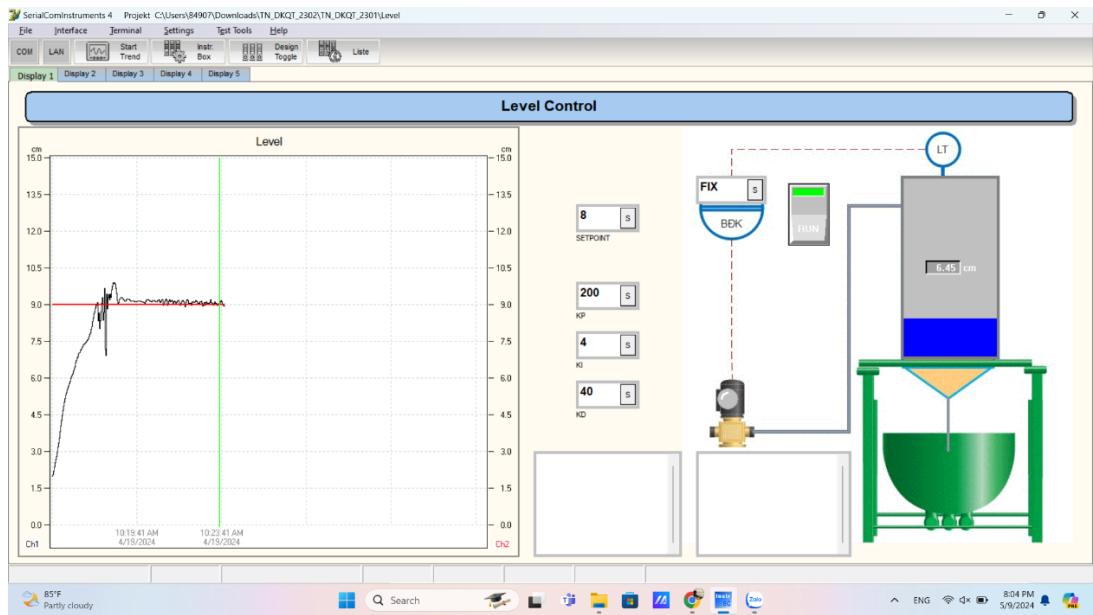
Bộ điều khiển ON/OFF cho giá trị cuối của mức chất lỏng không xác định mà dao động quanh giá trị xác lập với biên độ dao động từ xấp xỉ 8.4 - 10cm so với giá trị cài đặt 9cm, sai số tương đối lớn ($\pm 7\text{-}11\%$), chu kỳ dao động xấp xỉ 50s và không có thời gian xác lập. Do đó, chất lượng điều khiển của bộ điều khiển ON/OFF là rất thấp, gây ra sự thay đổi lưu lượng liên tục dẫn đến lưu lượng dòng ra từ bình chứa chất lỏng không ổn định, ảnh hưởng trực tiếp đến quá trình tiếp theo sau.

Do đó, có thể giảm thiểu mức độ dao động bằng cách sử dụng bộ điều khiển ON/OFF ba điểm hoặc chuyển sang bộ điều khiển PID để cho chất lượng điều khiển tốt nhất.

5.2. Thí nghiệm 2

a) Bộ điều khiển P

Bộ điều khiển P	y_{sp}
Thời gian xác lập	128s
Chu kỳ	16s
Độ quá điều chỉnh	9.9%



Hình 6. Đồ thị mức chất lỏng theo thời gian khi sử dụng phương thức phản hồi với bộ điều khiển P

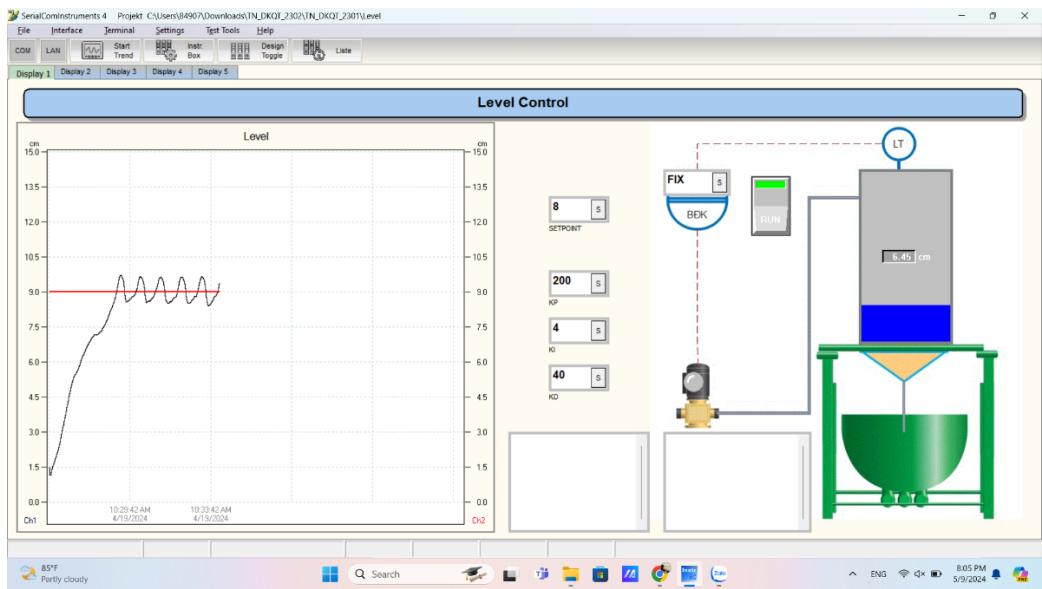
Bộ điều khiển PI cho giá trị cuối của mức chất lỏng gần bằng đúng giá trị xác lập ($y_{\infty} = y_{sp}$) với sai lệch tĩnh gần như tương đối ổn định ở giá trị cài đặt cho trước ($y_{sp} = 9\text{cm}$).

Thời gian xác lập của hệ được xác định trên hình là 128s (sai số xấp xỉ $\pm 1.5\%$ sau khi hệ đã ổn định). Tuy nhiên, độ quá điều chỉnh của hệ còn tương đối lớn (0.89 cm). Các số liệu trên cho thấy bộ điều khiển này cần chỉnh thông số k_p phù hợp với hệ

Ngoài ra, trên đồ thị đáp ứng cho thấy xuất hiện các dạng đường bậc thang xuống rồi lên trước khi hệ đạt ổn định. Nguyên nhân có thể do sử dụng cảm ứng siêu âm cùng với nhiều làm cho bề mặt thoảng chất lỏng trong bình chứa dao động dẫn đến sai số

b) Bộ điều khiển PI

Bộ điều khiển PI	y_{sp}
Thời gian xác lập	262s
Chu kỳ	60s
Độ quá điều chỉnh	7.9%



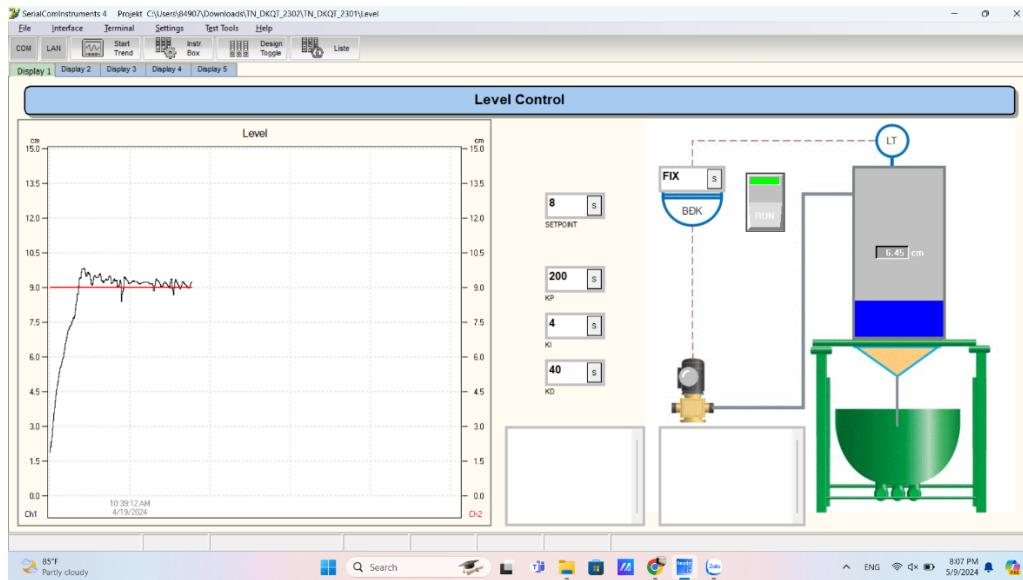
Hình 7. Đồ thị mức chất lỏng theo thời gian khi sử dụng phương thức phản hồi với bộ điều khiển PI

Bộ điều khiển PI cho giá trị cuối của mức chất lỏng không xác định mà dao động quanh giá trị xác lập với mức chất lỏng dao động từ xấp xỉ 8.65 – 9.65 cm so với giá trị cài đặt 9 cm, sai số tương đối lớn ($\pm 4\text{-}7\%$), chu kỳ dao động xấp xỉ 60s và không có thời gian xác lập. Do đó, chất lượng điều khiển của bộ điều khiển PI là rất thấp, gây ra sự thay đổi lưu lượng liên tục dẫn đến lưu lượng dòng ra từ bình chứa chất lỏng không ổn định, ảnh hưởng trực tiếp đến quá trình tiếp theo sau.

Do đó, có thể giảm thiểu mức độ dao động bằng cách thay đổi các thông số k_P và k_I sao cho hợp lý hoặc có thể chuyển sang bộ điều khiển PID để cho chất lượng điều khiển tốt nhất.

c) Bộ điều khiển PID

Bộ điều khiển PID	y_{sp}
Thời gian xác lập	84s
Chu kỳ	24s
Độ quá điều chỉnh	9.1%



Hình 8. Đồ thị mức chất lỏng theo thời gian khi sử dụng phương thức phản hồi với bộ điều khiển PID

Bộ điều khiển PID cho giá trị cuối của mức chất lỏng bằng đúng giá trị xác lập ($y_{\infty} = y_{sp}$) với sai lệch tĩnh gần như đạt đến ổn định ở giá trị cài đặt cho trước ($y_{sp} = 9\text{cm}$).

Thời gian xác lập của hệ được xác định trên hình là 84s (sai số xấp xỉ $\pm 3\%$). Các số liệu trên cho thấy bộ điều khiển này có chất lượng điều khiển tốt, các thông số k_P , k_I và k_D được chỉnh định tốt và phù hợp với hệ. Tuy nhiên để đạt được tốt nhất có thể thực hiện các bước hiệu chỉnh PID như Ziegler-Nichols hoặc Cohen-Coon để tối ưu hóa bộ điều khiển.

Ngoài ra, trên đồ thị đáp ứng cho thấy xuất hiện các dạng đường bậc thang xuống rồi lên. Nguyên nhân có thể do sử dụng cảm ứng siêu âm cùng với nhiều làm cho bề mặt thoáng chất lỏng trong bình chứa dao động dẫn đến sai số

6. Kết luận và kiến nghị

Trong các phương thức điều khiển đã sử dụng, phương thức phản hồi với bộ điều khiển PID được coi là mang lại chất lượng điều khiển cao nhất. Phương thức này tự động điều chỉnh đầu ra của hệ thống dựa trên phản hồi từ giá trị đo, sử dụng ba thành phần tỷ lệ, tích phân và vi phân để tối ưu hóa hiệu suất điều khiển. Trong khi đó, các phương thức phản hồi với bộ điều khiển ON/OFF và PI thường gặp khó khăn trong việc duy trì ổn định mức chất lỏng. Điều này có thể do sai lệch trong thiết bị đo (như lưu lượng kế hoặc motor của bơm), giới hạn vật lý của hệ thống (ví dụ: giới hạn quay của motor), hoặc các vấn đề khác như nhiễu từ môi trường bên ngoài.

Kết quả từ các đồ thị đáp ứng mới sau điều chỉnh có thể cung cấp thông tin quan trọng để so sánh hiệu suất của các phương thức điều khiển khác nhau. Trong quá trình này, sự ổn định của mức chất lỏng được xem xét là thước đo quan trọng, cùng với các yếu tố khác như thời gian đáp ứng và độ ổn định.

Trong hầu hết các trường hợp, đồ thị đáp ứng đều thể hiện một đường nhiễu dạng bậc thang, mà nguyên nhân có thể là do cảm biến siêu âm. Cảm biến siêu âm thường dễ bị ảnh hưởng bởi nhiều nguồn nhiễu khác nhau, như dao động của mặt nước, sự xuất hiện của bong bóng trên bề mặt, hoặc các vật thể ngoại lai như bụi, bẩn, và các vật thể khác nằm trên bề mặt.Thêm vào đó, sóng siêu âm có thể bị phản xạ lại từ thành bể và môi trường xung quanh, đặc biệt là khi cảm biến đặt ở khoảng cách cao. Do đó, một giải pháp để giảm thiểu độ nhạy của cảm biến siêu âm đối với các nguồn nhiễu là thay thế nó bằng một loại cảm biến mức khác như cảm biến điện dung, cảm biến mức, ... Cảm biến này có thể cung cấp kết quả đo ổn định hơn và ít bị ảnh hưởng bởi các yếu tố nhiễu từ môi trường xung quanh. Việc chọn lựa loại cảm biến phù hợp sẽ giúp cải thiện chất lượng của dữ liệu đo và tăng tính ổn định của hệ thống điều khiển chất lỏng.

7. Tài liệu tham khảo

- [1]. Điều khiển Quá trình Công nghệ Hóa học - Cơ sở điều khiển Quá trình – Quyển 1.
- [2]. Điều khiển Quá trình Công nghệ Hóa học – Hướng dẫn thí nghiệm, Thực hành cơ sở Điều khiển – Quyển 2.

BÀI THÍ NGHIỆM 4: THIẾT KẾ HỆ THỐNG ĐIỀU CHỈNH CÁC ĐẠI LUỢNG CƠ BẢN

1. Tóm tắt

Tìm hiểu cấu trúc của hệ thống thiết bị điều khiển các đại lượng cơ bản trong phòng thí nghiệm. Tìm hiểu nguyên lý hoạt động các thành phần trong hệ thống điều khiển như cảm biến đo các đại lượng công nghệ, thiết bị chấp hành, phần cứng bộ điều khiển, ... và sơ đồ kết nối giữa các thành phần này với nhau. Thực hiện cài đặt giao diện điều khiển và khảo sát ảnh hưởng của thông số bộ điều khiển đến đáp ứng của hệ. Thiết kế bộ điều khiển cho một hệ thống điều chỉnh.

2. Cơ sở lý thuyết

2.1. Ảnh hưởng các thông số bộ điều khiển

Bộ điều khiển PID là bộ điều khiển đưa ra tác động điều khiển dựa trên các tác động thành phần bao gồm P, I và D

- Thành phần P là thành phần có tác động tỉ lệ với độ lớn sai lệch điều khiển $e(t)$
- Thành phần I là thành phần có tác động tỉ lệ với tích phân sai lệch điều khiển $e(t)$
- Thành phần D là thành phần có tác động tỉ lệ với vi phân sai lệch điều khiển $e(t)$

Trong khi thành phần tỉ lệ P phản ứng với độ lớn sai lệch điều khiển ở thời điểm hiện tại, thành phần tích phân I phản ứng với khoảng thời gian tồn tại của sai lệch điều khiển đến thời điểm hiện tại, thì thành phần vi phân D phản ứng với tốc độ thay đổi của sai lệch để dự báo xu hướng thay đổi của nó trong tương lai và đưa ra các tác động điều khiển chống lại sự thay đổi đó.

Thuộc họ PID có những bộ điều khiển thông dụng như P, PI hay PID. Hai dạng mô tả bộ điều khiển PID thường được sử dụng: dạng truyền thống và dạng chuẩn

PID dạng truyền thống – hay còn gọi là dạng song song:

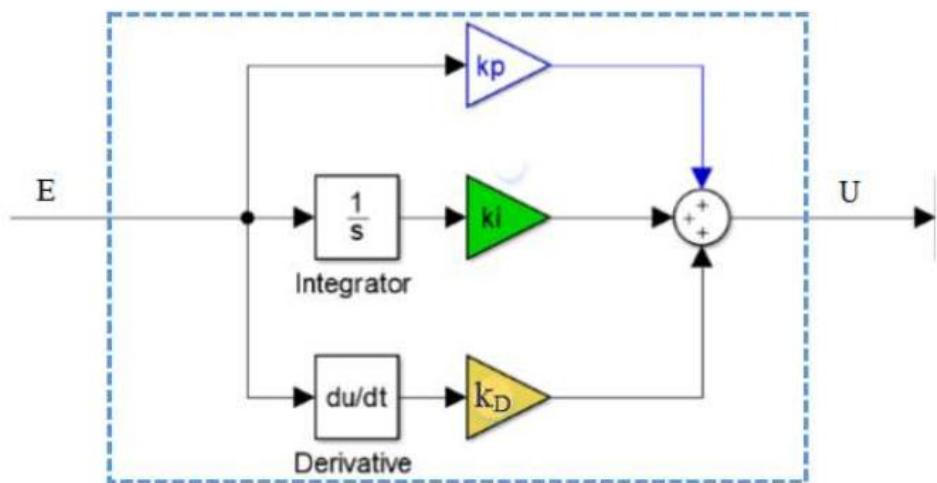
- Thuật toán PID truyền thống được biểu diễn bởi công thức:

$$u = \bar{u} + k_P e + k_I \int_0^t e dt + k_D \frac{de}{dt}$$

- Tương ứng với hàm truyền:

$$G_c = k_p + \frac{k_I}{s} + k_D \cdot s$$

- Biểu diễn trên sơ đồ khối:



Trong đó các thông số được gọi như sau:

k_p – hệ số khuếch đại tỉ lệ (proportional gain)

k_I – hệ số khuếch đại tích phân (integral gain)

k_D – hệ số khuếch đại vi phân (derivative gain)

PID dạng chuẩn:

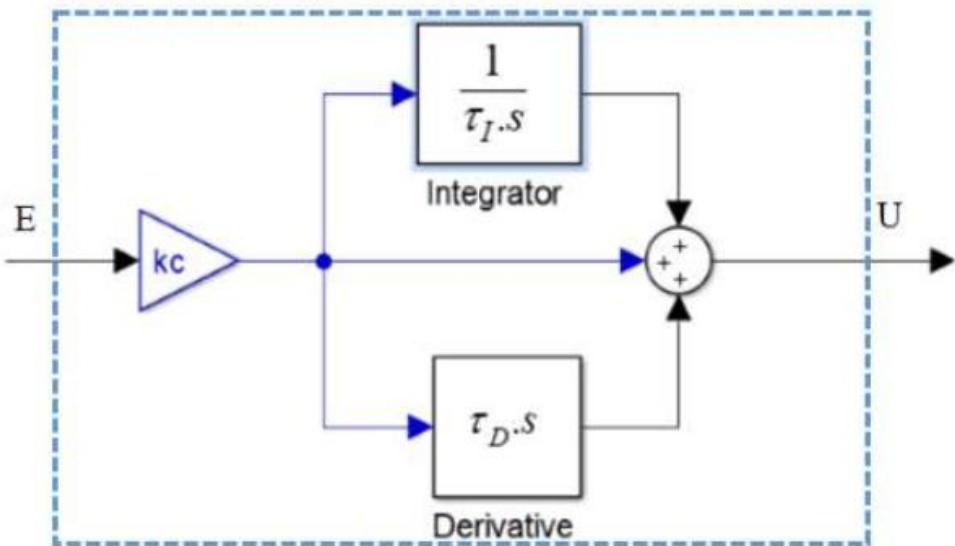
- Thuận toán PID dạng chuẩn được biểu diễn bởi công thức:

$$u = \bar{u} + k_c \left(e + \frac{1}{\tau_I} \int_0^t e dt + \tau_D \frac{de}{dt} \right)$$

- Tương ứng với hàm truyền:

$$G_c = k_c \left(1 + \frac{1}{\tau_I \cdot s} + \tau_D \cdot s \right)$$

- Biểu diễn sơ đồ khối:



Trong đó các thông số điều khiển:

$k_c = k_p$ là hệ số khuếch đại của bộ điều khiển

$\tau_I = \frac{k_c}{k_I}$ là thời gian tích phân

$\tau_D = \frac{k_D}{k_c}$ là thời gian vi phân

Các thông số bộ điều khiển có sự ảnh hưởng chéo lẫn nhau đến các chỉ tiêu chất lượng điều khiển. Đối với một số quá trình thông dụng, tác động của việc thay đổi một thông số độc lập có thể được tham khảo theo bảng sau:

Chỉ tiêu chất lượng	Thay đổi thông số bộ điều khiển		
	Tăng k_p	Giảm τ_I	Tăng τ_D
Sai lệch tĩnh	Giảm	Triệt tiêu	Ít thay đổi
Thời gian điều chỉnh	Giảm	Giảm	Giảm ít
Độ quá điều chỉnh	Tăng	Tăng	Giảm ít

2.2. Thiết kế bộ điều khiển

Phương pháp thứ nhất của Ziegler và Nichols (ZN-I) dùng để chỉnh định các thông số bộ điều khiển P, PI và PID dựa trên đường đáp ứng quá độ của đối tượng, thu được từ thực nghiệm. Quy trình thực nghiệm khá đơn giản:

- Đặt hệ thống ở chế độ điều khiển thủ công, điều chỉnh để hệ thống hoạt động ổn định tại điểm làm việc.

- Đưa một tác động nắc nhỏ (ví dụ: 3% đến 5%) vào quá trình thông qua thiết bị chấp hành và ghi lại diễn biến quá trình quá độ, đó là một đường cong đáp ứng của hệ hở (bao gồm cả quá trình, thiết bị chấp hành và thiết bị đo). Hai loại đường cong đáp ứng quá độ của hệ hở được thể hiện trên Hình 1.

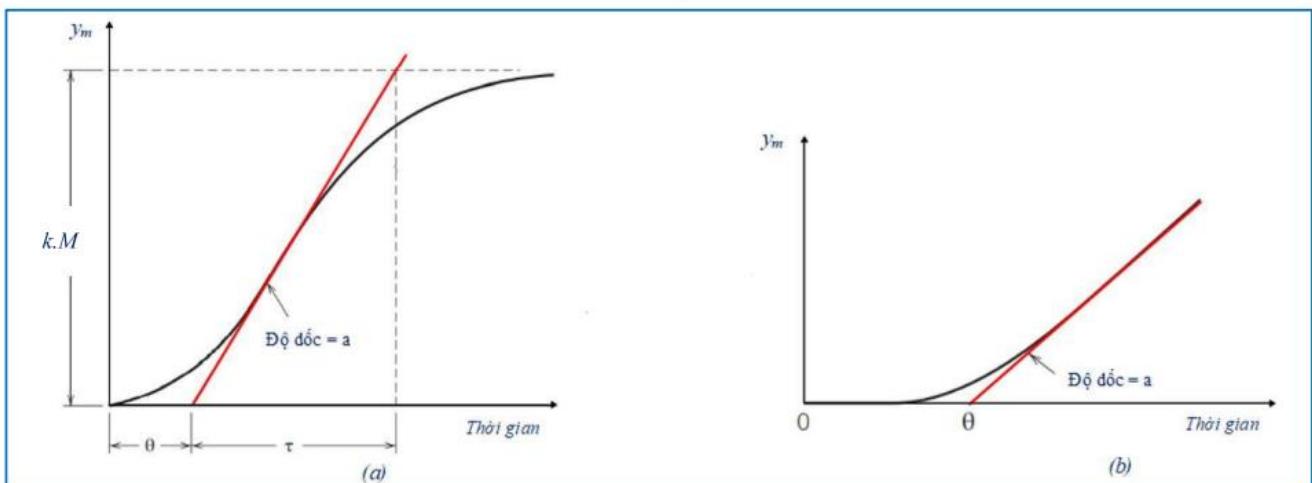
- Kẻ tiếp tuyến tại điểm có độ dốc lớn nhất của đường cong đáp ứng, xác định được hai giá trị là điểm cắt với trục hoành θ và độ dốc a .

Với đường đặc tính đáp ứng hàm nắc M/s như ở Hình 1a, ta tiến hành nhận dạng theo phương pháp kẻ tiếp tuyến. Mô hình ước lượng của quá trình là khâu quan tính bậc nhất có trễ với các tham số: hệ số khuếch đại k , hằng số thời gian t và thời gian trễ θ được xác định từ đồ thị.

$$G = \frac{k \times e^{-\theta s}}{\tau s + 1}$$

Với đường đặc tính đáp ứng hàm nắc M/s như trên Hình 1b, ta tiến hành nhận dạng Mô hình ước lượng của quá trình là khâu tích phân với các tham số $k' = a/M$ và thời gian trễ θ , trong đó a là độ dốc của đường tiếp tuyến.

$$G = \frac{k' \times e^{-\theta s}}{\tau s + 1}$$



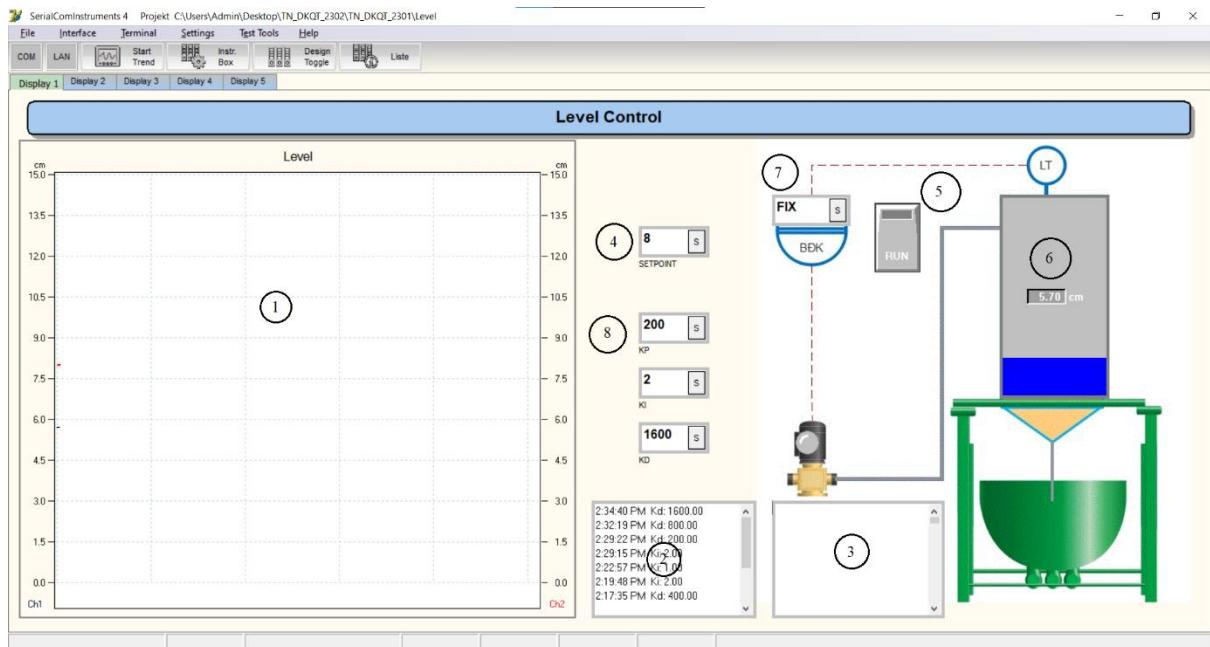
Hình 1. Đường đáp ứng quá độ của hệ hở

Các tham số của bộ điều khiển được chỉnh định theo công thức bảng sau:

Kiểu bộ điều khiển	G_c	k_c	τ_i	τ_d
Tỷ lệ (P)	k_c	$\frac{\tau}{k\theta}$ hay $\frac{1}{k'\theta}$	-	-
Tỷ lệ - tích phân (PI)	$k_c(1 + \frac{1}{\tau_i s})$	$\frac{0,9\tau}{k\theta}$ hay $\frac{0,9}{k'\theta}$	$3,3\theta$	-
Tỷ lệ - tích phân- vi phân (PID)	$k_c(1 + \frac{1}{\tau_i s} + \tau_d s)$	$\frac{1,2\tau}{k\theta}$ hay $\frac{1,2}{k'\theta}$	2θ	$0,5\theta$

Phương pháp ZN-I có nhược điểm là không áp dụng được cho quá trình không ổn định (trừ khi chỉ chứa một khâu tích phân bậc nhất) và phương pháp kế tiếp tuyến cho các số liệu kém chính xác. Theo kinh nghiệm phương pháp này chỉ áp dụng cho các quá trình có tỷ số θ/τ từ 0,1-0,6.

3. Mô tả thiết bị thí nghiệm



Hình 2. Giao diện điều khiển mức chất lỏng

- ① Đồ thị hiển thị giá trị đại lượng cần điều khiển theo thời gian thực.
- ② và ③ Đồ thị hiển thị các thông số trạng thái hệ thống
- ④ Giá trị cài đặt của đại lượng cần điều khiển

- ⑤ Khảo sát các bộ điều khiển tự động
- ⑦ Lựa chọn bộ điều khiển ON-OFF hoặc PID để khảo sát
- ⑧ Nhập các thông số k_P , k_I , k_D của bộ điều khiển PID

4. Quy trình thực hiện

- Nhập giá trị biến cần điều khiển mong muốn y_{sp}
- Chọn chế độ Tune trên giao diện điều khiển tại ô số ⑦
- Thay đổi các giá trị của các thông số của bộ điều khiển tại ô số ⑧
- Ghi nhận chỉ tiêu chất lượng vào bảng kết quả

5. Kết quả và bàn luận

5.1. Thí nghiệm 1: Khảo sát sự ảnh hưởng các thông số bộ điều khiển PID

a) Khảo sát ảnh hưởng của tham số k_P

k_P	100	200	400
Thời gian xác lập(s)	96	83	60
Sai lệch xác lập	1.36	0.72	0.81
Độ quá điều chỉnh (%)	3.85	2.71	3.02

b) Khảo sát ảnh hưởng của tham số k_I

k_I	2	4	8
Thời gian xác lập(s)	110	66	43
Sai lệch xác lập	0.19	0.12	0.8
Độ quá điều chỉnh (%)	1.52	2.39	3.4

c) Khảo sát ảnh hưởng của tham số k_D

k_D	20	40	80

<i>Thời gian xác lập(s)</i>	91	87	68
<i>Sai lệch xác lập</i>	0.11	0.06	0.04
<i>Độ quá điều chỉnh (%)</i>	1.82	0.91	0.83

5.2. Bàn luận

- Bộ điều khiển P: Giá trị tối ưu của hệ số k_P là xung quanh 200 vì tại đó thời gian xác lập là ngắn nhất, sai lệch xác lập là nhỏ nhất và độ quá điều chỉnh cũng thấp nhất.
- Bộ điều khiển PI: Khi giữ cố định giá trị $k_P = 200$ và tăng giá trị k_I , ta thấy thời gian xác lập và độ quá điều chỉnh tăng lên nhưng sai lệch xác lập giảm.
- Đối với bộ điều khiển PID: Theo số liệu thu được, với $k_P = 200$, $k_I = 4$ thì hệ số k_D tối ưu là 80 (độ quá điều chỉnh giảm, sai lệch xác lập cũng giảm).

6. Kết luận và kiến nghị

Các tham số cơ bản trong bộ điều khiển PID là tỷ lệ (k_P), tích phân (k_I) và vi phân (k_D).

Tỷ lệ (k_P): Tăng giá trị của k_P có thể làm tăng phản ứng tỷ lệ của hệ thống, giúp giảm tỷ lệ lỗi đáp ứng nhanh hơn. Tuy nhiên, tăng k_P quá mức có thể dẫn đến quá điều chỉnh và dao động. Giảm giá trị k_P có thể làm giảm dao động nhưng cũng làm tăng thời gian đáp ứng và tỷ lệ lỗi ổn định.

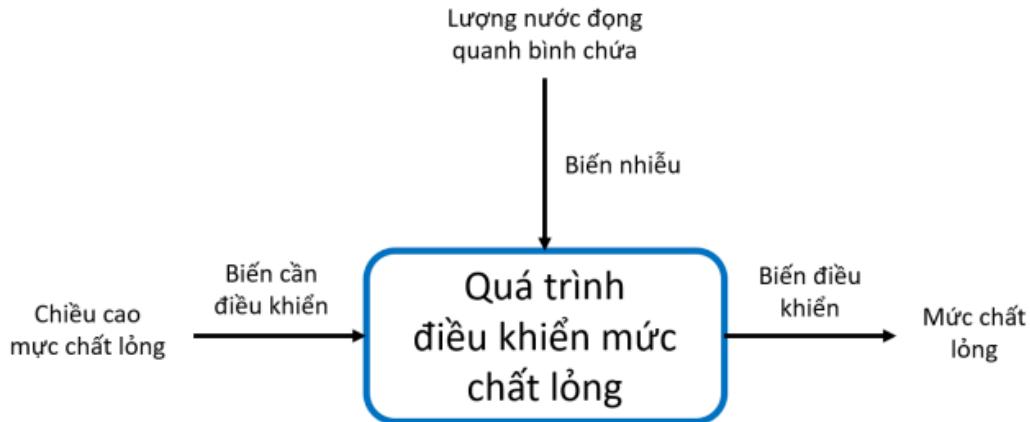
Tích phân (k_I): Tăng giá trị k_I giúp giảm tỷ lệ lỗi dài hạn và giữ cho hệ thống ổn định. Tuy nhiên, tăng k_I quá mức có thể dẫn đến dao động và quá điều chỉnh.

Vi phân (k_D): k_D giúp ổn định hệ thống bằng cách giảm dao động và quá điều chỉnh. Tăng k_D có thể làm giảm thời gian đáp ứng nhưng cũng có thể làm tăng nhiễu.

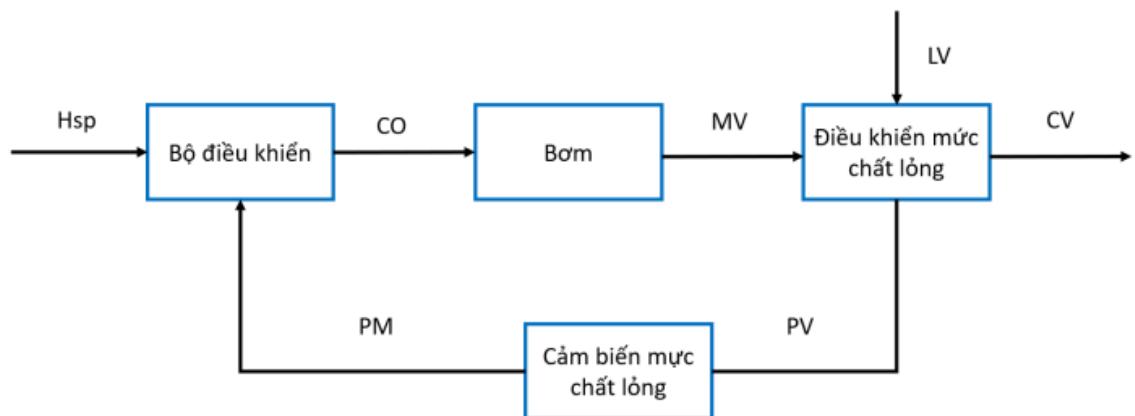
Khi điều chỉnh tham số, quan trọng là phải cân nhắc giữa độ ổn định và hiệu suất. Một số phương pháp phổ biến để điều chỉnh tham số bao gồm phương pháp thử và lỗi, phương pháp Ziegler-Nichols, phương pháp Cohen-Coon, và các phương pháp dựa trên mô hình. Tùy thuộc vào yêu cầu cụ thể của hệ thống và điều kiện vận hành, quá trình điều chỉnh tham số có thể mất thời gian và cần sự thử nghiệm và điều chỉnh liên tục để đạt được hiệu suất tối ưu.

7. Trả lời câu hỏi

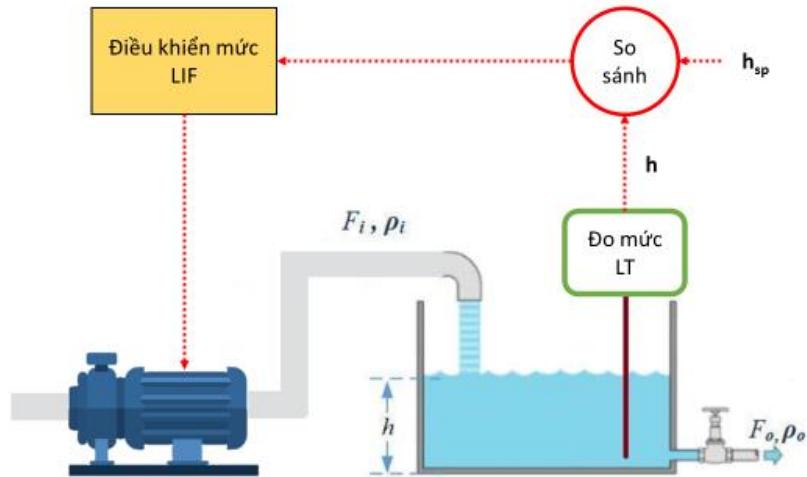
Câu 1: Vẽ sơ đồ khái tổng quát về các biến quá trình, sơ đồ khái và lưu đồ điều khiển?



Hình 3. Sơ đồ tổng quát về các biến quá trình điều khiển mức chất lỏng



Hình 4. Sơ đồ khái khái khái về quá trình điều khiển mức chất lỏng



Hình 5. Lưu đồ quá trình điều khiển mức chất lỏng

Câu 2: Mô tả thành phần của hệ thống điều khiển trong bài thí nghiệm này?

Hệ thống điều khiển gồm: Bộ điều khiển; Thiết bị chấp hành: máy bơm; Thiết bị đo: cảm biến siêu âm HC-SR04.

Câu 3: Nhận xét ảnh hưởng của các thông số bộ điều khiển đến chất lượng điều khiển?

* Nhận xét đối với từng bộ điều khiển

- Bộ điều khiển P: Thời gian xác lập nhanh nhưng chưa triệt tiêu được sai lệch xác lập, tức là sai lệch xác lập còn lớn.

- Bộ điều khiển PI: Thời gian xác lập chậm hơn so với bộ điều khiển P, nhưng sai lệch xác lập bé, có thể triệt tiêu được sai lệch tĩnh.

- Bộ điều khiển PID: Thời gian xác lập chậm nhất trong 3 bộ điều khiển đã nêu, nhưng sai lệch xác lập bé.

* Nhận xét khi thay đổi các thông số k_P , k_I và k_D

- Khi tăng k_P thì sai lệch xác lập giảm nhưng không thể triệt tiêu được nó.

- Khi tăng k_I (giữ nguyên k_P) thì thời gian xác lập và độ quá điều chỉnh tăng, nhưng sai lệch xác lập giảm.

- Khi tăng k_D (giữ nguyên k_P và k_I) thì độ quá điều chỉnh giảm, hệ chậm hơn và ổn định hơn (thời gian xác lập và sai lệch xác lập giảm).

* So sánh giá trị tối ưu từ số liệu đo được của 3 bộ điều khiển:

- Tại giá trị tối ưu của bộ điều khiển PI thì thời gian xác lập dài hơn so với bộ điều khiển P, nhưng sai lệch xác lập và độ quá điều chỉnh chính thấp hơn
 - So với bộ điều khiển PI, tại giá trị tối ưu, bộ điều khiển PID có thời gian xác lập ngắn hơn; sai lệch xác lập gần bằng nhưng độ quá điều chỉnh cao hơn
- So sánh cả 3 chỉ tiêu trên thì bộ điều khiển PI đáp ứng được tốt nhất.

8. Tài liệu tham khảo

- [1]. Điều khiển Quá trình Công nghệ Hóa học - Cơ sở điều khiển Quá trình – Quyển 1.
- [2]. Điều khiển Quá trình Công nghệ Hóa học – Hướng dẫn thí nghiệm, Thực hành cơ sở Điều khiển – Quyển 2.