

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM  
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG  
KHOA ĐIỆN – ĐIỆN TỬ



NGUYỄN HOÀNG VŨ

**THIẾT KẾ BỘ ĐIỀU KHIỂN PID BẰNG  
MONGODB REALM**

**ĐỒ ÁN TỔNG HỢP  
KỸ THUẬT ĐIỀU KHIỂN VÀ  
TỰ ĐỘNG HÓA**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM  
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG  
KHOA ĐIỆN – ĐIỆN TỬ



NGUYỄN HOÀNG VŨ - 41801295

**THIẾT KẾ BỘ ĐIỀU KHIỂN PID BẰNG  
MONGODB REALM**

**ĐỒ ÁN TỔNG HỢP  
KỸ THUẬT ĐIỀU KHIỂN VÀ TỰ ĐỘNG HÓA**

Người hướng dẫn  
**TS. Trần Việt Hùng**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

## LỜI CẢM ƠN

Đồ án tốt nghiệp là một môn học quan trọng và cấp thiết, giúp sinh viên Khoa Điện – Điện tử có thêm kỹ năng và kiến thức để tiếp cận sự phát triển vượt bật của ngành khoa học công nghệ thông qua việc nghiên cứu và học hỏi. Trong suốt quá trình thực hiện đồ án “*Thiết kế bộ điều khiển PID bằng Mongodb Realm*”, dưới sự hướng dẫn tận tâm của TS.Trần Việt Hùng đã giúp em hoàn thành đồ án một cách chỉnh chu và ít sai sót.

Em xin chân thành cảm ơn quý Thầy, Cô khoa Điện – Điện tử đã hỗ trợ sinh viên có cơ hội được tham gia thực hiện đồ án tốt nghiệp, cũng như những hướng dẫn và phò bồi cẩn thận về quá trình làm đồ án. Ngoài ra sự chỉ dẫn và những góp ý quý giá từ các giảng viên ngành Kỹ thuật Điều khiển và Tự động hóa đã giúp em hoàn thành đồ án một cách kịp thời và đạt được thành quả như mong đợi.

Thương chúc quý Thầy, Cô thật nhiều sức khỏe và thành công.

TP. Hồ Chí Minh, ngày tháng năm  
Tác giả

Nguyễn Hoàng Vũ

Công trình được hoàn thành tại Trường Đại học Tôn Đức Thắng  
Cán bộ hướng dẫn khoa học: Tiến sĩ Trần Việt Hùng

Đồ án tốt nghiệp/tổng hợp được bảo vệ tại **Hội đồng đánh giá Đồ án tốt nghiệp/tổng hợp của Trường Đại học Tôn Đức Thắng** vào ngày... /.../.....

Xác nhận của Chủ tịch Hội đồng đánh giá **Đồ án tốt nghiệp/tổng hợp và Trưởng khoa quản lý chuyên ngành** sau khi nhận **Đồ án tốt nghiệp/tổng hợp** đã được sửa chữ (nếu có).

**CHỦ TỊCH HỘI ĐỒNG**

**TRƯỞNG KHOA**

.....

.....

# CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi và được sự hướng dẫn khoa học của TS. Trần Việt Hùng. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong Đồ án tốt nghiệp/ tổng hợp còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung Đồ án tốt nghiệp/ tổng hợp của mình.** Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày tháng năm  
Tác giả

Nguyễn Hoàng Vũ

(Trang này dùng để đính kèm Nhiệm vụ Đồ án tốt nghiệp có chữ ký của Giảng viên hướng dẫn)



# THIẾT KẾ BỘ ĐIỀU KHIỂN PID BẰNG MONGODB REALM

## TÓM TẮT

Đồ án tính toán các thông số của bộ điều khiển PID. Bên cạnh đó áp dụng bộ lọc Kalman để so sánh tính ổn định của bộ điều khiển PID trước và sau khi có bộ lọc Kalman. Dùng phương pháp điều khiển online MongodB Realm để nhận các biến ngoài vào và thực thi tính toán trên MongodB Realm sau đó trả về các biến ngoài mong muốn.

# **DESIGN A PID CONTROLLER WITH MONGODB REALM**

## **ABSTRACT**

The project to calculate the parameters of the PID controller. Besides, the Kalman filter is applied to compare the stability of the PID controller before and after having the Kalman filter. Use Mongodb Realm online control method to get input variables and execute calculation on Mongodb Realm then return desired output variables.

## **MỤC LỤC**

<b>DANH MỤC HÌNH VẼ .....</b>	<b>IV</b>
<b>DANH MỤC BẢNG BIỂU .....</b>	<b>VIII</b>
<b>DANH MỤC CÁC CHỮ VIẾT TẮT .....</b>	<b>IX</b>
<b>CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI .....</b>	<b>1</b>
1.1. MỤC ĐÍCH THỰC HIỆN ĐỀ TÀI .....	1
1.2. Ý TƯỞNG VÀ PHƯƠNG PHÁP THỰC HIỆN .....	1
<b>CHƯƠNG 2. CƠ SỞ LÝ THUYẾT .....</b>	<b>3</b>
2.1. PHẦN MỀM LẬP TRÌNH ARDUINO IDE .....	3
2.1.1. <i>Arduino IDE là gì ?</i> .....	3
2.1.2. <i>Nguyên lý hoạt động</i> .....	3
2.2. NODE-RED .....	5
2.2.1. <i>Node RED là gì ?</i> .....	5
2.2.2. <i>Các tính năng của Node-RED</i> .....	8
2.3. MONGODB .....	9
2.3.1. <i>Mongodb là gì ?</i> .....	9
2.3.2. <i>Nguyên lý hoạt động</i> .....	10
2.3.3. <i>Ưu, nhược điểm của Mongodb</i> .....	11
2.4. PHƯƠNG THỨC TRUYỀN THÔNG HTTP .....	12
2.4.1. <i>HTTP là gì ?</i> .....	12
2.4.2. <i>Nguyên lý hoạt động</i> .....	12
2.4.3. <i>Uniform Resource Locator (URL)</i> .....	13
2.5. BỘ ĐIỀU KHIỂN PID .....	15
2.5.1. <i>PID là gì ?</i> .....	15
2.5.2. <i>Các loại bộ điều khiển</i> .....	17
2.6. BỘ LỌC KALMAN .....	17
2.7. ESP32 .....	21
2.7.1. <i>ESP32 là gì ?</i> .....	21

**ĐỒ ÁN TỔNG HỢP**  
**Trang - II**

---

2.7.2. CẤU TẠO .....	22
2.8. CẢM BIẾN SIÊU ÂM HC_SR04 .....	23
2.8.1. HC_SR04 là gì ? .....	23
2.8.2. Cấu tạo .....	24
2.8.3. Nguyên lý hoạt động .....	25
2.9. MODULE L298N .....	25
2.9.1. Module L298N là gì ? .....	25
2.9.2. Cấu tạo .....	26
2.9.3. Nguyên lý hoạt động .....	27
2.10. ĐỘNG CƠ BƠM NƯỚC R385 WATER PUMP 12VDC .....	27
2.11. NGUỒN ĐIỆN TỬ ONG .....	29
2.12. CẢM BIẾN LUU LUONG YF-S401 .....	29
<b>CHƯƠNG 3. THIẾT KẾ BỘ ĐIỀU KHIỂN .....</b>	<b>31</b>
3.1. THIẾT KẾ HỆ THỐNG .....	31
3.2. THIẾT KẾ MONGODB .....	32
3.3. THIẾT KẾ MONGODB REALM .....	33
3.4. THIẾT KẾ NODE-RED .....	34
3.5. THIẾT KẾ BỒN CHỨA CHẤT LỎNG .....	37
3.6. BẢN VẼ SƠ ĐỒ ĐẦU DÂY .....	39
<b>CHƯƠNG 4: LUU ĐÒ GIẢI THUẬT VÀ ĐIỀU KHIỂN .....</b>	<b>41</b>
4.1. HOẠT ĐỘNG CỦA HỆ THỐNG .....	41
4.2. LUU ĐÒ GIẢI THUẬT CỦA ESP32 .....	43
4.3. LUU ĐÒ GIẢI THUẬT PID TRÊN MONGODB REALM .....	44
4.4. TÍNH TOÁN LÝ THUYẾT .....	45
4.4.1. Bô P .....	45
4.4.2. Bô PI .....	50
4.4.3. Bô PD .....	62
4.4.4. Bô PID .....	72
4.4.5. Nhận xét .....	88
<b>CHƯƠNG 5: THỰC NGHIỆM .....</b>	<b>93</b>

**ĐỒ ÁN TỔNG HỢP**  
**Trang - III**

---

5.1. BỘ ĐIỀU KHIỂN P .....	93
5.2. BỘ ĐIỀU KHIỂN PI .....	95
5.3. BỘ ĐIỀU KHIỂN PD .....	97
5.4. BỘ ĐIỀU KHIỂN PID .....	99
5.5. KẾT LUẬN .....	101
5.6. SO SÁNH GIỮA SENSOR KHI CÓ KALMAN VÀ KHÔNG CÓ KALMAN .....	103
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>105</b>
<b>PHỤ LỤC A: CHƯƠNG TRÌNH ESP .....</b>	<b>106</b>
<b>PHỤ LỤC B CÔNG THỨC HÀM TRUYỀN LAPLACE CỦA HỆ THỐNG VÀ BỘ ĐIỀU KHIỂN P PI PD .....</b>	<b>112</b>

## **DANH MỤC HÌNH ẢNH**

<b>HÌNH 2.1.1: PHẦN MỀM ARDUINO IDE .....</b>	<b>3</b>
<b>HÌNH 2.2.1: PHẦN MỀM NODE-RED .....</b>	<b>5</b>
<b>HÌNH 2.2.1.1 : MÔ TẢ MỘT VÀI CHỨA NẮNG CỦA NODE-RED .....</b>	<b>6</b>
<b>HÌNH 2.2.1.2: GIAO DIỆN CỦA NODE-RED .....</b>	<b>7</b>
<b>HÌNH 2.3.1.1. PHẦN MỀM MONGODB .....</b>	<b>9</b>
<b>HÌNH 2.3.1.2: KIÊU DỮ LIỆU OBJECT .....</b>	<b>10</b>
<b>HÌNH 2.3.2.1: NGUYÊN LÝ HOẠT ĐỘNG CỦA MONGODB .....</b>	<b>11</b>
<b>HÌNH 2.4.2.1: SƠ ĐỒ NGUYÊN LÝ HOẠT ĐỘNG CỦA HTTP .....</b>	<b>13</b>
<b>HÌNH: 2.5.1.1: SƠ ĐỒ BỘ ĐIỀU KHIỂN PID .....</b>	<b>15</b>
<b>HÌNH 2.6.1: VÍ DỤ VỀ CHỐNG NHIỄU CỦA BỘ LỌC KALMAN .....</b>	<b>20</b>
<b>HÌNH 2.7.1.1. ESP32 DEV.BOARD .....</b>	<b>21</b>
<b>HÌNH 2.8.1.1. CẢM BIẾN SIÊU ÂM HC_SR04.....</b>	<b>23</b>
<b>HÌNH 2.8.3.1. NGUYÊN LÝ HOẠT ĐỘNG CỦA CẢM BIẾN SIÊU ÂM .....</b>	<b>25</b>
<b>HÌNH 2.9.1.1. MODULE L298N .....</b>	<b>26</b>
<b>HÌNH 2.10.1. BƠM NƯỚC R385 12VDC .....</b>	<b>28</b>
<b>HÌNH 2.12.1. CẢM BIẾN LUU LUQONG YF-S401 .....</b>	<b>30</b>
<b>HÌNH 3.1.1. SƠ ĐỒ HOẠT ĐỘNG CỦA HỆ THỐNG .....</b>	<b>31</b>
<b>HÌNH 3.2.1. DỮ LIỆU INPUT ĐƯỢC LUU TRÊN MONGODB .....</b>	<b>32</b>
<b>HÌNH 3.2.2. DỮ LIỆU ĐỂ THỰC HIỆN THUẬT TOÁN PID .....</b>	<b>32</b>
<b>HÌNH 3.2.3. DỮ LIỆU ĐỂ THỰC HIỆN THUẬT TOÁN KALMAN FILTER.....</b>	<b>33</b>
<b>HÌNH 3.3.1. MINH HOẠT LẬP TRÌNH TRÊN MONGODB REALM .....</b>	<b>33</b>
<b>HÌNH 3.4.1. CÁC FLOW CỦA NODE-RED .....</b>	<b>34</b>
<b>HÌNH 3.4.2. CÁC FLOW CỦA NODE-RED .....</b>	<b>35</b>
<b>HÌNH 3.4.3. CÁC FLOW CỦA NODE-RED .....</b>	<b>35</b>

<b>HÌNH 3.4.4. GIAO DIỆN DASHBOARD CỦA NODE-RED .....</b>	<b>36</b>
<b>HÌNH 3.5.1. MÔ HÌNH BỒN CHÚA CHẤT LỎNG .....</b>	<b>38</b>
<b>HÌNH 3.6.1. SƠ ĐỒ ĐẦU NỐI DÂY .....</b>	<b>39</b>
<b>HÌNH 3.6.2. BẢN VẼ MẠCH IN PCB .....</b>	<b>39</b>
<b>HÌNH 3.6.3. MẶT TRÊN CỦA BOARD .....</b>	<b>40</b>
<b>HÌNH 3.6.4. MẶT DƯỚI CỦA BOARD .....</b>	<b>40</b>
<b>HÌNH 4.1.1. THAO TÁC MỞ NODE-RED .....</b>	<b>41</b>
<b>HÌNH 4.1.2. THAO TÁC MỞ NODE-RED .....</b>	<b>42</b>
<b>HÌNH 4.1.3. GIAO DIỆN DASHBOARD CỦA NODE-RED .....</b>	<b>42</b>
<b>HÌNH 4.4.1.1. ĐỒ THỊ MÔ PHỎNG BỘ P (KHÔNG NHIỀU).....</b>	<b>46</b>
<b>HÌNH 4.1.1.2. ĐỒ THỊ MÔ PHỎNG BỘ P (CÓ NHIỀU).....</b>	<b>49</b>
<b>HÌNH 4.4.2.1. ĐỒ THỊ MÔ PHỎNG BỘ PI(KHÔNG NHIỀU).....</b>	<b>55</b>
<b>HÌNH 4.4.2.2. ĐỒ THỊ MÔ PHỎNG BỘ PI (CÓ NHIỀU).....</b>	<b>61</b>
<b>HÌNH 4.4.3.1. ĐỒ THỊ MÔ PHỎNG BỘ PD (KHÔNG NHIỀU).....</b>	<b>66</b>
<b>HÌNH 4.4.3.2. ĐỒ THỊ MÔ PHỎNG BỘ PD (CÓ NHIỀU).....</b>	<b>71</b>
<b>HÌNH 4.4.4.1. ĐỒ THỊ MÔ PHỎNG BỘ PID(KHÔNG NHIỀU).....</b>	<b>79</b>
<b>HÌNH 4.4.4.2. ĐỒ THỊ MÔ PHỎNG BỘ PID (CÓ NHIỀU).....</b>	<b>87</b>
<b>HÌNH 4.4.5.1. ĐỒ THỊ MÔ PHỎNG BỘ P (KHÔNG NHIỀU).....</b>	<b>88</b>
<b>HÌNH 4.4.5.2. ĐỒ THỊ MÔ PHỎNG BỘ P (CÓ NHIỀU).....</b>	<b>88</b>
<b>HÌNH 4.4.5.3. ĐỒ THỊ MÔ PHỎNG BỘ PI (KHÔNG NHIỀU).....</b>	<b>89</b>
<b>HÌNH 4.4.5.4. ĐỒ THỊ MÔ PHỎNG BỘ PI (CÓ NHIỀU).....</b>	<b>89</b>
<b>HÌNH 4.4.5.5. ĐỒ THỊ MÔ PHỎNG BỘ PD (KHÔNG NHIỀU).....</b>	<b>90</b>
<b>HÌNH 4.4.5.6. ĐỒ THỊ MÔ PHỎNG BỘ PD (CÓ NHIỀU).....</b>	<b>90</b>
<b>HÌNH 4.4.5.7. ĐỒ THỊ MÔ PHỎNG BỘ PID (KHÔNG NHIỀU).....</b>	<b>91</b>
<b>HÌNH 4.4.5.8. ĐỒ THỊ MÔ PHỎNG BỘ PID (CÓ NHIỀU).....</b>	<b>91</b>

<b>HÌNH 5.1.1. ĐỒ THỊ MỰC CHẤT LỎNG THEO THỜI GIAN CỦA BỘ ĐIỀU KHIỂN P TRÊN NODE-RED(KHÔNG NHIỄU) .....</b>	<b>93</b>
<b>HÌNH 5.1.2. ĐỒ THỊ MỰC CHẤT LỎNG THEO THỜI GIAN CỦA BỘ ĐIỀU KHIỂN P TRÊN MONGODB CHART(KHÔNG NHIỄU) .....</b>	<b>93</b>
<b>HÌNH 5.1.3. ĐỒ THỊ MỰC CHẤT LỎNG THEO THỜI GIAN CỦA BỘ ĐIỀU KHIỂN P TRÊN NODE-RED(CÓ NHIỄU) .....</b>	<b>94</b>
<b>HÌNH 5.1.4. ĐỒ THỊ MỰC CHẤT LỎNG THEO THỜI GIAN CỦA BỘ ĐIỀU KHIỂN P TRÊN MONGODB CHART(CÓ NHIỄU) .....</b>	<b>94</b>
<b>HÌNH 5.2.1. ĐỒ THỊ MỰC CHẤT LỎNG THEO THỜI GIAN CỦA BỘ ĐIỀU KHIỂN PI TRÊN NODE-RED(KHÔNG NHIỄU) .....</b>	<b>95</b>
<b>HÌNH 5.2.2. ĐỒ THỊ MỰC CHẤT LỎNG THEO THỜI GIAN CỦA BỘ ĐIỀU KHIỂN PI TRÊN MONGODB CHART(KHÔNG NHIỄU) .....</b>	<b>95</b>
<b>HÌNH 5.2.3. ĐỒ THỊ MỰC CHẤT LỎNG THEO THỜI GIAN CỦA BỘ ĐIỀU KHIỂN PI TRÊN NODE-RED(CÓ NHIỄU) .....</b>	<b>96</b>
<b>HÌNH 5.2.1. ĐỒ THỊ MỰC CHẤT LỎNG THEO THỜI GIAN CỦA BỘ ĐIỀU KHIỂN PI TRÊN MONGODB CHART(CÓ NHIỄU) .....</b>	<b>96</b>
<b>HÌNH 5.3.1. ĐỒ THỊ MỰC CHẤT LỎNG THEO THỜI GIAN CỦA BỘ ĐIỀU KHIỂN PD TRÊN NODE-RED(KHÔNG NHIỄU) .....</b>	<b>97</b>
<b>HÌNH 5.3.2. ĐỒ THỊ MỰC CHẤT LỎNG THEO THỜI GIAN CỦA BỘ ĐIỀU KHIỂN PD TRÊN MONGODB CHART(KHÔNG NHIỄU) .....</b>	<b>97</b>
<b>HÌNH 5.3.3. ĐỒ THỊ MỰC CHẤT LỎNG THEO THỜI GIAN CỦA BỘ ĐIỀU KHIỂN PD TRÊN NODE-RED(CÓ NHIỄU) .....</b>	<b>98</b>
<b>HÌNH 5.3.4. ĐỒ THỊ MỰC CHẤT LỎNG THEO THỜI GIAN CỦA BỘ ĐIỀU KHIỂN PD TRÊN MONGODB CHART(CÓ NHIỄU) .....</b>	<b>98</b>
<b>HÌNH 5.4.1. ĐỒ THỊ MỰC CHẤT LỎNG THEO THỜI GIAN CỦA BỘ ĐIỀU KHIỂN PID TRÊN NODE-RED(KHÔNG NHIỄU) .....</b>	<b>99</b>
<b>HÌNH 5.4.2. ĐỒ THỊ MỰC CHẤT LỎNG THEO THỜI GIAN CỦA BỘ ĐIỀU KHIỂN PID TRÊN MONGODB CHART(KHÔNG NHIỄU) .....</b>	<b>99</b>

**ĐỒ ÁN TỔNG HỢP**  
**Trang - VII**

---

<b>HÌNH 5.4.3. ĐỒ THỊ MỰC CHẤT LỎNG THEO THỜI GIAN CỦA BỘ ĐIỀU KHIỂN PID TRÊN NODE-RED(CÓ NHIỀU) .....</b>	<b>100</b>
<b>HÌNH 5.4.4. ĐỒ THỊ MỰC CHẤT LỎNG THEO THỜI GIAN CỦA BỘ ĐIỀU KHIỂN PID TRÊN MONGODB CHART(CÓ NHIỀU) .....</b>	<b>100</b>
<b>HÌNH 5.6.1. ĐỒ THỊ THỰC NGHIỆM CỦA CẢ 2 CẨM BIẾN .....</b>	<b>103</b>
<b>HÌNH 5.6.2. ĐỒ THỊ THỰC NGHIỆM KALMAN CHO CẨM BIẾN .....</b>	<b>104</b>

## **DANH MỤC BẢNG BIỂU**

<b>BẢNG 2.4.4.1: METHOD HTTP .....</b>	<b>14</b>
<b>BẢNG 2.5.1.1. TÁC ĐỘNG CỦA VIỆC TĂNG CÁC THÔNG SỐ ĐỘC LẬP CỦA PID17</b>	
<b>BẢNG 4.4.1.1. GIÁ TRỊ H CỦA BỘ P THEO THỜI GIAN T(S).....</b>	<b>46</b>
<b>BẢNG 4.4.1.2. GIÁ TRỊ H CỦA BỘ P THEO THỜI GIAN T(S) (CÓ NHIỀU).....</b>	<b>49</b>
<b>BẢNG 4.4.2.1. GIÁ TRỊ H CỦA BỘ PI THEO THỜI GIAN T(S) .....</b>	<b>55</b>
<b>BẢNG 4.4.2.2. GIÁ TRỊ H CỦA BỘ PI THEO THỜI GIAN T(S) (CÓ NHỀU) .....</b>	<b>61</b>
<b>BẢNG 4.4.3.1. GIÁ TRỊ H CỦA BỘ PD THEO THỜI GIAN T(S) .....</b>	<b>66</b>
<b>BẢNG 4.4.3.2. GIÁ TRỊ H CỦA BỘ PD THEO THỜI GIAN T(S) (CÓ NHIỀU) .....</b>	<b>71</b>
<b>BẢNG 4.4.4.1. GIÁ TRỊ H CỦA BỘ PID THEO THỜI GIAN T(S) .....</b>	<b>79</b>
<b>BẢNG 4.4.4.2. GIÁ TRỊ H CỦA BỘ PID THEO THỜI GIAN T(S) (CÓ NHIỀU).....</b>	<b>87</b>

## **DANH MỤC CÁC CHỮ VIẾT TẮT**

IoT: Internet of Thing

PID: Proportional, Integral, Derivative

## **CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI**

### **1.1. Mục đích thực hiện đề tài**

Điều khiển online cho các hệ thống cần bộ điều khiển PID để cân bằng một giá trị mong muốn nào đó. Ví dụ: hệ thống cần điều khiển đặt tại điểm A bất kỳ chỉ cần hệ thống có kết nối internet, thì người dùng chỉ cần đăng nhập vào web bằng internet thì vẫn có thể tác động đến hệ thống từ mọi nơi chỉ cần một thiết bị thông minh có khả năng truy cập vào web như máy tính, laptop, điện thoại,...

### **1.2. Ý tưởng và phương pháp thực hiện**

Do thời đại 4.0 internet phát triển mạnh nên chúng ta phải nắm bắt được xu hướng của thế giới, IoT cho mọi thứ từ các mô hình vườn cây thông minh phục vụ cho việc canh tác tự động trong nông nghiệp giúp người nông dân bỏ ra ít công sức hơn mà còn cho ra chất lượng sản phẩm tốt hơn, cho đến các mô hình nhà thông minh hiện đại ngày nay giúp chúng ta có nhiều tiện nghi hơn, nâng cao chất lượng cuộc sống của con người. Đủ cho ta thấy rằng, nhờ có Internet mà cuộc sống và công việc của con người ngày càng trở nên dễ dàng hơn. Chính vì lẽ đó nên em cũng sẽ áp dụng IoT vào đồ án của mình, giúp chúng ta có thể dễ dàng giám sát hoạt động của các thiết bị khi ta ở bất cứ đâu.

Mongodb Realm là một trong những app services giúp chúng ta có thể lập trình tính toán và xuất giá trị sau tính toán về hệ thống để thực hiện các yêu cầu mà người dùng mong muốn.

### **1.3. Đối tượng nghiêm túc**

- Phần mềm lập trình Arduino.
- Chương trình Node-RED.
- Web Mongodb Realm.
- Phương thức truyền thông HTTP.
- Truyền, nhận dữ liệu của ESP32 và Mongodb(database và Realm).
- Truyền, nhận dữ liệu của Node-RED và Mongodb(database và Realm).
- Bộ điều khiển PID.

- Bộ lọc Kalman.
- Mô hình: cảm biến siêu âm HC-SR04, cảm biến lưu lượng YF-S401, 2 bơm 12V MB385, module L298, ESP32, bể chứa, ống nhựa dẫn chất lỏng.

#### **1.4. Phạm vi nghiên cứu**

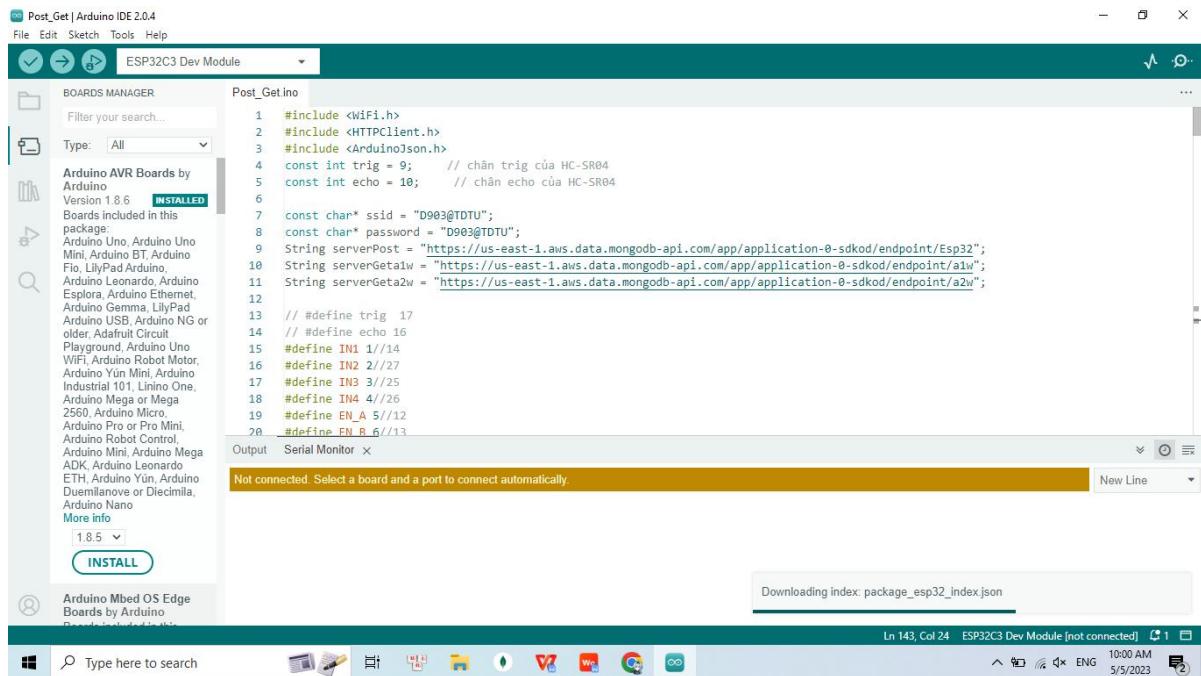
Với góc độ em đang thực hiện đồ án tổng hợp, hạn chế về kinh phí, thời gian nên mô hình hệ thống điều khiển quá trình chất lỏng xây dựng dựa trên bể chứa nhỏ, máy bơm công suất nhỏ, ống nhựa dẫn nước có đường kính nhỏ.

#### **1.5. Dự kiến kết quả**

Hệ thống sẽ được điều khiển bằng các nút mềm và đồ thị trên Node RED, thuật toán điều khiển PID được lập trình trên Mongodb Realm bằng ngôn ngữ Javascript và lưu trữ, giám sát thông tin và dữ liệu trên Mongodb Database.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

### 2.1. Phần mềm lập trình Arduino IDE



Hình 2.1.1: Phần mềm Arduino IDE

#### 2.1.1. Arduino IDE là gì ?

- Arduino IDE là một phần mềm với một mã nguồn mở, được sử dụng chủ yếu để viết và biên dịch mã vào module Arduino. Nó bao gồm phần cứng và phần mềm. Phần cứng chứa đến 300,000 board mạch được thiết kế sẵn với các cảm biến, linh kiện. Phần mềm giúp bạn có thể sử dụng các cảm biến, linh kiện ấy của Arduino một cách linh hoạt phù hợp với mục đích sử dụng.

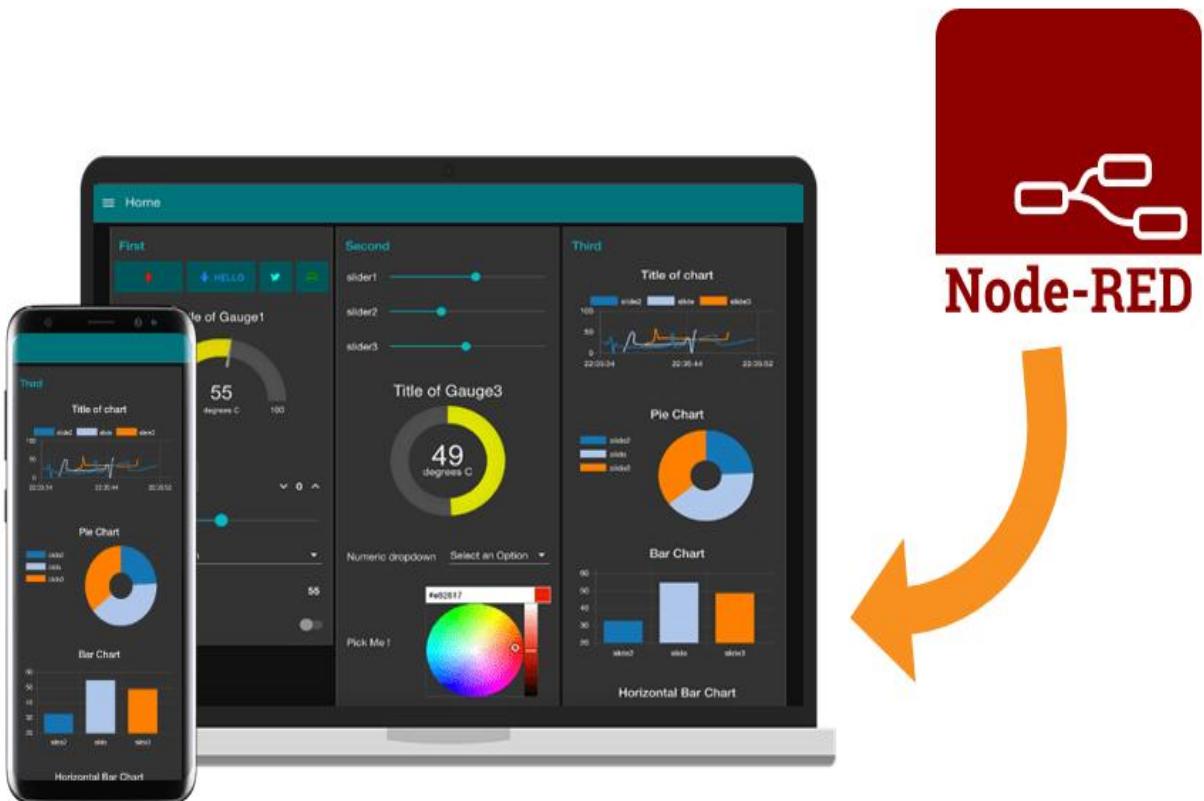
- Đây là một phần mềm Arduino chính thống, giúp cho việc biên dịch mã trở nên dễ dàng, ngay cả một người bình thường không có kiến thức kỹ thuật cũng có thể làm được.

#### 2.1.2. Nguyên lý hoạt động

- Khi người dùng viết mã và biên dịch, IDE sẽ tạo file Hex cho mã. File Hex là các file thập phân Hexa được Arduino hiểu và gửi đến bo mạch bằng cáp USB. Mỗi board

Arduino, ESP đều được tích hợp một bộ vi điều khiển, bộ vi điều khiển sẽ nhận file Hex và chạy theo mã được viết.

## 2.2. Node-RED

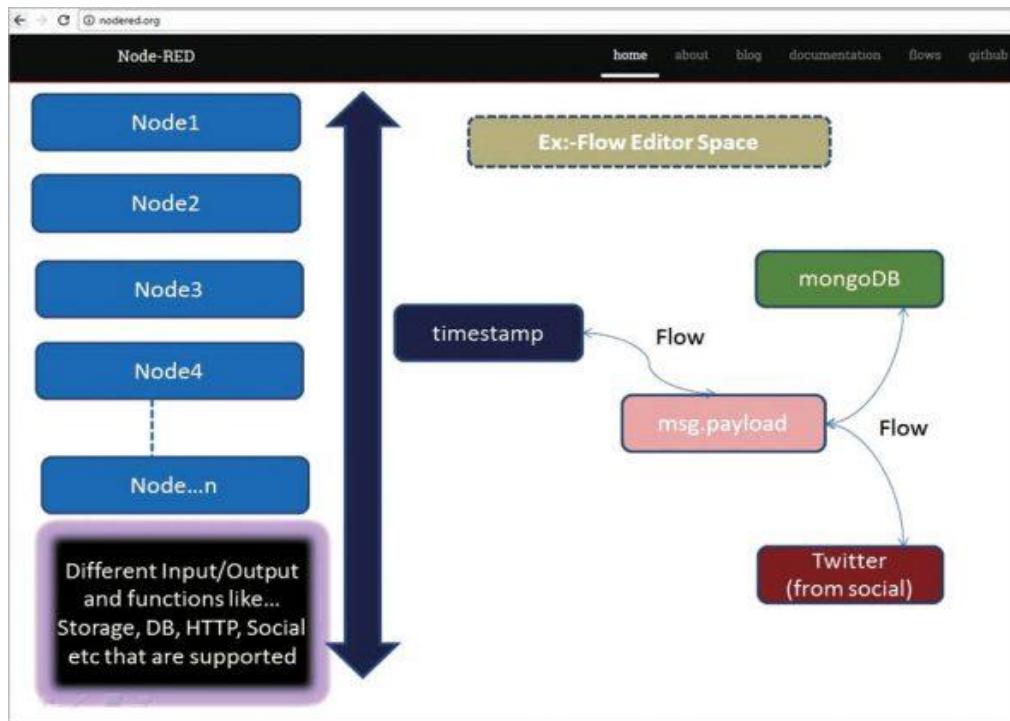


Hình 2.2.1: Phần mềm Node-RED

### 2.2.1. Node RED là gì ?

- Node RED là một công cụ lập trình dùng để kết nối các thiết bị phần cứng, API và các dịch vụ trực tuyến với nhau. Về cơ bản, đây là một công cụ trực quan được thiết kế cho IoT (Internet of Things), nhưng cũng có thể được sử dụng cho các ứng dụng khác nhằm liên kết nhanh các luồng (flow) dịch vụ khác nhau.
- Node-RED là mã nguồn mở và được phát triển bởi Emerging Technology Services của IBM và được tích hợp trong gói ứng dụng khởi động IoT Bluemix của IBM (Platform-as-a-Service hoặc PaaS). Node-RED cũng có thể được triển khai riêng bằng ứng dụng Node.js. Hiện tại, Node-RED là một dự án của JS Foundation.

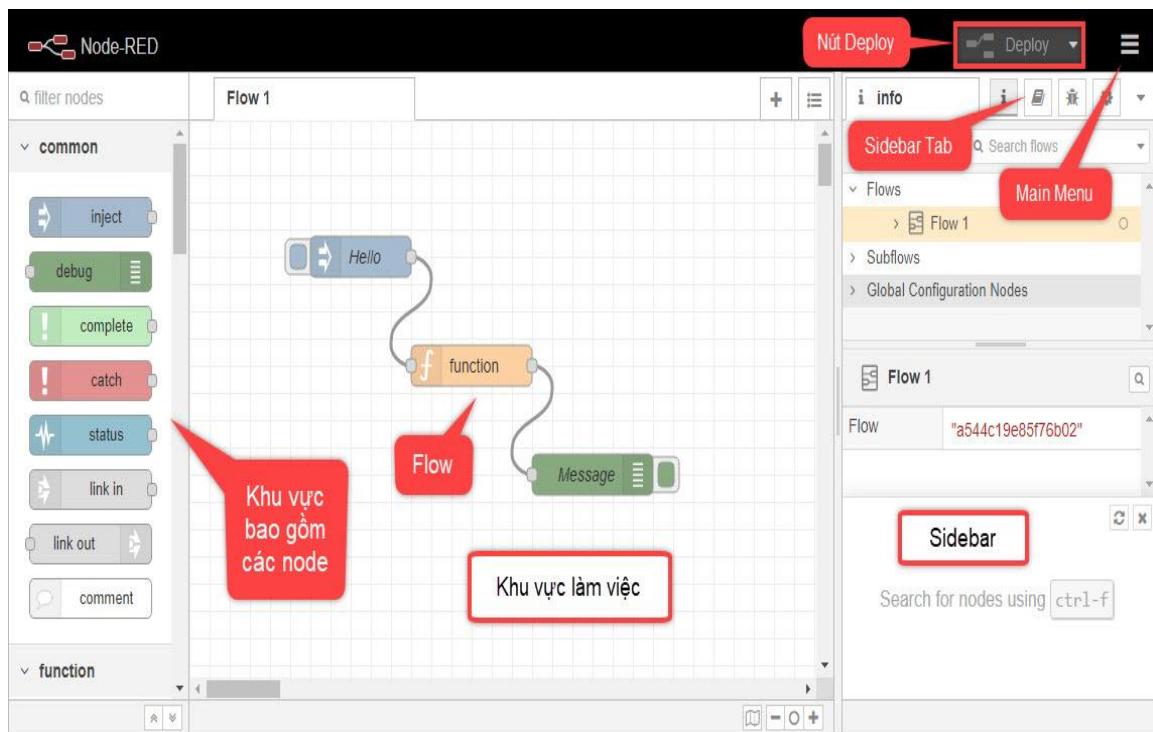
- Node-RED cho phép người dùng kết hợp các dịch vụ Web và phần cứng bằng cách thay thế các tác vụ mã hóa cấp thấp phổ biến (như một dịch vụ đơn giản giao tiếp với một cổng nối tiếp) và điều này có thể được thực hiện với giao diện kéo thả trực quan. Các thành phần khác nhau trong Node-RED được kết nối với nhau để tạo ra một luồng (flow). Hầu hết mã lệnh (code) cần thiết được tạo tự động.



Hình 2.2.1.1 : Mô tả một vài chức năng của Node-RED

# ĐỒ ÁN TỔNG HỢP

## Trang - 7



Hình 2.2.1.2: Giao diện của Node-RED

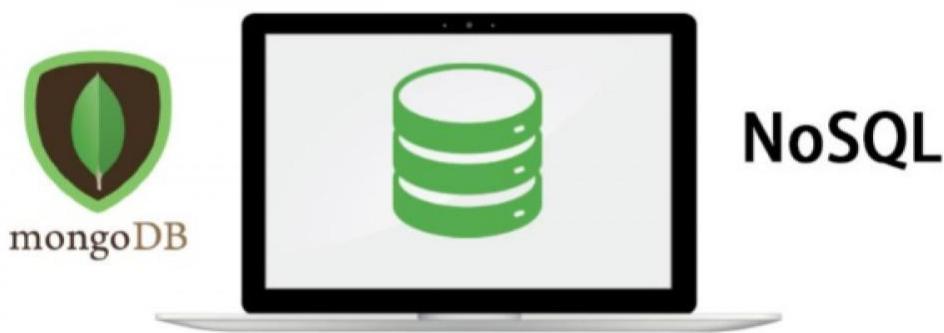
### **2.2.2. Các tính năng của Node-RED**

- Cho phép chỉnh sửa luồng (flow) ngay trên trình duyệt.
- Vì được xây dựng dựa trên Node.js, nên Node-RED hỗ trợ môi trường thời gian thực nhẹ cùng với mô hình hướng sự kiện và không chặn.
- Các luồng khác nhau được tạo trong Node-RED được lưu trữ bằng JSON, có thể dễ dàng nhập và xuất để chia sẻ với những người khác.
- Bạn có thể chạy Node-RED cục bộ (hỗ trợ Docker, v.v.).
- Node-RED có thể dễ dàng phù hợp với hầu hết các thiết bị được sử dụng rộng rãi như Raspberry Pi, BeagleBone Black, Arduino, các thiết bị dựa trên Android, v.v.
- Node-RED có thể chạy trong môi trường điện toán đám mây như IBM Cloud, AWS, Microsoft Azure, v.v.

## 2.3. Mongodb

### 2.3.1. Mongodb là gì ?

## Is MongoDB NoSQL



Hình 2.3.1.1. Phần mềm Mongodb

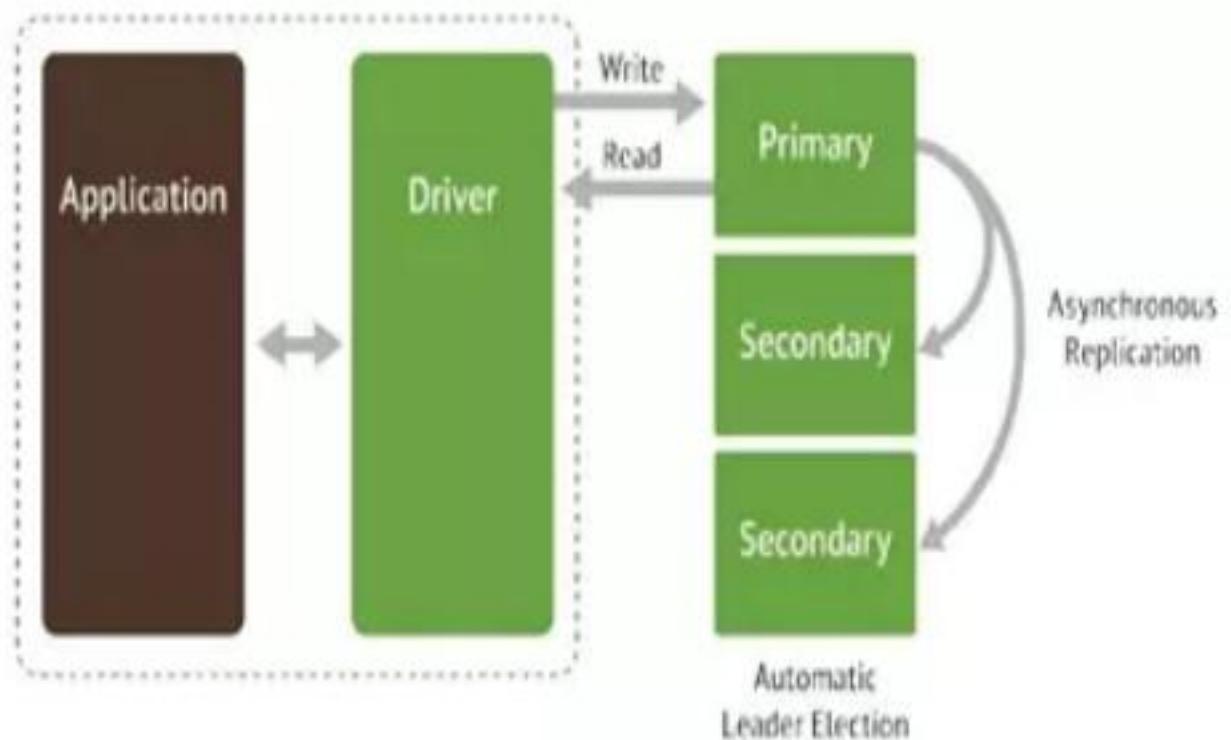
- MongoDB là một mã nguồn mở và là một tập tài liệu dùng cơ chế NoSQL để truy vấn, nó được viết bởi ngôn ngữ C++. Chính vì được viết bởi C++ nên nó có khả năng tính toán với tốc độ cao chứ không giống như các hệ quản trị CSDL hiện nay.
- Mỗi một table (bảng dữ liệu) trong SQL sử dụng thì trong MongoDB gọi là collection (tập hợp).
- Mỗi một record (bản ghi) trong MongoDB được gọi là document (tài liệu). Một bản ghi của MongoDB được lưu trữ dưới dạng document (tài liệu), nó được ghi xuống với cấu trúc field (trường) và value (giá trị). Nó giống như là một đối tượng JSON có dạng như sau:

```
{  
    name: "sue",           ← field: value  
    age: 26,              ← field: value  
    status: "A",           ← field: value  
    groups: [ "news", "sports" ] ← field: value  
}  
}
```

Hình 2.3.1.2: kiểu dữ liệu Object

### 2.3.2. Nguyên lý hoạt động

- MongoDB hoạt động dưới một tiến trình ngầm service luôn mở một cổng (Cổng mặc định là 27017) để lắng nghe các yêu cầu truy vấn, thao tác từ các ứng dụng gửi vào sau đó mới tiến hành xử lý.
- Mỗi một bản ghi của MongoDB được tự động gắn thêm một field có tên “\_id” thuộc kiểu dữ liệu ObjectId mà nó quy định để xác định được tính duy nhất của bản ghi này so với bản ghi khác, cũng như phục vụ các thao tác tìm kiếm và truy vấn thông tin về sau. Trường dữ liệu “\_id” luôn được tự động đánh index (chỉ mục) để tốc độ truy vấn thông tin đạt hiệu suất cao nhất.
- Mỗi khi có một truy vấn dữ liệu, bản ghi được cache (ghi đệm) lên bộ nhớ Ram, để phục vụ lượt truy vấn sau diễn ra nhanh hơn mà không cần phải đọc từ ổ cứng. Khi có yêu cầu thêm/sửa/xóa bản ghi, để đảm bảo hiệu suất của ứng dụng mặc định MongoDB sẽ chưa cập nhật xuống ổ cứng ngay, mà sau 60 giây MongoDB mới thực hiện ghi toàn bộ dữ liệu thay đổi từ RAM xuống ổ cứng.



Hình 2.3.2.1: Nguyên lý hoạt động của Mongod

### 2.3.3. Ưu, nhược điểm của Mongod

#### A. Ưu điểm:

- Dữ liệu lưu trữ phi cấu trúc, không có tính ràng buộc, toàn vẹn nên tính sẵn sàng cao, hiệu suất lớn và dễ dàng mở rộng lưu trữ.
- Dữ liệu được caching (ghi đệm) lên RAM, hạn chế truy cập vào ổ cứng nên tốc độ đọc và ghi cao.

#### B. Nhược điểm:

- Không ràng buộc, toàn vẹn nên không ứng dụng được cho các mô hình giao dịch yêu cầu độ chính xác cao.
- Không có cơ chế transaction (giao dịch) để phục vụ các ứng dụng ngân hàng.
- Dữ liệu được caching, lấy RAM làm trọng tâm hoạt động vì vậy khi hoạt động yêu cầu một bộ nhớ RAM lớn.

- Như đã giới thiệu ở trên, mọi thay đổi về dữ liệu mặc định đều chưa được ghi xuống ổ cứng ngay lập tức vì vậy khả năng bị mất dữ liệu từ nguyên nhân mất điện đột xuất là rất cao.

## **2.4. Phương thức truyền thông HTTP**

### **2.4.1. HTTP là gì ?**

- HTTP (HyperText Transfer Protocol - Giao thức truyền tải siêu văn bản) là một trong các giao thức chuẩn về mạng Internet, được dùng để liên hệ thông tin giữa Máy cung cấp dịch vụ (Web server) và Máy sử dụng dịch vụ (Web client), là giao thức Client/Server dùng cho World Wide Web – WWW

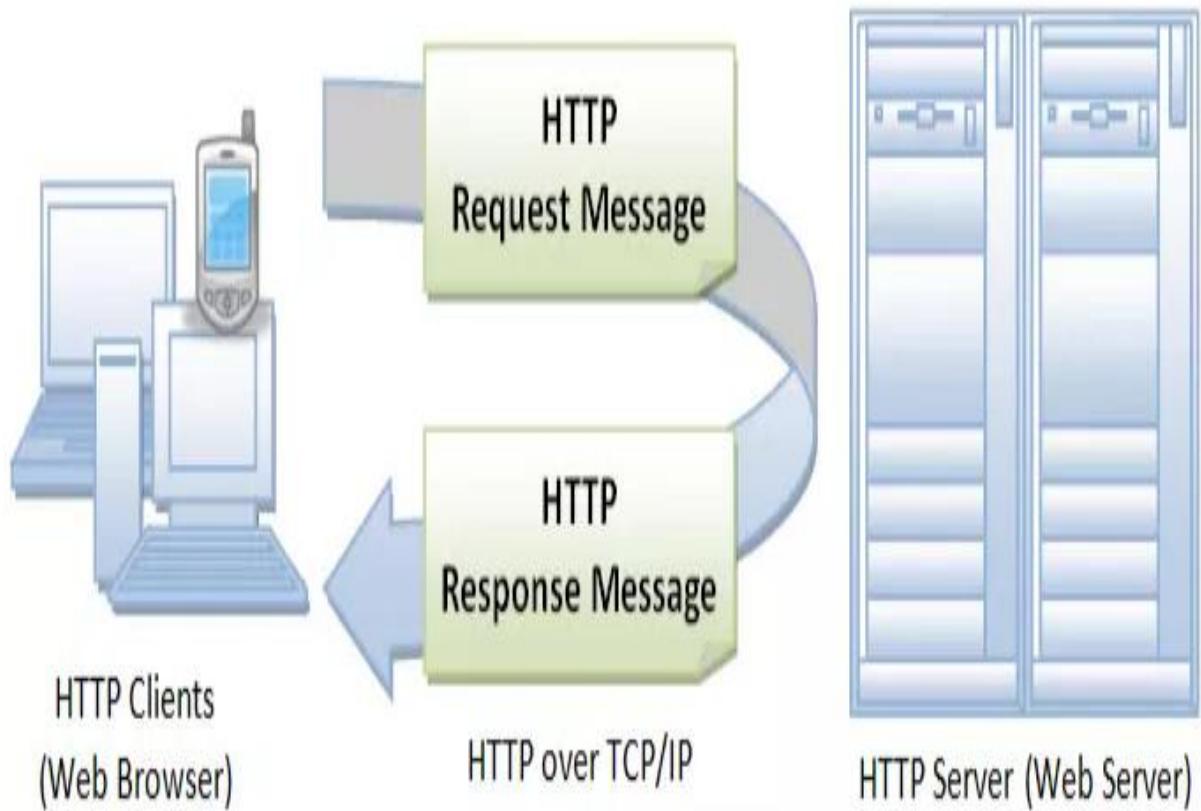
- HTTP là một giao thức ứng dụng của bộ giao thức TCP/IP (các giao thức nền tảng cho Internet).

### **2.4.2. Nguyên lý hoạt động**

- HTTP hoạt động dựa trên mô hình Client – Server. Trong mô hình này, các máy tính của người dùng sẽ đóng vai trò làm máy khách (Client). Sau một thao tác nào đó của người dùng, các máy khách sẽ gửi yêu cầu đến máy chủ (Server) và chờ đợi câu trả lời từ những máy chủ này.

- HTTP là một stateless protocol. Hay nói cách khác, request hiện tại không biết những gì đã hoàn thành trong request trước đó.

- HTTP cho phép tạo các yêu cầu gửi và nhận các kiểu dữ liệu, do đó cho phép xây dựng hệ thống độc lập với dữ liệu được truyền giao.



Hình 2.4.2.1: Sơ đồ nguyên lý hoạt động của HTTP

#### 2.4.3. Uniform Resource Locator (URL)

Một URL (Uniform Resource Locator) được sử dụng để xác định duy nhất một tài nguyên trên Web. Một URL có cấu trúc như sau:

“protocol://hostname:port/path-and-file-name”

**Trong một URL có 4 thành phần:**

- Protocol: giao thức tầng ứng dụng được sử dụng bởi client và server
- Hostname: tên DNS domain
- Port: Cổng TCP để server lắng nghe request từ client
- Path-and-file-name: Tên và vị trí của tài nguyên yêu cầu.

#### 2.4.4. Method HTTP

Method	Hoạt động	Chú thích
--------	-----------	-----------

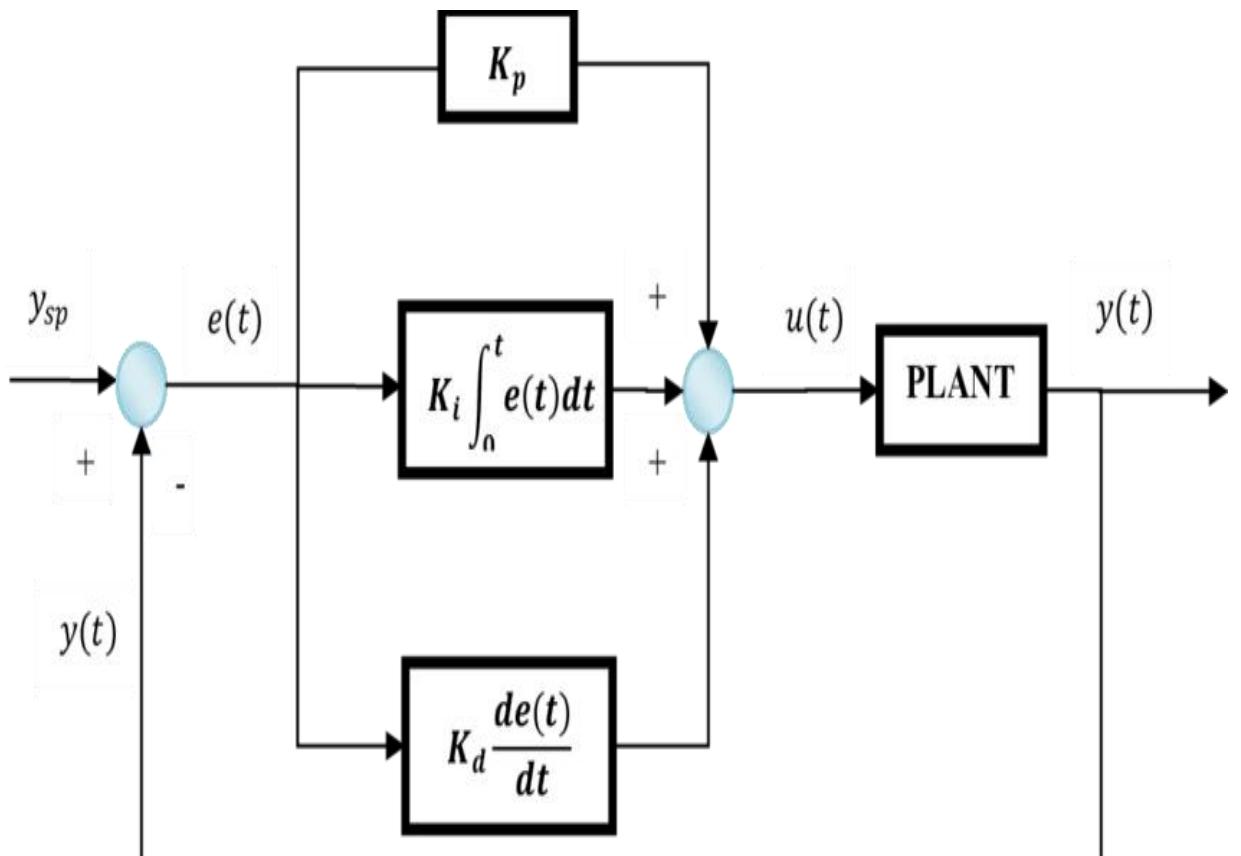
<b>GET</b>	Được sử dụng để lấy lại thông tin từ Server một tài nguyên xác định.	
<b>POST</b>	Yêu cầu máy chủ chẩn nhận thực thể được đính kèm trong request được xác định bởi URL.	Các yêu cầu sử dụng GET chỉ nhận dữ liệu và không nên có ảnh hưởng gì tới dữ liệu
<b>PUT</b>	Nếu URL đề cập đến một tài nguyên đã có, nó sẽ bị sửa đổi; nếu URL không trỏ đến một tài nguyên hiện có, thì máy chủ sẽ có thể tạo ra tài nguyên với URL đó.	
<b>DELETE</b>	Xóa bỏ tất cả các đại diện của tài nguyên được chỉ định bởi URL.	
<b>PATCH</b>	Áp dụng cho việc sửa đổi một phần của tài nguyên được xác định.	

**Bảng 2.4.4.1: Method HTTP**

## 2.5. Bộ điều khiển PID

### 2.5.1. PID là gì ?

- **PID (Proportional Integral Derivative)** là một cơ chế phản hồi vòng điều khiển được sử dụng rộng rãi trong các hệ thống điều khiển công nghiệp. Bộ điều khiển PID được sử dụng nhiều nhất trong các hệ thống điều khiển vòng kín (có tín hiệu phản hồi). Bộ điều khiển PID sẽ tính toán giá trị sai số là hiệu số giữa giá trị đo thông số biến đổi và giá trị đặt mong muốn. Bộ điều khiển sẽ thực hiện giảm tối đa sai số bằng cách điều chỉnh giá trị điều khiển đầu vào. Để đạt được kết quả tốt nhất, các thông số PID sử dụng trong tính toán phải điều chỉnh theo tính chất của hệ thống-trong khi kiểu điều khiển là giống nhau, các thông số phải phụ thuộc vào đặc thù của hệ thống.



Hình: 2.5.1.1: Sơ đồ bộ điều khiển PID

Trong đó:

- **P (Proportional):** là phương pháp điều chỉnh tỉ lệ, giúp tạo ra tín hiệu điều chỉnh tỉ lệ với sai lệch đầu vào theo thời gian lấy mẫu.

- **I (Integral):** là tích phân của sai lệch theo thời gian lấy mẫu. Điều khiển tích phân là phương pháp điều chỉnh để tạo ra các tín hiệu điều chỉnh sao cho độ sai lệch giảm về 0. Từ đó cho ta biết tổng sai số tức thời theo thời gian hay sai số tích lũy trong quá khứ. Khi thời gian càng nhỏ thì hiện tác động điều chỉnh tích phân càng mạnh, tương ứng với độ lệch càng nhỏ.

- **D (Derivative):** là vi phân của sai lệch. Điều khiển vi phân tạo ra tín hiệu điều chỉnh sao cho tỉ lệ với tốc độ thay đổi sai lệch đầu vào. Thời gian càng lớn thì phạm vi điều chỉnh vi phân càng mạnh, tương ứng với bộ điều chỉnh đáp ứng với thay đổi đầu vào càng nhanh.

\*Công thức tổng quát:

$$\begin{aligned} u(t) &= K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \\ &= K_p e(t) + \frac{1}{Ti} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \\ &= K_p (e(t) + \frac{1}{Ti} \int_0^t e(\sigma) d\sigma) + T_d \frac{de(t)}{dt} \end{aligned}$$

Trong đó:

- $u(t)$ : Đầu ra hệ thống
- $e(t)$ : Sai số giữa tín hiệu thực tế và tín hiệu đặt
- $K_p$ : Độ lợi tỉ lệ
- $K_i = \frac{1}{Ti}$  : Độ lợi tích phân
- $K_d = T_d$ : Độ lợi vi phân

Thông số	Thời gian khởi động	Quá độ	Thời gian xác lập	Sai số ổn định	Độ ổn định
Kp	Giảm	Tăng	Thay đổi nhỏ	Giảm	Giảm cấp
Ki	Giảm	Tăng	Tăng	Giảm lớn	Giảm cấp
Kd	Giảm ít	Giảm ít	Giảm ít	Không tác động về mặc lý thuyết	Cải thiện nếu Kd nhỏ

**Bảng 2.5.1.1. Tác động của việc tăng các thông số độc lập của PID**

### **2.5.2. Các loại bộ điều khiển**

- Bộ điều khiển tỉ lệ – P (Proportional Controller).
- PI ( Proportinal and Integral Controller) gọi là bộ điều khiển tỉ lệ và tích phân.
- PD (Proportional and Derivative (PD) Controller ) gọi là bộ điều khiển đạo hàm.
- PID (Proportional, Integral, and Derivative (PID) Controller) là bộ điều khiển tỉ lệ – tích phân- đạo hàm (vi phân).

### **2.6. Bộ lọc Kalman**

- Bộ lọc Kalman theo dõi trạng thái ước tính của hệ thống và phương sai hoặc độ không đảm bảo của ước tính. Ước tính được cập nhật bằng cách sử dụng mô hình chuyển đổi.
- Đối với lý thuyết thống kê và điều khiển , lọc Kalman , còn được gọi là ước lượng bậc hai tuyến tính ( LQE ), là một thuật toán sử dụng một loạt các phép đo được quan sát theo thời gian, bao gồm nhiều thông kê và các điểm không chính xác khác, đồng thời tạo ra ước tính của các biến chưa biết có xu hướng lớn hơn chính xác hơn so với những kết quả chỉ dựa trên một phép đo duy nhất, bằng cách ước tính phân phối xác suất chung trên các biến cho từng khung thời gian. Bộ lọc được đặt theo tên của Rudolf E. Kalman , một trong những người phát triển chính lý thuyết của nó.

- Bộ lọc kỹ thuật số này đôi khi được gọi là bộ lọc Stratonovich–Kalman–Bucy vì nó là trường hợp đặc biệt của bộ lọc phi tuyến, tổng quát hơn được phát triển trước đó bởi nhà toán học Liên Xô Ruslan Stratonovich. Trên thực tế, một số phương trình của bộ lọc tuyến tính trường hợp đặc biệt đã xuất hiện trong các bài báo của Stratonovich được xuất bản trước mùa hè năm 1960, khi Kalman gặp Stratonovich trong một hội nghị ở Moscow.

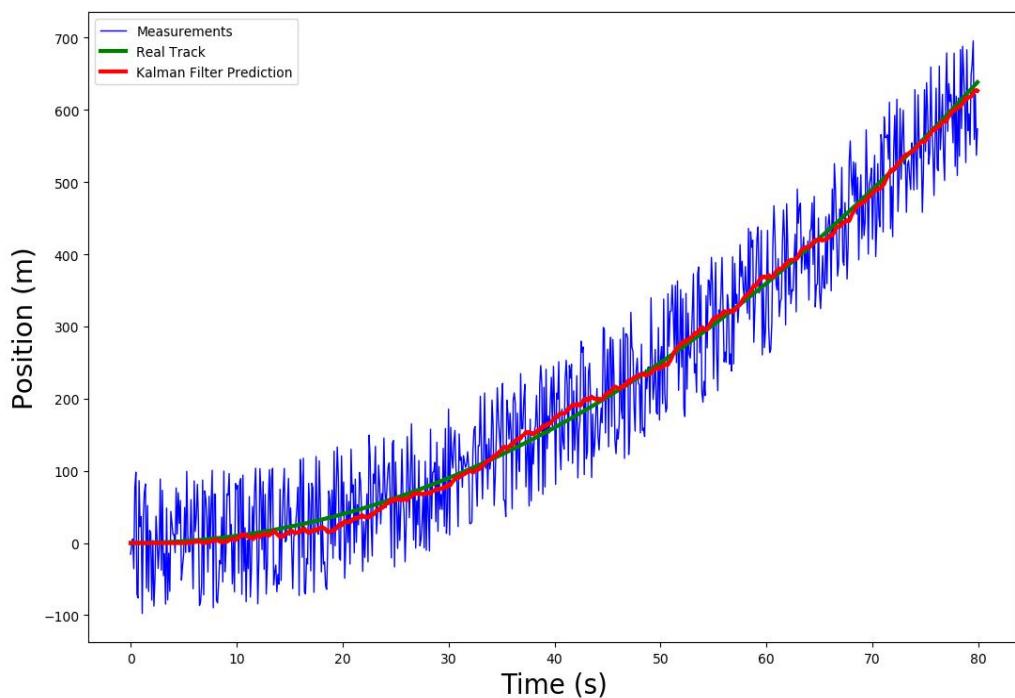
- Lọc Kalman có nhiều ứng dụng công nghệ. Một ứng dụng phổ biến là hướng dẫn, điều hướng và điều khiển các phương tiện, đặc biệt là máy bay, tàu vũ trụ và tàu được định vị động .Hơn nữa, lọc Kalman là một khái niệm được áp dụng nhiều trong phân tích chuỗi thời gian được sử dụng cho các chủ đề như xử lý tín hiệu và kinh tế lượng . Lọc Kalman cũng là một trong những chủ đề chính của lập kế hoạch và điều khiển chuyển động của robot và có thể được sử dụng để tối ưu hóa quỹ đạo .Lọc Kalman cũng hoạt động để lập mô hình kiểm soát chuyển động của hệ thống thần kinh trung ương . Do thời gian trễ giữa việc đưa ra các lệnh vận động và nhận phản hồi cảm giác , việc sử dụng bộ lọc Kalman cung cấp một mô hình thực tế để ước tính trạng thái hiện tại của hệ thống vận động và đưa ra các lệnh cập nhật.

- Thuật toán hoạt động theo quy trình hai giai đoạn. Đối với giai đoạn dự đoán, bộ lọc Kalman tạo ra các ước tính về các biến trạng thái hiện tại , cùng với độ không đảm bảo của chúng. Sau khi quan sát thấy kết quả của phép đo tiếp theo (chắc chắn bị hỏng với một số lỗi, bao gồm cả nhiễu ngẫu nhiên), các ước tính này được cập nhật bằng cách sử dụng trung bình có trọng số , với trọng số lớn hơn được trao cho các ước tính có độ chắc chắn cao hơn. Thuật toán là đệ quy . Nó có thể hoạt động trong thời gian thực , chỉ sử dụng các phép đo đầu vào hiện tại và trạng thái được tính toán trước đó và ma trận độ không đảm bảo của nó; không cần thêm thông tin quá khứ.

- Tính tối ưu của lọc Kalman giả định rằng các lỗi có phân phối chuẩn (Gaussian) . Theo lời của Rudolf E. Kálmán : "Tóm lại, các giả định sau được đưa ra về các quá

trình ngẫu nhiên: Các hiện tượng ngẫu nhiên vật lý có thể được coi là do các nguồn ngẫu nhiên sơ cấp kích thích các hệ động lực. Các nguồn sơ cấp được giả định là các quá trình ngẫu nhiên gaussian độc lập với giá trị trung bình bằng 0; các hệ thống động sẽ tuyến tính."Mặc dù không tính đến Gaussianity, nếu biết hiệp phương sai của quá trình và phép đo, bộ lọc Kalman là công cụ ước tính tuyến tính tốt nhất có thể theo nghĩa sai số trung bình bình phương tối thiểu .Đó là một quan niệm sai lầm phổ biến (tồn tại trong tài liệu) rằng bộ lọc Kalman không thể được áp dụng một cách nghiêm ngặt trừ khi tất cả các quá trình nhiễu được giả định là Gaussian.

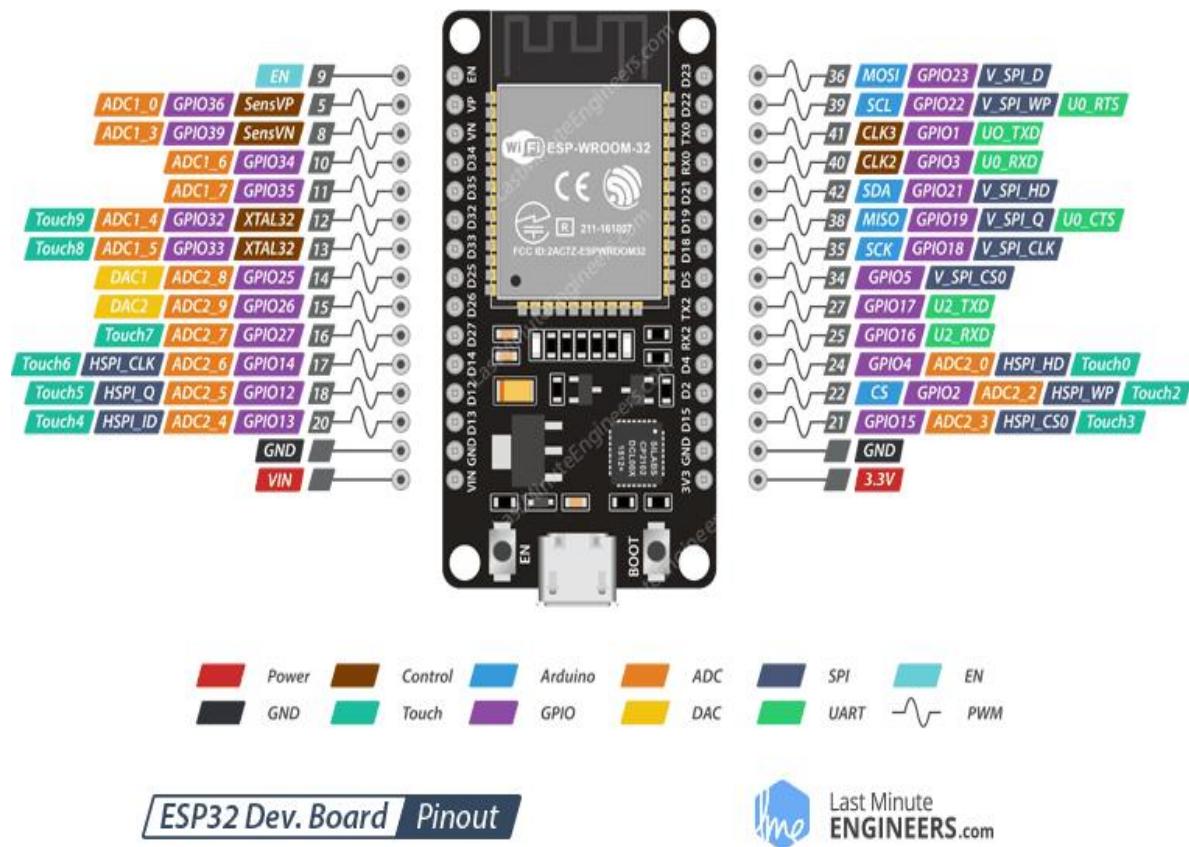
- Các phần mở rộng và khái quát hóa của phương pháp cũng đã được phát triển, chẳng hạn như bộ lọc Kalman mở rộng và bộ lọc Kalman không mìu hoạt động trên các hệ thống phi tuyến tính . Cơ sở là một mô hình Markov ẩn sao cho không gian trạng thái của các biến tiềm ẩn là liên tục và tất cả các biến tiềm ẩn và biến quan sát đều có phân bố Gaussian. Lọc Kalman đã được sử dụng thành công trong phản ứng tổng hợp đa cảm biến, và các mạng cảm biến phân tán để phát triển lọc Kalman phân tán hoặc đồng thuận .



**Hình 2.6.1:** Ví dụ về chống nhiễu của bộ lọc Kalman

## 2.7. ESP32

### 2.7.1. ESP32 là gì ?



### Hình 2.7.1.1. ESP32 Dev.Board

- ESP32 là một hệ thống vi điều khiển trên chip (SoC) giá rẻ của Espressif Systems, nhà phát triển của ESP8266 SoC. Nó là sự kế thừa của SoC ESP8266 và có cả hai biến thể lõi đơn và lõi kép của bộ vi xử lý 32-bit Xtensa LX6 của Tensilica với Wi-Fi và Bluetooth tích hợp.

- Điểm tốt về ESP32, giống như ESP8266 là các thành phần RF tích hợp của nó như bộ khuếch đại công suất, bộ khuếch đại nhận tiếng ồn thấp, công tắc ăng-ten, bộ lọc và Balun RF. Điều này làm cho việc thiết kế phần cứng xung quanh ESP32 rất dễ dàng vì bạn cần rất ít thành phần bên ngoài.

- Một điều quan trọng khác cần biết về ESP32 là nó được sản xuất bằng công nghệ 40 nm công suất cực thấp của TSMC. Vì vậy, việc thiết kế các ứng dụng hoạt động bằng pin như thiết bị đeo, thiết bị âm thanh, đồng hồ thông minh, ..., sử dụng ESP32 sẽ rất dễ dàng.

### **2.7.2. Cấu tạo**

- ESP32 có nhiều tính năng hơn ESP8266 và rất khó để đưa tất cả các thông số kỹ thuật vào bài này. Vì vậy, Điện Tử Tương Lai đã lập danh sách một số thông số kỹ thuật quan trọng của ESP32 tại đây. Nhưng để có bộ thông số kỹ thuật hoàn chỉnh, bạn nên tham khảo datasheet.

- Bộ vi xử lý LX6 32-bit lõi đơn hoặc lõi kép với xung nhịp lên đến 240 MHz.

- 520 KB SRAM, 448 KB ROM và 16 KB SRAM RTC.

- Hỗ trợ kết nối Wi-Fi 802.11 b / g / n với tốc độ lên đến 150 Mbps.

- Hỗ trợ cho cả thông số kỹ thuật Bluetooth v4.2 và BLE cổ điển.

- 34 GPIO có thể lập trình.

- 18 kênh SAR ADC 12 bit và 2 kênh DAC 8 bit

- Kết nối nối tiếp bao gồm 4 x SPI, 2 x I2C, 2 x I2S, 3 x UART.

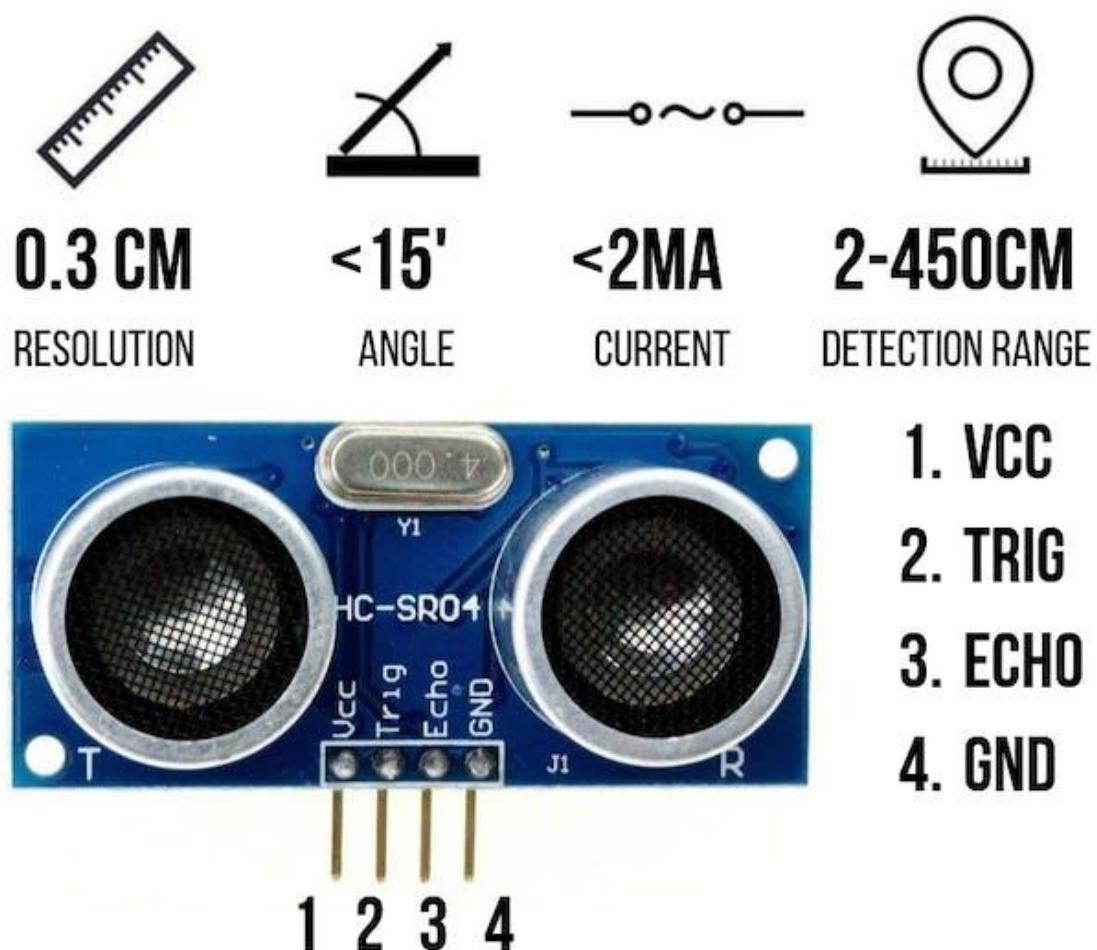
- Ethernet MAC cho giao tiếp mạng LAN vật lý (yêu cầu PHY bên ngoài).

- 1 bộ điều khiển host cho SD / SDIO / MMC và 1 bộ điều khiển slave cho SDIO / SPI.

- Động cơ PWM và 16 kênh LED PWM.
- Khởi động an toàn và mã hóa Flash.
- Tăng tốc phần cứng mật mã cho AES, Hash (SHA-2), RSA, ECC và RNG.

## 2.8. Cảm biến siêu âm HC\_SR04

### 2.8.1. HC\_SR04 là gì ?



Hình 2.8.1.1. Cảm biến siêu âm HC\_SR04

- Cảm biến siêu âm HC-SR04 thường được kết hợp với các bộ arduino, PIC, AVR,... để chạy một số ứng dụng như : phát hiện vật cản trên xe robot, đo khoảng cách vật,...

- Cảm biến siêu âm HC-SR04 là một dạng cảm biến module. Cảm biến này thường chỉ là một bản mạch, hoạt động theo nguyên lý thu phát sóng siêu âm bởi 2 chiếc loa cao tần.

- Chính vì là một cảm biến siêu âm dạng module, cho nên hầu như ứng dụng hay độ chính xác của cảm biến đều phụ thuộc vào phần code mà người sử dụng lập trình và nạp vào bản mạch điều khiển.

### **2.8.2. Cấu tạo**

- HC\_SR04 gồm 3 phần chính:

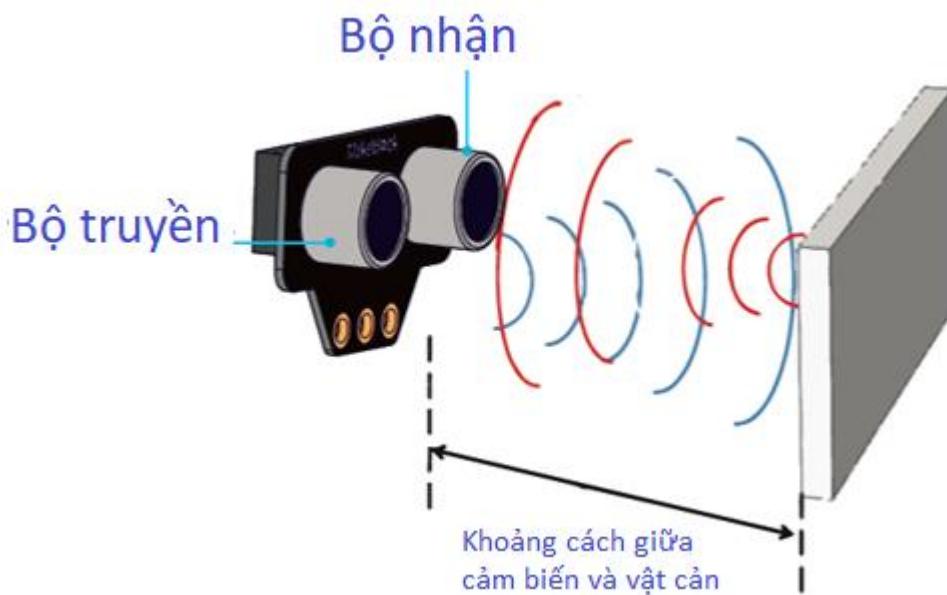
+ Bộ phận phát sóng siêu âm: Cấu tạo của các đầu phát và đầu thu siêu âm là các loa gồm đặc biệt, phát siêu âm có cường độ cao ở tần số thường là 40kHz cho nhu cầu đo khoảng cách.

+ Bộ phận thu sóng siêu âm phản xạ: Thiết bị thu là dạng loa gồm có cấu tạo chỉ nhạy với một tần số chẵng hạn như 40KHz. Qua một loạt các linh kiện như OPAM TL072, transistor NPN... Tín hiệu này liên tục được khuếch đại biên độ và cuối cùng là đưa qua một bộ so sánh, kết hợp với tín hiệu từ bộ điều khiển để đưa về bộ điều khiển.

+ Bộ phận xử lý, điều khiển tín hiệu: Vi điều khiển (PIC16F688, STC11,...) được sử dụng làm nhiệm vụ phát xung, xử lý tính toán thời gian từ khi phát đến khi thu được sóng siêu âm nếu nhận được tín hiệu TRIG.

### 2.8.3. Nguyên lý hoạt động

- Khi một chùm siêu âm được phát đi gặp một vật thì sẽ có hiện tượng phản hồi âm, tần số của chùm siêu âm phản hồi về sẽ thay đổi so với tần số của chùm phát đi nếu khoảng cách tương đối giữa nguồn phát và vật thay đổi: Tần số tăng nếu khoảng cách giảm và ngược lại.

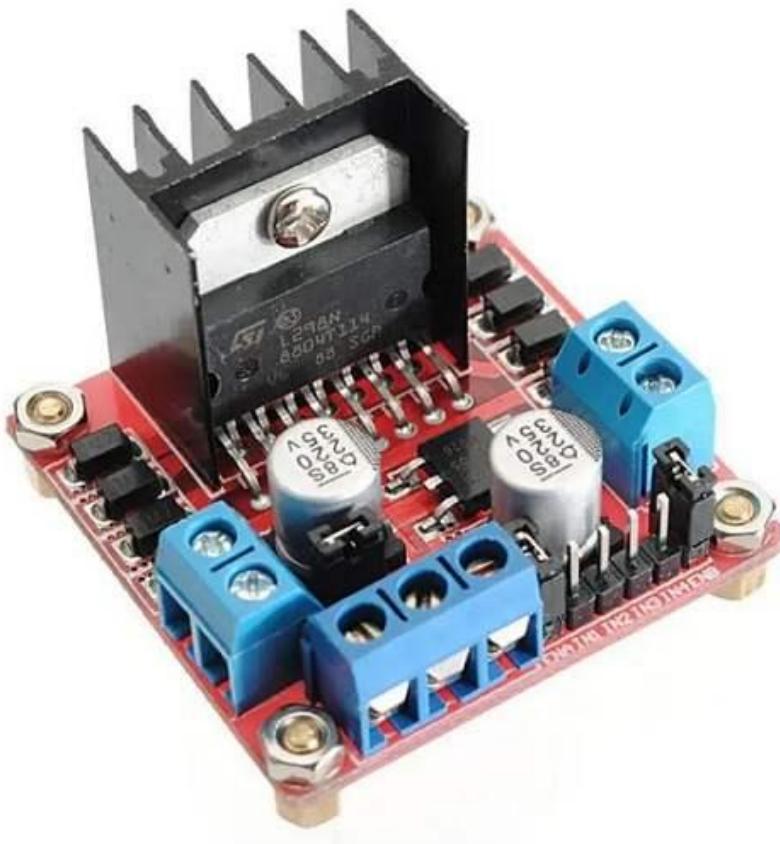


Hình 2.8.3.1. Nguyên lý hoạt động của cảm biến siêu âm

## 2.9. Module L298N

### 2.9.1. Module L298N là gì ?

- L298N là module điều khiển động cơ trong các xe DC và động cơ bước. Module có một IC điều khiển động cơ L298 và một bộ điều chỉnh điện áp 5V 78M05.
- Module L298N có thể điều khiển tối đa 4 động cơ DC hoặc 2 động cơ DC với khả năng điều khiển hướng và tốc độ.



Hình 2.9.1.1. Module L298N

## 2.9.2. Cấu tạo

*L298 gồm các chân:*

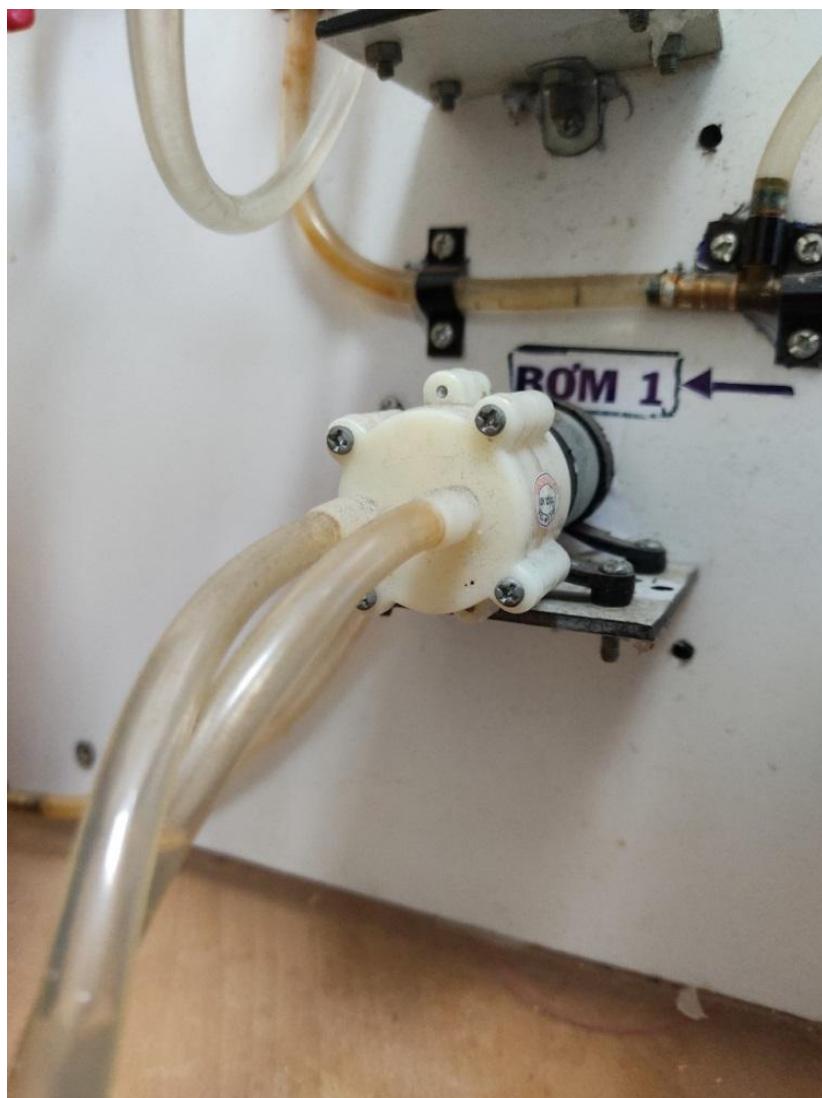
- + 12V power, 5V power. Đây là 2 chân cấp nguồn trực tiếp đến động cơ.
- + Bạn có thể cấp nguồn 9-12V ở 12V.
- + Bên cạnh đó có jumper 5V, nếu bạn để như hình ở trên thì sẽ có nguồn 5V ra ở cổng 5V power, ngược lại thì không. Bạn để như hình thì ta chỉ cần cấp nguồn 12V vô ở 12V power là có 5V ở 5V power.

- + Power GND chân này là GND của nguồn cấp cho Động cơ. Nếu chơi Arduino thì nhớ nối với GND của Arduino.
- + 2 Jump A enable và B enable.
- + Gồm có 4 chân Input. IN1, IN2, IN3, IN4.
- + Output A: nối với động cơ A.
- + Output B: nối với động cơ B.

### **2.9.3. Nguyên lý hoạt động**

- Chân IN1, IN2 ứng với ngõ ra OUT A.
  - Chân ENA ứng với ngõ ra OUT A.
  - Chân IN3, IN4 ứng với ngõ ra OUT B.
  - Chân ENB ứng với ngõ ra OUT B.
- Khi:
- + IN1 = high, IN2 = low thì OUT A = V+
  - + IN1 = low, IN2 = high thì OUT A = V-
  - + IN1 = low, IN2 = low thì OUT A = 0
  - + IN1 = high, IN2 = high thì OUT A = 0

### **2.10. Động Cơ Bơm Nước R385 Water Pump 12VDC**



Hình 2.10.1. Bơm nước R385 12VDC

- Bơm R385 có nhiệm vụ là cấp và hồi của chất lỏng trong bồn.

**Thông số kỹ thuật:**

- Kích cỡ là 90 x 50 x 38mm.
- Công suất là 5 đến 12W.
- Lưu lượng tầm 1 đến 2 lít trên phút.
- Dòng tiêu thụ ở mức 0.6 đến 2A.
- Điện áp làm việc là 6 đến 12VDC.

## **2.11. Nguồn điện tổ ong**

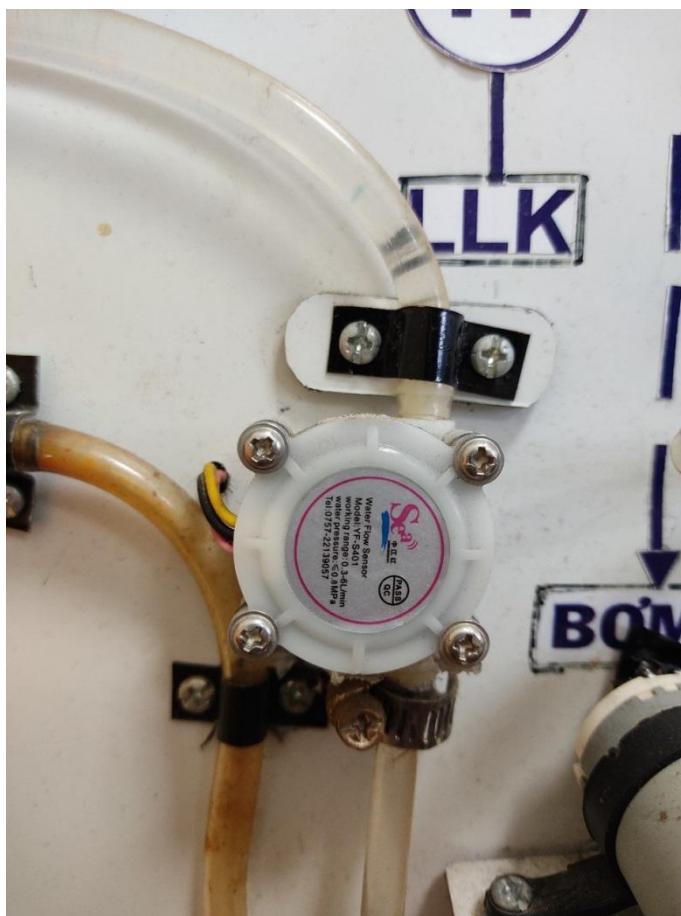


**Hình 2.11.1. Nguồn tổ ong 12V\_5A**

- Nguồn tổ ong 12V\_5A có nhiệm vụ là nguồn nuôi của các máy bơm, ESP32, flowmeter, distance sensor,...
- Nguồn tổ ong có nhiệm vụ là chuyển dòng điện xoay chiều 220V thành dòng điện 1 chiều 12V\_5A.

## **2.12. Cảm biến lưu lượng YF-S401**

- Cảm biến YF-S401 có công dụng đo lượng chất lỏng thoát trong quá trình thoát nước do chênh lệch áp suất giữa hai bồn.



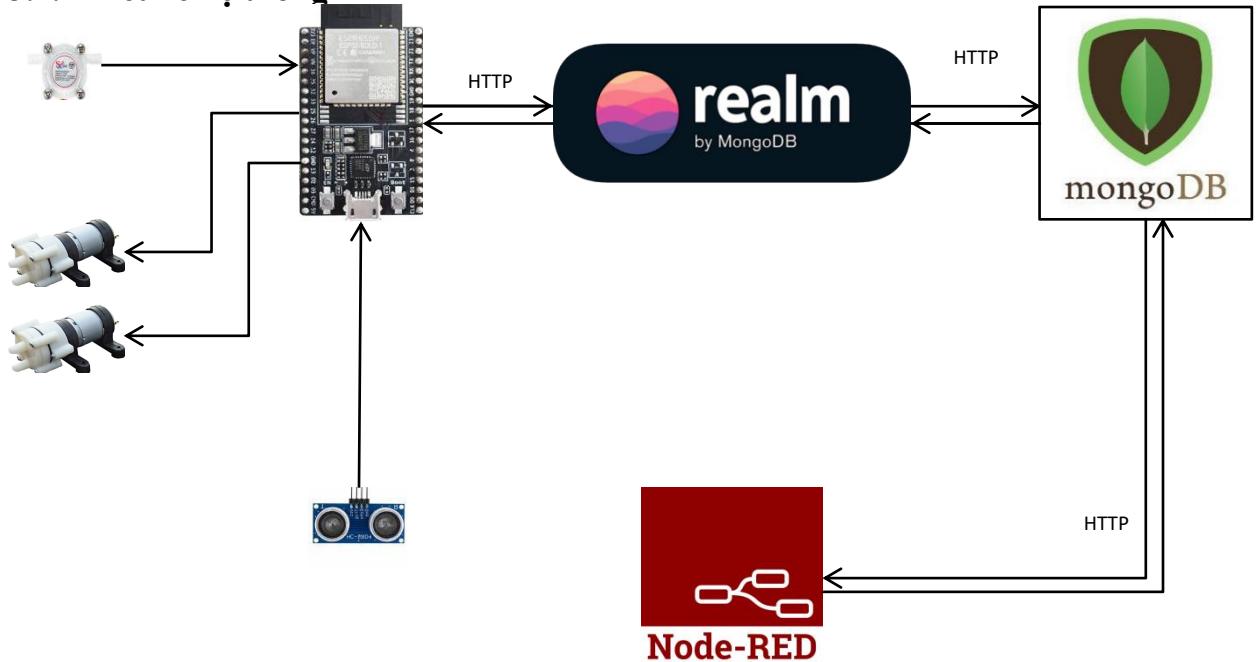
Hình 2.12.1. Cảm biến lưu lượng YF-S401

**Thông số kỹ thuật:**

- Nhiệt độ làm việc nên bé hơn 70 độ C.
- Lưu lượng là 0.5 đến 7 lít trên phút.
- Áp suất tối đa 1.8Mpa.
- Tín hiệu ngõ ra là dạng xung sử dụng hiệu ứng Hall.
- Điện áp làm việc ở 4 đến 12VDC.
- Dòng điện tiêu thụ ở mức 20mA.

## CHƯƠNG 3. THIẾT KẾ BỘ ĐIỀU KHIỂN

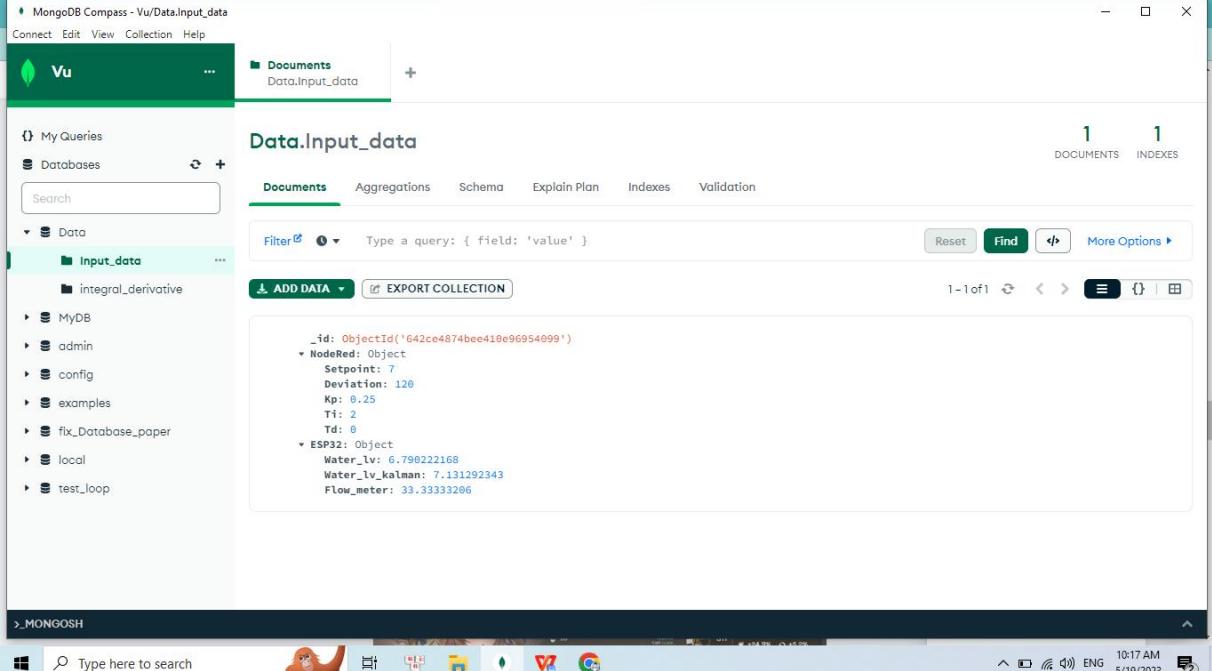
### 3.1. Thiết kế hệ thống



Hình 3.1.1. Sơ đồ hoạt động của hệ thống

### 3.2. Thiết kế MongoDB

- Thiết kế MongoDB là việc thiết kế một Database, nơi mà chúng ta sẽ lưu trữ dữ liệu được truyền lên từ ESP32 và Node-RED, đây cũng là cơ sở dữ liệu chính dùng để thực hiện các thuật toán PID và Kalman filter.

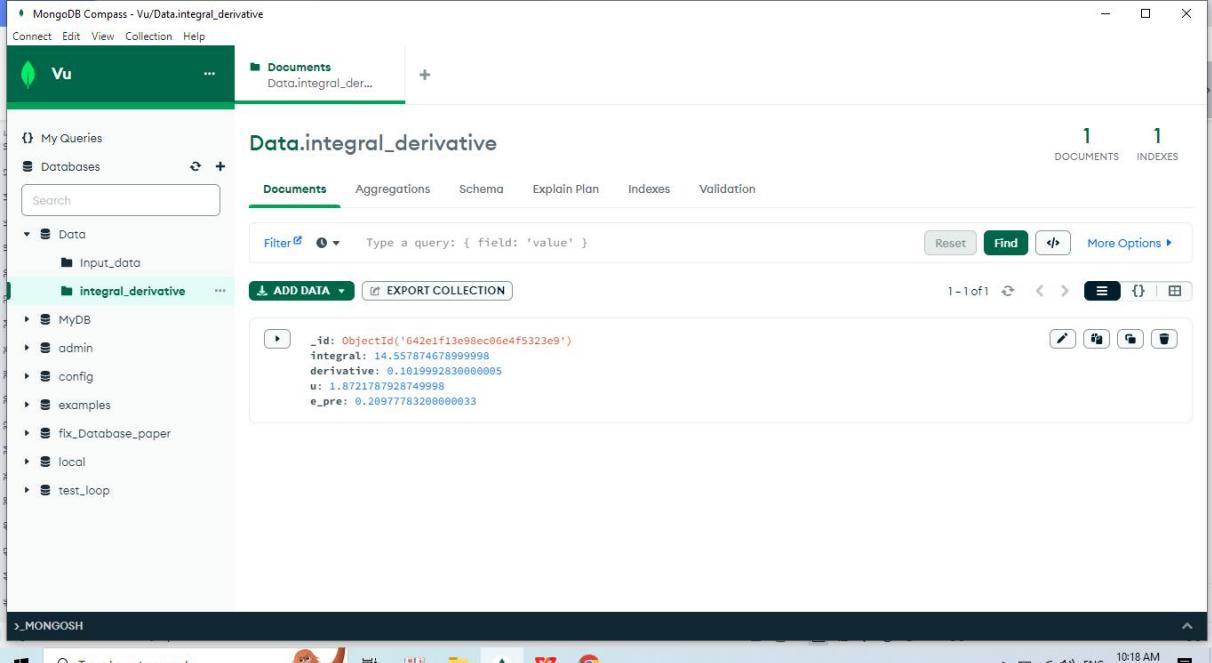


```

{
  "_id": ObjectId("642ce4874bee410e96954099"),
  "NodeRed": Object {
    "Setpoint": 7,
    "Deviation": 120,
    "Kp": 0.25,
    "Ti": 2,
    "Td": 0
  },
  "ESP32": Object {
    "Water_lv": 6.790222168,
    "Water_lv_kalman": 7.131292343,
    "Flow_meter": 33.33333206
  }
}

```

Hình 3.2.1. Dữ liệu input được lưu trên MongoDB



```

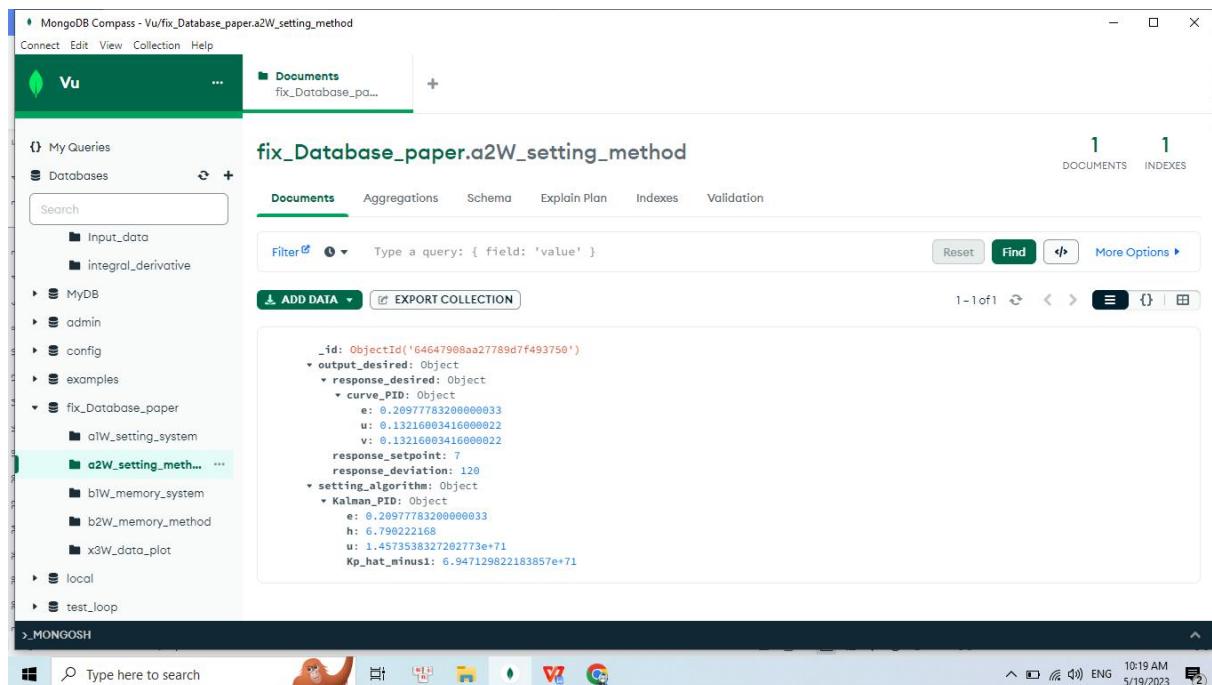
{
  "_id": ObjectId("642e1f13e98ec06e4f5323e9"),
  "integral": 14.557874678999998,
  "derivative": 0.10199928300000005,
  "u": 1.8721787928749998,
  "e_pre": 0.20977783200000033
}

```

Hình 3.2.2. Dữ liệu để thực hiện thuật toán PID

# ĐỒ ÁN TỔNG HỢP

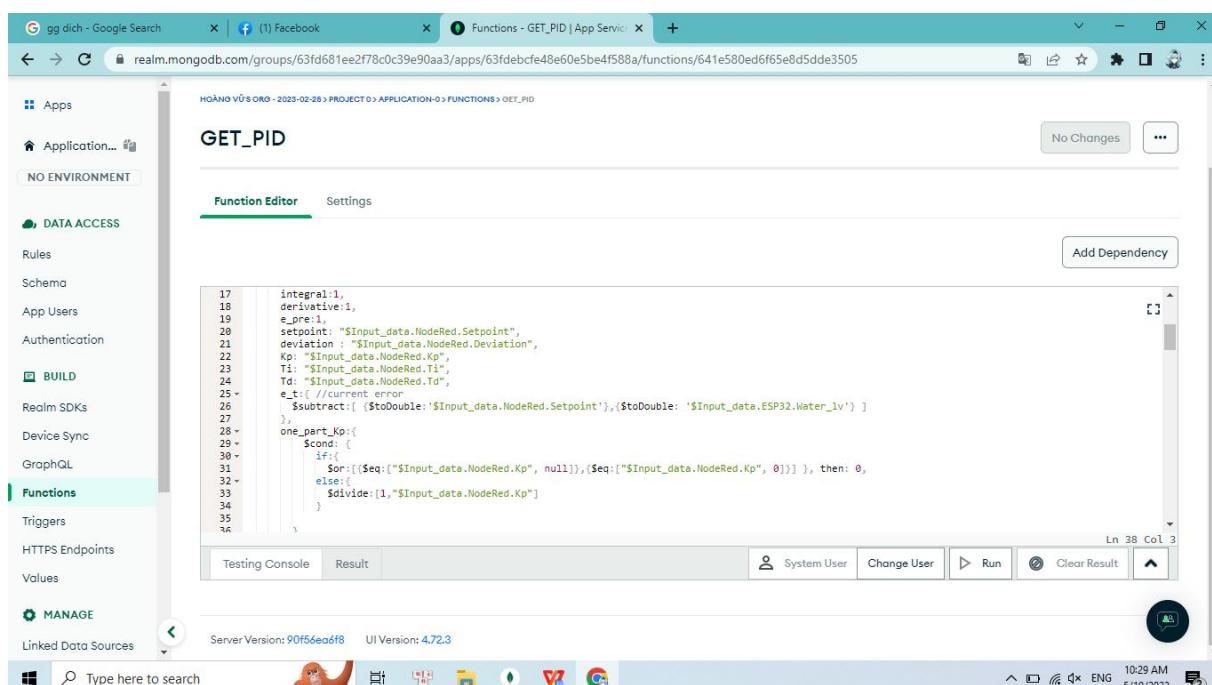
## Trang - 33



Hình 3.2.3. Dữ liệu để thực hiện thuật toán Kalman filter

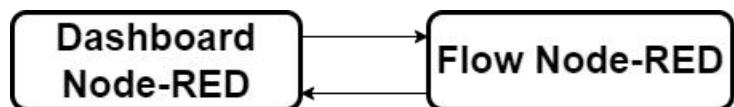
### 3.3. Thiết kế MongoDB Realm

- MongoDB Realm là nơi để chúng ta thực hiện lập trình các giải thuật từ các dữ liệu có trong Database ở trên và những thuật đó đều được thực hiện online bằng internet.



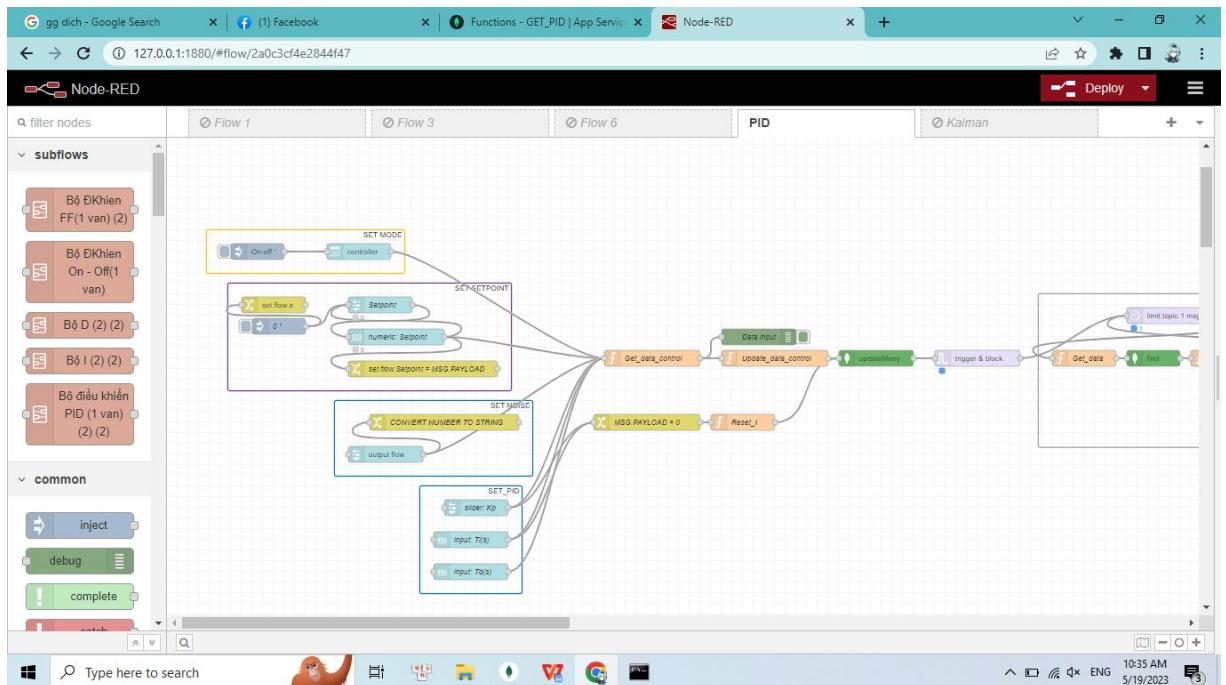
Hình 3.3.1. Minh họat lập trình trên MongoDB Realm

### 3.4. Thiết kế Node-RED



- Node-RED bao gồm 2 phần chính là Flow và Dashboard

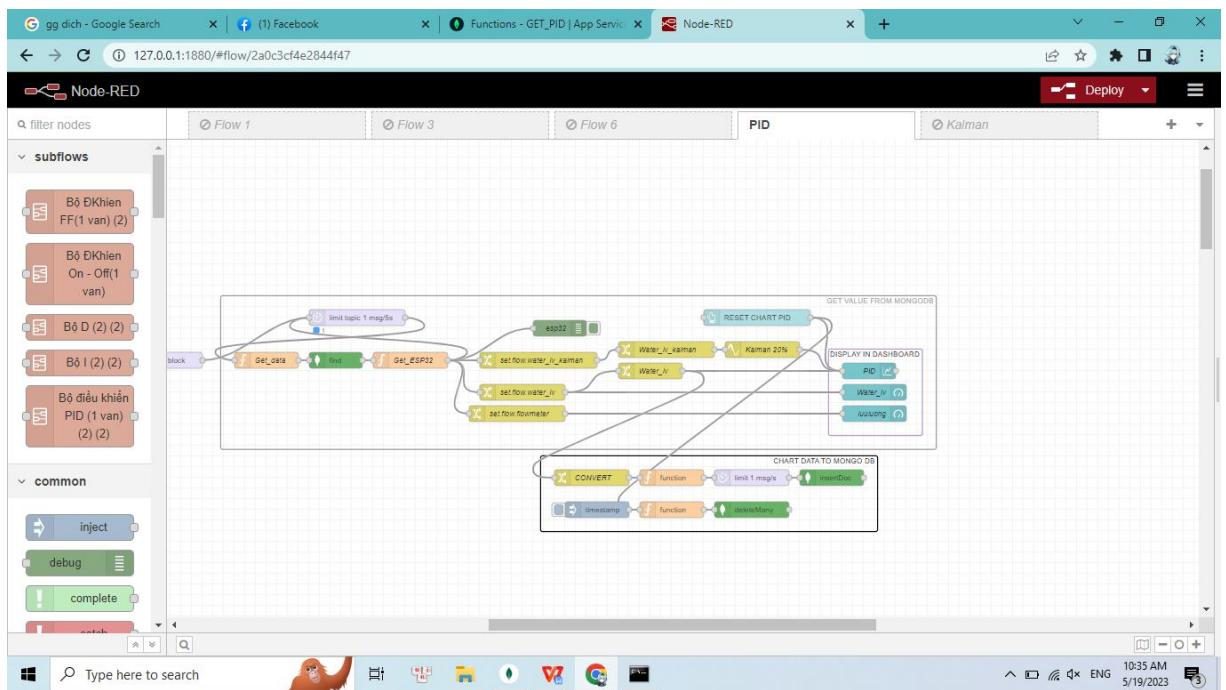
+ Flow: là nơi chứa các khối lệnh dùng để truyền và nhận dữ liệu từ MongoDB



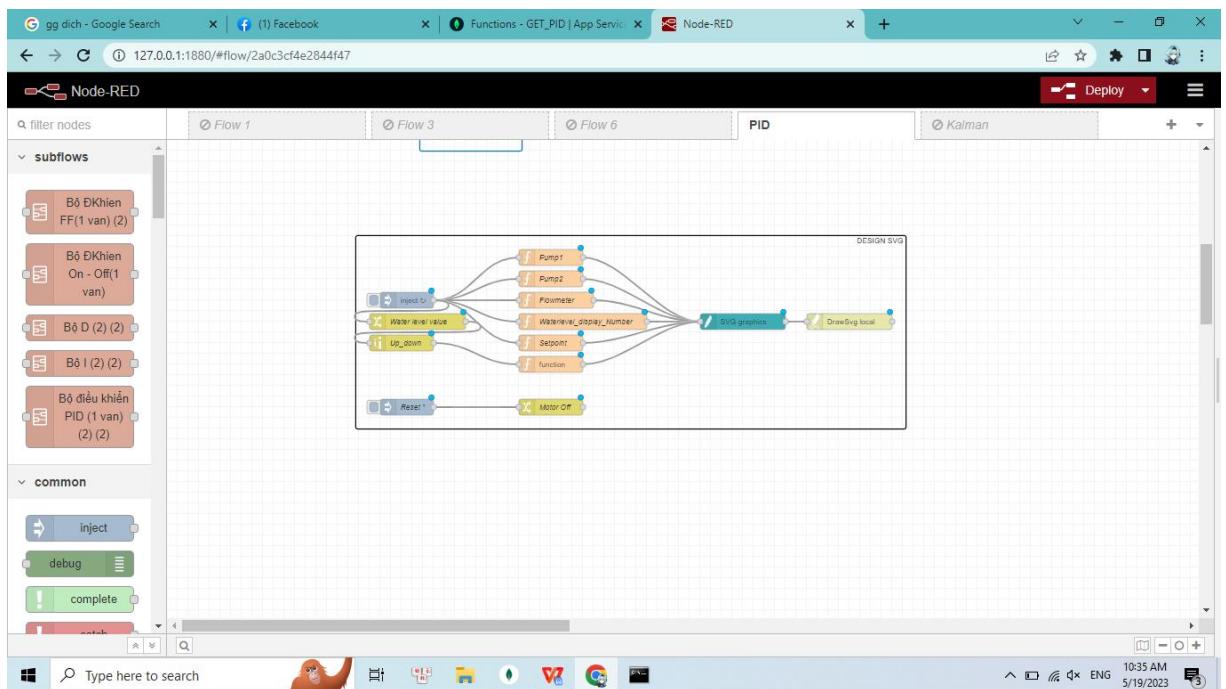
Hình 3.4.1. Các Flow của Node-RED

# ĐỒ ÁN TỔNG HỢP

## Trang - 35

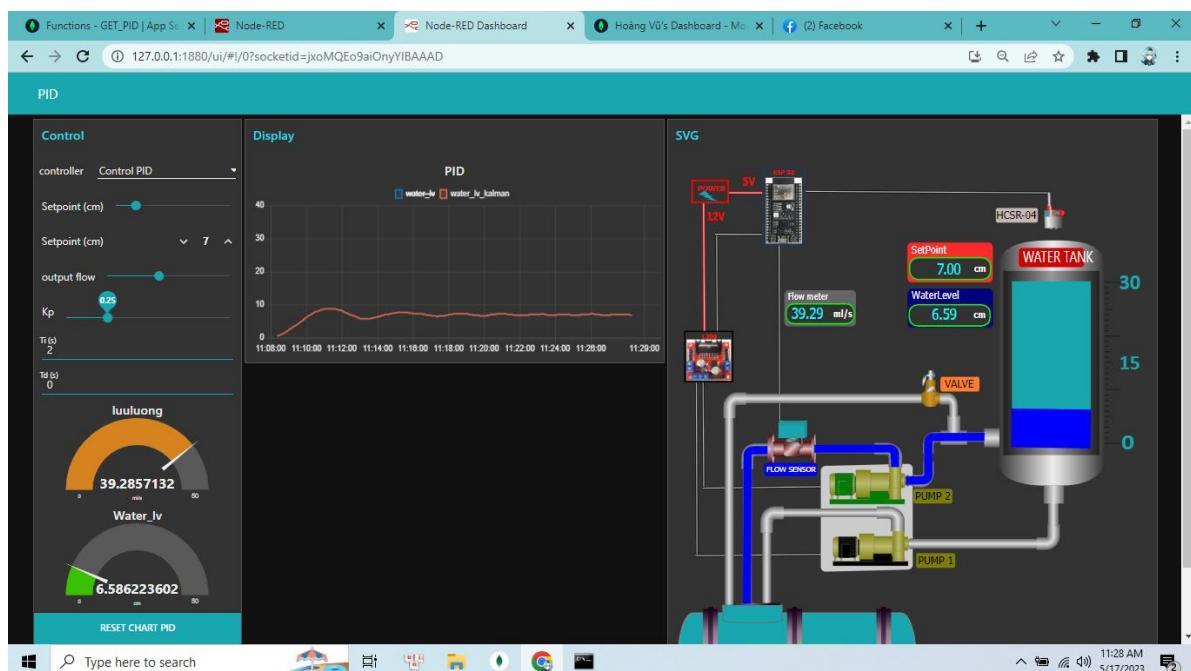


Hình 3.4.2. Các Flow của Node-RED



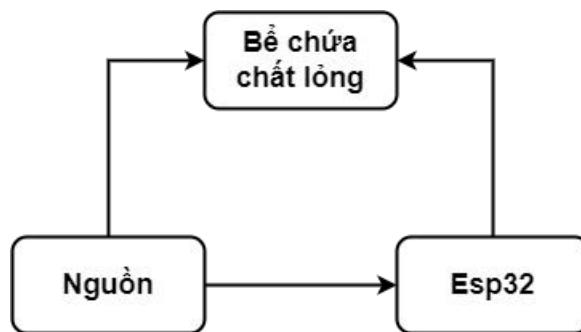
Hình 3.4.3. Các Flow của Node-RED

+ Dashboard: là giao diện điều khiển của phần mềm Node-RED, người dùng có thể sử dụng dễ dàng vì giao diện được thiết kế đơn giản, giúp người dùng dễ tiếp cận. Người dùng thiết lập các thông số ( chế độ điều khiển, mực chất lỏng mong muốn, thông số của bộ điều khiển P, thông số bộ điều khiển I – Ti, thông số bộ điều khiển D – Td, công suất máy bơm để xả ra ) trên hệ thống bể chứa chất lỏng sẽ được phần mềm Node-RED xử lý, tính toán đưa ra giá trị, thông số phù hợp rồi gửi về hệ thống sau đó máy bơm khởi động đến khi đạt được mực chất lỏng ta mong muốn. Giá trị mực chất lỏng được biểu thị dưới dạng biểu đồ để người sử dụng dễ theo dõi, giám sát và được gửi qua web MongoDB để lưu trữ và giám sát từ xa.



**Hình 3.4.4. Giao diện Dashboard của Node-RED**

### 3.5. Thiết kế bồn chứa chất lỏng

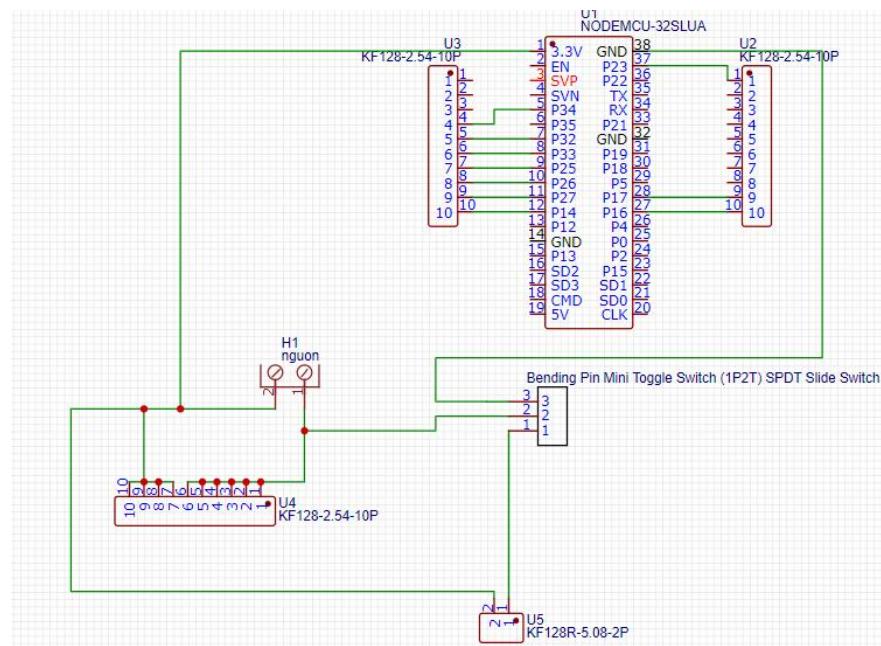


- **ESP32:** có nhiệm vụ đọc và xử lý sơ bộ tín hiệu nhận được từ cảm biến khoảng cách HC\_SR04 và cảm biến lưu lượng YF-S401, đồng thời xuất tín hiệu điều khiển cho các cơ cấu chấp hành. Ngoài ra ESP32 còn có nhiệm vụ là truyền dữ liệu cho MongoDB để lưu trữ chúng trên Database của MongoDB.
- **Nguồn:** Là thứ sẽ cung cấp điện áp và dòng điện để nuôi ESP32, các sensor và các thiết bị chấp hành.
- **Bể chứa chất lỏng:** gồm có bể chứa chất lỏng, cảm biến siêu âm HC-SR04, cảm biến lưu lượng YF-S401, Esp32, máy bơm 12V, module L298, ống nhựa dẫn chất lỏng.

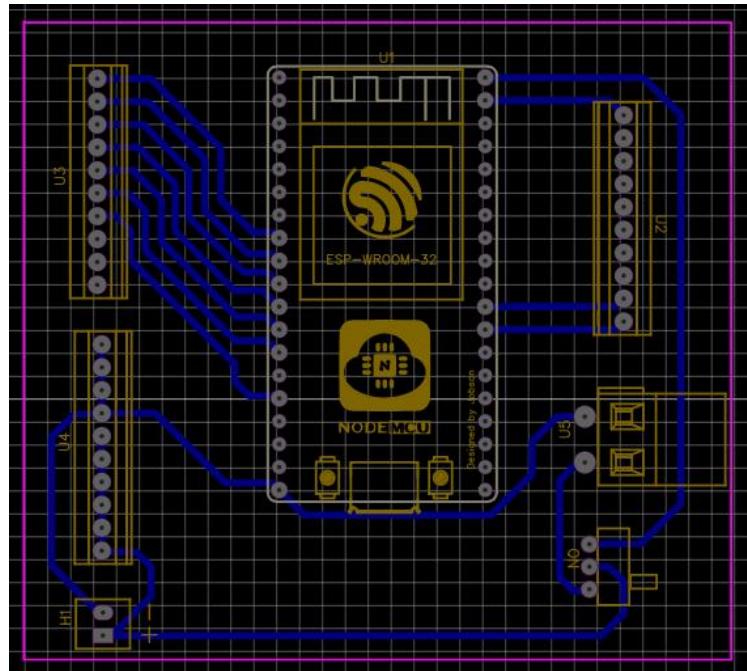


Hình 3.5.1. Mô hình bồn chứa chất lỏng

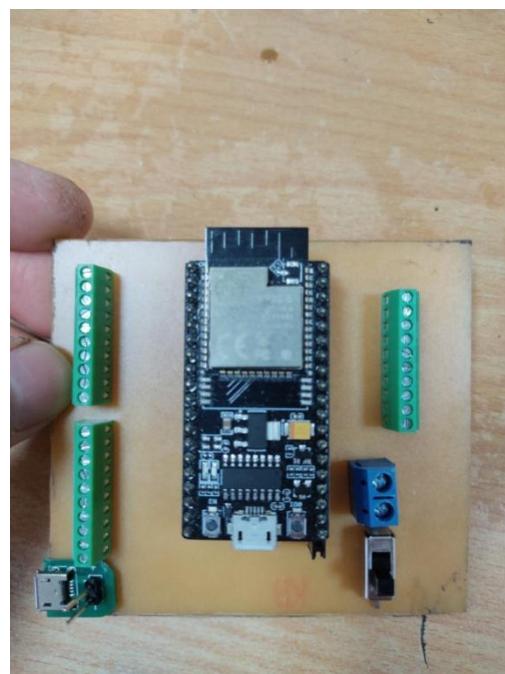
### **3.6.Bản vẽ sơ đồ đấu dây**



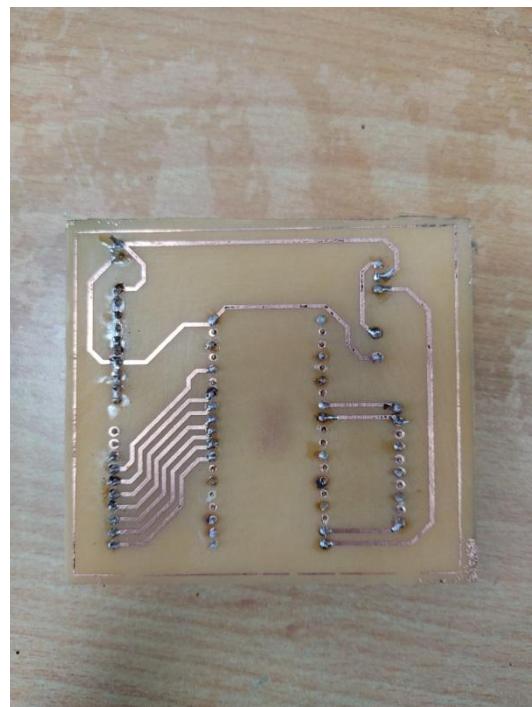
Hình 3.6.1. Sơ đồ đấu nối dây



Hình 3.6.2. Bản vẽ mạch in PCB



**Hình 3.6.3. Mặt trên của board**

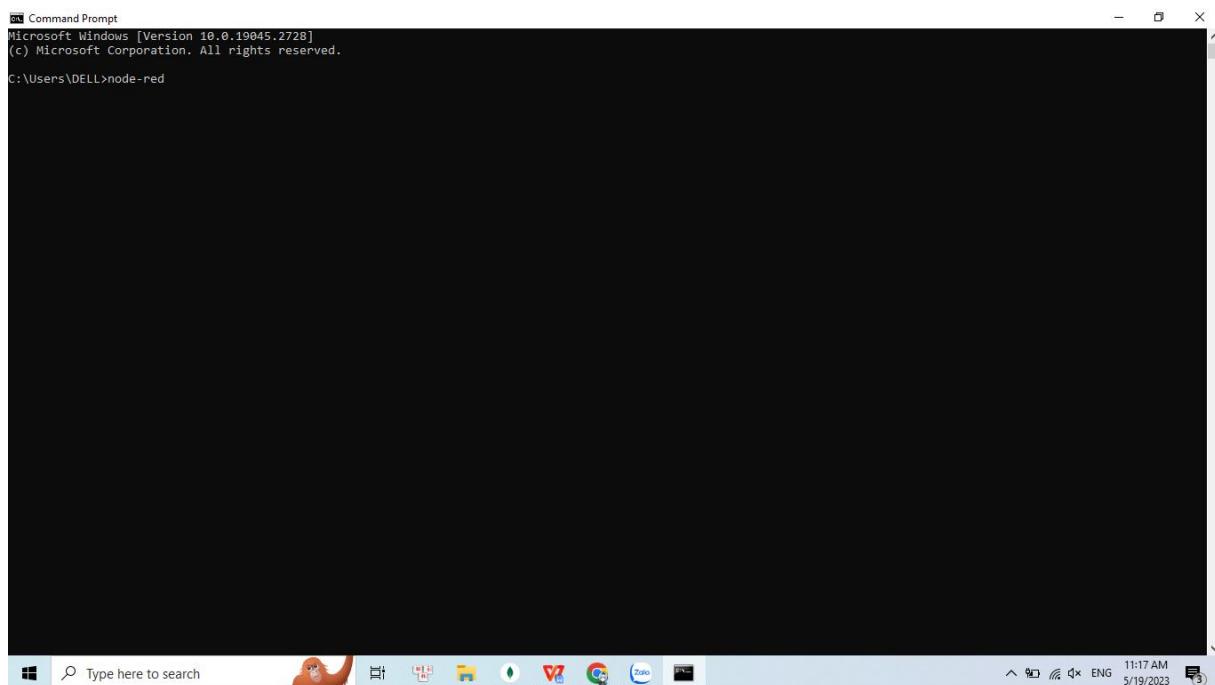


**Hình 3.6.4. Mặt dưới của board**

## **CHƯƠNG 4: LUU ĐO GIẢI THUẬT VÀ ĐIỀU KHIỂN**

### **4.1. Hoạt động của hệ thống**

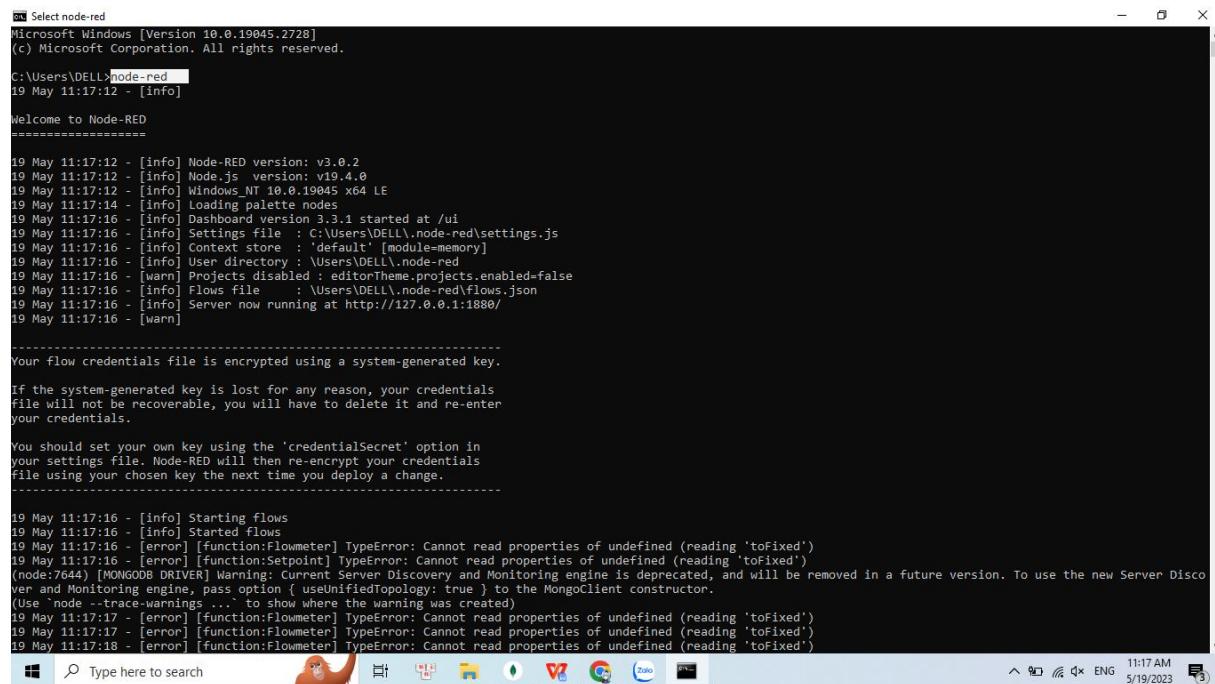
- B1: chúng ta cấp nguồn cho hệ thống thông qua bộ nguồn tổng để chuyển đổi Điện áp 220VAC thành 12VDC.
- B2: chúng ta mở chương trình điều khiển và giám sát Node-RED bằng cách mở “Command Prompt” tại search có biểu tượng trên máy tính và nhập “node-red” vào.



**Hình 4.1.1. Thao tác mở Node-RED**

# ĐỒ ÁN TỔNG HỢP

## Trang - 42



```
Select node-red
Microsoft Windows [Version 10.0.19045.2728]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL\Node-RED
19 May 11:17:12 - [info]

Welcome to Node-RED
=====
19 May 11:17:12 - [info] Node-RED version: v3.0.2
19 May 11:17:12 - [info] Node.js version: v19.4.0
19 May 11:17:12 - [info] Windows_NT 10.0.19045 x64 LE
19 May 11:17:14 - [info] Loading palette nodes
19 May 11:17:16 - [info] Dashboard version 3.3.1 started at /ui
19 May 11:17:16 - [info] Settings file : C:\Users\DELL\.node-red\settings.json
19 May 11:17:16 - [info] Context store : 'default' [module=memory]
19 May 11:17:16 - [info] User directory : C:\Users\DELL\.node-red
19 May 11:17:16 - [warn] Projects disabled : editorTheme.projects.enabled=false
19 May 11:17:16 - [info] Flows file : C:\Users\DELL\.node-red\flows.json
19 May 11:17:16 - [info] Server now running at http://127.0.0.1:1880/
19 May 11:17:16 - [warn]

Your flow credentials file is encrypted using a system-generated key.

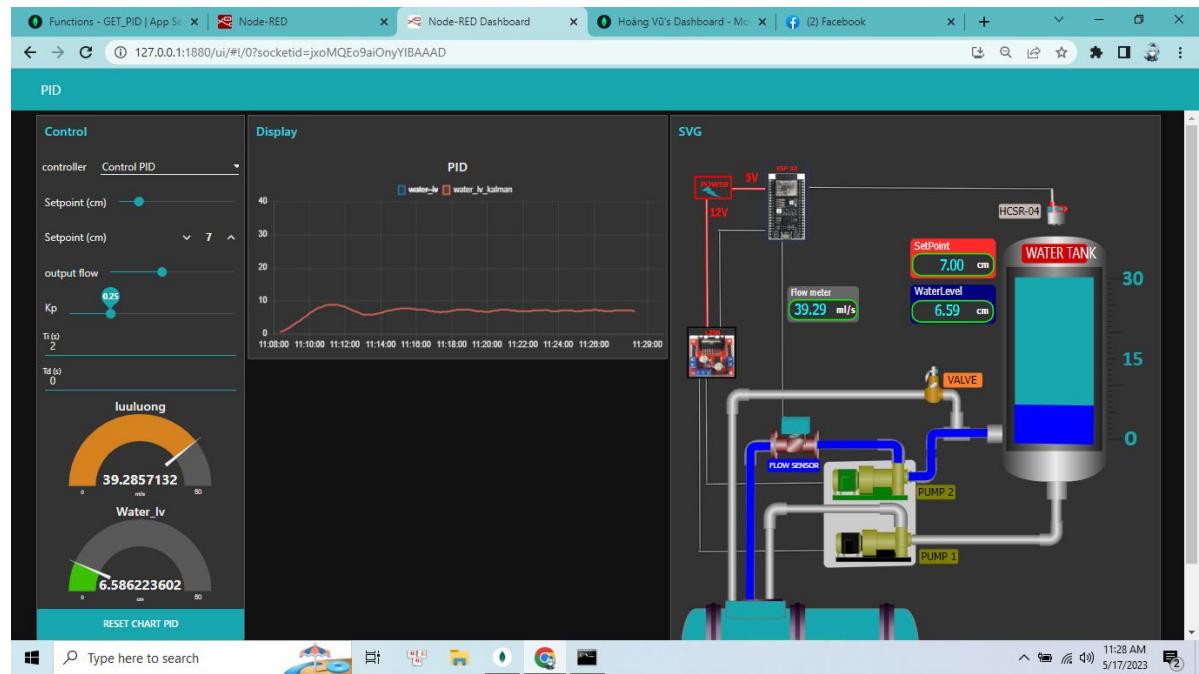
If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credential
file using your chosen key the next time you deploy a change.
=====

19 May 11:17:16 - [info] Starting flows
19 May 11:17:16 - [info] Started flows
19 May 11:17:16 - [error] [function:Flowmeter] TypeError: Cannot read properties of undefined (reading 'toFixed')
19 May 11:17:16 - [error] [function:Setpoint] TypeError: Cannot read properties of undefined (reading 'toFixed')
(node:7644) [MONGODB DRIVER] Warning: Current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discoverer and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
(Use `node --trace-warnings ...` to show where the warning was created)
19 May 11:17:17 - [error] [function:Flowmeter] TypeError: Cannot read properties of undefined (reading 'toFixed')
19 May 11:17:17 - [error] [function:Flowmeter] TypeError: Cannot read properties of undefined (reading 'toFixed')
19 May 11:17:18 - [error] [function:Flowmeter] TypeError: Cannot read properties of undefined (reading 'toFixed')
```

Hình 4.1.2. Thao tác mở Node-RED

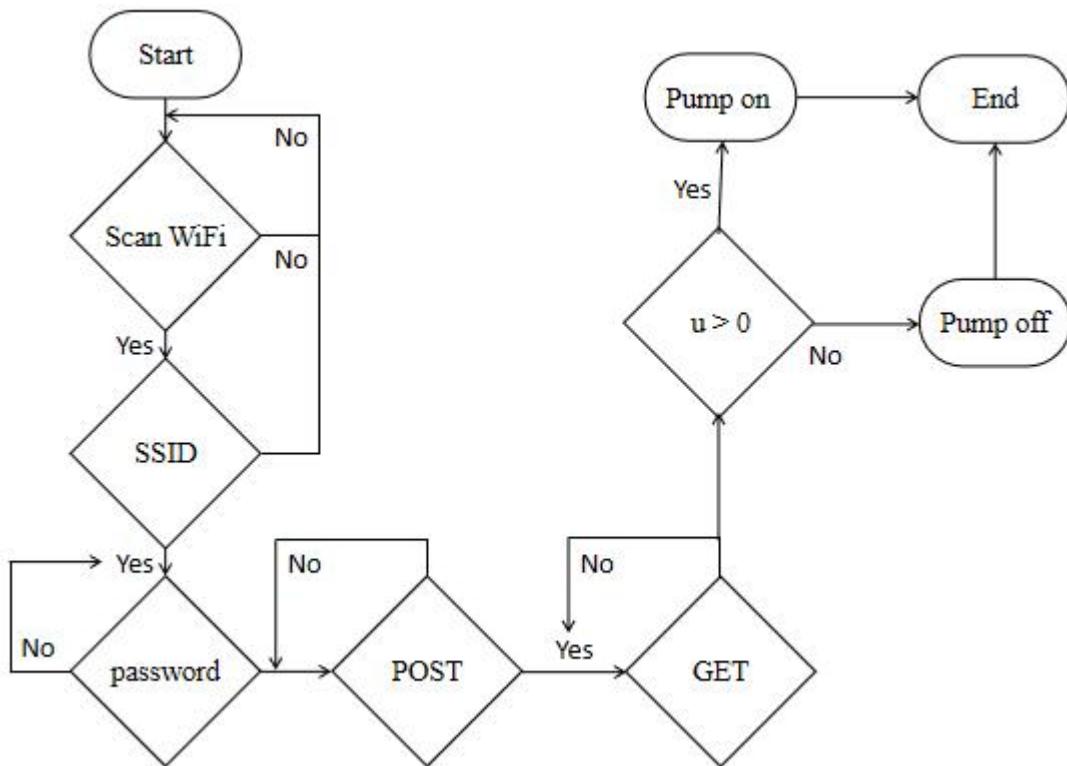
- B3: Sao khi cmd chạy xong sẽ hiện đường dẫn “<http://127.0.0.1:1880/>”, chúng ta sẽ mở Google Chrome và nhập đường dẫn này vào và thêm “/ui” ở cuối đường dẫn để mở Dashboard của Node-RED.



Hình 4.1.3. Giao diện dashboard của Node-RED

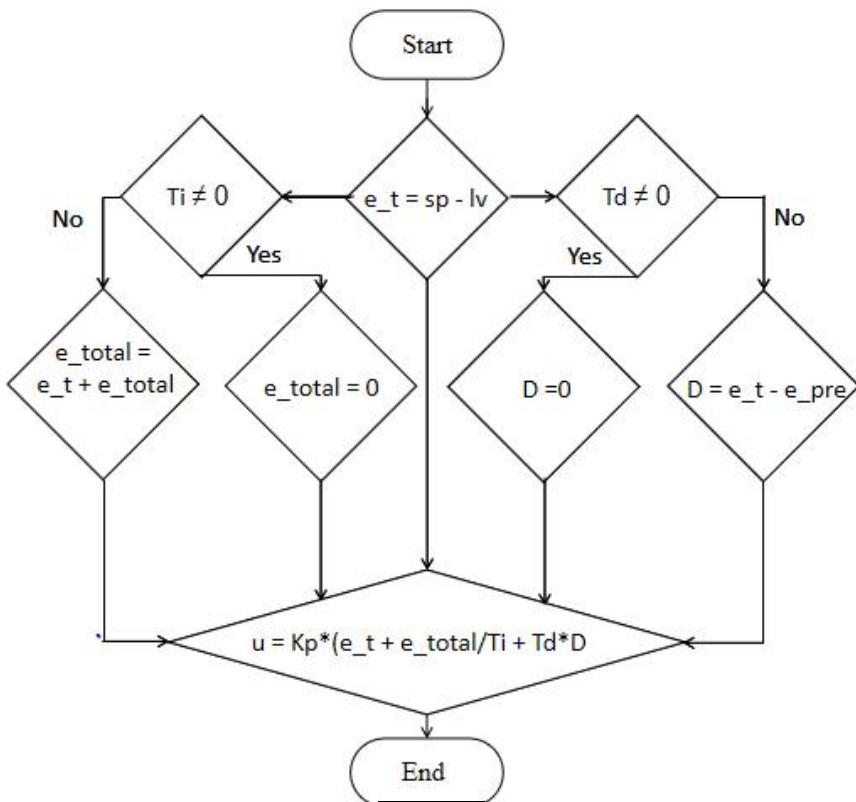
- B4: tại đây chúng ta có thể điều chỉnh các thông số của hệ thống (setpoint, output, Kp, Ti, Td).

#### 4.2. Lưu đồ giải thuật của ESP32



- Khi hệ thống được cấp nguồn và chương trình chạy, ESP32 dò quét tên WiFi đã được thiết lập. Nếu chưa tìm thấy thì sẽ bắt đầu dò quét đến khi tìm thấy. Khi tìm thấy được tên WiFi, chương trình tiếp tục quét để tìm SSID trên các WiFi đến khi tìm thấy SSID đã được cài đặt trong chương trình. Khi SSID đã phù hợp, trùng khớp với WiFi đã thiết lập, chương trình sẽ nhập mật khẩu và đăng nhập vào WiFi đó. Sau khi đăng nhập thành công thì ESP32 chuyển thành chế độ online và thực hiện quá trình “POST” dữ liệu đo được từ cảm biến lên MongoDB và “GET” Giá trị “u” sau khi đã tính toán trên Realm về ESP để so sánh và xuất tín hiệu cho các bơm.

### 4.3. Lưu đồ giải thuật PID trên MongoDB Realm



- B1: Chúng ta tính  $e_t$  bằng cách lấy giá trị “setpoint” - “level” đo được từ cảm biến.
- B2:

- + Nếu chúng ta chỉ chọn bộ điều khiển P thì  $u = K_p \cdot e_t$
- + Nếu chúng ta chọn bộ PI thì  $u = K_p \cdot (e_t + e_{total}/T_i)$
- + Nếu chúng ta chọn bộ PD thì  $u = K_p \cdot (e_t + T_d \cdot D)$
- + Nếu chúng ta chọn bộ PID thì  $u = K_p \cdot (e_t + e_{total}/T_i + T_d \cdot D)$

**Trong đó:**

**$e_t$ :** lỗi tại thời điểm hiện tại.

**$e_{pre}$ :** lỗi trước đó.

**$e_{total}$ :** tổng lỗi tích lũy theo thời gian( bộ tích phân theo thời gian).

**D:** bù lỗi( bộ đạo hàm theo thời gian).

**u:** tín hiệu ngõ ra của bộ điều khiển PID.

#### 4.4. Tính toán lý thuyết

Error = Setpoint ( mục chất lỏng mong muốn ) – H ( mục chất lỏng thực tế ).

$$H = U_t + h_{pre}$$

$h_{pre}$ : mục nước trước đó

##### 4.4.1. Bộ P

\**Trường hợp không nhiều*

$$\text{setpoint} = 10, \text{sensor} = 0$$

$$\text{Chọn } K_p = 0.6$$

Tại  $t = 0s$

$$H = 0$$

Tại  $t = 1s$ :

$$U_t = K_p * (\text{setpoint} - H) = 0.6 * (10 - 0) = 6$$

$$\Rightarrow H = U_t + h_{pre} = 6 + 0 = 6$$

Tại  $t = 2s$ :

$$U_t = K_p * (\text{setpoint} - H) = 0.6 * (10 - 6) = 2.4$$

$$\Rightarrow H = U_t + h_{pre} = 2.4 + 6 = 8.4$$

Tại  $t = 3s$ :

$$U_t = K_p * (\text{setpoint} - H) = 0.6 * (10 - 8.4) = 0.96$$

$$\Rightarrow H = U_t + h_{pre} = 0.96 + 8.4 = 9.36$$

Tại  $t = 4s$ :

$$U_t = K_p * (\text{setpoint} - H) = 0.6 * (10 - 9.36) = 0.384$$

$$\Rightarrow H = U_t + h_{pre} = 0.384 + 9.36 = 9.744$$

Tại  $t = 5s$ :

$$U_t = K_p * (\text{setpoint} - H) = 0.6 * (10 - 9.744) = 0.1536$$

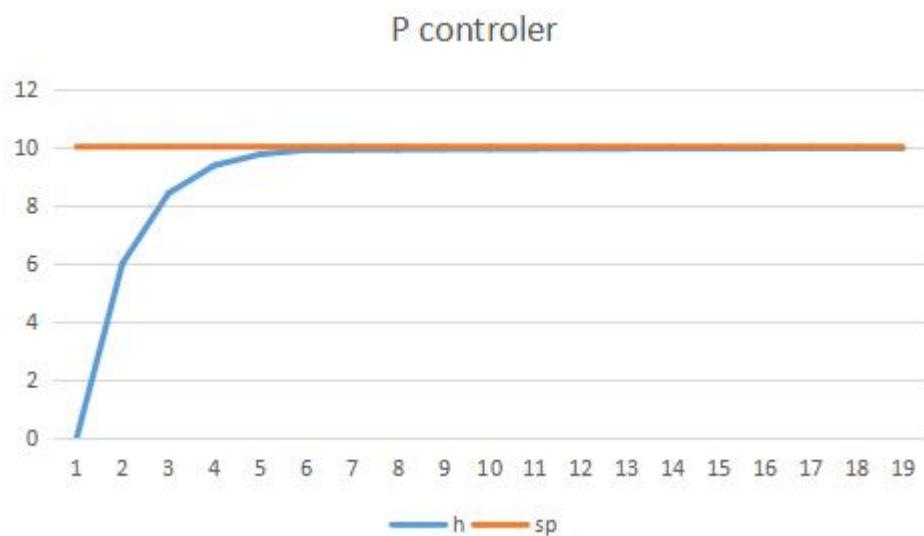
$$\Rightarrow H = U_t + h_{pre} = 0.1536 + 9.744 = 9.8976$$

Tại  $t = Ns$

$$\Rightarrow H \Rightarrow \text{setpoint} = 10$$

T(s)	0	1	2	3	4	5	6	7	8	9
H	0	6	8.4	9.36	9.74	9.89	9.9	9.91	9.92	9.92
T(s)	10	11	12	13	14	15	16	17	18	19
H	9.93	9.93	9.94	9.94	9.94	9.95	9.95	9.95	9.95	9.96

**Bảng 4.4.1.1. Giá trị H của bộ P theo thời gian t(s)**



**Hình 4.4.1.1. Đồ thị mô phỏng bộ P (không nhiễu)**

\**Trường hợp có nhiễu*

**setpoint = 10, sensor = 0**

**Giả sử nhiễu Uout = 1.2**

**Chọn Kp = 0.6**

Tại t = 0s

$$H = 0$$

Tại t = 1s:

$$U_t = Kp * (setpoint - H) - Uout = 0.6 * (10 - 0) - 1.2 = 4.8$$

$$\Rightarrow H = U_t + h_{pre} = 4.8 + 0 = 4.8$$

Tại t = 2s:

$$U_t = Kp * (setpoint - H) - Uout = 0.6 * (10 - 4.8) - 1.2 = 1.92$$

$$\Rightarrow H = U_t + h_{pre} = 1.92 + 4.8 = 6.72$$

Tại t = 3s:

$$U_t = Kp * (setpoint - H) - Uout = 0.6 * (10 - 6.72) - 1.2 = 0.768$$

$$\Rightarrow H = U_t + h_{pre} = 0.768 + 6.72 = 7.488$$

Tại t = 4s:

$$U_t = Kp * (setpoint - H) - Uout = 0.6 * (10 - 7.488) - 1.2 = 0.307$$

$$\Rightarrow H = U_t + h_{pre} = 0.307 + 7.488 = 7.795$$

Tại t = 5s:

$$U_t = Kp * (setpoint - H) - Uout = 0.6 * (10 - 7.795) - 1.2 = 0.123$$

$$\Rightarrow H = U_t + h_{pre} = 0.123 + 7.795 = 7.918$$

Tại t = 6s:

$$U_t = Kp * (setpoint - H) - Uout = 0.6 * (10 - 7.918) - 1.2 = 0.05$$

$$\Rightarrow H = U_t + h_{pre} = 0.05 + 7.918 = 7.968$$

Tại t = 7s:

$$U_t = Kp * (setpoint - H) - Uout = 0.6 * (10 - 7.968) - 1.2 = 0.02$$

$$\Rightarrow H = U_t + h_{pre} = 0.02 + 7.968 = 7.988$$

Tại t = 8s:

$$\begin{aligned} U_t &= K_p * (\text{setpoint} - H) - U_{out} = 0.6 * (10 - 7.988) - 1.2 = 0.007 \\ \Rightarrow H &= U_t + h_{pre} = 0.007 + 7.988 = 7.995 \end{aligned}$$

Tại t = 9s:

$$\begin{aligned} U_t &= K_p * (\text{setpoint} - H) - U_{out} = 0.6 * (10 - 7.995) - 1.2 = 0.003 \\ \Rightarrow H &= U_t + h_{pre} = 0.003 + 7.995 = 7.998 \end{aligned}$$

Tại t = 10s:

$$\begin{aligned} U_t &= K_p * (\text{setpoint} - H) - U_{out} = 0.6 * (10 - 7.998) - 1.2 = 0.002 \\ \Rightarrow H &= U_t + h_{pre} = 0.002 + 7.998 = 8 \end{aligned}$$

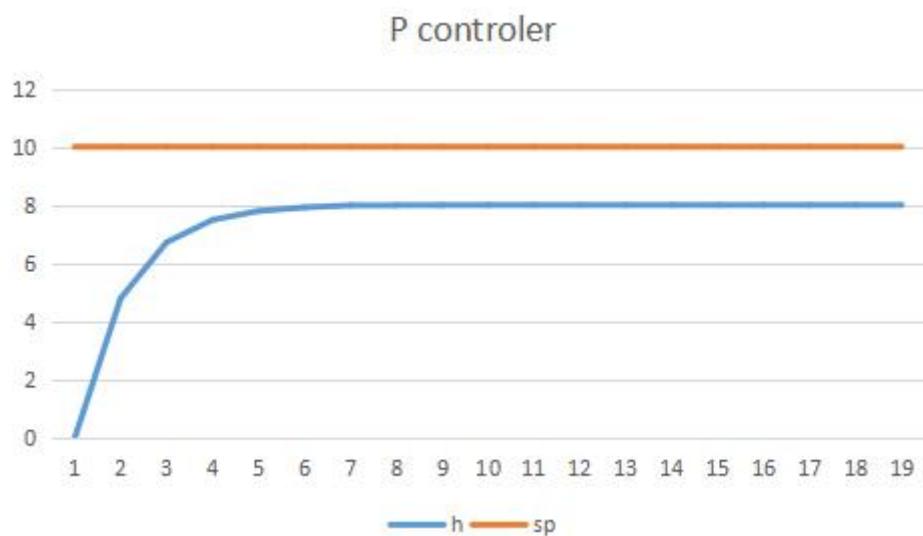
Tại t = 11s:

$$\begin{aligned} U_t &= K_p * (\text{setpoint} - H) - U_{out} = 0.6 * (10 - 8) - 1.2 = 0 \\ \Rightarrow H &= U_t + h_{pre} = 0 + 8 = 8 \end{aligned}$$

**ĐỒ ÁN TỔNG HỢP**  
**Trang - 49**

T(s)	0	1	2	3	4	5	6	7	8	9
H	0	4.8	6.72	7.488	7.795	7.918	7.988	7.995	7.998	8
T(s)	10	11	12	13	14	15	16	17	18	19
H	8	8	8	8	8	8	8	8	8	8

**Bảng 4.4.1.2. Giá trị H của bộ P theo thời gian t(s) (có nhiễu)**



**Hình 4.1.1.2. Đồ thị mô phỏng bộ P (có nhiễu)**

#### 4.4.2. Bộ PI

\**Trường hợp không nhiễu*

**setpoint = 10, sensor = 0**

**Chọn Kp = 0.4, Ti = 2**

Tại t = 0s

$$H = 0$$

$$e_{\text{total}} = 0$$

Tại t = 1s:

$$e_t = \text{setpoint} - H = 10 - 0 = 10$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 10 + 0 = 10$$

$$U_t = K_p * (e_t + e_{\text{total}}/T_i)$$

$$= 0.4 * (10 + 10/2) = 6$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 6 + 0 = 6$$

Tại t = 2s:

$$e_t = \text{setpoint} - H = 10 - 6 = 4$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 4 + 10 = 14$$

$$U_t = K_p * (e_t + e_{\text{total}}/T_i)$$

$$= 0.4 * (4 + 14/2) = 4.4$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 4.4 + 6 = 10.4$$

Tại t = 3s:

$$e_t = \text{setpoint} - H = 10 - 10.4 = -0.6$$

$$e_{\text{total}} = e_t + e_{\text{total}} = -0.6 + 14 = 13.4$$

$$U_t = K_p * (e_t + e_{\text{total}}/T_i)$$

$$= 0.4 * (-0.6 + 13.4/2) = 2.44$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 2.44 + 10.4 = 12.84$$

Tại t = 4s:

$$e_t = \text{setpoint} - H = 10 - 12.84 = -2.84$$

$$e_{\text{total}} = e_t + e_{\text{total}} = -2.84 + 13.4 = 10.56$$

$$U_t = K_p * (e_t + e_{\text{total}}/T_i)$$

$$= 0.4 * (-2.84 + 10.56/2) = 0.976$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0.976 + 12.84 = 13.816 \text{ (với lỗ max)}$$

Tại t = 5s:

$$e_t = \text{setpoint} - H = 10 - 13.816 = -3.816$$

$$e_{\text{total}} = e_t + e_{\text{total}} = -3.816 + 10.56 = 6.744$$

$$U_t = K_p * (e_t + e_{\text{total}}/T_i)$$

$$= 0.4 * (-3.816 + 6.744/2) = -0.1776$$

$$\Rightarrow H = U_t + h_{\text{pre}} = -0.1776 + 13.816 = 13.6384$$

Tại t = 6s:

$$e_t = \text{setpoint} - H = 10 - 13.6384 = -3.6384$$

$$e_{\text{total}} = e_t + e_{\text{total}} = -3.6384 + 6.744 = 3.1056$$

$$U_t = K_p * (e_t + e_{\text{total}}/T_i)$$

$$= 0.4 * (-3.6384 + 3.1056/2) = -0.83424$$

$$\Rightarrow H = U_t + h_{\text{pre}} = -0.83424 + 13.6384 = 12.804$$

Tại t = 7s:

$$e_t = \text{setpoint} - H = 10 - 12.804 = -2.804$$

$$e_{\text{total}} = e_t + e_{\text{total}} = -2.804 + 3.1056 = 0.3016$$

$$U_t = K_p * (e_t + e_{\text{total}}/T_i)$$

$$= 0.4 * (-2.804 + 0.3016/2) = -1.061$$

$$\Rightarrow H = U_t + h_{\text{pre}} = -1.061 + 12.804 = 11.743$$

Tại t = 8s:

$$\begin{aligned}e_t &= \text{setpoint} - H = 10 - 11.743 = -1.743 \\e_{\text{total}} &= e_t + e_{\text{total}} = -1.743 + 0.3016 = -1.441\end{aligned}$$

$$\begin{aligned}U_t &= K_p * (e_t + e_{\text{total}} / T_i) \\&= 0.4 * (-1.743 - 1.441 / 2) = -0.985 \\ \Rightarrow H &= U_t + h_{\text{pre}} = \mathbf{-0.985 + 11.743 = 10.758}\end{aligned}$$

Tại t = 9s:

$$\begin{aligned}e_t &= \text{setpoint} - H = 10 - 10.758 = -0.758 \\e_{\text{total}} &= e_t + e_{\text{total}} = -0.758 - 1.441 = -2.199\end{aligned}$$

$$\begin{aligned}U_t &= K_p * (e_t + e_{\text{total}} / T_i) \\&= 0.4 * (-0.758 - 2.199 / 2) = -0.743 \\ \Rightarrow H &= U_t + h_{\text{pre}} = \mathbf{-0.743 + 10.758 = 10.015}\end{aligned}$$

Tại t = 10s:

$$\begin{aligned}e_t &= \text{setpoint} - H = 10 - 10.015 = -0.015 \\e_{\text{total}} &= e_t + e_{\text{total}} = -0.015 - 2.199 = -2.214\end{aligned}$$

$$\begin{aligned}U_t &= K_p * (e_t + e_{\text{total}} / T_i) \\&= 0.4 * (-0.015 - 2.214 / 2) = -0.449 \\ \Rightarrow H &= U_t + h_{\text{pre}} = \mathbf{-0.449 + 10.015 = 9.566}\end{aligned}$$

Tại t = 11s:

$$\begin{aligned}e_t &= \text{setpoint} - H = 10 - 9.566 = 0.434 \\e_{\text{total}} &= e_t + e_{\text{total}} = 0.434 - 2.214 = -1.78\end{aligned}$$

$$\begin{aligned}U_t &= K_p * (e_t + e_{\text{total}} / T_i) \\&= 0.4 * (0.434 - 1.78 / 2) = -0.182 \\ \Rightarrow H &= U_t + h_{\text{pre}} = \mathbf{-0.182 + 9.566 = 9.384}\end{aligned}$$

Tại t = 12s:

$$e_t = \text{setpoint} - H = 10 - 9.384 = 0.616$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 0.616 - 1.78 = -1.164$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i) \\ &= 0.4 * (0.616 - 1.164 / 2) = 0.014 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0.014 + 9.384 = 9.398$$

Tại t = 13s:

$$e_t = \text{setpoint} - H = 10 - 9.398 = 0.602$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 0.602 - 1.164 = -0.562$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i) \\ &= 0.4 * (0.602 - 0.562 / 2) = 0.129 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0.129 + 9.398 = 9.527$$

Tại t = 14s:

$$e_t = \text{setpoint} - H = 10 - 9.527 = 0.473$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 0.473 - 0.562 = -0.089$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i) \\ &= 0.4 * (0.473 - 0.089 / 2) = 0.171 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0.171 + 9.527 = 9.698$$

Tại t = 15s:

$$e_t = \text{setpoint} - H = 10 - 9.698 = 0.302$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 0.302 - 0.089 = 0.213$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i) \\ &= 0.4 * (0.302 + 0.213 / 2) = 0.163 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0.163 + 9.698 = 9.861$$

Tại t = 16s:

$$e_t = \text{setpoint} - H = 10 - 9.861 = 0.139$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 0.139 + 0.213 = 0.352$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i) \\ &= 0.4 * (0.139 + 0.352 / 2) = 0.126 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0.126 + 9.861 = 9.987$$

Tại t = 17s:

$$e_t = \text{setpoint} - H = 10 - 9.987 = 0.013$$

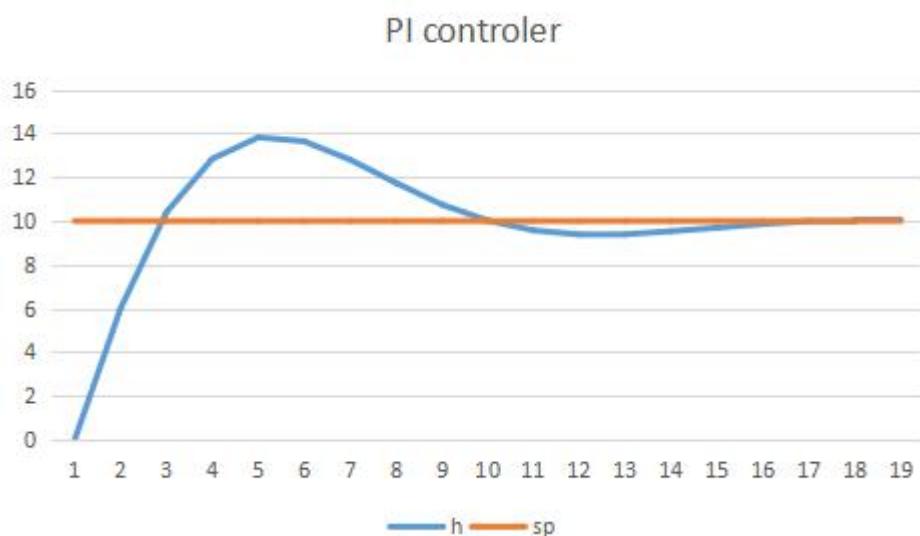
$$e_{\text{total}} = e_t + e_{\text{total}} = 0.013 + 0.352 = 0.365$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i) \\ &= 0.4 * (0.013 + 0.352 / 2) = 0.078 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0.078 + 9.987 = 10.065$$

T(s)	0	1	2	3	4	5	6	7	8	9
H	0	6	10.4	12.84	<u>13.82</u>	13.64	12.8	11.74	10.75	10.02
T(s)	10	11	12	13	14	15	16	17	18	19
H	9.57	9.38	9.39	9.53	9.69	9.86	9.99	10.06	10.01	10

**Bảng 4.4.2.1. Giá trị H của bộ PI theo thời gian t(s)**



**Hình 4.4.2.1. Đồ thị mô phỏng bộ PI(không nhiễu)**

Định: 13.82

Thời gian xác lập: 16(s)

\**Trường hợp có nhiễu*

**setpoint = 10, sensor = 0**

**Chọn Kp = 0.4, Ti = 2**

**Giả sử Uout = 1.2**

Tại t = 0s

$$H = 0$$

$$e_{\text{total}} = 0$$

Tại t = 1s:

$$e_t = \text{setpoint} - H = 10 - 0 = 10$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 10 + 0 = 10$$

$$\begin{aligned} U_t &= Kp * (e_t + e_{\text{total}}/Ti) - Uout \\ &= 0.4 * (10 + 10/2) - 1.2 = 4.8 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 4.8 + 0 = 4.8$$

Tại t = 2s:

$$e_t = \text{setpoint} - H = 10 - 4.8 = 5.2$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 5.2 + 10 = 15.2$$

$$\begin{aligned} U_t &= Kp * (e_t + e_{\text{total}}/Ti) - Uout \\ &= 0.4 * (5.2 + 15.2/2) - 1.2 = 3.92 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 3.92 + 4.8 = 8.72$$

Tại t = 3s:

$$e_t = \text{setpoint} - H = 10 - 8.72 = 1.28$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 1.28 + 15.2 = 16.48$$

$$\begin{aligned} U_t &= Kp * (e_t + e_{\text{total}}/Ti) - Uout \\ &= 0.4 * (1.28 + 16.48/2) - 1.2 = 2.608 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 2.608 + 8.72 = 11.328$$

Tại t = 4s:

$$\begin{aligned}e_t &= \text{setpoint} - H = 10 - 11.328 = -1.328 \\e_{\text{total}} &= e_t + e_{\text{total}} = -1.328 + 16.48 = 15.152\end{aligned}$$

$$\begin{aligned}U_t &= K_p * (e_t + e_{\text{total}} / T_i) - U_{\text{out}} \\&= 0.4 * (-1.328 + 15.152 / 2) - 1.2 = 1.299 \\ \Rightarrow H &= U_t + h_{\text{pre}} = 1.299 + 11.328 = 12.627\end{aligned}$$

Tại t = 5s:

$$\begin{aligned}e_t &= \text{setpoint} - H = 10 - 12.627 = -2.627 \\e_{\text{total}} &= e_t + e_{\text{total}} = -2.627 + 15.152 = 12.525\end{aligned}$$

$$\begin{aligned}U_t &= K_p * (e_t + e_{\text{total}} / T_i) - U_{\text{out}} \\&= 0.4 * (-2.627 + 12.525 / 2) - 1.2 = 0.254 \\ \Rightarrow H &= U_t + h_{\text{pre}} = 0.254 + 12.627 = 12.881\end{aligned}$$

Tại t = 6s:

$$\begin{aligned}e_t &= \text{setpoint} - H = 10 - 12.881 = -2.881 \\e_{\text{total}} &= e_t + e_{\text{total}} = -2.881 + 12.525 = 9.644\end{aligned}$$

$$\begin{aligned}U_t &= K_p * (e_t + e_{\text{total}} / T_i) - U_{\text{out}} \\&= 0.4 * (-2.881 + 9.644 / 2) - 1.2 = -0.424 \\ \Rightarrow H &= U_t + h_{\text{pre}} = -0.424 + 12.881 = 12.457\end{aligned}$$

Tại t = 7s:

$$\begin{aligned}e_t &= \text{setpoint} - H = 10 - 12.457 = -2.457 \\e_{\text{total}} &= e_t + e_{\text{total}} = -2.457 + 9.644 = 7.187\end{aligned}$$

$$\begin{aligned}U_t &= K_p * (e_t + e_{\text{total}} / T_i) - U_{\text{out}} \\&= 0.4 * (-2.457 + 7.187 / 2) - 1.2 = -0.745 \\ \Rightarrow H &= U_t + h_{\text{pre}} = -0.745 + 12.457 = 11.712\end{aligned}$$

Tại t = 8s:

$$e_t = \text{setpoint} - H = 10 - 11.712 = -1.712$$

$$e_{\text{total}} = e_t + e_{\text{total}} = -1.712 + 7.187 = 5.475$$

$$U_t = K_p * (e_t + e_{\text{total}}/T_i) - U_{\text{out}}$$

$$= 0.4 * (-1.712 + 5.475/2) - 1.2 = -0.79$$

$$\Rightarrow H = U_t + h_{\text{pre}} = -0.79 + 11.712 = 10.922$$

Tại t = 9s:

$$e_t = \text{setpoint} - H = 10 - 10.922 = -0.922$$

$$e_{\text{total}} = e_t + e_{\text{total}} = -0.922 + 5.475 = 4.553$$

$$U_t = K_p * (e_t + e_{\text{total}}/T_i) - U_{\text{out}}$$

$$= 0.4 * (-0.922 + 4.553/2) - 1.2 = -0.658$$

$$\Rightarrow H = U_t + h_{\text{pre}} = -0.658 + 10.922 = 10.264$$

Tại t = 10s:

$$e_t = \text{setpoint} - H = 10 - 10.264 = -0.264$$

$$e_{\text{total}} = e_t + e_{\text{total}} = -0.264 + 4.553 = 4.289$$

$$U_t = K_p * (e_t + e_{\text{total}}/T_i) - U_{\text{out}}$$

$$= 0.4 * (-0.264 + 4.289/2) - 1.2 = -0.448$$

$$\Rightarrow H = U_t + h_{\text{pre}} = -0.448 + 10.264 = 9.816$$

Tại t = 11s:

$$e_t = \text{setpoint} - H = 10 - 9.816 = 0.184$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 0.184 + 4.289 = 4.473$$

$$U_t = K_p * (e_t + e_{\text{total}}/T_i) - U_{\text{out}}$$

$$= 0.4 * (0.184 + 4.473/2) - 1.2 = -0.232$$

$$\Rightarrow H = U_t + h_{\text{pre}} = -0.232 + 9.816 = 9.584$$

Tại t=12s:

$$e_t = \text{setpoint} - H = 10 - 9.584 = 0.416$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 0.416 + 4.473 = 4.889$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}}/T_i) - U_{\text{out}} \\ &= 0.4 * (0.416 + 4.889/2) - 1.2 = -0.056 \\ \Rightarrow H &= U_t + h_{\text{pre}} = -0.056 + 9.584 = 9.528 \end{aligned}$$

Tại t=13s:

$$e_t = \text{setpoint} - H = 10 - 9.528 = 0.472$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 0.472 + 4.889 = 5.361$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}}/T_i) - U_{\text{out}} \\ &= 0.4 * (0.472 + 5.361/2) - 1.2 = 0.061 \\ \Rightarrow H &= U_t + h_{\text{pre}} = 0.061 + 9.528 = 9.589 \end{aligned}$$

Tại t=14s:

$$e_t = \text{setpoint} - H = 10 - 9.589 = 0.411$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 0.411 + 5.361 = 5.772$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}}/T_i) - U_{\text{out}} \\ &= 0.4 * (0.411 + 5.772/2) - 1.2 = 0.119 \\ \Rightarrow H &= U_t + h_{\text{pre}} = 0.119 + 9.589 = 9.708 \end{aligned}$$

Tại t=15s:

$$e_t = \text{setpoint} - H = 10 - 9.708 = 0.292$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 0.292 + 5.772 = 6.064$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}}/T_i) - U_{\text{out}} \\ &= 0.4 * (0.292 + 6.064/2) - 1.2 = 0.13 \\ \Rightarrow H &= U_t + h_{\text{pre}} = 0.13 + 9.708 = 9.838 \end{aligned}$$

Tại t = 16s:

$$e_t = \text{setpoint} - H = 10 - 9.838 = 0.162$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 0.162 + 6.064 = 6.226$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i) - U_{\text{out}} \\ &= 0.4 * (0.162 + 6.226 / 2) - 1.2 = 0.11 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0.11 + 9.838 = 9.948$$

Tại t = 17s:

$$e_t = \text{setpoint} - H = 10 - 9.948 = 0.052$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 0.052 + 6.226 = 6.278$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i) - U_{\text{out}} \\ &= 0.4 * (0.052 + 6.278 / 2) - 1.2 = 0.076 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0.076 + 9.948 = 10.024$$

Tại t = 18s:

$$e_t = \text{setpoint} - H = 10 - 10.024 = -0.024$$

$$e_{\text{total}} = e_t + e_{\text{total}} = -0.024 + 6.278 = 6.254$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i) - U_{\text{out}} \\ &= 0.4 * (-0.024 + 6.254 / 2) - 1.2 = 0.041 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0.041 + 10.024 = 10.065$$

Tại t = 19s:

$$e_t = \text{setpoint} - H = 10 - 10.065 = -0.065$$

$$e_{\text{total}} = e_t + e_{\text{total}} = -0.065 + 6.254 = 6.189$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i) - U_{\text{out}} \\ &= 0.4 * (-0.065 + 6.189 / 2) - 1.2 = 0.01 \end{aligned}$$

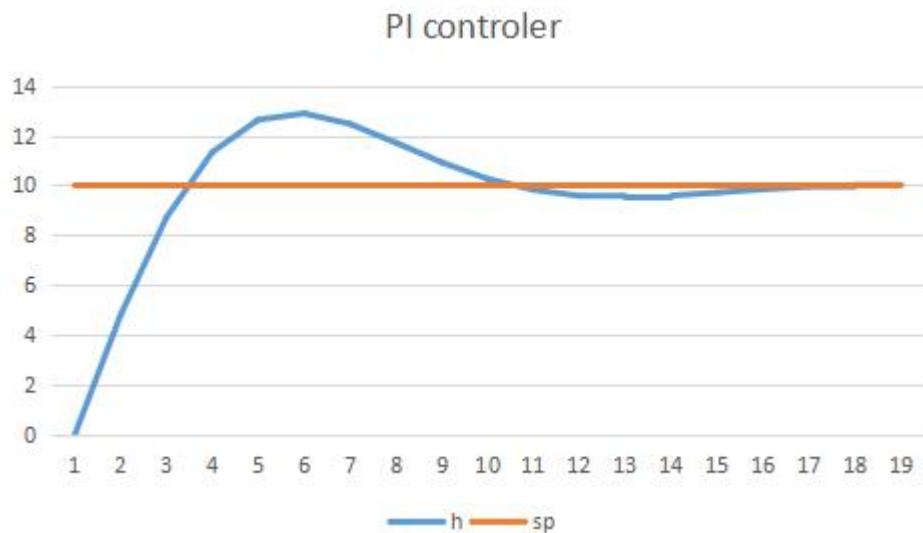
$$\Rightarrow H = U_t + h_{\text{pre}} = 0.01 + 10.065 = 10.075$$

**ĐỒ ÁN TỔNG HỢP**  
**Trang - 61**

---

T(s)	0	1	2	3	4	5	6	7	8	9
H	0	4.8	8.72	11.33	12.63	<u>12.88</u>	12.46	11.71	10.92	10.26
T(s)	10	11	12	13	14	15	16	17	18	19
H	9.82	9.58	9.52	9.59	9.71	9.84	9.95	10.02	10.07	10.08

**Bảng 4.4.2.2. Giá trị H của bộ PI theo thời gian t(s) (có nhiễu)**



**Hình 4.4.2.2. Đồ thị mô phỏng bộ PI (có nhiễu)**

Đỉnh: 12.88

Thời gian xác lập: 16(s)

#### 4.4.3. Bộ PD

\**Trường hợp không nhiễu*

**setpoint = 10, sensor = 0**

**Chọn Kp = 0.4, Td = 1**

Tại t = 0s

$$H = 0$$

Tại t = 1s:

$$e_t = \text{setpoint} - H = 10 - 0 = 10$$

$$D = e_t - e_{\text{pre}} = 10 - 10 = 0$$

$$U_t = K_p * (e_t + T_d * D)$$

$$= 0.4 * (10 + 0) = 4$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 4 + 0 = 4$$

Tại t = 2s:

$$e_t = \text{setpoint} - H = 10 - 4 = 6$$

$$D = e_t - e_{\text{pre}} = 6 - 10 = -4$$

$$U_t = K_p * (e_t + T_d * D)$$

$$= 0.4 * (6 - 4) = 0.8$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0.8 + 4 = 4.8$$

Tại t = 3s:

$$e_t = \text{setpoint} - H = 10 - 4.8 = 5.2$$

$$D = e_t - e_{\text{pre}} = 5.2 - 6 = -0.8$$

$$U_t = K_p * (e_t + T_d * D)$$

$$= 0.4 * (5.2 - 0.8) = 1.76$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 1.76 + 4.8 = 6.56$$

Tại t = 4s:

$$e_t = \text{setpoint} - H = 10 - 6.56 = 3.44$$

$$D = e_t - e_{\text{pre}} = 3.44 - 5.2 = -1.76$$

$$U_t = K_p * (e_t + T_d * D)$$

$$= 0.4 * (3.44 - 1.76) = 0.672$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0.672 + 6.56 = 7.232$$

Tại t = 5s:

$$e_t = \text{setpoint} - H = 10 - 7.232 = 2.768$$

$$D = e_t - e_{\text{pre}} = 2.768 - 3.44 = -0.672$$

$$U_t = K_p * (e_t + T_d * D)$$

$$= 0.4 * (2.768 - 0.672) = 0.838$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0.838 + 7.232 = 8.07$$

Tại t = 6s:

$$e_t = \text{setpoint} - H = 10 - 8.07 = 1.93$$

$$D = e_t - e_{\text{pre}} = 1.93 - 2.768 = -0.838$$

$$U_t = K_p * (e_t + T_d * D)$$

$$= 0.4 * (1.93 - 0.838) = 0.437$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0.437 + 8.07 = 8.507$$

Tại t = 7s:

$$e_t = \text{setpoint} - H = 10 - 8.507 = 1.493$$

$$D = e_t - e_{\text{pre}} = 1.493 - 1.93 = -0.437$$

$$U_t = K_p * (e_t + T_d * D)$$

$$= 0.4 * (1.493 - 0.437) = 0.422$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0.422 + 8.507 = 8.929$$

Tại t = 8s:

$$e_t = \text{setpoint} - H = 10 - 8.929 = 1.071$$

$$D = e_t - e_{\text{pre}} = 1.071 - 1.493 = -0.422$$

$$\begin{aligned} U_t &= K_p * (e_t + T_d * D) \\ &= 0.4 * (1.071 - 0.422) = 0.26 \\ \Rightarrow H &= U_t + h_{\text{pre}} = 0.26 + 8.929 = 9.189 \end{aligned}$$

Tại t = 9s:

$$e_t = \text{setpoint} - H = 10 - 9.189 = 0.811$$

$$D = e_t - e_{\text{pre}} = 0.811 - 1.071 = -0.26$$

$$\begin{aligned} U_t &= K_p * (e_t + T_d * D) \\ &= 0.4 * (0.811 - 0.26) = 0.22 \\ \Rightarrow H &= U_t + h_{\text{pre}} = 0.22 + 9.189 = 9.409 \end{aligned}$$

Tại t = 10s:

$$e_t = \text{setpoint} - H = 10 - 9.409 = 0.591$$

$$D = e_t - e_{\text{pre}} = 0.591 - 0.811 = -0.22$$

$$\begin{aligned} U_t &= K_p * (e_t + T_d * D) \\ &= 0.4 * (0.591 - 0.22) = 0.148 \\ \Rightarrow H &= U_t + h_{\text{pre}} = 0.148 + 9.409 = 9.557 \end{aligned}$$

Tại t = 11s:

$$e_t = \text{setpoint} - H = 10 - 9.557 = 0.443$$

$$D = e_t - e_{\text{pre}} = 0.443 - 0.591 = -0.148$$

$$\begin{aligned} U_t &= K_p * (e_t + T_d * D) \\ &= 0.4 * (0.443 - 0.148) = 0.118 \\ \Rightarrow H &= U_t + h_{\text{pre}} = 0.118 + 9.557 = 9.675 \end{aligned}$$

Tại t = 12s:

$$e_t = \text{setpoint} - H = 10 - 9.675 = 0.325$$

$$D = e_t - e_{\text{pre}} = 0.325 - 0.443 = -0.118$$

$$\begin{aligned} U_t &= K_p * (e_t + T_d * D) \\ &= 0.4 * (0.325 - 0.118) = 0.083 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0.083 + 9.675 = 9.758$$

Tại t = 13s:

$$e_t = \text{setpoint} - H = 10 - 9.758 = 0.242$$

$$D = e_t - e_{\text{pre}} = 0.242 - 0.325 = -0.083$$

$$\begin{aligned} U_t &= K_p * (e_t + T_d * D) \\ &= 0.4 * (0.242 - 0.083) = 0.064 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0.064 + 9.758 = 9.822$$

Tại t = 14s:

$$e_t = \text{setpoint} - H = 10 - 9.822 = 0.178$$

$$D = e_t - e_{\text{pre}} = 0.178 - 0.242 = -0.064$$

$$\begin{aligned} U_t &= K_p * (e_t + T_d * D) \\ &= 0.4 * (0.178 - 0.064) = 0.046 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0.046 + 9.822 = 9.868$$

Tại t = 15s:

$$e_t = \text{setpoint} - H = 10 - 9.868 = 0.132$$

$$D = e_t - e_{\text{pre}} = 0.132 - 0.178 = -0.046$$

$$\begin{aligned} U_t &= K_p * (e_t + T_d * D) \\ &= 0.4 * (0.132 - 0.046) = 0.034 \end{aligned}$$

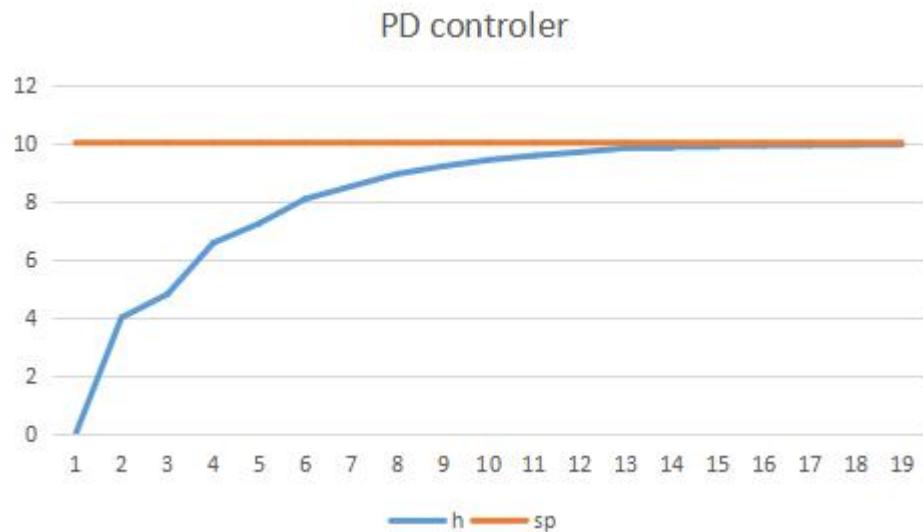
$$\Rightarrow H = U_t + h_{\text{pre}} = 0.034 + 9.868 = 9.902$$

Tại t = N  $\Rightarrow H \Rightarrow 10$

**ĐỒ ÁN TỔNG HỢP**  
**Trang - 66**

T(s)	0	1	2	3	4	5	6	7	8	9
H	0	4	4.8	6.56	7.232	8.07	8.507	8.929	9.189	9.409
T(s)	10	11	12	13	14	15	16	17	18	19
H	9.557	9.675	9.758	9.822	9.868	9.902	9.912	9.922	9.930	9.939

Bảng 4.4.3.1. Giá trị H của bộ PD theo thời gian t(s)



Hình 4.4.3.1. Đồ thị mô phỏng bộ PD (không nhiễu)

\**Trường hợp có nhiễu*

**setpoint = 10, sensor = 0**

**Chọn Kp = 0.4, Td = 1**

**Giả sử Uout = 1.2**

Tại t = 0s

$$H = 0$$

Tại t = 1s:

$$e_t = \text{setpoint} - H = 10 - 0 = 10$$

$$D = e_t - e_{\text{pre}} = 10 - 10 = 0$$

$$U_t = K_p * (e_t + T_d * D) - U_{\text{out}}$$

$$= 0.4 * (10 + 0) - 1.2 = 2.8$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 2.8 + 0 = 2.8$$

Tại t = 2s:

$$e_t = \text{setpoint} - H = 10 - 2.8 = 7.2$$

$$D = e_t - e_{\text{pre}} = 7.2 - 10 = -2.8$$

$$U_t = K_p * (e_t + T_d * D) - U_{\text{out}}$$

$$= 0.4 * (7.2 - 2.8) - 1.2 = 0.56$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0.56 + 2.8 = 3.36$$

Tại t = 3s:

$$e_t = \text{setpoint} - H = 10 - 3.36 = 6.64$$

$$D = e_t - e_{\text{pre}} = 6.64 - 7.2 = -0.56$$

$$U_t = K_p * (e_t + T_d * D) - U_{\text{out}}$$

$$= 0.4 * (6.64 - 0.56) - 1.2 = 1.232$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 1.232 + 3.36 = 4.592$$

Tại t = 4s:

$$e_t = \text{setpoint} - H = 10 - 4.592 = 5.408$$

$$D = e_t - e_{\text{pre}} = 5.408 - 6.64 = -1.232$$

$$\begin{aligned} U_t &= K_p * (e_t + T_d * D) - U_{\text{out}} \\ &= 0.4 * (5.408 - 1.232) - 1.2 = 0.47 \\ \Rightarrow H &= U_t + h_{\text{pre}} = 0.47 + 4.592 = 5.062 \end{aligned}$$

Tại t = 5s:

$$e_t = \text{setpoint} - H = 10 - 5.062 = 4.938$$

$$D = e_t - e_{\text{pre}} = 4.938 - 5.408 = -0.47$$

$$\begin{aligned} U_t &= K_p * (e_t + T_d * D) - U_{\text{out}} \\ &= 0.4 * (4.938 - 0.47) - 1.2 = 0.587 \\ \Rightarrow H &= U_t + h_{\text{pre}} = 0.587 + 5.062 = 5.649 \end{aligned}$$

Tại t = 6s:

$$e_t = \text{setpoint} - H = 10 - 5.649 = 4.351$$

$$D = e_t - e_{\text{pre}} = 4.351 - 4.938 = -0.587$$

$$\begin{aligned} U_t &= K_p * (e_t + T_d * D) - U_{\text{out}} \\ &= 0.4 * (4.351 - 0.587) - 1.2 = 0.306 \\ \Rightarrow H &= U_t + h_{\text{pre}} = 0.306 + 5.649 = 5.955 \end{aligned}$$

Tại t = 7s:

$$e_t = \text{setpoint} - H = 10 - 5.955 = 4.045$$

$$D = e_t - e_{\text{pre}} = 4.045 - 4.351 = -0.306$$

$$\begin{aligned} U_t &= K_p * (e_t + T_d * D) - U_{\text{out}} \\ &= 0.4 * (4.045 - 0.306) - 1.2 = 0.296 \\ \Rightarrow H &= U_t + h_{\text{pre}} = 0.296 + 5.955 = 6.251 \end{aligned}$$

Tại t = 8s:

$$e_t = \text{setpoint} - H = 10 - 6.251 = 3.749$$

$$D = e_t - e_{\text{pre}} = 3.749 - 4.045 = -0.296$$

$$\begin{aligned} U_t &= K_p * (e_t + T_d * D) - U_{\text{out}} \\ &= 0.4 * (3.749 - 0.296) - 1.2 = 0.181 \\ \Rightarrow H &= U_t + h_{\text{pre}} = 0.181 + 6.251 = 6.432 \end{aligned}$$

Tại t = 9s:

$$e_t = \text{setpoint} - H = 10 - 6.432 = 3.568$$

$$D = e_t - e_{\text{pre}} = 3.568 - 3.749 = -0.181$$

$$\begin{aligned} U_t &= K_p * (e_t + T_d * D) - U_{\text{out}} \\ &= 0.4 * (3.568 - 0.181) - 1.2 = 0.155 \\ \Rightarrow H &= U_t + h_{\text{pre}} = 0.155 + 6.432 = 6.587 \end{aligned}$$

Tại t = 10s:

$$e_t = \text{setpoint} - H = 10 - 6.587 = 3.413$$

$$D = e_t - e_{\text{pre}} = 3.413 - 3.568 = -0.155$$

$$\begin{aligned} U_t &= K_p * (e_t + T_d * D) - U_{\text{out}} \\ &= 0.4 * (3.413 - 0.155) - 1.2 = 0.103 \\ \Rightarrow H &= U_t + h_{\text{pre}} = 0.103 + 6.587 = 6.69 \end{aligned}$$

Tại t = 11s:

$$e_t = \text{setpoint} - H = 10 - 6.69 = 3.31$$

$$D = e_t - e_{\text{pre}} = 3.31 - 3.413 = -0.103$$

$$\begin{aligned} U_t &= K_p * (e_t + T_d * D) - U_{\text{out}} \\ &= 0.4 * (3.31 - 0.103) - 1.2 = 0.083 \\ \Rightarrow H &= U_t + h_{\text{pre}} = 0.083 + 6.69 = 6.773 \end{aligned}$$

Tại t = 12s:

$$e_t = \text{setpoint} - H = 10 - 6.773 = 3.227$$

$$D = e_t - e_{\text{pre}} = 3.227 - 3.31 = -0.083$$

$$\begin{aligned} U_t &= K_p * (e_t + T_d * D) - U_{\text{out}} \\ &= 0.4 * (3.227 - 0.083) - 1.2 = 0.058 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0.058 + 6.773 = 6.831$$

Tại t = 13s:

$$e_t = \text{setpoint} - H = 10 - 6.831 = 3.169$$

$$D = e_t - e_{\text{pre}} = 3.169 - 3.227 = -0.058$$

$$\begin{aligned} U_t &= K_p * (e_t + T_d * D) - U_{\text{out}} \\ &= 0.4 * (3.169 - 0.058) - 1.2 = 0.044 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0.044 + 6.831 = 6.875$$

Tại t = 14s:

$$e_t = \text{setpoint} - H = 10 - 6.875 = 3.125$$

$$D = e_t - e_{\text{pre}} = 3.125 - 3.169 = -0.044$$

$$\begin{aligned} U_t &= K_p * (e_t + T_d * D) - U_{\text{out}} \\ &= 0.4 * (3.125 - 0.044) - 1.2 = 0.032 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0.032 + 6.875 = 6.907$$

Tại t = 15s:

$$e_t = \text{setpoint} - H = 10 - 6.907 = 3.093$$

$$D = e_t - e_{\text{pre}} = 3.093 - 3.125 = -0.032$$

$$\begin{aligned} U_t &= K_p * (e_t + T_d * D) - U_{\text{out}} \\ &= 0.4 * (3.093 - 0.032) - 1.2 = 0.024 \end{aligned}$$

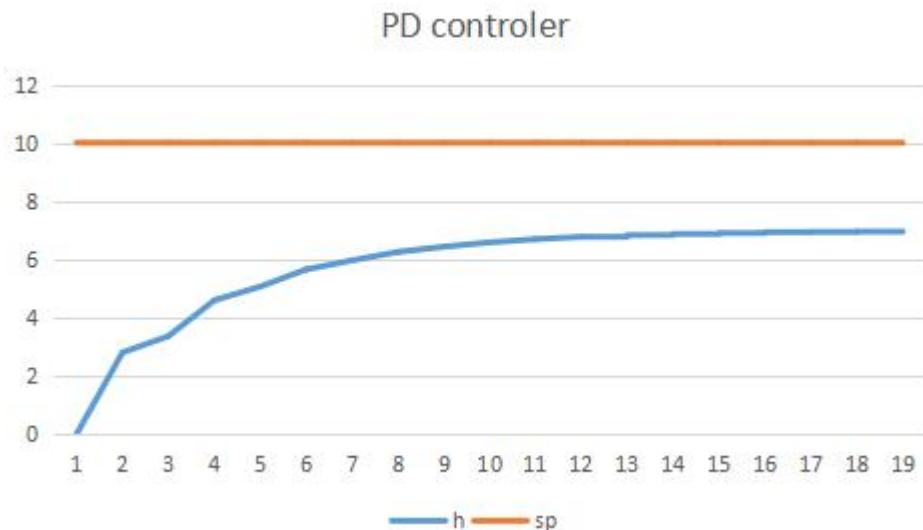
$$\Rightarrow H = U_t + h_{\text{pre}} = 0.024 + 6.907 = 6.931$$

Tại t = N, H=> 7

**ĐỒ ÁN TỔNG HỢP**  
**Trang - 71**

T(s)	0	1	2	3	4	5	6	7	8	9
H	0	2.8	3.36	4.592	5.062	5.649	5.955	6.251	6.432	6.587
T(s)	10	11	12	13	14	15	16	17	18	19
H	6.69	6.773	6.831	6.875	6.907	6.931	6.94	6.945	6.95	6.955

**Bảng 4.4.3.2. Giá trị H của bộ PD theo thời gian t(s) (có nhiễu)**



**Hình 4.4.3.2. Đồ thị mô phỏng bộ PD (có nhiễu)**

#### 4.4.4. Bộ PID

\**Trường hợp không nhiễu*

**setpoint = 10, sensor = 0**

**Chọn Kp = 0.4, Ti = 2, Td = 1**

Tại t = 0s

$$H = 0$$

$$e_{\text{total}} = 0$$

Tại t = 1s:

$$e_t = \text{setpoint} - H = 10 - 0 = 10$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 10 + 0 = 10$$

$$D = e_t - e_{\text{pre}} = 10 - 10 = 0$$

$$U_t = K_p * (e_t + e_{\text{total}} / T_i + T_d * D)$$

$$= 0.4 * (10 + 10/2 + 0) = 6$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 6 + 0 = 6$$

Tại t = 2s:

$$e_t = \text{setpoint} - H = 10 - 6 = 4$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 4 + 10 = 14$$

$$D = e_t - e_{\text{pre}} = 4 - 10 = -6$$

$$U_t = K_p * (e_t + e_{\text{total}} / T_i + T_d * D)$$

$$= 0.4 * (4 + 14/2 - 6) = 2$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 2 + 6 = 8$$

Tại t = 3s:

$$e_t = \text{setpoint} - H = 10 - 8 = 2$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 2 + 14 = 16$$

$$D = e_t - e_{\text{pre}} = 2 - 4 = -2$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) \\ &= 0.4 * (2 + 16 / 2 - 2) = 3.2 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 3.2 + 8 = 11.2$$

Tại t = 4s:

$$e_t = \text{setpoint} - H = 10 - 11.2 = -1.2$$

$$e_{\text{total}} = e_t + e_{\text{total}} = -1.2 + 16 = 14.8$$

$$D = e_t - e_{\text{pre}} = -1.2 - 2 = -3.2$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) \\ &= 0.4 * (-1.2 + 14.8 / 2 - 3.2) = 1.2 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 1.2 + 11.2 = 12.4$$

Tại t = 5s:

$$e_t = \text{setpoint} - H = 10 - 12.4 = -2.4$$

$$e_{\text{total}} = e_t + e_{\text{total}} = -2.4 + 14.8 = 12.4$$

$$D = e_t - e_{\text{pre}} = -2.4 - (-1.2) = -1.2$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) \\ &= 0.4 * (-2.4 + 12.4 / 2 - 1.2) = 1.04 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 1.04 + 12.4 = 13.44$$

Tại t = 6s:

$$e_t = \text{setpoint} - H = 10 - 13.44 = -3.44$$

$$e_{\text{total}} = e_t + e_{\text{total}} = -3.44 + 12.4 = 8.96$$

$$D = e_t - e_{\text{pre}} = -3.44 - (-2.4) = -1.04$$

$$U_t = K_p * (e_t + e_{\text{total}} / T_i + T_d * D)$$

$$= 0.4 * (-3.44 + 8.96 / 2 - 1.04) = 0$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0 + 13.44 = 13.44 \text{ (vượt lõi max)}$$

Tại t = 7s:

$$e_t = \text{setpoint} - H = 10 - 13.44 = -3.44$$

$$e_{\text{total}} = e_t + e_{\text{total}} = -3.44 + 8.96 = 5.52$$

$$D = e_t - e_{\text{pre}} = -3.44 - (-3.44) = 0$$

$$U_t = K_p * (e_t + e_{\text{total}} / T_i + T_d * D)$$

$$= 0.4 * (-3.44 + 5.52 / 2 + 0) = -0.272$$

$$\Rightarrow H = U_t + h_{\text{pre}} = -0.272 + 13.44 = 13.168$$

Tại t = 8s:

$$e_t = \text{setpoint} - H = 10 - 13.168 = -3.168$$

$$e_{\text{total}} = e_t + e_{\text{total}} = -3.168 + 5.52 = 2.352$$

$$D = e_t - e_{\text{pre}} = -3.168 - (-3.44) = 0.272$$

$$U_t = K_p * (e_t + e_{\text{total}} / T_i + T_d * D)$$

$$= 0.4 * (-3.168 + 2.352 / 2 + 0.272) = -0.688$$

$$\Rightarrow H = U_t + h_{\text{pre}} = -0.688 + 13.168 = 12.48$$

Tại t = 9s:

$$e_t = \text{setpoint} - H = 10 - 12.48 = -2.48$$

$$e_{\text{total}} = e_t + e_{\text{total}} = -2.48 + 2.352 = -0.128$$

$$D = e_t - e_{\text{pre}} = -2.48 - (-3.168) = 0.688$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) \\ &= 0.4 * (-2.48 - 0.128 / 2 + 0.688) = -0.742 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = -0.742 + 12.48 = 11.738$$

Tại t = 10s:

$$e_t = \text{setpoint} - H = 10 - 11.738 = -1.738$$

$$e_{\text{total}} = e_t + e_{\text{total}} = -1.738 - 0.128 = -1.856$$

$$D = e_t - e_{\text{pre}} = -1.738 - (-2.48) = 0.742$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) \\ &= 0.4 * (-1.738 - 1.856 / 2 + 0.742) = -0.77 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = -0.77 + 11.738 = 10.968$$

Tại t = 11s:

$$e_t = \text{setpoint} - H = 10 - 10.968 = -0.968$$

$$e_{\text{total}} = e_t + e_{\text{total}} = -0.968 - 1.856 = -2.824$$

$$D = e_t - e_{\text{pre}} = -0.968 - (-1.738) = 0.77$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) \\ &= 0.4 * (-0.968 - 2.824 / 2 + 0.77) = -0.644 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = -0.644 + 10.968 = 10.324$$

Tại t = 12s:

$$e_t = \text{setpoint} - H = 10 - 10.324 = -0.324$$

$$e_{\text{total}} = e_t + e_{\text{total}} = -0.324 - 2.824 = -3.148$$

$$D = e_t - e_{\text{pre}} = -0.324 - (-0.968) = 0.644$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) \\ &= 0.4 * (-0.324 - 3.148 / 2 + 0.644) = -0.5 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = -0.5 + 10.324 = 9.824$$

Tại t = 13s:

$$e_t = \text{setpoint} - H = 10 - 9.824 = 0.176$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 0.176 - 3.148 = -2.972$$

$$D = e_t - e_{\text{pre}} = 0.176 - (-0.324) = 0.5$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) \\ &= 0.4 * (0.176 - 2.972 / 2 + 0.5) = -0.324 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = -0.324 + 9.824 = 9.5$$

Tại t = 14s:

$$e_t = \text{setpoint} - H = 10 - 9.5 = 0.5$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 0.5 - 2.972 = -2.472$$

$$D = e_t - e_{\text{pre}} = 0.5 - 0.176 = 0.324$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) \\ &= 0.4 * (0.5 - 2.472 / 2 + 0.324) = -0.165 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = -0.165 + 9.5 = 9.335$$

Tại t = 15s:

$$e_t = \text{setpoint} - H = 10 - 9.335 = 0.665$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 0.665 - 2.472 = -1.807$$

$$D = e_t - e_{\text{pre}} = 0.665 - 0.5 = 0.165$$

$$U_t = K_p * (e_t + e_{\text{total}} / T_i + T_d * D)$$

$$= 0.4 * (0.665 - 1.807 / 2 + 0.165) = -0.029$$

$$\Rightarrow H = U_t + h_{\text{pre}} = -0.029 + 9.335 = 9.306$$

Tại t = 16s:

$$e_t = \text{setpoint} - H = 10 - 9.306 = 0.694$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 0.694 - 1.807 = -1.113$$

$$D = e_t - e_{\text{pre}} = 0.694 - 0.665 = 0.029$$

$$U_t = K_p * (e_t + e_{\text{total}} / T_i + T_d * D)$$

$$= 0.4 * (0.694 - 1.113 / 2 + 0.029) = 0.067$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0.067 + 9.306 = 9.373$$

Tại t = 17s:

$$e_t = \text{setpoint} - H = 10 - 9.373 = 0.627$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 0.627 - 1.113 = -0.486$$

$$D = e_t - e_{\text{pre}} = 0.627 - 0.694 = -0.067$$

$$U_t = K_p * (e_t + e_{\text{total}} / T_i + T_d * D)$$

$$= 0.4 * (0.627 - 0.486 / 2 - 0.067) = 0.127$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0.127 + 9.373 = 9.5$$

Tại t = 18s:

$$\begin{aligned}e_t &= \text{setpoint} - H = 10 - 9.5 = 0.5 \\e_{\text{total}} &= e_t + e_{\text{total}} = 0.5 - 0.486 = 0.014 \\D &= e_t - e_{\text{pre}} = 0.5 - 0.627 = -0.127\end{aligned}$$

$$\begin{aligned}U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) \\&= 0.4 * (0.5 + 0.014 / 2 - 0.127) = 0.152\end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0.152 + 9.5 = 9.652$$

Tại t = 19s:

$$\begin{aligned}e_t &= \text{setpoint} - H = 10 - 9.652 = 0.348 \\e_{\text{total}} &= e_t + e_{\text{total}} = 0.348 + 0.014 = 0.362 \\D &= e_t - e_{\text{pre}} = 0.348 - 0.5 = -0.152\end{aligned}$$

$$\begin{aligned}U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) \\&= 0.4 * (0.348 + 0.362 / 2 - 0.152) = 0.151\end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0.151 + 9.652 = 9.8$$

Tại t = 20s:

$$\begin{aligned}e_t &= \text{setpoint} - H = 10 - 9.8 = 0.2 \\e_{\text{total}} &= e_t + e_{\text{total}} = 0.2 + 0.362 = 0.562 \\D &= e_t - e_{\text{pre}} = 0.2 - 0.348 = -0.148\end{aligned}$$

$$\begin{aligned}U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) \\&= 0.4 * (0.2 + 0.562 / 2 - 0.148) = 0.133\end{aligned}$$

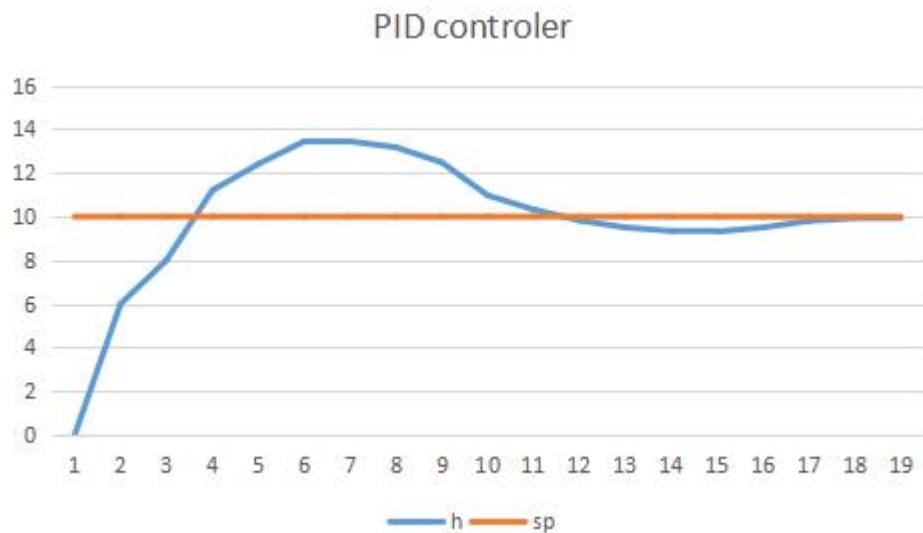
$$\Rightarrow H = U_t + h_{\text{pre}} = 0.133 + 9.8 = 9.933$$

**ĐỒ ÁN TỔNG HỢP**  
**Trang - 79**

---

T(s)	0	1	2	3	4	5	6	7	8	9	10
H	0	6	8	11.2	12.4	<u>13.44</u>	<u>13.44</u>	13.17	12.48	11.74	10.97
T(s)	11	12	13	14	15	16	17	18	19	20	21
H	10.32	9.82	9.5	9.34	9.31	9.37	9.5	9.65	9.8	9.93	10

Bảng 4.4.4.1. Giá trị H của bộ PID theo thời gian t(s)



Hình 4.4.4.1. Đồ thị mô phỏng bộ PID(không nhiễu)

Định: 13.44

Thời gian xác lập: 21(s)

\**Trường hợp có nhiễu*

**setpoint = 10, sensor = 0**

**Chọn Kp = 0.4, Ti = 2, Td = 1**

**Giả sử Uout = 1.2**

Tại t = 0s

$$H = 0$$

$$e_{\text{total}} = 0$$

Tại t = 1s:

$$e_t = \text{setpoint} - H = 10 - 0 = 10$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 10 + 0 = 10$$

$$D = e_t - e_{\text{pre}} = 10 - 10 = 0$$

$$U_t = K_p * (e_t + e_{\text{total}} / T_i + T_d * D) - U_{\text{out}}$$

$$= 0.4 * (10 + 10/2 + 0) - 1.2 = 4.8$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 4.8 + 0 = 4.8$$

Tại t = 2s:

$$e_t = \text{setpoint} - H = 10 - 4.8 = 5.2$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 5.2 + 10 = 15.2$$

$$D = e_t - e_{\text{pre}} = 5.2 - 10 = -4.8$$

$$U_t = K_p * (e_t + e_{\text{total}} / T_i + T_d * D) - U_{\text{out}}$$

$$= 0.4 * (5.2 + 15.2/2 - 4.8) - 1.2 = 2$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 2 + 4.8 = 6.8$$

Tại t = 3s:

$$\begin{aligned}e_t &= \text{setpoint} - H = 10 - 6.8 = 3.2 \\e_{\text{total}} &= e_t + e_{\text{total}} = 3.2 + 15.2 = 18.4 \\D &= e_t - e_{\text{pre}} = 3.2 - 5.2 = -2\end{aligned}$$

$$\begin{aligned}U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) - U_{\text{out}} \\&= 0.4 * (3.2 + 18.4 / 2 - 2) - 1.2 = 2.96 \\&\Rightarrow H = U_t + h_{\text{pre}} = 2.96 + 6.8 = 9.76\end{aligned}$$

Tại t = 4s:

$$\begin{aligned}e_t &= \text{setpoint} - H = 10 - 9.76 = 0.24 \\e_{\text{total}} &= e_t + e_{\text{total}} = 0.24 + 18.4 = 18.64 \\D &= e_t - e_{\text{pre}} = 0.24 - 3.2 = -2.96\end{aligned}$$

$$\begin{aligned}U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) - U_{\text{out}} \\&= 0.4 * (0.24 + 18.64 / 2 - 2.96) - 1.2 = 1.44 \\&\Rightarrow H = U_t + h_{\text{pre}} = 1.44 + 9.76 = 11.2\end{aligned}$$

Tại t = 5s:

$$\begin{aligned}e_t &= \text{setpoint} - H = 10 - 11.2 = -1.2 \\e_{\text{total}} &= e_t + e_{\text{total}} = -1.2 + 18.64 = 17.44 \\D &= e_t - e_{\text{pre}} = -1.2 - 0.24 = -1.44\end{aligned}$$

$$\begin{aligned}U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) - U_{\text{out}} \\&= 0.4 * (-1.2 + 17.44 / 2 - 1.44) - 1.2 = 1.232 \\&\Rightarrow H = U_t + h_{\text{pre}} = 1.232 + 11.2 = 12.432\end{aligned}$$

Tại t = 6s:

$$\begin{aligned}e_t &= \text{setpoint} - H = 10 - 12.432 = -2.432 \\e_{\text{total}} &= e_t + e_{\text{total}} = -2.432 + 17.44 = 15.008 \\D &= e_t - e_{\text{pre}} = -2.432 - (-1.2) = -1.232\end{aligned}$$

$$\begin{aligned}U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) - U_{\text{out}} \\&= 0.4 * (-2.432 + 15.008 / 2 - 1.232) - 1.2 = 0.336 \\&\Rightarrow H = U_t + h_{\text{pre}} = 0.336 + 12.432 = 12.768\end{aligned}$$

Tại t = 7s:

$$\begin{aligned}e_t &= \text{setpoint} - H = 10 - 12.768 = -2.768 \\e_{\text{total}} &= e_t + e_{\text{total}} = -2.768 + 15.008 = 12.24 \\D &= e_t - e_{\text{pre}} = -2.768 - (-2.432) = -0.336\end{aligned}$$

$$\begin{aligned}U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) - U_{\text{out}} \\&= 0.4 * (-2.768 + 12.24 / 2 - 0.336) - 1.2 = 0.006 \\&\Rightarrow H = U_t + h_{\text{pre}} = 0.006 + 12.768 = 12.774\end{aligned}$$

Tại t = 8s:

$$\begin{aligned}e_t &= \text{setpoint} - H = 10 - 12.774 = -2.774 \\e_{\text{total}} &= e_t + e_{\text{total}} = -2.774 + 12.24 = 9.466 \\D &= e_t - e_{\text{pre}} = -2.774 - (-2.768) = -0.006\end{aligned}$$

$$\begin{aligned}U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) - U_{\text{out}} \\&= 0.4 * (-2.774 + 9.466 / 2 - 0.006) - 1.2 = -0.419 \\&\Rightarrow H = U_t + h_{\text{pre}} = -0.419 + 12.774 = 12.355\end{aligned}$$

Tại t = 9s:

$$e_t = \text{setpoint} - H = 10 - 12.355 = -2.355$$

$$e_{\text{total}} = e_t + e_{\text{total}} = -2.355 + 9.466 = 7.111$$

$$D = e_t - e_{\text{pre}} = -2.355 - (-2.774) = 0.419$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) - U_{\text{out}} \\ &= 0.4 * (-2.355 + 7.111 / 2 + 0.419) - 1.2 = -0.552 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = -0.552 + 12.355 = 11.803$$

Tại t = 10s:

$$e_t = \text{setpoint} - H = 10 - 11.803 = -1.803$$

$$e_{\text{total}} = e_t + e_{\text{total}} = -1.803 + 7.111 = 5.308$$

$$D = e_t - e_{\text{pre}} = -1.803 - (-2.355) = 0.552$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) - U_{\text{out}} \\ &= 0.4 * (-1.803 + 5.308 / 2 + 0.552) - 1.2 = -0.639 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = -0.639 + 11.803 = 11.164$$

Tại t = 11s:

$$e_t = \text{setpoint} - H = 10 - 11.164 = -1.164$$

$$e_{\text{total}} = e_t + e_{\text{total}} = -1.164 + 5.308 = 4.144$$

$$D = e_t - e_{\text{pre}} = -1.164 - (-1.803) = 0.639$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) - U_{\text{out}} \\ &= 0.4 * (-1.164 + 4.144 / 2 + 0.639) - 1.2 = -0.581 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = -0.581 + 11.164 = 10.583$$

Tại t = 12s:

$$e_t = \text{setpoint} - H = 10 - 10.583 = -0.583$$

$$e_{\text{total}} = e_t + e_{\text{total}} = -0.583 + 4.144 = 3.561$$

$$D = e_t - e_{\text{pre}} = -0.583 - (-1.164) = 0.581$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) - U_{\text{out}} \\ &= 0.4 * (-0.583 + 3.561 / 2 + 0.581) - 1.2 = -0.489 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = \mathbf{-0.489 + 10.583 = 10.094}$$

Tại t = 13s:

$$e_t = \text{setpoint} - H = 10 - 10.094 = -0.094$$

$$e_{\text{total}} = e_t + e_{\text{total}} = -0.094 + 3.561 = 3.467$$

$$D = e_t - e_{\text{pre}} = -0.094 - (-0.583) = 0.489$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) - U_{\text{out}} \\ &= 0.4 * (-0.094 + 3.467 / 2 + 0.489) - 1.2 = -0.349 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = \mathbf{-0.349 + 10.094 = 9.745}$$

Tại t = 14s:

$$e_t = \text{setpoint} - H = 10 - 9.745 = 0.255$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 0.255 + 3.467 = 3.722$$

$$D = e_t - e_{\text{pre}} = 0.255 - (-0.094) = 0.349$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) - U_{\text{out}} \\ &= 0.4 * (0.255 + 3.722 / 2 + 0.349) - 1.2 = -0.214 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = \mathbf{-0.214 + 9.745 = 9.531}$$

Tại t = 15s:

$$e_t = \text{setpoint} - H = 10 - 9.531 = 0.469$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 0.469 + 3.772 = 4.241$$

$$D = e_t - e_{\text{pre}} = 0.469 - 0.255 = 0.214$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) - U_{\text{out}} \\ &= 0.4 * (0.469 + 4.241 / 2 + 0.214) - 1.2 = -0.079 \\ \Rightarrow H &= U_t + h_{\text{pre}} = -0.079 + 9.531 = 9.452 \end{aligned}$$

Tại t = 16s:

$$e_t = \text{setpoint} - H = 10 - 9.452 = 0.548$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 0.548 + 4.241 = 4.789$$

$$D = e_t - e_{\text{pre}} = 0.548 - 0.469 = 0.079$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) - U_{\text{out}} \\ &= 0.4 * (0.548 + 4.789 / 2 + 0.079) - 1.2 = 0.008 \\ \Rightarrow H &= U_t + h_{\text{pre}} = 0.008 + 9.452 = 9.46 \end{aligned}$$

Tại t = 17s:

$$e_t = \text{setpoint} - H = 10 - 9.46 = 0.54$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 0.54 + 4.789 = 5.329$$

$$D = e_t - e_{\text{pre}} = 0.54 - 0.548 = -0.008$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) - U_{\text{out}} \\ &= 0.4 * (0.54 + 5.329 / 2 - 0.008) - 1.2 = 0.079 \\ \Rightarrow H &= U_t + h_{\text{pre}} = 0.079 + 9.46 = 9.539 \end{aligned}$$

Tại t = 18s:

$$e_t = \text{setpoint} - H = 10 - 9.539 = 0.461$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 0.461 + 5.329 = 5.97$$

$$D = e_t - e_{\text{pre}} = 0.461 - 0.54 = -0.079$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) - U_{\text{out}} \\ &= 0.4 * (0.461 + 5.97 / 2 - 0.079) - 1.2 = 0.147 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0.147 + 9.539 = 9.686$$

Tại t = 19:

$$e_t = \text{setpoint} - H = 10 - 9.686 = 0.314$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 0.314 + 5.97 = 6.284$$

$$D = e_t - e_{\text{pre}} = 0.314 - 0.461 = -0.147$$

$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) - U_{\text{out}} \\ &= 0.4 * (0.314 + 6.284 / 2 - 0.147) - 1.2 = 0.124 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0.124 + 9.686 = 9.81$$

Tại t = 20:

$$e_t = \text{setpoint} - H = 10 - 9.81 = 0.19$$

$$e_{\text{total}} = e_t + e_{\text{total}} = 0.19 + 6.284 = 6.474$$

$$D = e_t - e_{\text{pre}} = 0.19 - 0.314 = -0.124$$

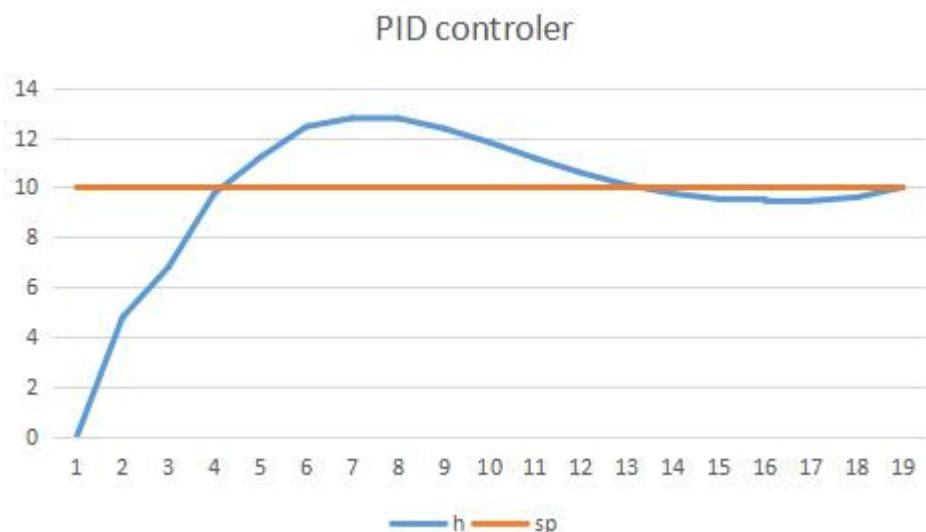
$$\begin{aligned} U_t &= K_p * (e_t + e_{\text{total}} / T_i + T_d * D) - U_{\text{out}} \\ &= 0.4 * (0.19 + 6.474 / 2 - 0.124) - 1.2 = 0.121 \end{aligned}$$

$$\Rightarrow H = U_t + h_{\text{pre}} = 0.121 + 9.81 = 9.931$$

**ĐỒ ÁN TỔNG HỢP**  
**Trang - 87**

T(s)	0	1	2	3	4	5	6	7	8	9	10
H	0	4.8	6.8	9.76	11.2	12.43	<u>12.77</u>	<u>12.77</u>	12.36	11.8	11.16
T(s)	11	12	13	14	15	16	17	18	19	20	21
H	10.58	10.09	9.75	9.53	9.45	9.46	9.54	9.69	9.81	9.93	9.99

**Bảng 4.4.4.2. Giá trị H của bộ PID theo thời gian t(s) (có nhiễu)**



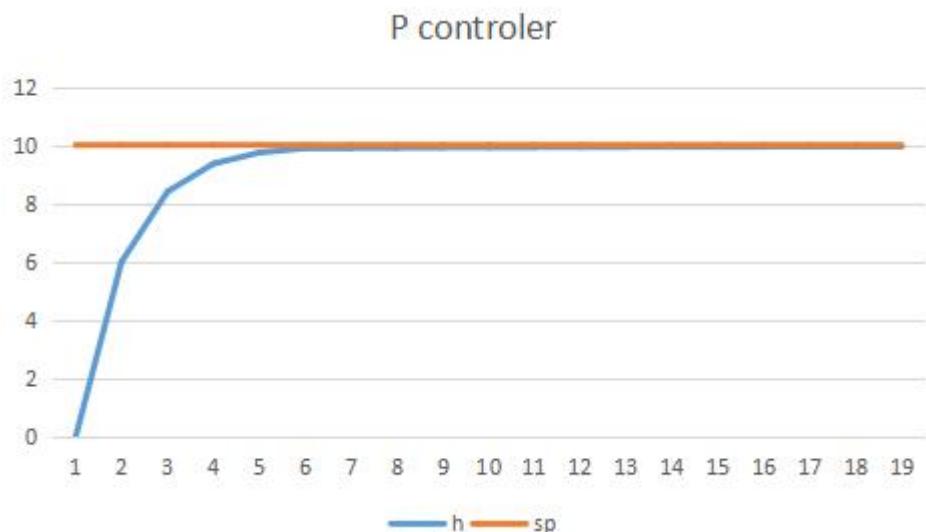
**Hình 4.4.4.2. Đồ thị mô phỏng bộ PID (có nhiễu)**

Đỉnh: 12.77

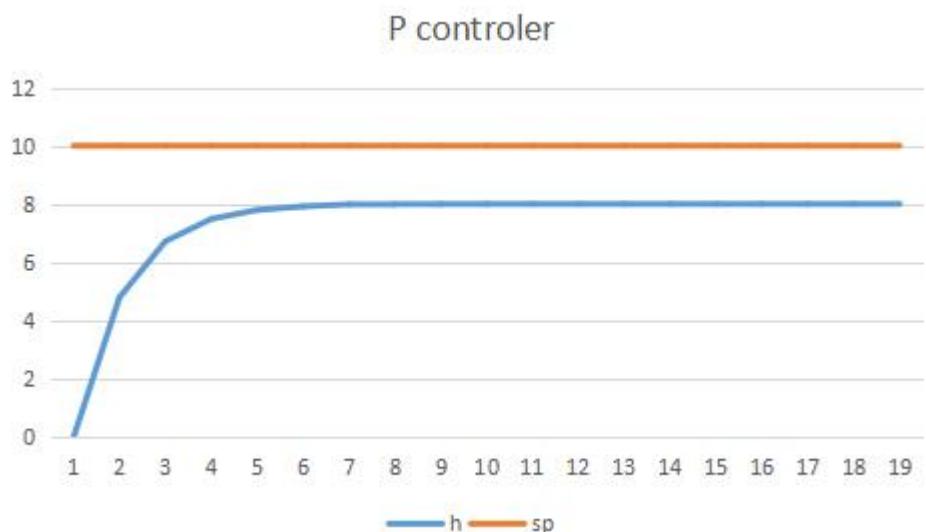
Thời gian xác lập: 21(s)

#### 4.4.5. Nhận xét

Bộ P:



Hình 4.4.5.1. Đồ thị mô phỏng bộ P (không nhiễu)

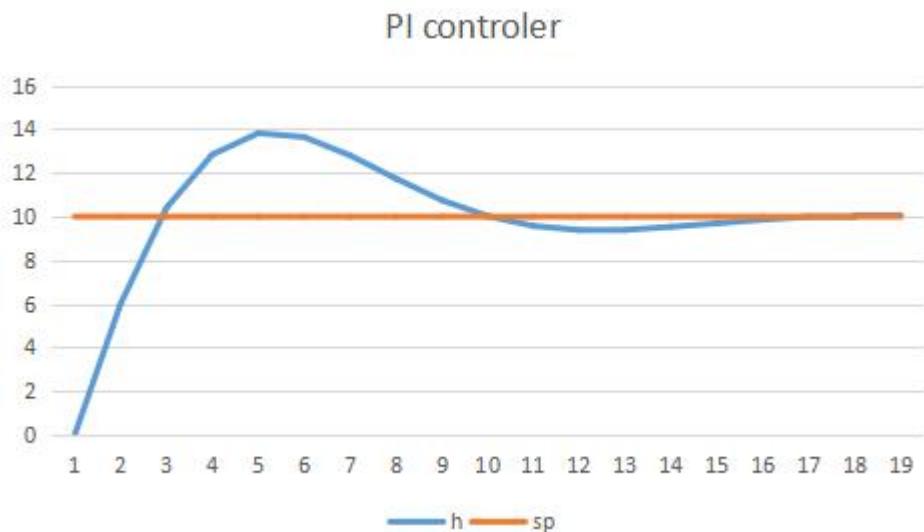


Hình 4.4.5.2. Đồ thị mô phỏng bộ P (có nhiễu)

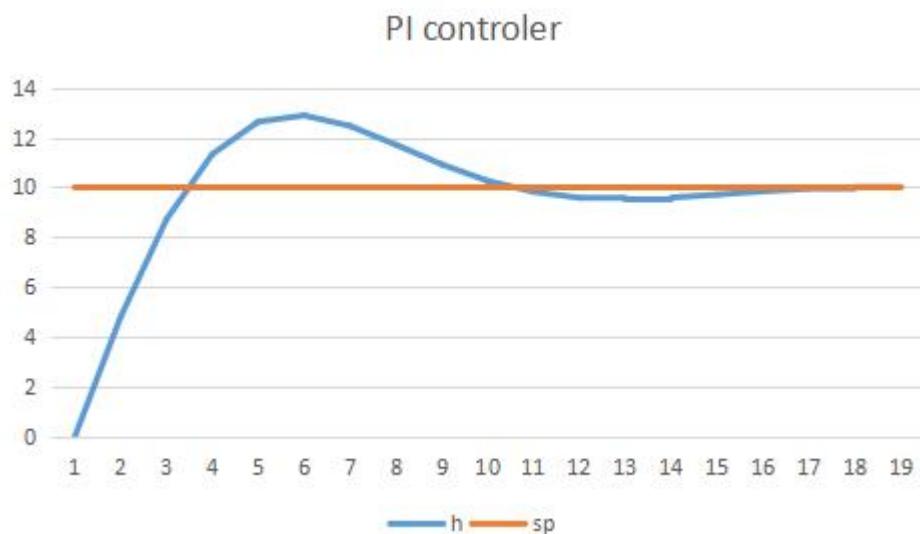
Vọt lố: không có vọt lố

Xác lập: xác lập khi không có nhiễu , nếu có nhiễu thì không xác lập được

Bộ PI:



Hình 4.4.5.3. Đồ thị mô phỏng bộ PI (không nhiễu)



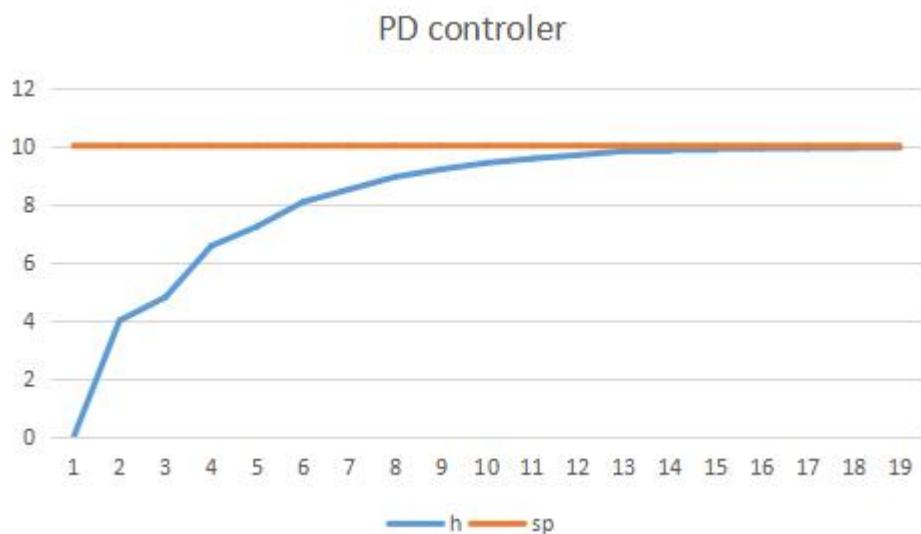
Hình 4.4.5.4. Đồ thị mô phỏng bộ PI (có nhiễu)

Vọt lồ:  $13.82 - 10 = 3.82$  (không nhiễu)

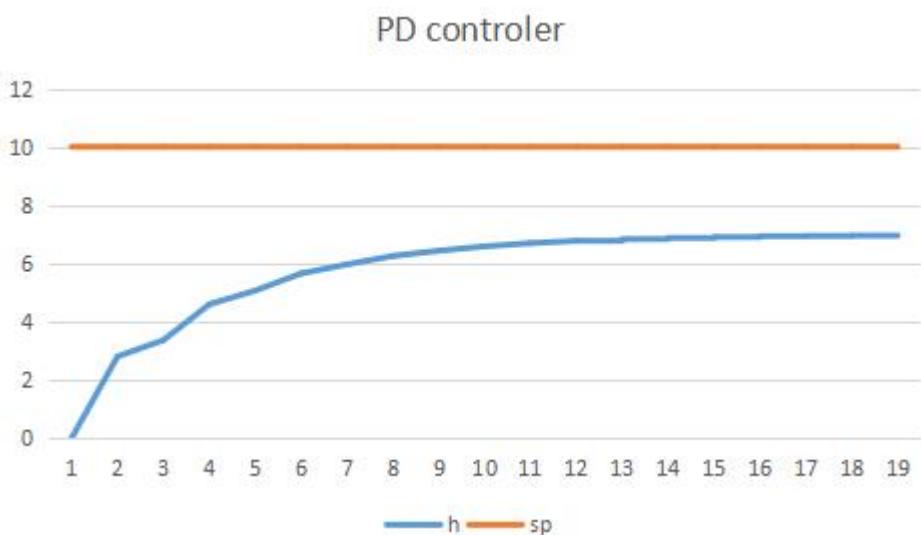
$12.88 - 10 = 2.88$  (có nhiễu)

Xác lập: 18(s), cả 2 trường hợp không nhiễu hay có nhiễu đều xác lập được

Bộ PD:



Hình 4.4.5.5. Đồ thị mô phỏng bộ PD (không nhiễu)

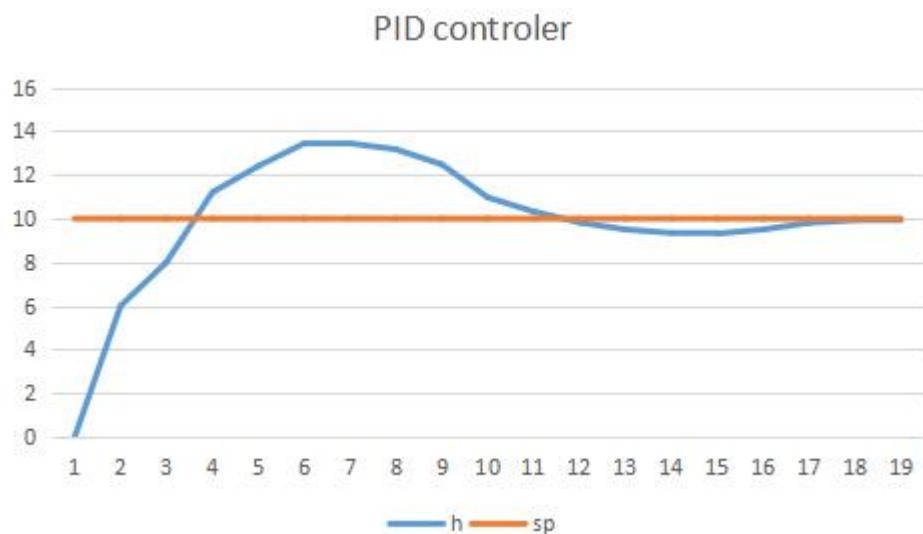


Hình 4.4.5.6. Đồ thị mô phỏng bộ PD (có nhiễu)

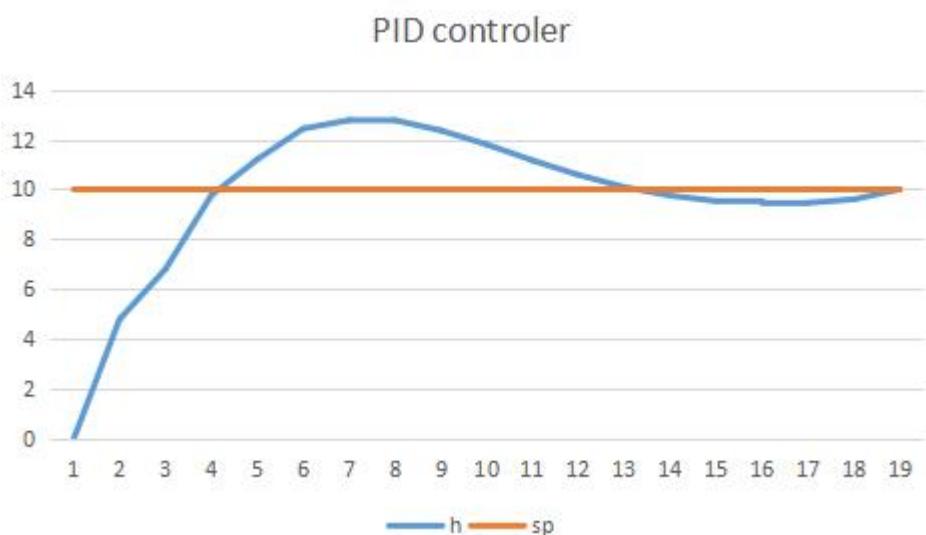
Vọt lố: không có vọt lố

Xác lập: xác lập khi không có nhiễu , nếu có nhiễu thì không xác lập được

Bộ PID:



Hình 4.4.5.7. Đồ thị mô phỏng bộ PID (không nhiễu)



Hình 4.4.5.8. Đồ thị mô phỏng bộ PID (có nhiễu)

Vọt lô:  $13.44 - 10 = 3.44$  (không nhiễu)

$12.77 - 10 = 2.77$  (có nhiễu)

Xác lập: 21(s), cả 2 trường hợp không nhiễu hay có nhiễu đều xác lập được

**\*Nhận xét:**

- Đối với bộ điều khiển P, PD ta thấy rằng bộ cả 2 bộ điều khiển này có thể tiệm cận đến giá trị mà ta mong muốn khi không có nhiễu, không đạt được giá trị mong muốn khi có nhiễu, không có vọt lô.
  - Đối với bộ điều khiển PID thì sẽ có **độ vọt lô thấp hơn** bộ điều khiển PI nhưng đổi lại thì PID sẽ **tăng thời gian xác lập** của hệ thống.
  - Cả 2 bộ điều khiển PI và PID điều có thể đạt được giá trị mong muốn trong cả 2 trường hợp hệ thống không nhiễu và hệ thống có nhiễu.
- \*Kết luận:** tùy vào yêu cầu của từng hệ thống như: mức vọt lô cho phép, thời gian đáp ứng nhanh hay chậm, hệ thống có nhiễu động hay không mà ta sẽ chọn một trong các bộ điều khiển trên sao cho phù hợp nhất.

## CHƯƠNG 5: THỰC NGHIỆM

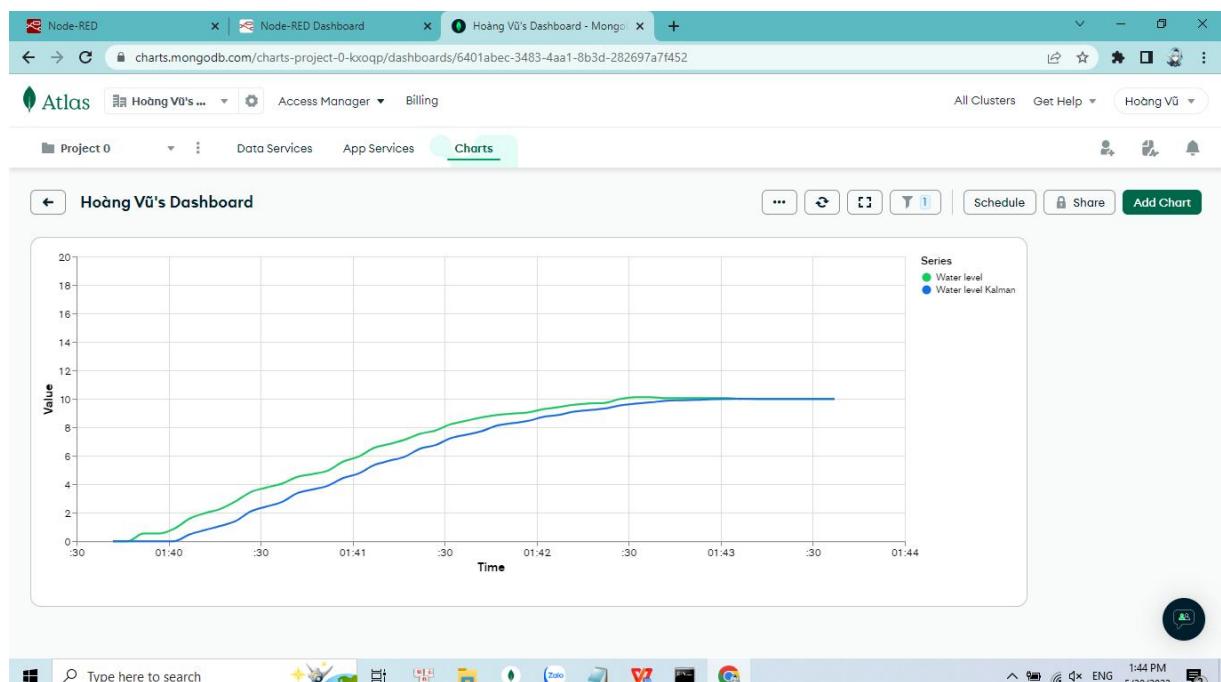
### 5.1. Bộ điều khiển P

Setpoint = 10, K<sub>p</sub> = 0.6

*Không nhiễu:*

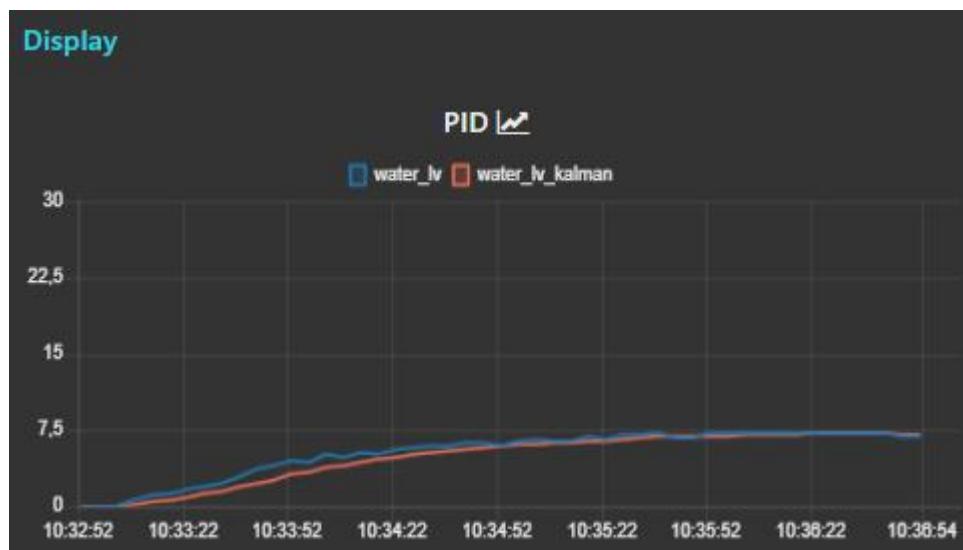


Hình 5.1.1. Đồ thị mục chất lỏng theo thời gian của bộ điều khiển P trên Node-RED(không nhiễu)

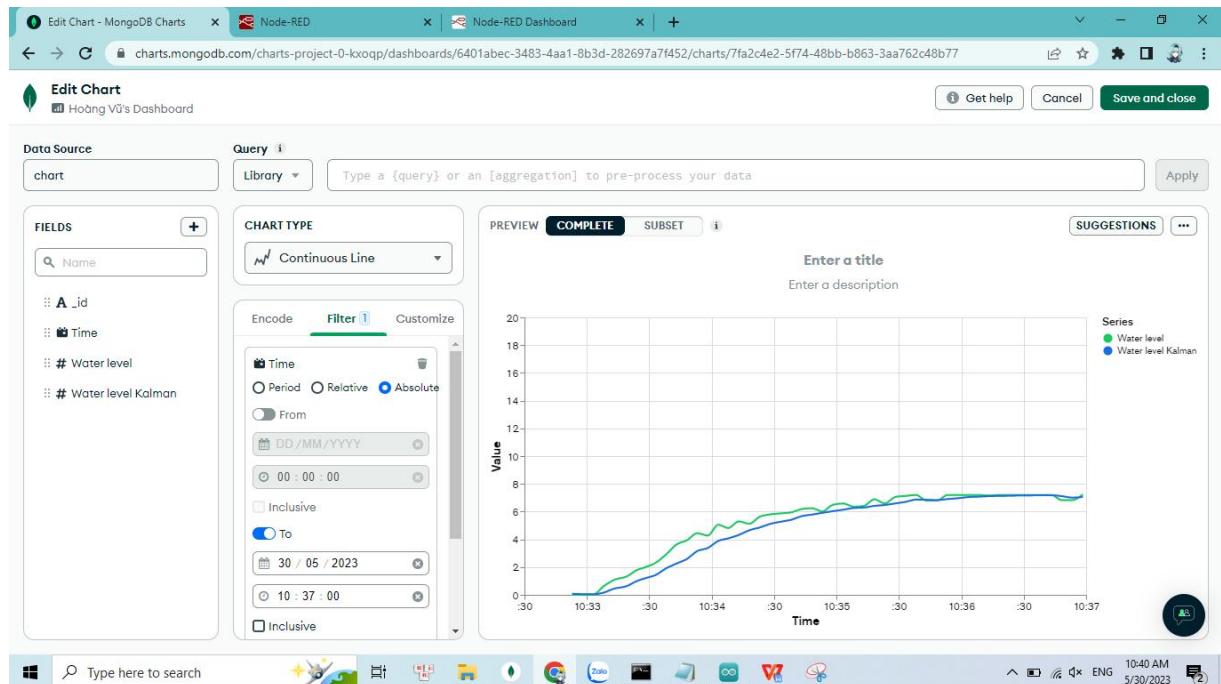


Hình 5.1.2. Đồ thị mục chất lỏng theo thời gian của bộ điều khiển P trên MongoDB Chart(không nhiễu)

**Có nhiều:**



**Hình 5.1.3. Đồ thị mực chất lỏng theo thời gian của bộ điều khiển P trên Node-RED(có nhiều)**

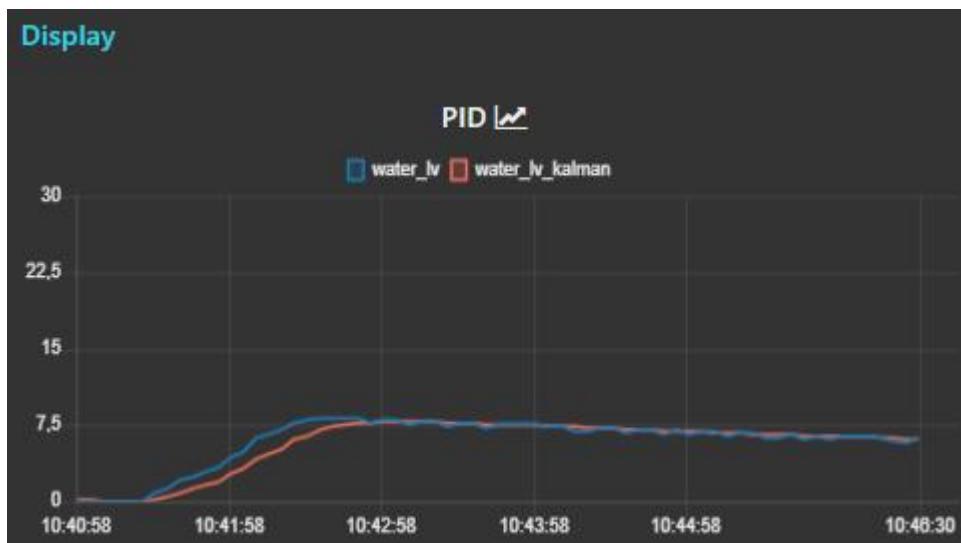


**Hình 5.1.4. Đồ thị mực chất lỏng theo thời gian của bộ điều khiển P trên MongoDB Chart(có nhiều)**

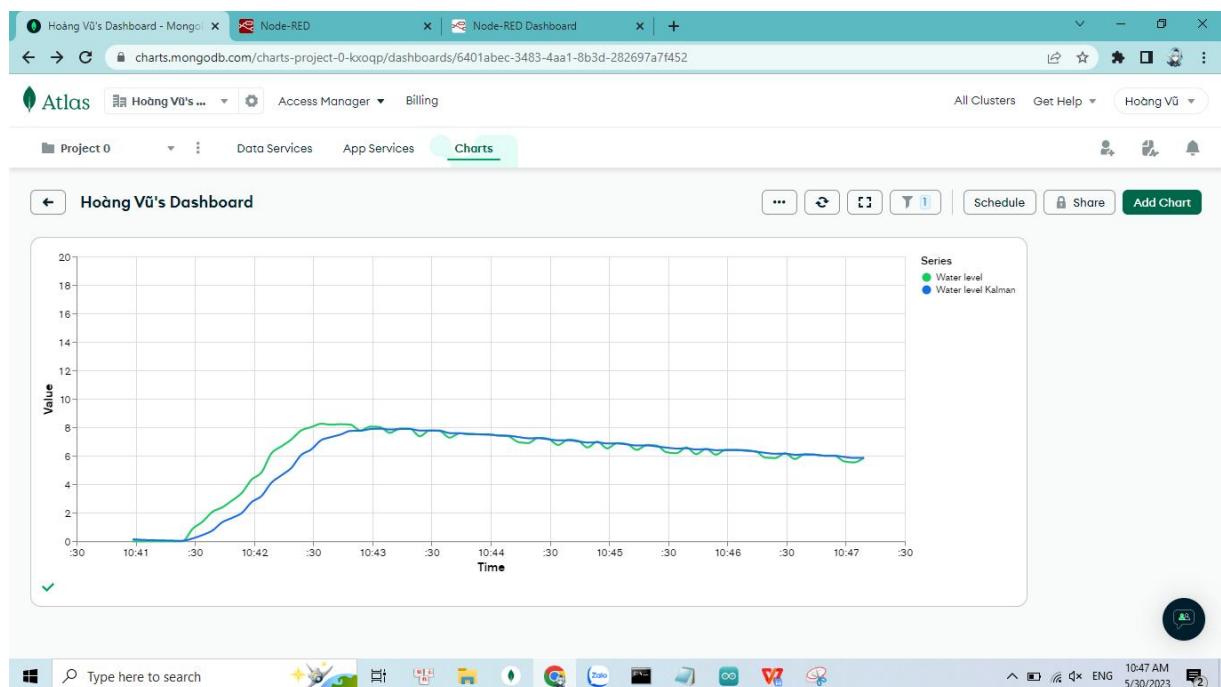
## 5.2. Bộ điều khiển PI

Setpoint = 6, K<sub>P</sub> = 0.25, T<sub>I</sub> = 2

*Không nhiễu:*

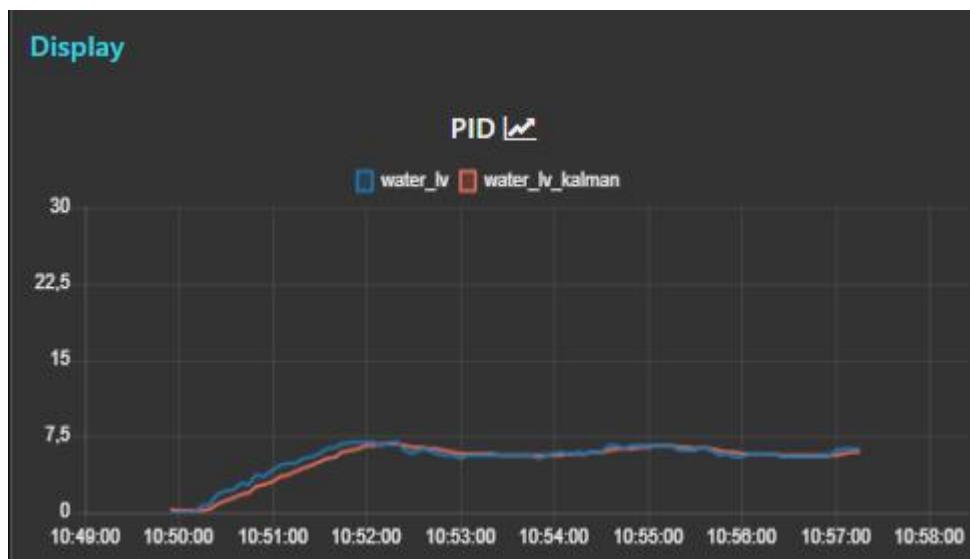


Hình 5.2.1. Đồ thị mực chất lỏng theo thời gian của bộ điều khiển PI trên Node-RED(không nhiễu)

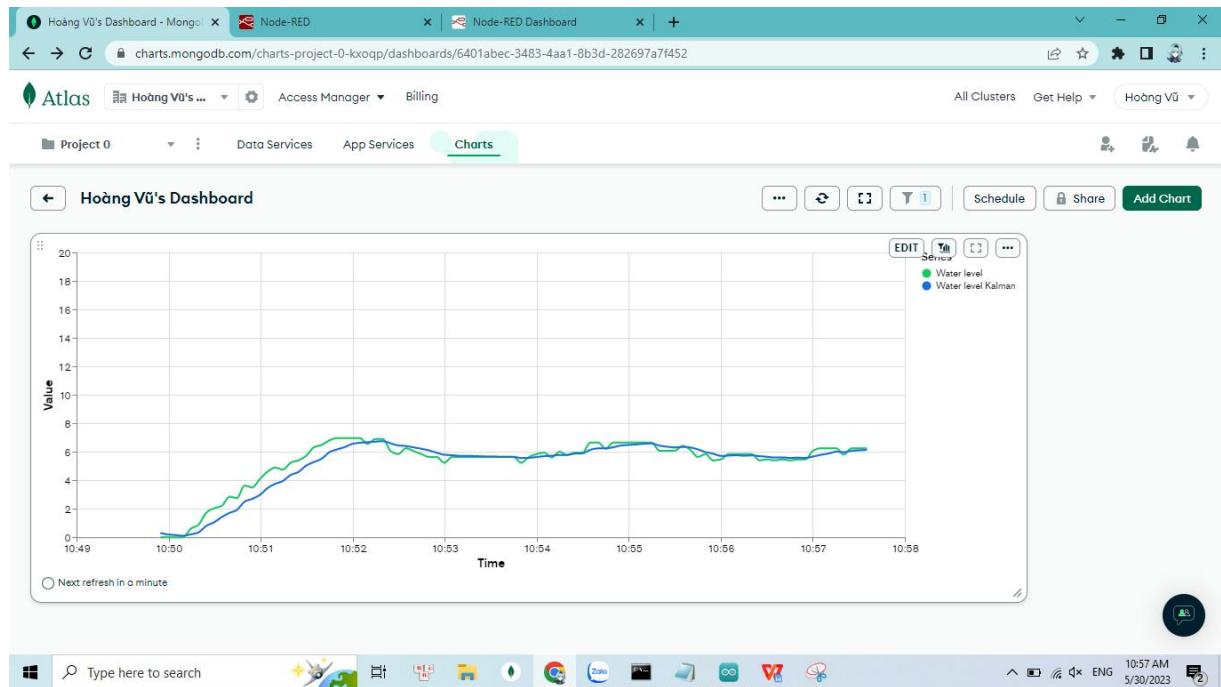


Hình 5.2.2. Đồ thị mực chất lỏng theo thời gian của bộ điều khiển PI trên MongoDB Chart(không nhiễu)

**Có nhiễu:**



**Hình 5.2.3. Đồ thị mực chất lỏng theo thời gian của bộ điều khiển PI trên Node-RED(có nhiễu)**

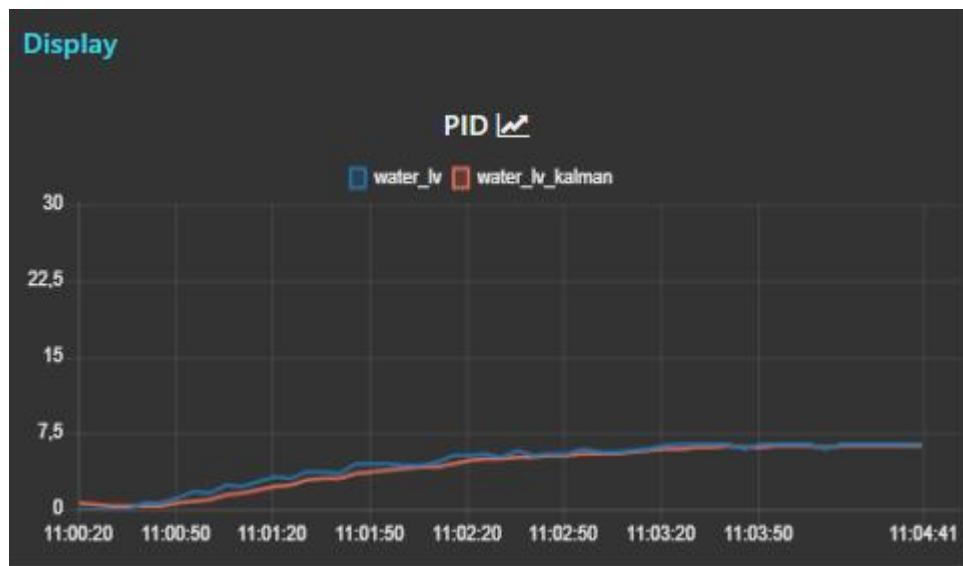


**Hình 5.2.1. Đồ thị mực chất lỏng theo thời gian của bộ điều khiển PI trên MongoDB Chart(có nhiễu)**

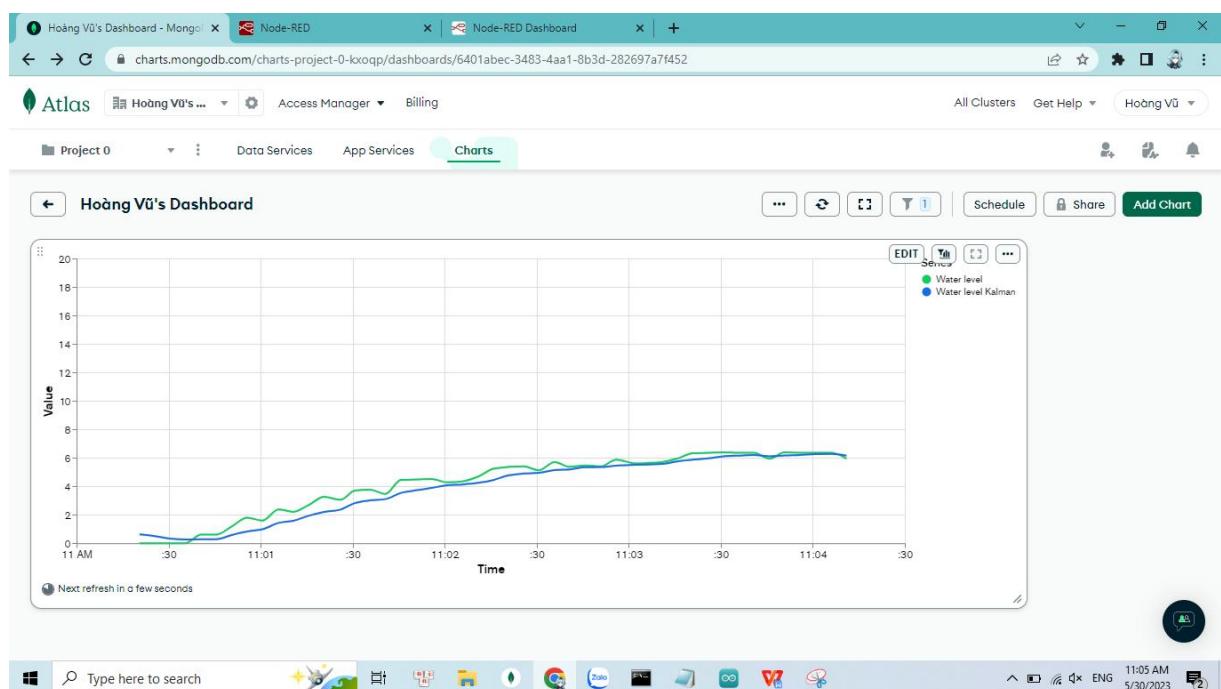
### 5.3. Bộ điều khiển PD

Setpoint = 6, Kp = 0.4, Td = 3

*Không nhiễu:*

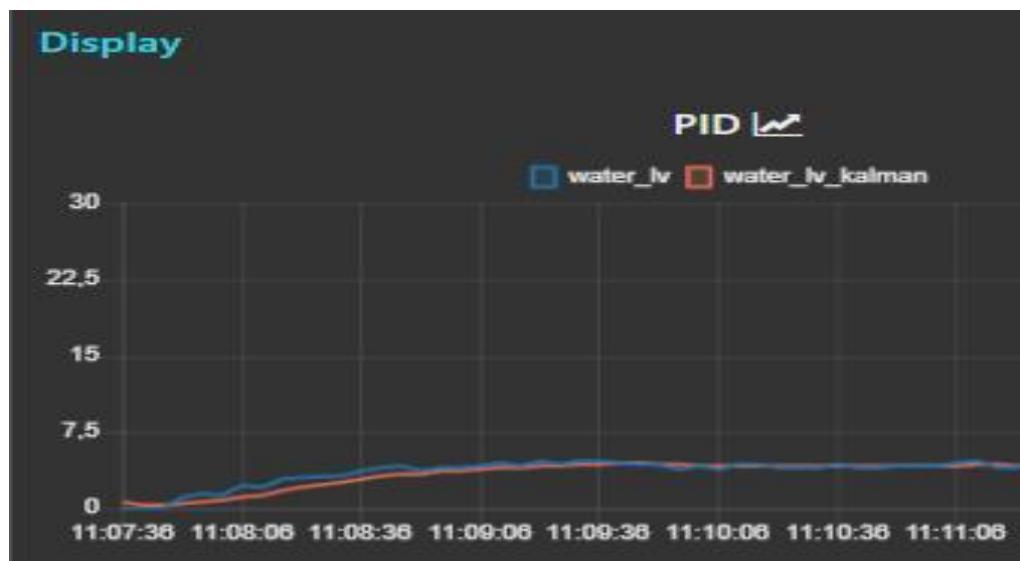


Hình 5.3.1. Đồ thị mực chất lỏng theo thời gian của bộ điều khiển PD trên Node-RED(không nhiễu)

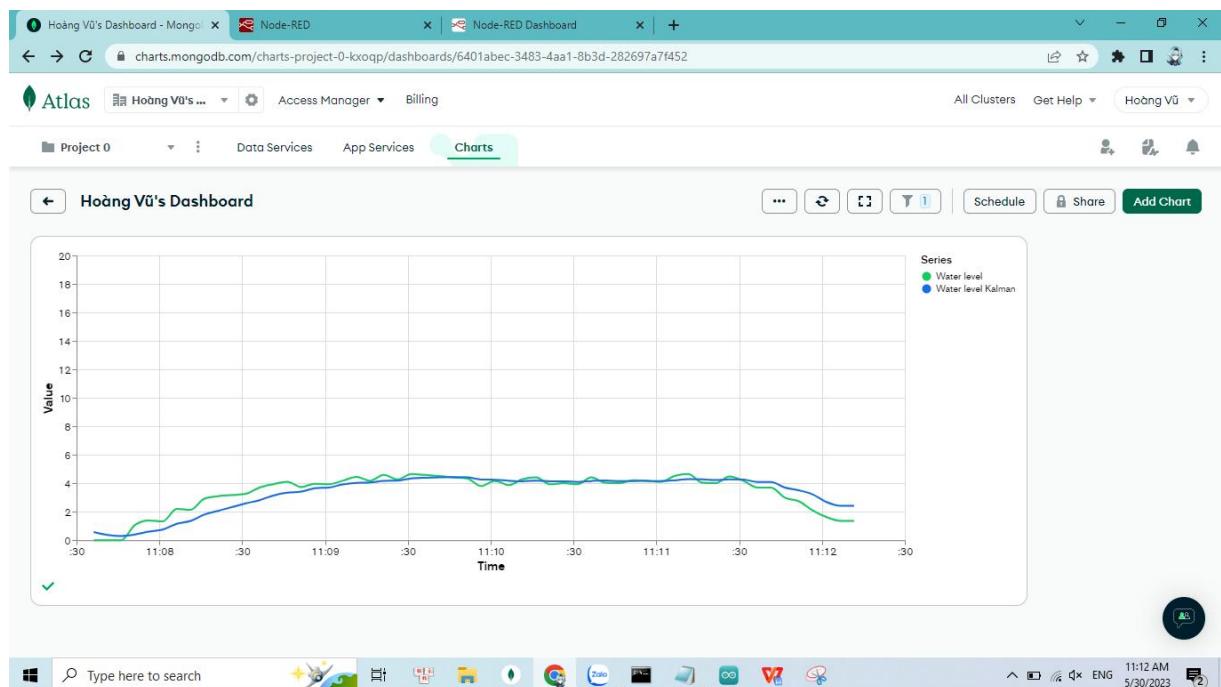


Hình 5.3.2. Đồ thị mực chất lỏng theo thời gian của bộ điều khiển PD trên MongoDB Chart(không nhiễu)

Có nhiễu:



Hình 5.3.3. Đồ thị mục chất lỏng theo thời gian của bộ điều khiển PD trên Node-RED(có nhiễu)

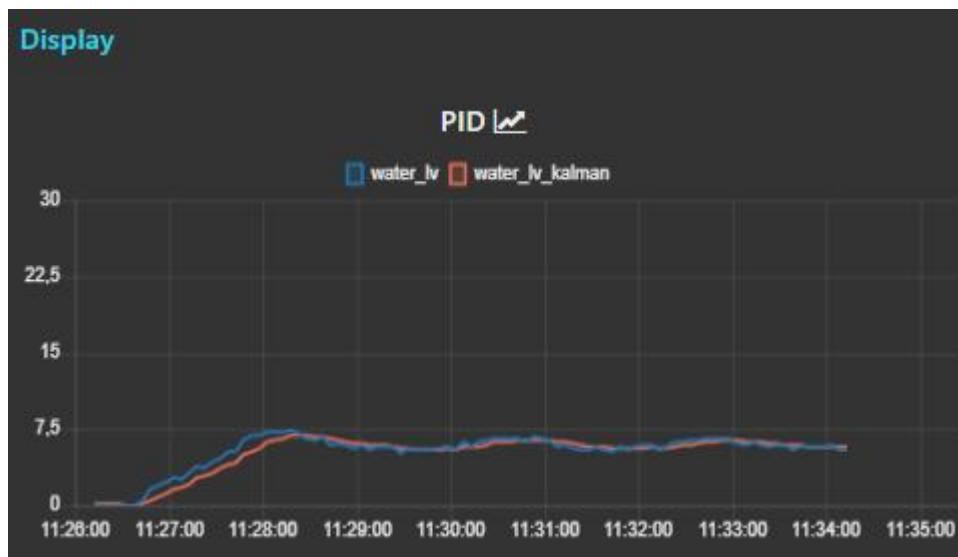


Hình 5.3.4. Đồ thị mục chất lỏng theo thời gian của bộ điều khiển PD trên MongoDB Chart(có nhiễu)

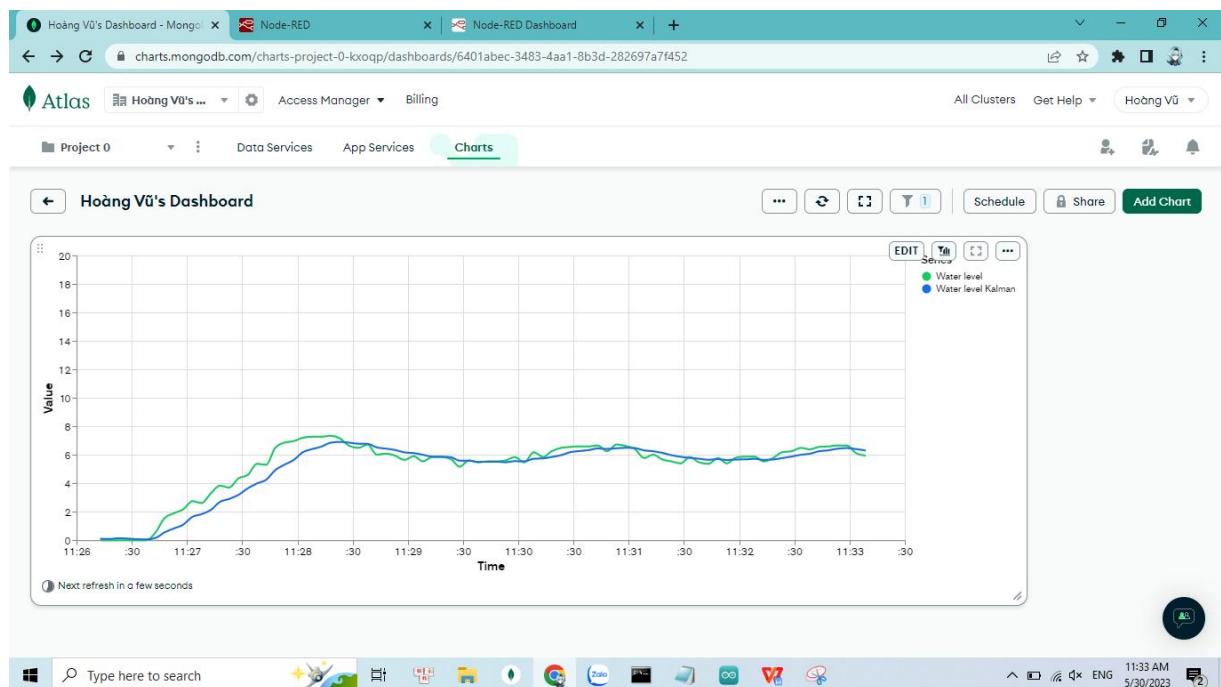
#### 5.4. Bộ điều khiển PID

Setpoint = 6, K<sub>p</sub> = 0.25, T<sub>i</sub> = 2, T<sub>d</sub> = 3

*Không nhiễu:*



Hình 5.4.1. Đồ thị mục chất lỏng theo thời gian của bộ điều khiển PID trên Node-RED(không nhiễu)

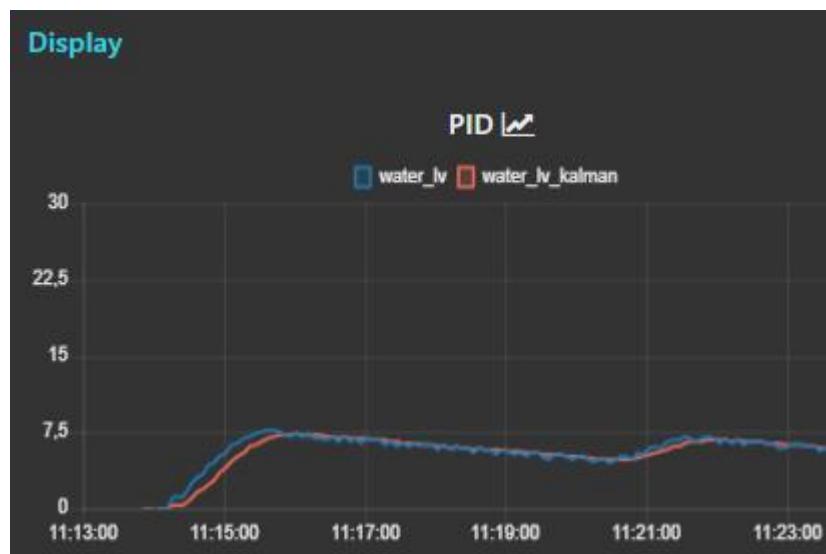


Hình 5.4.2. Đồ thị mục chất lỏng theo thời gian của bộ điều khiển PID trên MongoDB Chart(không nhiễu)

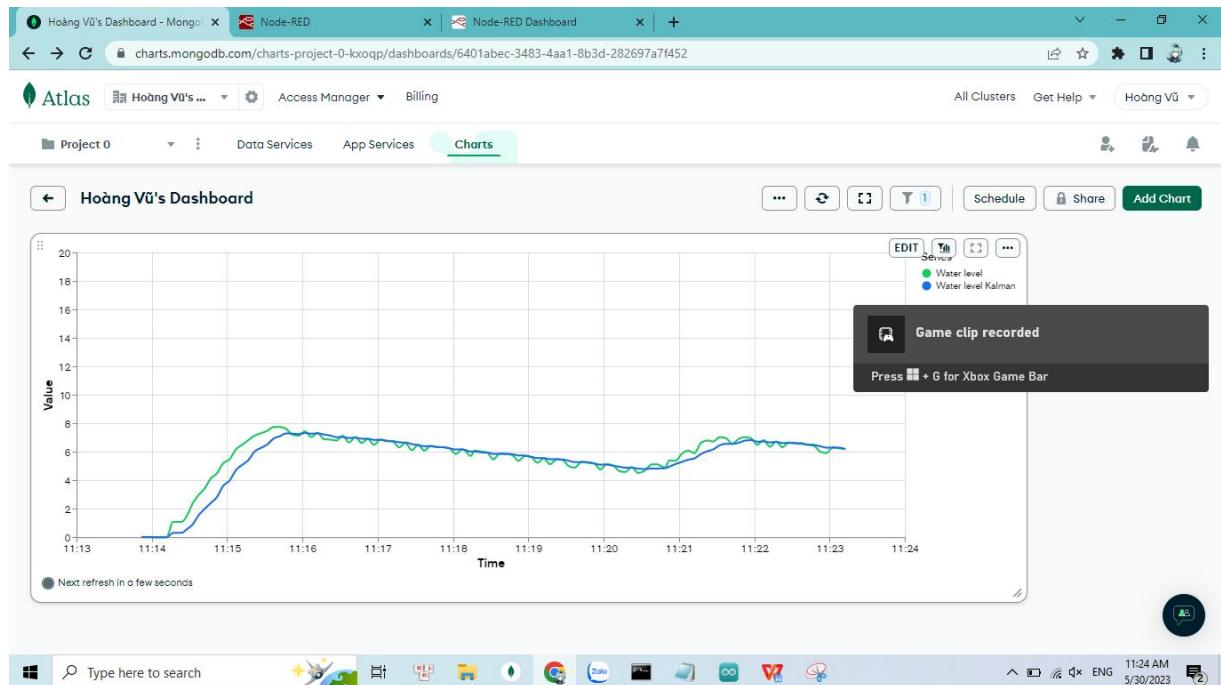
# ĐỒ ÁN TỔNG HỢP

## Trang - 100

Có nhiều:



Hình 5.4.3. Đồ thị mục chất lỏng theo thời gian của bộ điều khiển PID trên Node-RED(có nhiều)



Hình 5.4.4. Đồ thị mục chất lỏng theo thời gian của bộ điều khiển PID trên MongoDB Chart(có nhiều)

### 5.5. Kết luận

$$\begin{aligned} u(t) &= K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \\ &= K_p e(t) + \frac{1}{Ti} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \\ &= K_p(e(t) + \frac{1}{Ti} \int_0^t e(\sigma) d\sigma + T_d \frac{de(t)}{dt}) \end{aligned}$$

Sau quá trình thử nghiệm thì bể chứa vận hành rất tốt, ổn định. Các chế độ điều khiển hoạt động tốt, các số liệu được tính toán chính xác. Tuy nhiên đôi khi hệ thống còn bị nhiễu do ảnh hưởng các tác động bởi môi trường xung quanh.

#### Bộ điều khiển P:

- Khi không bật van xả thì thời gian đáp ứng chậm và đạt được mục chất lỏng mong muốn.
- Khi bật van xả hệ thống bị nhiễu và không đạt được mục chất lỏng mong muốn.
- Kp càng lớn thì tốc độ đáp ứng của hệ thống càng nhanh.

#### Bộ điều khiển PI:

- Khi không bật van xả thì hệ thống đáp ứng nhanh, độ vọt lô cao và đạt được mục chất lỏng mong muốn.
- Khi có bật van xả thì thời gian đáp ứng chậm, nhiễu cao, đạt mục chất lỏng mong muốn, hệ thống ổn định.
  - Do bộ P có giới hạn tồn tại sai số giữa biến quy trình và điểm đặt, nên cần có bộ I bù lại, nhằm loại bỏ sai số để hệ thống ổn định.
  - Kp càng lớn thì sai số xác lập càng nhỏ (nhưng không thể triệt tiêu).
  - Kp càng lớn thì độ vọt lô càng cao.
  - Ti càng lớn thì đáp ứng quá độ càng chậm.
  - Ti càng lớn thì độ vọt lô càng cao.
  - Khâu tích phân không thể sử dụng độc lập mà phải dùng kết hợp với bộ P.

### **Bộ điều khiển PD:**

Khi không bật van xả thì thời gian đáp ứng chậm và đạt được mực chất lỏng mong muốn.

- Khi bật van xả hệ thống bị nhiễu và không đạt được mực chất lỏng mong muốn.
- Td càng lớn thì đáp ứng quá độ càng chậm.
- Td càng lớn thì độ vọt lô càng thấp.
- Khâu vi phân không thể sử dụng độc lập mà phải dùng kết hợp với các bộ P hoặc bộ PI.

### **Kết luận:**

Qua quá trình thực nghiệm và các nhận xét bên trên thì ta thấy là tùy thuộc vào yêu cầu của hệ thống như: có nhiễu hay không, độ vọt lô cho phép, thời gian để hệ thống ổn định,... mà ta sẽ chọn bộ điều khiển cho đơn giản và phù hợp nhất.

### **Vd:**

- Nếu hệ thống không có nhiễu thì chúng ta nên chọn bộ điều khiển P vì nó đáp ứng được yêu cầu của hệ thống và bộ điều khiển P đơn giản, không có vọt lô.
- Nếu hệ thống có nhiễu thì ta sẽ chọn bộ điều khiển PI hoặc PID tùy vào yêu cầu của hệ thống.

Trong trường hợp bồn chứa chất lỏng mà ta đang thực nghiệm thì có 2 bộ điều khiển khả dụng nhất là bộ điều khiển PI và PID, nó có thể đạt được mực chất lỏng mong muốn trong tất cả trường hợp thực nghiệm trên. Bộ PI thì cho ta thời gian xác lập nhanh hơn, PID giảm được độ vọt lô khá đáng kể nhưng bù lại thì thời gian xác lập sẽ lâu hơn bộ PI.

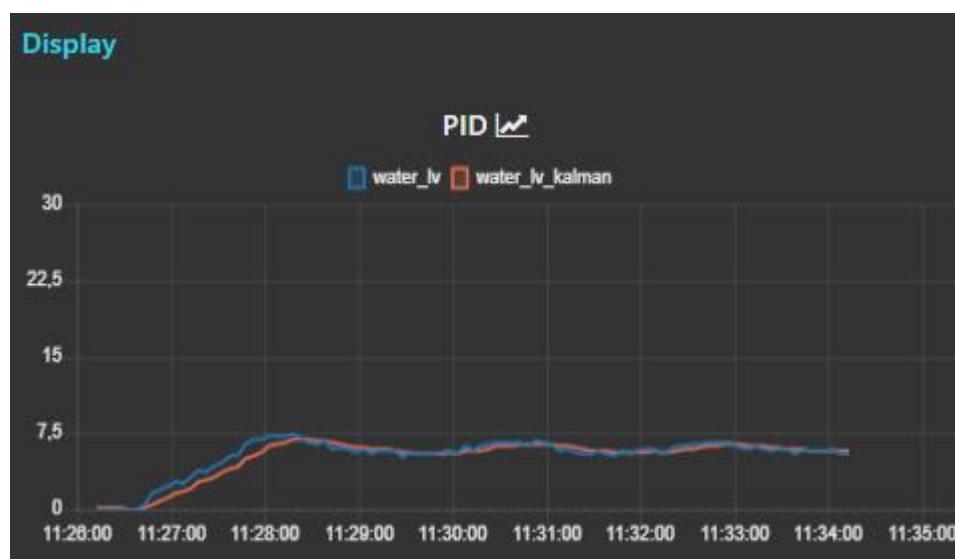
### **Ưu điểm:**

- Hệ thống bể chứa chất lỏng vận hành tương đối nhanh.
- Bộ điều khiển PID hoạt động tương đối chính xác.
- Kết quả giữa lý thuyết, mô phỏng và thực nghiệm giống nhau.
- Điều khiển từ Dashboard của phần mềm Node-RED.
- Giải thuật online trên web server.
- Giải quyết được vấn đề nhiễu cảm biến bằng thuật toán Kalman.

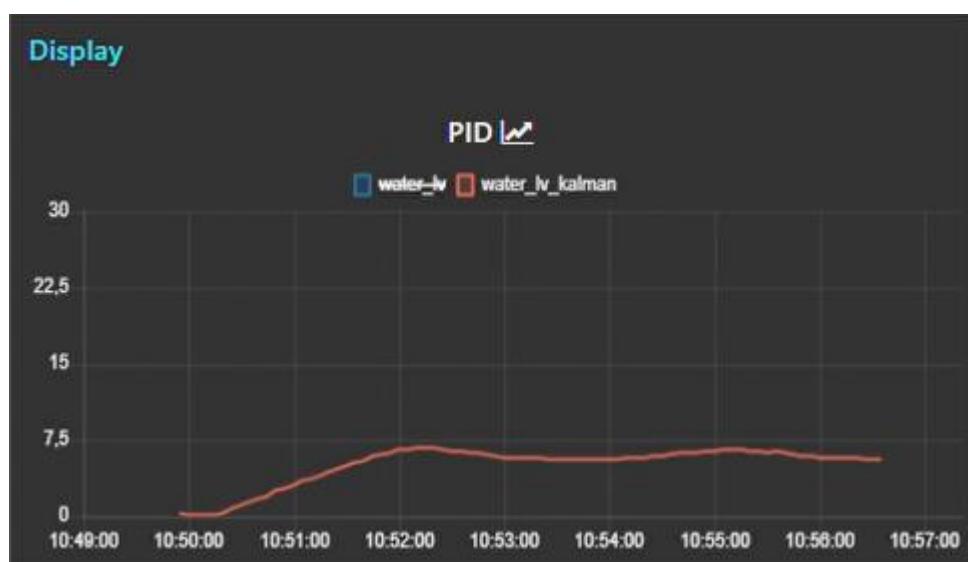
### Nhược điểm:

- Công suất máy bơm còn quá yếu so với bồn chứa chất lỏng, dẫn đến hệ thống đáp ứng chậm.
- Có thể điều khiển trên cả 2 thiết bị nhưng chưa tạo được quyền cho máy nào được quyền điều khiển, giám sát(admin) và những máy nào chỉ được giám sát(only view). Dẫn tới việc máy A đang điều khiển nhưng máy B tác động vào làm cho hệ thống bị sai lệch và hoạt động không chính xác.

### 5.6. So sánh giữa sensor khi có kalman và không có kalman



Hình 5.6.1. Đồ thị thực nghiệm của cả 2 cảm biến



**Hình 5.6.2. Đồ thị thực nghiệm kalman cho cảm biến**

- Từ các đồ thị trên ta có thể thấy được rằng, giá trị đọc được từ cảm biến lúc ban đầu còn bị nhiễu động khá nhiều nhưng sau khi có bộ lọc Kalman thì cảm biến của ta sẽ ít nhiễu hơn, ổn định hơn trước đáng kể.
- Ta có thể kết luận rằng, bộ lọc Kalman có thể giúp cho phép đo trở nên chính xác hơn, từ đó giúp cho quá trình tính toán trở nên hiệu quả hơn trước đó.

## **TÀI LIỆU THAM KHẢO**

[1] Nguyen Quang Dung - Le Hoang Trung - Le Hoang Thien - Tran Viet Hung, Moving Update Kalman Algorithm in Low-cost Node-RED IoT Network for Estimating Flood Water Level, “JOURNAL OF ADVANCED ENGINEERING AND COMPUTATION”, trang 200 - 211

<https://jaec.vn/index.php/JAEC/article/view/367/181>

[2] Le Huu Tai - Ngo Thai Phu - Tran Van Ngu - Tran Viet Hung, Liquid Level Detection Via Node-RED and Automatic Image Processing System, “JOURNAL OF ADVANCED ENGINEERING AND COMPUTATION”, trang 75 - 82

<https://jaec.vn/index.php/JAEC/article/download/351/175>

[3] Phần mềm Arduino

<https://quantrimang.com/cong-nghe/arduino-la-gi-va-ung-dung-cua-no-trong-cuoc-song-145388>

[4] Bộ điều khiển PID

<https://adi-jsc.com.vn/tin-tuc/dieu-khien-onoff-va-pid.html>

[5] MongoDB

<https://viblo.asia/p/tim-hieu-ve-mongodb-4P856ajGIY>

## **PHỤ LỤC A: CHƯƠNG TRÌNH ESP**

```
#include <WiFi.h>
#include <HTTPClient.h>
#include <ArduinoJson.h>
// const int trig = 9;      // chân trig của HC-SR04
// const int echo = 10;      // chân echo của HC-SR04

const char* ssid = "TEC_wifi";
const char* password = "TEC_wifi";
String serverPost = "https://us-east-1.aws.data.mongodb-
api.com/app/application-0-sdkod/endpoint/Esp32";
String serverGet_PID = "https://us-east-1.aws.data.mongodb-
api.com/app/application-0-sdkod/endpoint/GET_PID";
String serverGeta1w = "https://us-east-1.aws.data.mongodb-
api.com/app/application-0-sdkod/endpoint/a1w";
String serverGeta2w = "https://us-east-1.aws.data.mongodb-
api.com/app/application-0-sdkod/endpoint/a2w";

#define trig 17
#define echo 16
#define IN1 14
#define IN2 27
#define IN3 25
#define IN4 26
#define EN_A 12
#define EN_B 13
#define SENSOR 23

float x = 0;
int map_pump;
float flowRate;
float water_lv = 0;
long currentMillis;
long previousMillis = 0;
int interval = 1000;
int calibrationFactor = 5880;
volatile byte pulseCount;
float pwm_pumper;
String u_star, u, setpoint, deviation,Ti;
float u_num;
int setpoint_num, deviation_num, Ti_num;
String payload;
```

```
void IRAM_ATTR pulseCounter();  
  
void initWiFi() {  
    WiFi.mode(WIFI_STA);  
    WiFi.begin(ssid, password);  
    Serial.print("Connecting to WiFi ..");  
    while (WiFi.status() != WL_CONNECTED) {  
        Serial.print('.');  
        delay(1000);  
    }  
    Serial.print("Connected to WiFi network with IP Address: ");  
    Serial.println(WiFi.localIP());  
}  
void setup() {  
    Serial.begin(9600);  
    initWiFi();  
    Serial.print("RRSI: ");  
    Serial.println(WiFi.RSSI());  
    pinMode(trig,OUTPUT);  
    pinMode(echo,INPUT);  
    pinMode(EN_A,OUTPUT);  
    pinMode(EN_B,OUTPUT);  
    pinMode(IN1, OUTPUT);  
    pinMode(IN2, OUTPUT);  
    pinMode(IN3, OUTPUT);  
    pinMode(IN4, OUTPUT);  
  
    pinMode(SENSOR, INPUT_PULLUP);  
    pulseCount = 0;  
    flowRate = 0.0;  
    attachInterrupt(digitalPinToInterrupt(SENSOR), pulseCounter, FALLING);  
}  
  
void loop() {  
    //-----Flowrate-----  
    currentMillis = millis();  
    if (currentMillis - previousMillis > interval) {  
        flowRate = (1000.0 * pulseCount) / calibrationFactor;  
        previousMillis = millis();  
        pulseCount = 0;  
    //-----Distance-----  
    water_lv = 34.5 - HC_SR04();  
    if(water_lv > (-0.5) && water_lv < (0.5))
```

```
{  
    water_lv = 0;  
}  
  
x = x + 0.2*(water_lv - x); // kalman  
}  
  
// mo phong  
// water_lv += u_num;  
// Serial.println("H:");  
// Serial.println(water_lv);  
  
  
if(WiFi.status() == WL_CONNECTED){  
    POST(water_lv, x, flowRate);  
    GETa1w(); // data kalman  
    GETa2w(); // kalman for Kp  
    payload = GET_PID();  
    Ti = splitString(splitString(payload, "," , 1), ":" , 1);  
    Ti_num = Ti.toInt();  
    u = splitString(splitString(payload, "," , 2), ":" , 1);  
    u_num = u.toFloat();  
    setpoint = splitString(splitString(payload, "," , 3), ":" , 1);  
    setpoint_num = setpoint.toInt();  
    deviation = splitString(splitString(payload, "," , 4), ":" , 1);  
    deviation_num = deviation.toInt();  
}  
Serial.println(payload);  
Serial.print("u:");  
Serial.println(u);  
Serial.print("setpoint:");  
Serial.println(setpoint);  
Serial.print("Ti:");  
Serial.println(Ti_num);  
  
if(u_num <= 0.03){  
    digitalWrite(IN3, LOW);  
    digitalWrite(IN4, LOW);  
}  
else{  
    if(Ti_num != 0)  
    {  
        map_pump = deviation_num;  
    } else  
    {
```

```
    map_pump = 125;
}
pwm_pumper = map(u_num,0.03,0.6*setpoint_num, map_pump ,350);
if (pwm_pumper > 255)
{
    pwm_pumper = 255;
}
analogWrite(EN_B,pwm_pumper);
digitalWrite(IN3,LOW);
digitalWrite(IN4,HIGH);
}

//pwm_error_num_FF=map(pwm_error_num, 70, 190, 70, 160);
if(deviation_num != 0)
{
    digitalWrite(IN1,LOW);
    digitalWrite(IN2,HIGH);
    analogWrite(EN_A,deviation_num);
    Serial.print("Devi:");
    Serial.println(deviation_num);
} else
{
    digitalWrite(IN1,LOW);
    digitalWrite(IN2,LOW);
}
}

void POST(float n, float k, float m){
HTTPClient http;
http.begin(serverPost);
http.addHeader("Content-Type", "application/json");
StaticJsonDocument<200> doc;

// Tạo các đối tượng con của JSON document
JsonObject esp32 = doc.createNestedObject("ESP32");
esp32["Water_lv"] = n;
esp32["Water_lv_kalman"] = k;
esp32["Flow_meter"] = m;

String requestBody;
serializeJson(doc, requestBody);
int httpResponseCode = http.POST(requestBody);
if(httpResponseCode>0){
    String response = http.getString();                                //Get the
response to the request
    Serial.println(httpResponseCode); //Print return code
}
```

```
    Serial.println(response);           //Print request answer
}else{
    Serial.print("Error on sending POST: ");
    Serial.println(httpResponseCode);
}
http.end();
}
String GET_PID(){
HTTPClient http;
http.begin(serverGet_PID);
int httpCode = http.GET(); //Make the request
if (httpCode > 0) { //Check for the returning code
    String payload = http.getString();
    Serial.println(httpCode);
    return payload;
}else{
    Serial.println("Error on HTTP request");
}
http.end();
}
void GETa1w(){
HTTPClient http;
http.begin(serverGeta1w);
int httpCode = http.GET(); //Make the request
if (httpCode > 0) { //Check for the returning code
    String payload = http.getString();
    Serial.println(httpCode);
    Serial.println(payload);
}else{
    Serial.println("Error on HTTP request");
}
http.end();
}
void GETa2w(){
HTTPClient http;
http.begin(serverGeta2w);
int httpCode = http.GET(); //Make the request
if (httpCode > 0) { //Check for the returning code
    String payload = http.getString();
    Serial.println(httpCode);
    Serial.println(payload);
}else{
    Serial.println("Error on HTTP request");
}
http.end();
}
```

```
float HC_SR04(){
    unsigned long duration; // biến đo thời gian
    float distance; // biến lưu khoảng cách
    /* Phát xung từ chân trig */
    digitalWrite(trig,0); // tắt chân trig
    delayMicroseconds(2);
    digitalWrite(trig,1); // phát xung từ chân trig
    delayMicroseconds(5); // xung có độ dài 5 microSeconds
    digitalWrite(trig,0); // tắt chân trig
    /* Tính toán thời gian */
    // Đo độ rộng xung HIGH ở chân echo.
    duration = pulseIn(echo,HIGH);
    // Tính khoảng cách đến vật.
    distance = float(duration/2/29.412);
    return distance;
}
//Hàm tách chuỗi theo 1 ký tự
String splitString(String str, String delim, uint16_t pos){
    String tmp = str;
    for(int i=0; i<pos; i++){
        tmp = tmp.substring(tmp.indexOf(delim)+1);
        if(tmp.indexOf(delim)== -1
        && i != pos -1 )
            return "";
    }
    return tmp.substring(0,tmp.indexOf(delim));
}
void IRAM_ATTR pulseCounter()
{
    pulseCount++;
}
```

## PHỤ LỤC B CÔNG THỨC HÀM TRUYỀN LAPLACE CỦA HỆ THỐNG VÀ BỘ ĐIỀU KHIỂN P PI PD

Hàm truyền của bộ điều khiển P:

$$u(t) = \bar{u} + k_c e(t)$$

trong đó

$u(t)$  – đầu ra của bộ phận điều khiển (tín hiệu điều khiển thực)

$\bar{u}$  – độ dịch

$e(t)$  – sai lệch điều khiển,  $e(t) = r(t) - y(t)$

$k_c$  – hệ số khuếch đại của bộ điều khiển,  $k_c$

Nếu coi ban đầu hệ thống ở trạng thái xác lập và  $\bar{u} = 0$ , thì hàm truyền đạt của bộ điều khiển P được viết đơn giản là:

$$K_P(s) = k_c$$

Hàm truyền của bộ điều khiển PI:

$$u(t) = \bar{u} + k_c \left( e(t) + \frac{1}{\tau_i} \int_0^t e(\tau) d\tau \right)$$

Trong đó  $\tau_i$  là thời gian tích phân và  $\bar{u}$  là giá trị đầu ra của bộ điều khiển tại thời điểm  $t = 0$ . Hàm truyền đạt của bộ điều khiển PI là:

$$K_{PI}(s) = \frac{u(s)}{e(s)} = k_c \left( 1 + \frac{1}{\tau_i s} \right)$$

Hàm truyền của bộ điều khiển PID:

$$u(t) = \bar{u} + k_c \left( e(t) + \frac{1}{\tau_i} \int_0^t e(\tau) d\tau + \tau_d \frac{de(t)}{dt} \right)$$

Trong đó

$k_c$  – hệ số khuếch đại của bộ điều khiển

$\tau_i$  – thời gian tích phân

$\tau_d$  – thời gian vi phân

$\bar{u}$  – giá trị đầu ra của bộ điều khiển tại điểm làm việc.

$\bar{u} = 0$  thì hàm truyền đạt của bộ điều khiển sẽ được viết như sau:

$$K_{PID}(s) = \frac{u(s)}{e(t)} = k_c \left( 1 + \frac{1}{\tau_i s} + \tau_d s \right)$$