

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT  
THÀNH PHỐ HỒ CHÍ MINH**

**CHUYÊN ĐỀ**

**THIẾT KẾ BỘ ĐIỀU KHIỂN TỐC ĐỘ QUẠT SỬ DỤNG  
FUZZY LOGIC**

**NGÀNH: KỸ THUẬT ĐIỆN TỬ**

**MÃ SỐ HỌC VIÊN: 2080701**

**HỌC VIÊN: PHẠM QUỐC BẢO**

**GIẢNG VIÊN HƯỚNG DẪN: PGS.TS. TRẦN THU HÀ**

**Tp. Hồ Chí Minh, tháng 12 năm 2020**

## MỤC LỤC

<b>Chương MỞ ĐẦU .....</b>	<b>4</b>
1. Tính cần thiết của đề tài .....	4
2. Các nghiên cứu trong và ngoài nước.....	5
a. Nghiên cứu trong nước .....	5
b. Nghiên cứu ngoài nước .....	5
3. Nhiệm vụ của đề tài.....	6
4. Giới hạn của đề tài .....	6
5. Các bước tiến hành .....	6
6. Điểm mới của đề tài .....	6
7. Tính thực tiễn .....	7
8. Phương pháp nghiên cứu .....	7
9. Nội dung đề tài.....	7
10. Tiến độ thực hiện đề tài .....	7
<b>Chương 1.....</b>	<b>8</b>
<b>TỔNG QUAN .....</b>	<b>8</b>
1.1. Tổng quan Fuzzy Logic .....	8
1.2. Định nghĩa tập mờ.....	8
1.3. Xây dựng mô hình mờ .....	10
1.4. Nguyên lý hoạt động của Fuzzy Logic.....	11
1.5. Cơ sở ứng dụng cho mô hình .....	12
1.6. Giới thiệu phần mềm MATLAB.....	12
<b>Chương 2.....</b>	<b>13</b>
<b>XÂY DỰNG MÔ HÌNH .....</b>	<b>13</b>
2.1 Sơ đồ khối.....	13
2.2 Tạo giao diện.....	15
2.3 Tạo các khối liên kết .....	21
<b>Chương 3.....</b>	<b>30</b>
<b>MÔ PHỎNG THỰC TẾ .....</b>	<b>30</b>
3.1 Mô phỏng thực tế của hệ thống trên phần mềm MATLAB.....	30
<b>Chương 4.....</b>	<b>34</b>

<b>KẾT LUẬN VÀ HƯỚNG NGHIÊN CỨU PHÁT TRIỂN.....</b>	<b>34</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>35</b>

### MỤC LỤC HÌNH ẢNH

Hình 1: Mô hình hóa ngõ vào.....	9
Hình 2: Mô hình mờ Mamdani và Sugeno .....	10
Hình 3: Nguyên lý hoạt động khối Fuzzy Logic .....	11
Hình 4: Giao diện MATLAB.....	12
Hình 5: Mô hình giao tiếp của hệ thống.....	13
Hình 6: Lưu đồ giải thuật của hệ thống .....	14
Hình 7: Giao diện người dùng.....	15
Hình 8: Thay đổi tên tag: "edtxt" .....	16
Hình 9: Thay đổi tên tag: "edtxt2" .....	17
Hình 10: Thay đổi tên tag: "editbox" .....	18
Hình 11: Khai báo khối kết nối.....	18
Hình 12: Lấy giá trị Tin từ giao diện.....	19
Hình 13: Hiển thị giá trị Tin lên giao diện.....	19
Hình 14: Lấy giá trị Tout từ giao diện .....	20
Hình 15: Hiển thị giá trị Tout trên giao diện.....	20
Hình 16: Tạo khối liên kết.....	21
Hình 17: Gọi chương trình Fuzzy Logic .....	22
Hình 18: Tạo khối Fuzzy Logic.....	23
Hình 19: Mờ hóa các ngõ vào Fuzzy Logic (Tin) .....	24
Hình 20: Mờ hóa các ngõ vào Fuzzy Logic (Tout) .....	25
Hình 21: Mờ hóa ngõ ra.....	26
Hình 22: Tạo luật mờ cho khối Fuzzy Logic .....	27
Hình 23: Khai báo chương trình liên kết.....	28
Hình 24: Lấy giá trị cho "editbox".....	28
Hình 25: Cập nhật giá trị cho hàm input_sf.....	29
Hình 26: Gán giá trị cho khối S-function.....	29
Hình 27: Các thành phần của giao diện.....	30
Hình 28: Giao diện khi chạy mô phỏng MATLAB .....	31
Hình 29: Chạy thử nghiệm lần 1.....	31
Hình 30: Chạy thử nghiệm lần 1 khối xử lý trung tâm .....	32
Hình 31: Chạy thử nghiệm lần 2.....	33
Hình 32: Chạy thử nghiệm lần 2 khối xử lý trung tâm .....	33

**LỜI CẢM ƠN**

Sự thành công nào cũng đều gắn liền với những sự hỗ trợ, giúp đỡ của những người xung quanh dù cho sự giúp đỡ đó là ít hay nhiều, trực tiếp hay gián tiếp.

Một lần nữa em xin chân thành cảm ơn PGS.TS Trần Thu Hà, sau khi được học môn xử lý tín hiệu số nâng cao của cô và được tiếp xúc với cách làm việc từ cô, em rất vui và hào hứng vì được cô gợi mở nhiều hướng cho tiểu luận cũng như chuyên đề trong quá trình học.

Chúc cô thật nhiều sức khỏe, nhiều niềm vui và mai mắn trong cuộc sống. Sẽ còn mãi nhiệt huyết, tận tâm vì sự nghiệp giáo dục.

Em xin chân thành cảm ơn!

Trân trọng

Tp.HCM Tháng 12 năm 2020

Học viên: Phạm Quốc Bảo

## Chương MỞ ĐẦU

### 1. Tính cần thiết của đề tài

Ngày nay việc chuyển đổi điều khiển thủ công sang tự động đang phát triển mạnh mẽ. Giúp con người nâng cao năng suất lao động, có thêm thời gian nghỉ ngơi so với việc điều khiển bằng phương pháp thủ công như trước đây. Rất nhiều thiết bị được điều khiển bằng ứng dụng Fuzzy Logic đang dần trở nên phổ biến trong các lĩnh vực công nghiệp hoặc dân dụng chẳng hạn như bộ điều khiển lái tàu, Máy giặt, tủ lạnh... Sở dĩ có sự chuyển đổi mạnh mẽ như vậy cũng vì một phần những bộ điều khiển này dựa trên suy luận mờ. Cho phép người điều khiển không cần phải thao tác nhiều trên một thiết bị hay quá trình. Logic mờ được xây dựng dựa trên kinh nghiệm của những người điều khiển, qua đó giúp tạo ra được bộ điều khiển đủ tin cậy. Thay thế được họ nhưng hiệu quả thì vẫn giữ được như khi người điều khiển đang thực hiện.

Cùng với sự tiến bộ vượt bậc của khoa học kỹ thuật nói chung cũng như những bộ điều khiển sử dụng Fuzzy Logic nói riêng thì việc điều khiển tốc độ quạt trong nhà xưởng hiện nay chủ yếu thực hiện bằng tay và đa số chưa có thể điều chỉnh tốc độ hợp lý bằng việc đọc các giá trị cảm biến nhiệt độ. Do đó yêu cầu phải tạo ra được một hệ thống điều khiển được tốc độ của quạt bằng cách đọc giá trị của những cảm biến được lắp bên trong và bên ngoài nhà máy để thu thập dữ liệu nhiệt độ. Ngoài ra còn đảm bảo hoạt động liên tục, tin cậy và phải thay đổi trạng thái sao cho phù hợp với mỗi lần có giá trị nhiệt độ mới. Bởi vì bản thân giá trị cảm biến nhiệt độ trả về kết quả sẽ không phải ở một mức cố định mà giá trị ở đây chính là những con số thay đổi liên tục theo thời gian.

Để giúp nhân viên cảm thấy thoải mái hơn trong môi trường làm việc nhà xưởng, việc chuyển đổi điều khiển tốc độ quạt từ thủ công sang tự động là một vấn đề cần có sự đầu tư nghiêm túc để tìm ra một hướng giải pháp tối ưu, hợp lý. Đề tài: “Điều khiển tốc độ quạt trên phần mềm MATLAB sử dụng Fuzzy Logic” được học viên tiến hành nghiên cứu nhằm giải quyết những vấn đề trên.

## 2. Các nghiên cứu trong và ngoài nước

### a. Nghiên cứu trong nước

✓ Sử dụng Logic Mờ điều khiển động cơ không đồng bộ, Huỳnh Thanh Tuấn, Dương Hoài Nghĩa.

Đề tài nghiên cứu sử dụng Fuzzy Logic điều khiển động cơ không đồng bộ và mô phỏng trên phần mềm MatLab.

✓ Logic Mờ trong hệ thống điều khiển tốc độ động cơ không đồng bộ ba pha, Nguyễn Thị Hiên.

Đề tài sử dụng nghiên cứu một hệ thống điều khiển tốc độ các động cơ không đồng bộ sử dụng bộ điều khiển logic mờ.

Các công trình nghiên cứu trên áp dụng phần mềm MatLab để giải quyết các bài toán điều khiển các hệ thống. Tuy nhiên, giá trị thực tiễn còn hạn chế vì chỉ tính toán cho một kịch bản ít dữ liệu đầu vào, và chưa xét đến kịch bản phát triển nhiều tham số đầu vào trong tương lai.

### b. Nghiên cứu ngoài nước

✓ Simulation of fuzzy logic control for DC servo motor using Arduino based on MATLAB/Simulink. ([Simulation of fuzzy logic control for DC servo motor using Arduino based on MATLAB/Simulink - IEEE Conference Publication](#))

Bài báo này nghiên cứu vấn đề điều khiển DC Servo trên Arduino và MatLab

✓ Implementation of a Fuzzy Logic Speed Controller For a Permanent Magnet DC Motor Using a LowCost Arduino Platform. ( [Implementation of a fuzzy logic speed controller for a permanent magnet dc motor using a low-cost Arduino platform - IEEE Conference Publication](#))

Bài báo này nghiên cứu vấn đề giảm chi phí điều khiển tốc độ động cơ bằng việc sử dụng Fuzzy Logic và mô phỏng trên MatLab.

✓ Caldo, R. B., Seranilla, J. T., Castillo, D. J., Diocales, K. S., Gulle, W. D., Nunez, B. L., & Parreno, C. T. (2015). Design and development of fuzzy logic controlled dimming lighting system using Arduino microcontroller. 2015

International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM).

Bài báo này nghiên cứu việc điều khiển độ sáng của đèn sử dụng Fuzzy Logic trên Arduino.

Các công trình nghiên cứu trên đây chủ yếu áp dụng phần mềm MatLab kết hợp với Fuzzy Logic để thiết kế hệ thống điều khiển tốc độ của động cơ hoặc độ sáng của đèn chứ chưa xét đến bài toán có những cảm biến đầu vào để việc điều chỉnh phù hợp với điều kiện trong thực tế.

### **3. Nhiệm vụ của đề tài**

- Nghiên cứu mô hình điều khiển tốc độ quạt bằng Fuzzy Logic.
- Nghiên cứu giải thuật Fuzzy Logic trong phần mềm MatLab.
- Lựa chọn luật mờ cho khối Fuzzy Logic.
- Nghiên cứu cách mô phỏng giao diện trên phần mềm MatLab.
- Nghiên cứu cách kết hợp giữa giao diện và Fuzzy Logic trong phần mềm MatLab.

### **4. Giới hạn của đề tài**

Nghiên cứu và thực hiện mô phỏng thay đổi tốc độ quạt từ giá trị hai cảm biến bên trong và bên ngoài phòng.09ug

### **5. Các bước tiến hành**

- Tổng quan về phần mềm MATLAB và Fuzzy Logic.
- Nghiên cứu phần mềm MATLAB.
- Xác định sơ đồ khối.
- Tạo giao diện.
- Tạo các khối liên kết.
- Kết luận và hướng nghiên cứu phát triển.

### **6. Điểm mới của đề tài**

- Đề xuất phương pháp mô phỏng hệ thống điều khiển tốc độ quạt từ những giá trị cảm biến theo thời gian thực.
- Tạo giao diện để việc mô phỏng diễn ra dễ dàng.

**7. Tính thực tiễn**

- Kết quả nghiên cứu của luận văn được sử dụng làm tài liệu tham khảo cho các học viên cao học, nghiên cứu sinh Ngành Kỹ thuật điện tử.
- Đề xuất các giải pháp khả thi trong vận hành quạt nhằm tiết kiệm điện năng trong giới hạn cho phép.

**8. Phương pháp nghiên cứu**

Đề tài sử dụng các phương pháp sau:

- Phương pháp thu thập và nghiên cứu tài liệu.
- Phương pháp mô phỏng.
- Phương pháp phân tích và tổng hợp.

**9. Nội dung đề tài**

**Chương:** Mở Đầu.

**Chương 1:** Tổng quan.

**Chương 2:** Xây dựng mô hình.

**Chương 3:** Mô phỏng thực tế.

**Chương 4:** Kết luận và hướng nghiên cứu phát triển.

**Tài liệu tham khảo.**

**Phụ lục**

**10. Tiến độ thực hiện đề tài**

- Tháng 9 năm 2020: Thu thập tài liệu, viết Chương Mở đầu, Chương 1.
- Tháng 10,11 năm 2022: Viết Chương 2.
- Tháng 11 năm 2020: Viết Chương 3.
- Tháng 12 năm 2020: Viết Chương 4.
- Tháng 12 năm 2020: Hoàn thiện đề tài, viết tóm tắt, soạn trình chiếu.



## Chương 1

# TỔNG QUAN

### 1.1. Tổng quan Fuzzy Logic

Fuzzy Logic hay còn gọi là “logic mờ” khác so với các phương pháp điều khiển truyền thống chính là sử dụng khối điều khiển mờ. Cách Fuzzy Logic điều khiển một đối tượng thông qua các điều kiện đầu vào cũng giống như cách con người suy nghĩ và giải quyết một vấn đề đó chính là: “Nếu ... Thì ...”. Bản chất của điều khiển fuzzy logic chính là tổng quát hóa các điều kiện và những hệ quả từ những điều kiện này lại thành một tập giá trị có thể xảy ra từ đó áp dụng cho hệ thống sử dụng fuzzy logic. Điều khiển mờ đang chiếm một vị trí quan trọng trong việc điều khiển hiện nay. Bởi vì đối với những phương pháp điều khiển truyền thống sẽ không thể nào áp dụng hiệu quả cho những bài toán phức tạp, thường xuyên thay đổi trạng thái hoặc có hệ số đầu vào để điều khiển thay đổi liên tục, không ổn định. Bộ điều khiển ứng dụng fuzzy logic tỏ ra giải quyết vấn đề trên cực kỳ đơn giản. Ngoài ra còn có những lợi thế hơn những phương pháp điều khiển truyền thống như sau:

- Mô hình hóa các hàm phi tuyến có tính phức tạp cao.
- Có thể kết hợp Fuzzy Logic với nhiều cách điều khiển khác.
- Gần gũi với suy nghĩ tư duy của người vận hành.
- Dễ thay đổi khi hệ thống cần sự nâng cấp.

### 1.2. Định nghĩa tập mờ

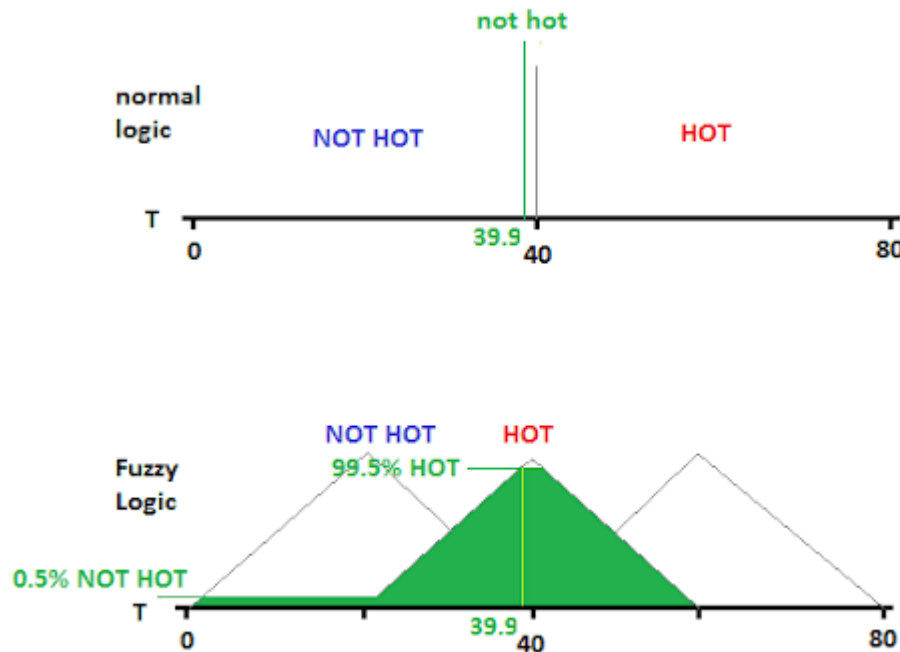
Khái niệm tập mờ là mở rộng của những tập hợp cổ điển, với mục đích biểu diễn những mối quan hệ không rõ ràng. Trong lý thuyết tập hợp cổ điển, quan hệ thành viên của các phần tử được đánh giá theo kiểu nhị phân rõ ràng. Ví dụ như phần Tử A chắc chắn thuộc tập hợp C hoặc chắc chắn không thuộc tập C. Như vậy để kết luận cho điều vừa nói ở trên ta sẽ gán cho A giá trị 1 nếu thuộc tập C và 0 nếu không thuộc tập C. Mô hình hóa khái niệm này ta có dạng như sau:

$$\forall u \in U, \mu(u) = \begin{cases} 1, & u \in C \\ 0, & u \notin C \end{cases}$$

Như vậy, ta có thể thấy giá trị của  $\mu(u)$  chỉ có thể nhận được 2 giá trị đó là  $\{0,1\}$ . Nhưng đối với tập mờ, cho phép chúng ta đánh giá theo tỉ lệ khả năng A có nằm trong C hay không. Bằng cách sử dụng một hàm liên thuộc để thể hiện tỉ lệ mà A có thuộc về tập C hay không. Nói cách khác tỉ lệ này sẽ có

khoảng giá trị chạy từ 0 đến 1, không phải cố định tại một giá trị. Mô hình hóa ta có dạng như sau:

$$\forall u \in U, 0 \leq \mu(u) \leq 1$$



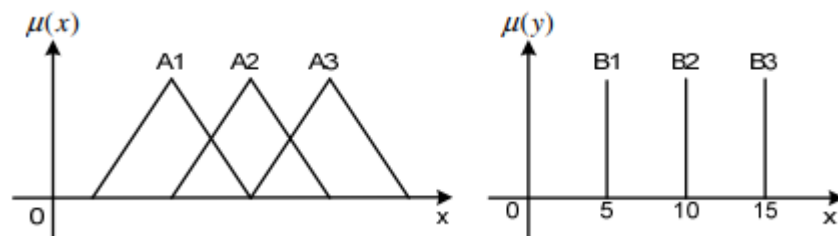
Hình 1: Mô hình hóa ngõ vào

Xét ví dụ như trên ta thấy giá trị sẽ thay đổi từ  $[0:1]$  và khoảng nhiệt độ lạnh  $[0:40]$ , vừa  $[20:60]$  và nóng  $[40:80]$ . Xét trong khái niệm tập hợp cổ điển với trường hợp 39.9 độ C ta kết luận ngay nhiệt độ này là không nóng nhưng thực tế nhiệt độ này có 99.5% là nóng. Bởi vì theo tập hợp cổ điển ta chỉ có thể kết luận nhiệt độ không nóng khi nhỏ hơn 40 độ C và Nóng khi từ 40 độ C trở lên. Nhưng khi xét với khái niệm tập mờ, có thể kết luận tỉ lệ 39.9 độ C ở khoảng không nóng là 0.5% và nóng là 99.5%. Ngoài ra, cũng có thể dựa vào tập mờ để ra các khái niệm: “tương đối lạnh” khi có tỉ lệ nhiệt độ ở khoảng lạnh cao hơn tỉ lệ của nhiệt độ ở khoảng vừa. “Hơi hơi ấm” khi có tỉ lệ nhiệt độ ở khoảng vừa cao hơn tỉ lệ của nhiệt độ ở khoảng nóng.

### 1.3. Xây dựng mô hình mờ

Xét theo luật điều khiển mờ, ta có thể phân loại thành hai loại mô hình mờ: Mô hình mờ Mamdani và mô hình mờ Sugeno.

Mô hình mờ Mamdani là bộ điều khiển trong đó kết luận của điều khiển là mệnh đề mờ. Mô hình mờ Sugeno là bộ điều khiển trong đó kết luận của luật điều khiển là hàm các tín hiệu vào của hệ mờ<sup>1</sup>. Trong các ứng dụng điều khiển liên quan đến các đối tượng không xác định được mô hình toán học hay ngõ ra không cụ thể thì mô hình Mamdani là một lựa chọn tối ưu. Trong khi mô hình Sugeno thích hợp với các điều khiển có mô hình không chính xác hoặc các mô hình phi tuyến. Tuy hai mô hình đã nêu khác nhau hoàn toàn. Nhưng trong nhiều trường hợp kết luận ngõ ra của mô hình Mamdani và Sugeno cho ra hai giá trị tương đương nhau.



Hình 2: Mô hình mờ Mamdani và Sugeno

Ta cùng xét bộ điều khiển Fuzzy logic có một ngõ vào và một ngõ ra. Khi sử dụng mô hình Mamdani, các luật điều khiển được thể hiện như sau:

- + y sẽ là B1 nếu x là A1.
- + y sẽ là B2 nếu x là A2.
- + y sẽ là B3 nếu x là A3.

Khi sử dụng mô hình Sugeno ta cũng thể hiện các luật điều khiển như sau:

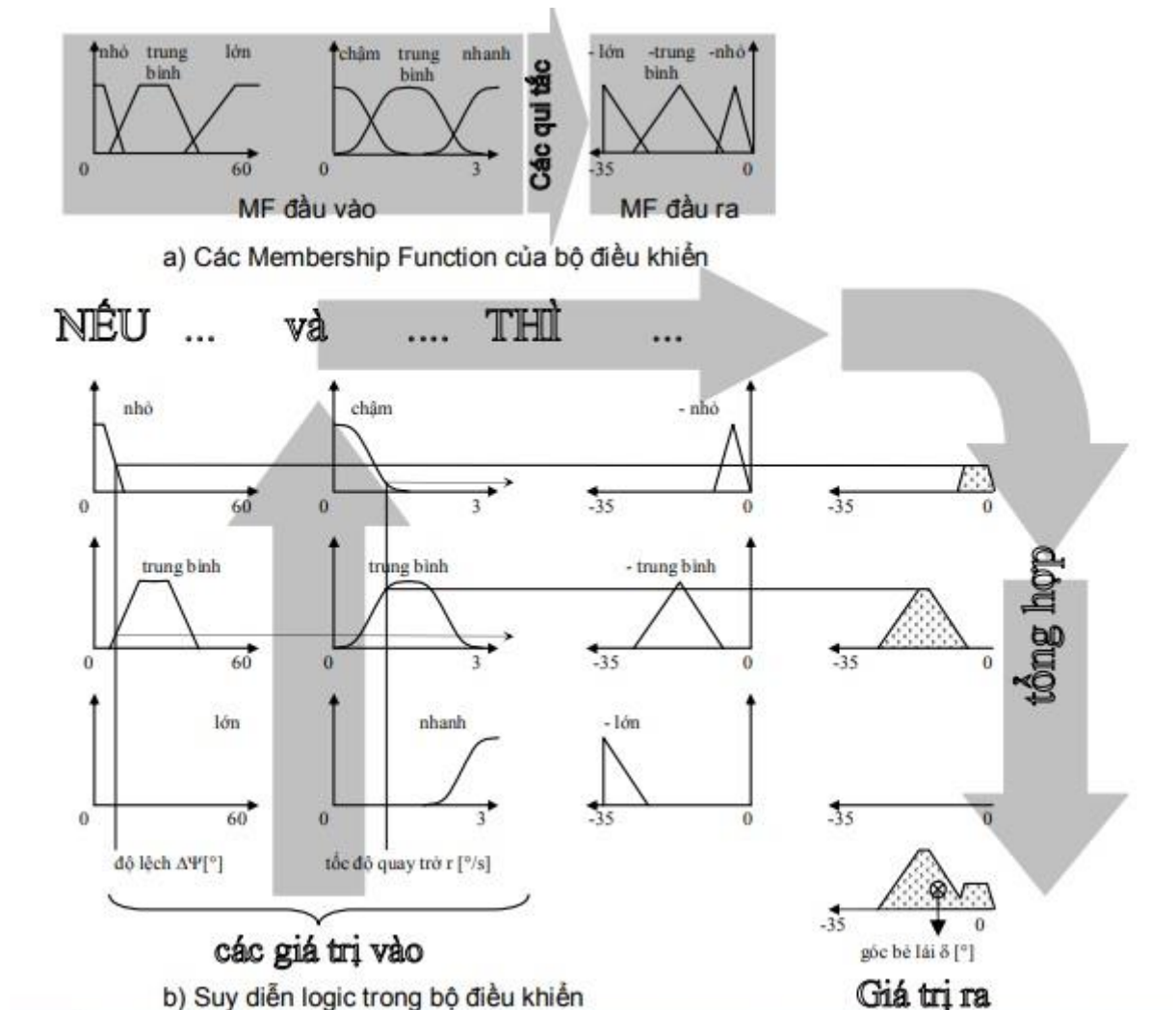
- + y sẽ bằng 5 khi x bằng A1.
- + y sẽ bằng 10 khi x bằng A2.
- + y sẽ bằng 15 khi x bằng A3.

Trong trường hợp này thì khi sử dụng một trong hai dạng mô hình thì giá trị ở ngõ ra cũng đều ra chung một kết quả.

<sup>1</sup> [Microsoft Word - Giao trinh Dieu khien mo va mang noron\\_DaSua\\_080519 \(uneti.edu.vn\)](http://Microsoft Word - Giao trinh Dieu khien mo va mang noron_DaSua_080519 (uneti.edu.vn))

### 1.4. Nguyên lý hoạt động của Fuzzy Logic

Bộ điều khiển Fuzzy Logic được cấu tạo gồm một hoặc nhiều các Membership Function (MF) đầu vào, tiếp theo là các luật mờ đã được định nghĩa, ngõ ra dựa trên sự liên kết giữa các đầu vào và luật mờ.



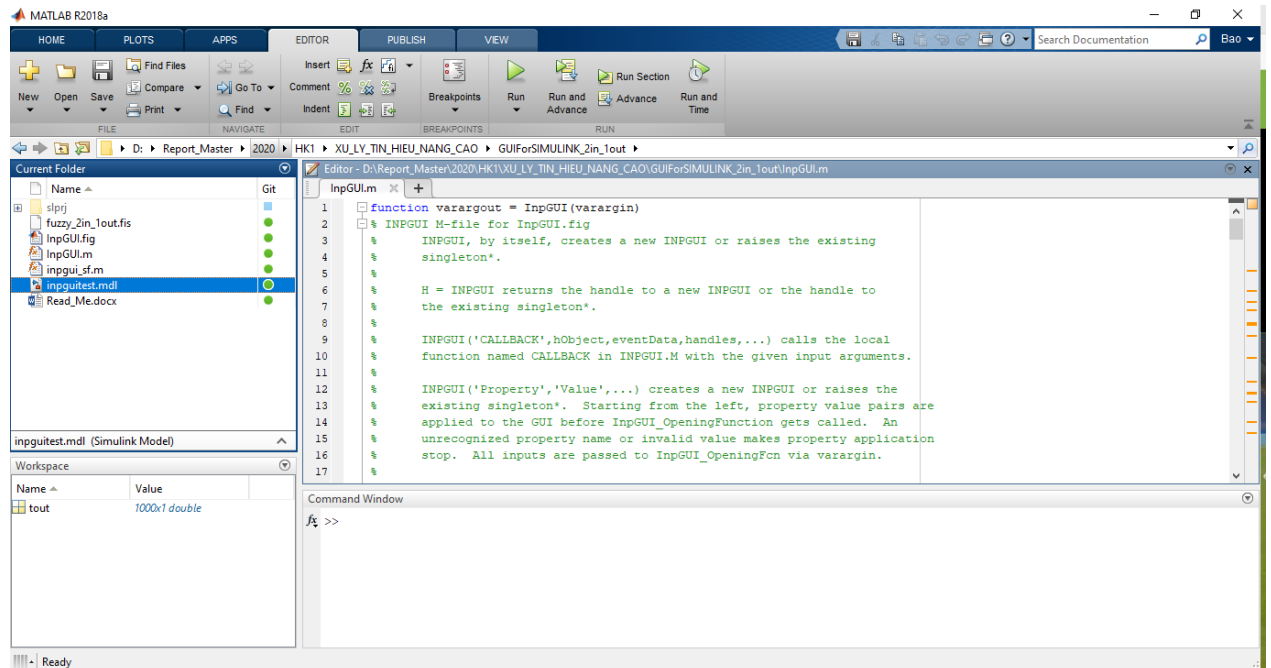
Hình 3: Nguyên lý hoạt động khối Fuzzy Logic

Từ một giá trị cụ thể ở ngõ vào, khối Fuzzy Logic bắt đầu mờ hóa ngõ vào chính là đánh giá các tín hiệu thuộc mức nào trong các mức đã định nghĩa từ trước trong các MF đầu vào. Sau bước này thì các dạng tín hiệu đầu vào sẽ được biểu diễn theo các khái niệm: “nhỏ, trung bình, lớn”. Giống như ví dụ đã nêu ở phần trên một điều kiện đầu vào có thể nhiều hơn một mức của bộ điều khiển. Thì ở ngõ ra cũng sẽ có nhiều hơn một giá trị. Khối Fuzzy logic tiến hành giải mờ và tổng hợp lại các kết quả, sau đó sử dụng thuật toán tìm trọng tâm để tính toán ra giá trị chính xác nhất.

### 1.5. Cơ sở ứng dụng cho mô hình

Cơ sở ứng dụng cho mô hình:

- Matlab: Tạo Giao diện và mô phỏng sự thay đổi của 2 giá trị nhiệt độ trong phòng và nhiệt độ ngoài phòng. Sử dụng của MatLab vào đồ án.
- Fuzzy Logic toolbox: Tạo khối xử lý chính với 2 ngõ vào 1 ngõ ra.



Hình 4: Giao diện MATLAB

## 1.6. Giới thiệu phần mềm MATLAB

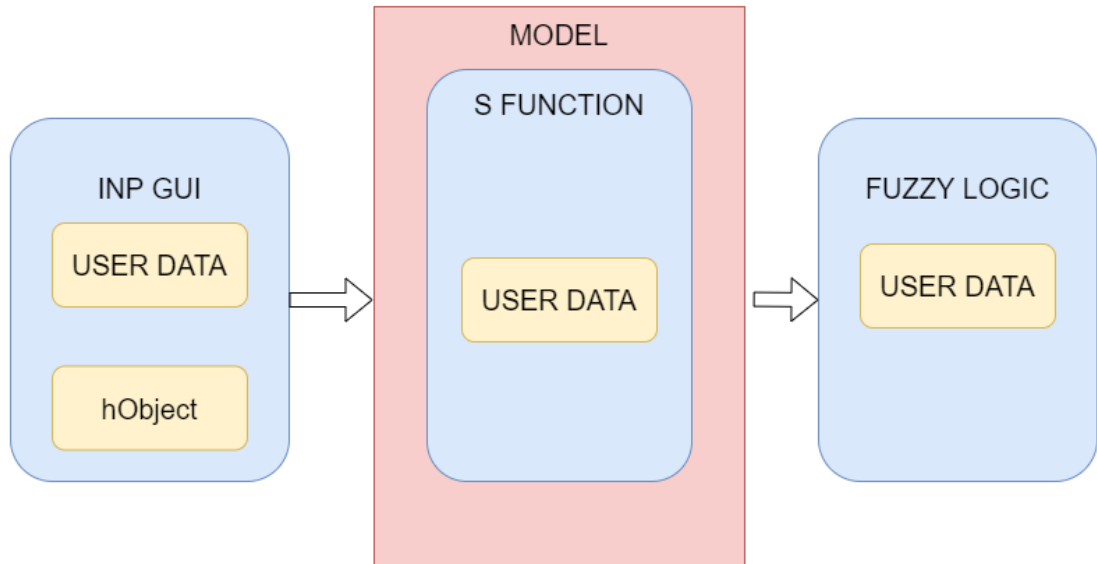
MatLab là một môi trường tính toán số và lập trình, được thiết kế bởi công ty MathWorks. MATLAB cho phép tính toán số với ma trận, vẽ đồ thị hàm số hay biểu đồ thông tin, thực hiện thuật toán, tạo các giao diện người dùng và liên kết với những chương trình máy tính viết trên nhiều ngôn ngữ lập trình khác. MATLAB giúp đơn giản hóa việc giải quyết các bài toán tính toán kỹ thuật so với các ngôn ngữ lập trình truyền thống như C, C++, và Fortran.<sup>2</sup> MatLab hiện nay đang được sử dụng nhiều trong các lĩnh vực: xử lý tín hiệu, xử lý ảnh, hệ thống điều khiển tự động, mô phỏng hệ thống ...

<sup>2</sup> Giới thiệu ngắn gọn về Matlab - Matlab4vn (google.com)

## Chương 2

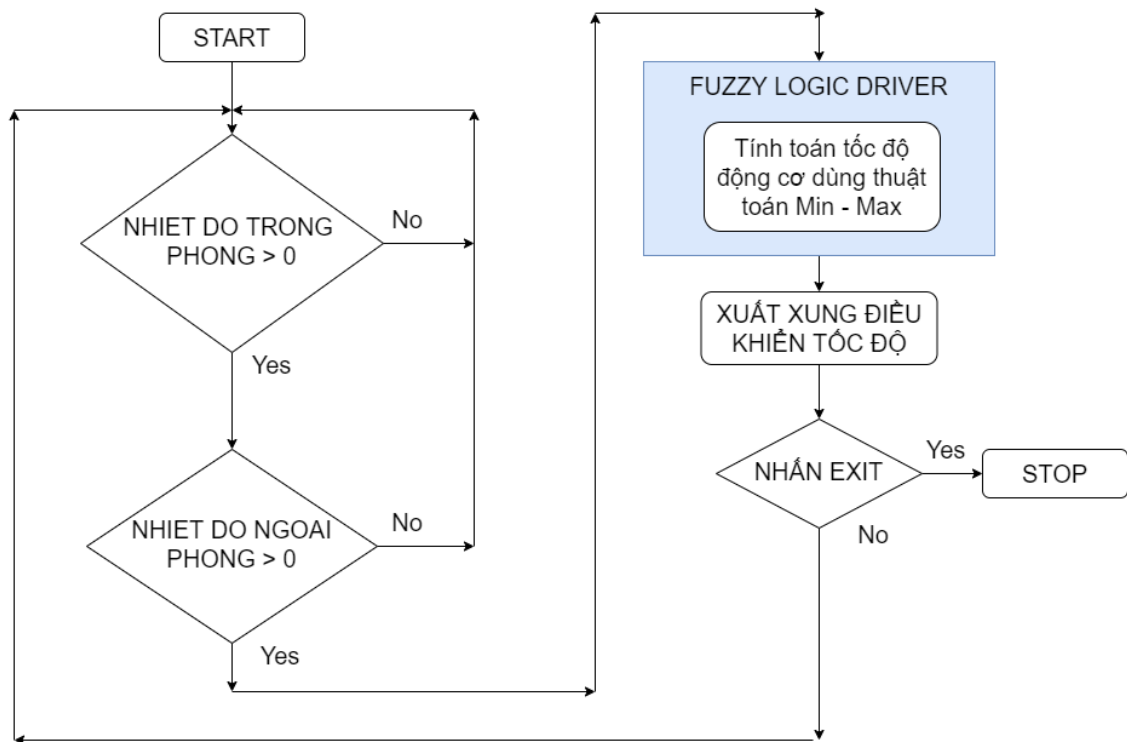
# XÂY DỰNG MÔ HÌNH

### 2.1 Sơ đồ khối



Hình 5: Mô hình giao tiếp của hệ thống

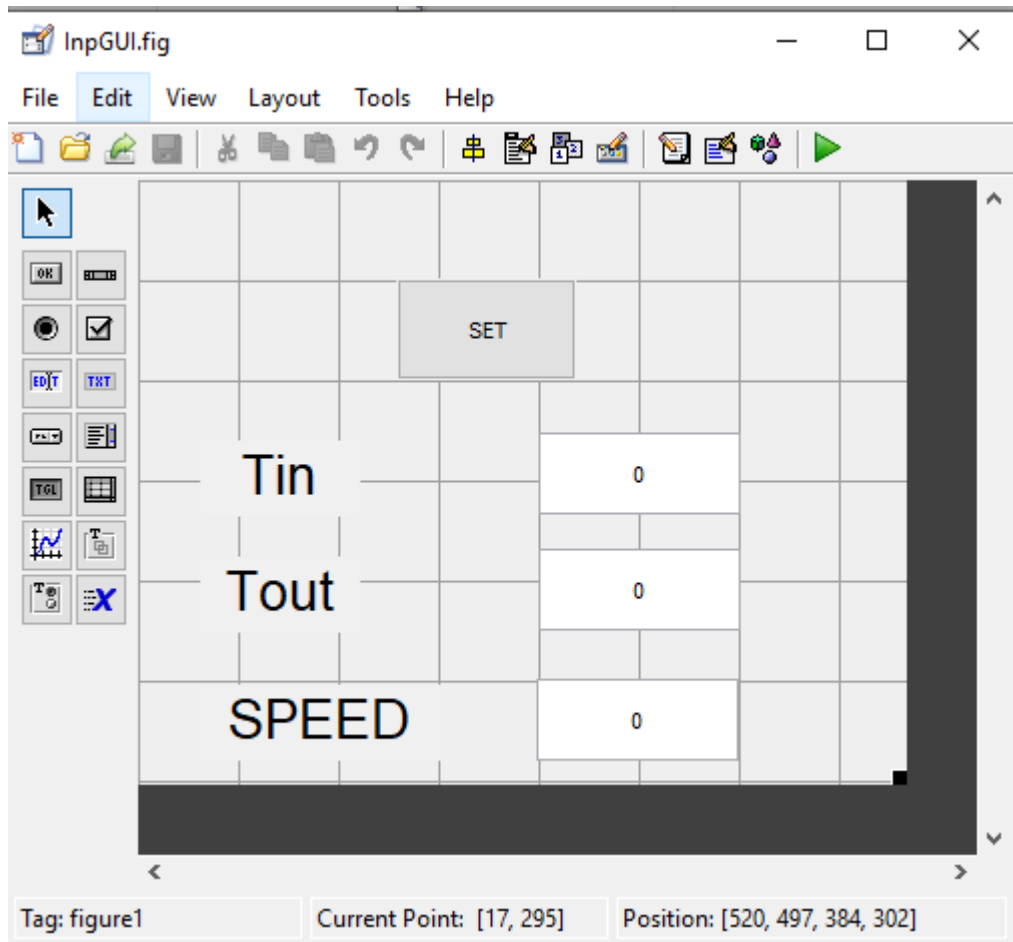
Trong đề tài này mô hình được chia thành ba khối với từng chức năng khác nhau: "Khối INPUT GUI", "khối MODEL", "Khối FUZZY LOGIC". Trong đó khối INPUT GUI có chức năng nhận dữ liệu mô phỏng mà người dùng nhập vào. Tiếp theo khối MODEL có nhiệm vụ xử lý những dữ liệu được chuyển đến từ khối INPUT GUI chẳng hạn như việc chuyển đổi những giá trị nhiệt độ từ dạng chuỗi sang dạng số. Sau đó chuyển dữ liệu này sang khối FUZZY LOGIC để tính toán tốc độ hợp lý cho quạt.



Hình 6: Lưu đồ giải thuật của hệ thống

Hệ thống gồm 2 cảm biến dùng để xác định giá trị nhiệt độ bên trong phòng và bên ngoài phòng. Nếu nhiệt độ bên trong phòng và bên ngoài phòng đều lớn hơn 0 thì hai giá trị này sẽ được đưa vào bộ Fuzzy Logic Driver, sử dụng thuật toán tìm “Max – Min”, từ đó tính toán ra được tốc độ cần thiết cho quạt. Sau khi hoàn tất một chu kỳ thì chương trình kiểm tra xe người dùng có nhấn nút thoát để thoát khỏi chương trình điều khiển.

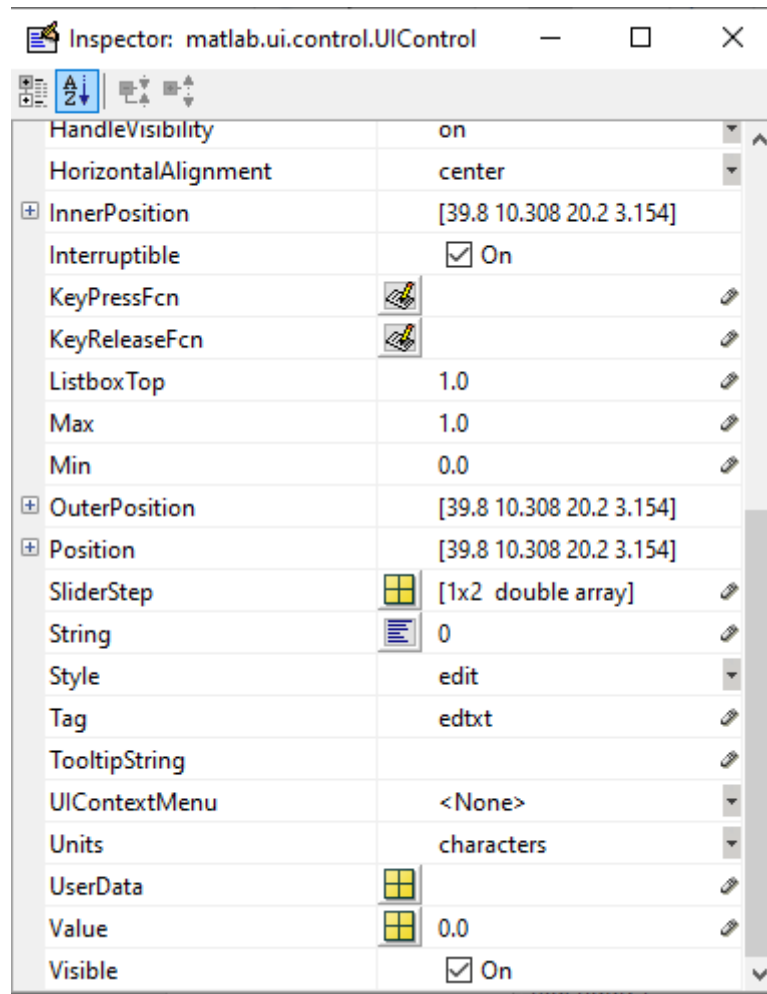
## 2.2 Tạo giao diện



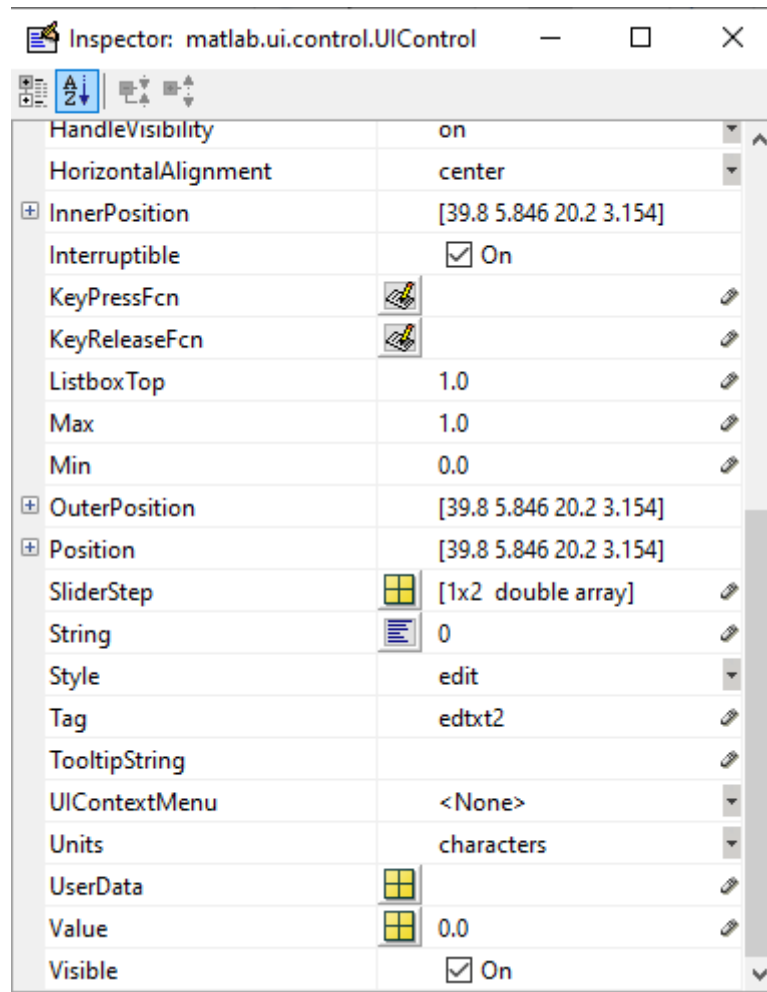
Hình 7: Giao diện người dùng

Để tiến hành tạo giao diện tương tác giữa người dùng và MatLab trước hết ta chọn “guide” trên cửa sổ gõ lệnh của MatLab, sau đó lần lượt sử dụng các khối: Text để hiển thị văn bản: “Tin, Tout, SPEED”, Button để tạo nút “SET” và EditText để hiển thị các giá trị mà mình muốn thể hiện.

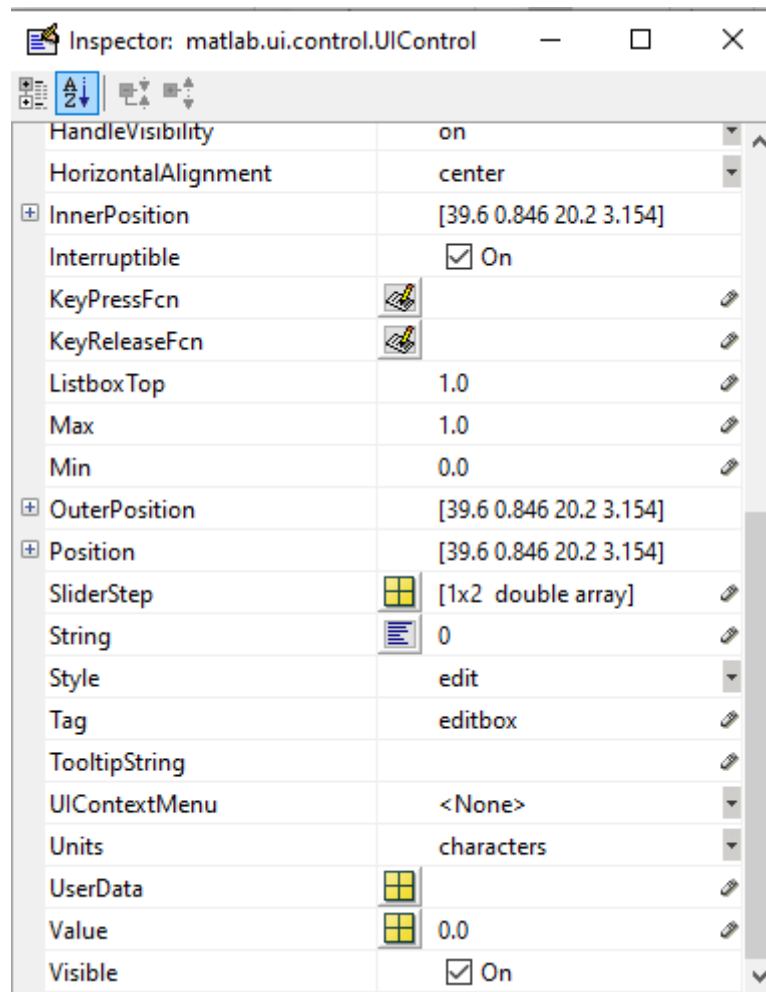




Hình 8: Thay đổi tên tag: "edtxt"



Hình 9: Thay đổi tên tag: "edtxt2"



Hình 10: Thay đổi tên tag: "editbox"

Sau khi đã tạo được giao diện như hình ta tiến hành đặt tên cho các thành phần để dễ dàng sử dụng trong phần “Xử lý trung tâm”. Đầu tiên ta đặt lại Tag\_Id cho từng thành phần EditText như đã trình bày ở trên. Ở đây chúng ta dùng edtxt cho Tin, edtxt2 cho Tout và editbox cho SPEED. Việc đặt tên như vậy giúp dễ dàng liên kết giữa hai thành phần “Giao diện người dùng” và “Xử lý trung tâm” trong đề tài này. Tiếp theo tiến hành lập trình cho giao diện, tiến hành khai báo các giá trị.

```
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @InpGUI_OpeningFcn, ...
                  'gui_OutputFcn',  @InpGUI_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
```

Hình 11: Khai báo khối kết nối

Khi nhập giá trị vào hai giá trị: Tin, Tout. Nếu định dạng nhập vào đúng với như đã định nghĩa là kiểu số. Giá trị này sẽ được chuyển đổi sang dạng chuỗi và hiển thị trên giao diện. Bằng cách lưu giá trị vào biến “ hObject” trong Matlab.

```
function edtxt_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edtxt (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%-----part2-2-----% display initial value
ud =get(gcf,'UserData'); % get ud stored in fig userdata field
set(hObject,'String', num2str(ud.editfield)) ; %write value in userdata to field
guidata(hObject, handles); %update view
%-----part2-2 ends-----%
```

Hình 12: Lấy giá trị Tin từ giao diện

```
function edtxt2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edtxt2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%-----part2-2-----% display initial value
ud =get(gcf,'UserData'); % get ud stored in fig userdata field
set(hObject,'String', num2str(ud.edit2field)) ; %write value in userdata to field
guidata(hObject, handles); %update view
%-----part2-2 ends-----%
```

Hình 13: Hiển thị giá trị Tin lên giao diện

Ngoài ra vì khối “Xử lý trung tâm” cũng sử dụng giá trị này như những giá trị đầu vào cho việc tính toán của khối Fuzzy Logic. Nên khi nhập một giá trị vào editText Box thì ta tiến hành chuyển đổi giá trị từ dạng chuỗi sang dạng số sau đó lưu giá trị vào biến: “editfield, edit2field”. Nếu giá trị nhập vào không đúng như định dạng đã khai báo từ trước thì chương trình sẽ thông báo giá trị bị lỗi và ngừng chương trình.

```

function edtxt_Callback(hObject, eventdata, handles)
% hObject    handle to edtxt (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edtxt as text
%        str2double(get(hObject,'String')) returns contents of edtxt as a double

%---part 2-1 -----%
val = get(hObject,'String'); %get the string in edit field

ud =get(gcf,'UserData'); % get ud stored in fig userdata field
if str2num(val) %if number is stored in edit field
|   ud.editfield = str2num(val);
else % if its not number
|   ud.editfield = 0; % store 0
end
set(gcf,'UserData',ud) ; %update userdata

```

Hình 14: Lấy giá trị Tout từ giao diện

```

function edtxt2_Callback(hObject, eventdata, handles)
% hObject    handle to edtxt2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

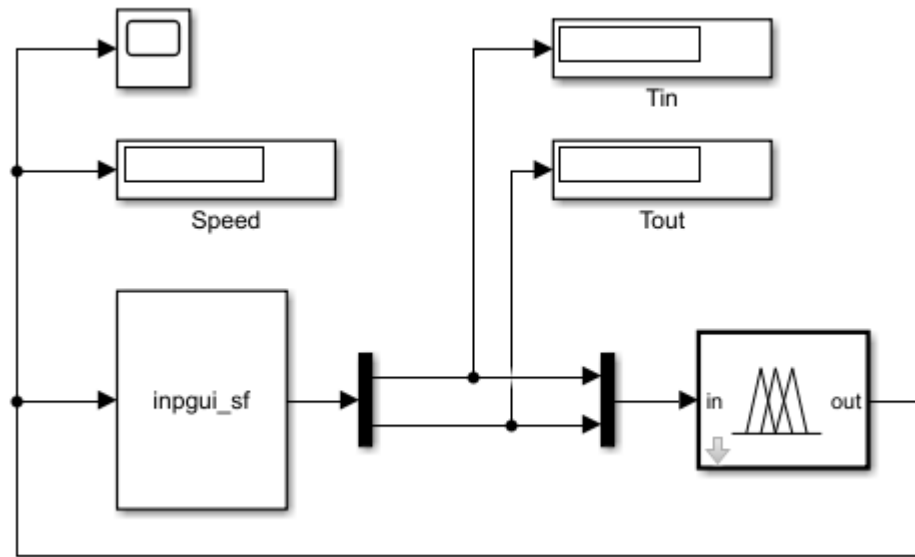
% Hints: get(hObject,'String') returns contents of edtxt2 as text
%        str2double(get(hObject,'String')) returns contents of edtxt2 as a double
val = get(hObject,'String'); %get the string in edit field

ud =get(gcf,'UserData'); % get ud stored in fig userdata field
if str2num(val) %if number is stored in edit field
|   ud.edit2field = str2num(val);
else % if its not number
|   ud.edit2field = 0; % store 0
end
set(gcf,'UserData',ud) ; %update userdata

```

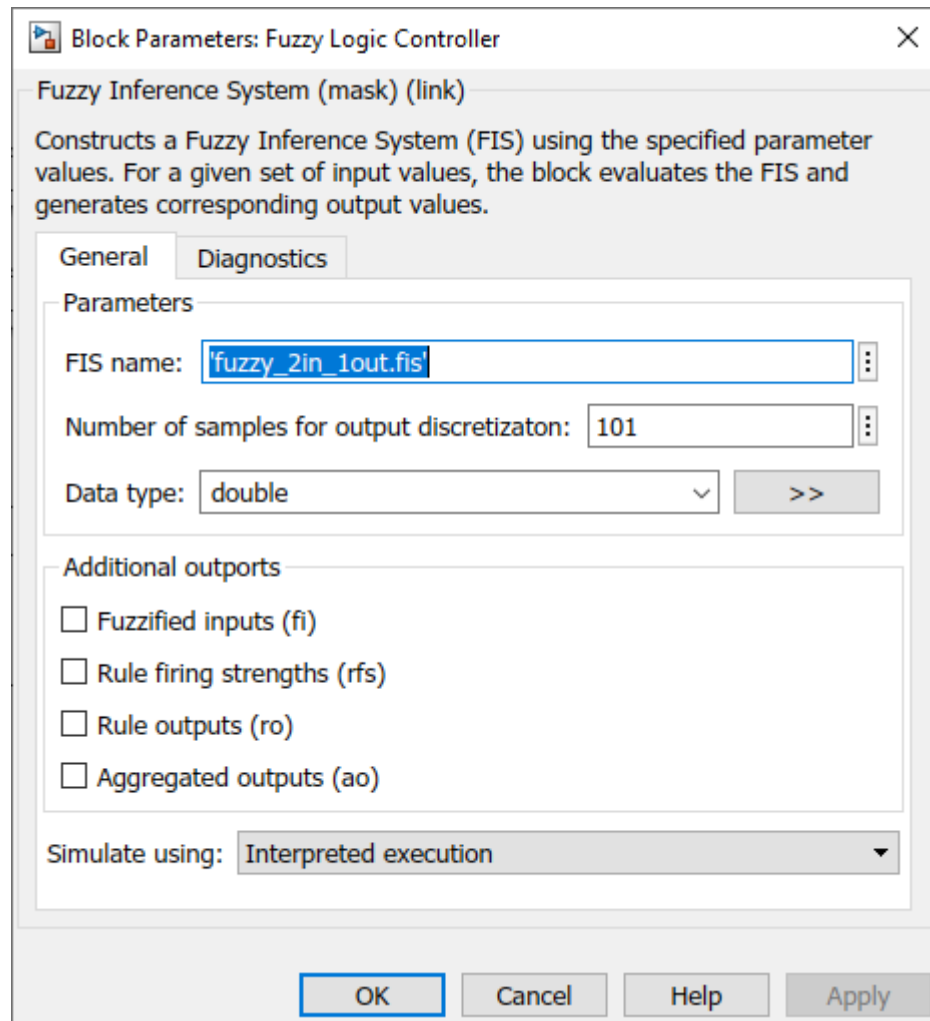
Hình 15: Hiển thị giá trị Tout trên giao diện

### 2.3 Tạo các khối liên kết



Hình 16: Tạo khối liên kết

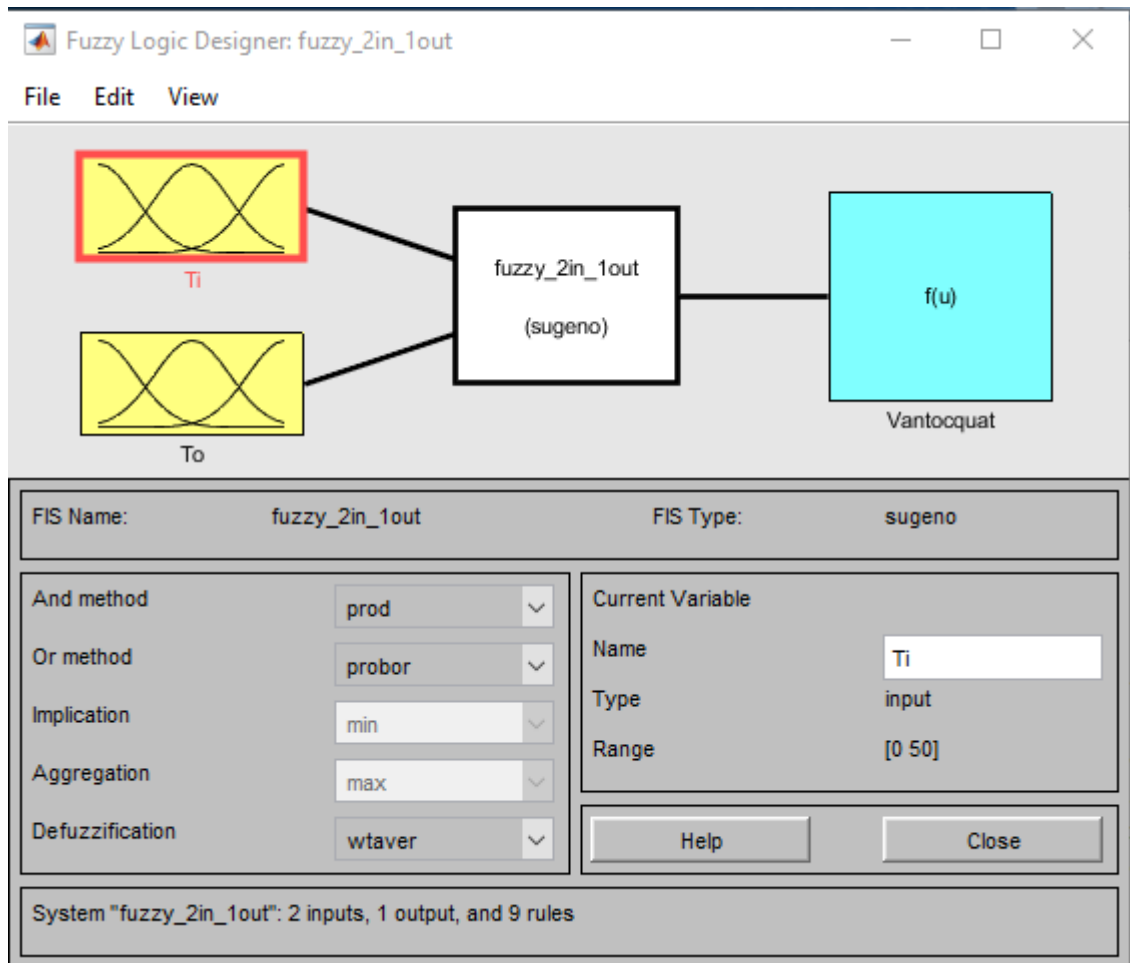
Ta tiếp tục tạo phần Simulink cho hệ thống bằng cách gõ: “Simulink” trên thanh gõ lệnh của MatLab. Gõ các khối: “Display” để hiển thị các giá trị Nhiệt độ bên trong phòng và ngoài phòng và Tốc độ hiện tại của quạt giống như các khối Textedit đã trình bày ở phần: “tạo giao diện tương tác người dùng”. Sử dụng khối: “Scope” để hiển thị tốc độ đo được dùng để kiểm tra kết quả hiển thị.



Hình 17: Gọi chương trình Fuzzy Logic

Gọi khối Fuzzy-Logic và tiến hành gán chương trình fuzzy\_logic: “fuzzy\_2in\_1out.fis”. Chương trình này giúp chúng ta tính toán được giá trị quạt bằng hai giá trị ở ngõ vào với các điều kiện được khai báo như hình bên dưới.

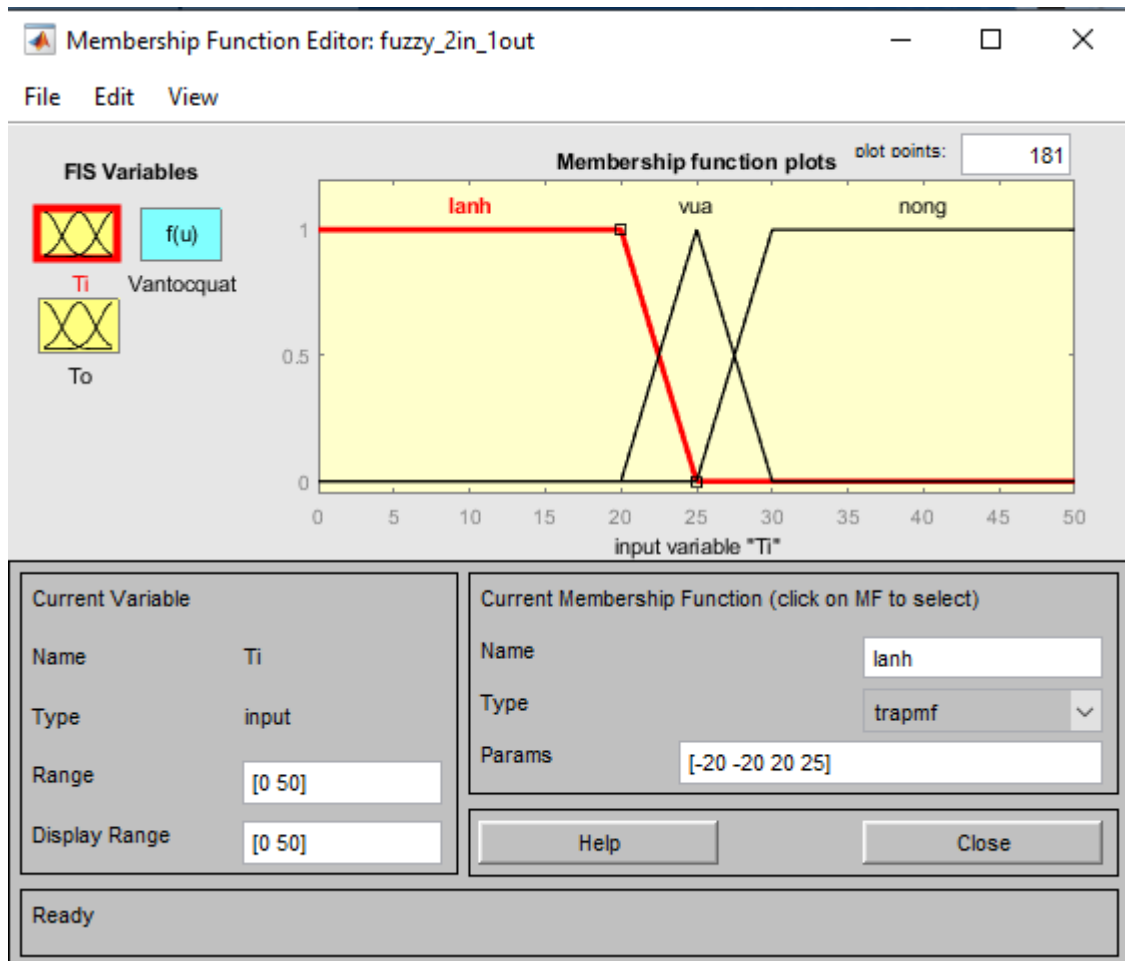
Chương trình “fuzzy\_2in\_1out” được khai báo như sau. Ở cửa sổ gõ lệnh của MatLab ta gọi fuzzy. Đối với yêu cầu ban đầu của đề tài khối Fuzzy logic sẽ có 2 ngõ vào và 1 ngõ ra. Sử dụng hàm truyền: “sugeno” cho khối Fuzzy Logic.



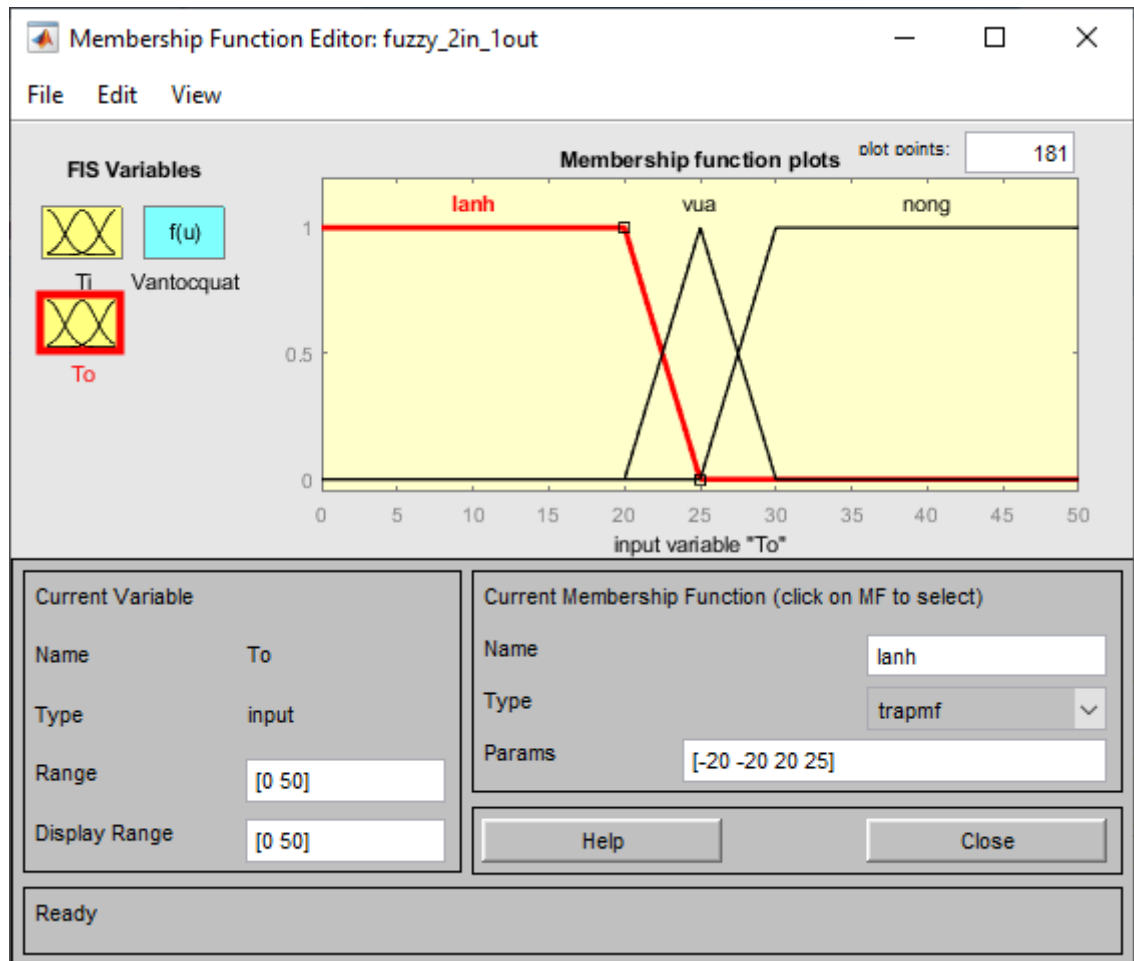
Hình 18: Tạo khối Fuzzy Logic

Tiến hành mờ hóa ngõ vào bằng các điều kiện đã được đặt ra từ trước. Đối với Tin Nhiệt độ lạnh từ [0:20] dạng tín hiệu hình thang, nhiệt độ vừa từ [20:30] dạng tín hiệu hình tam giác và nóng là trên 30 độ C dạng tín hiệu hình thang. Trường hợp của Tout khi mờ hóa ngõ vào các điều kiện giống như với Tin.



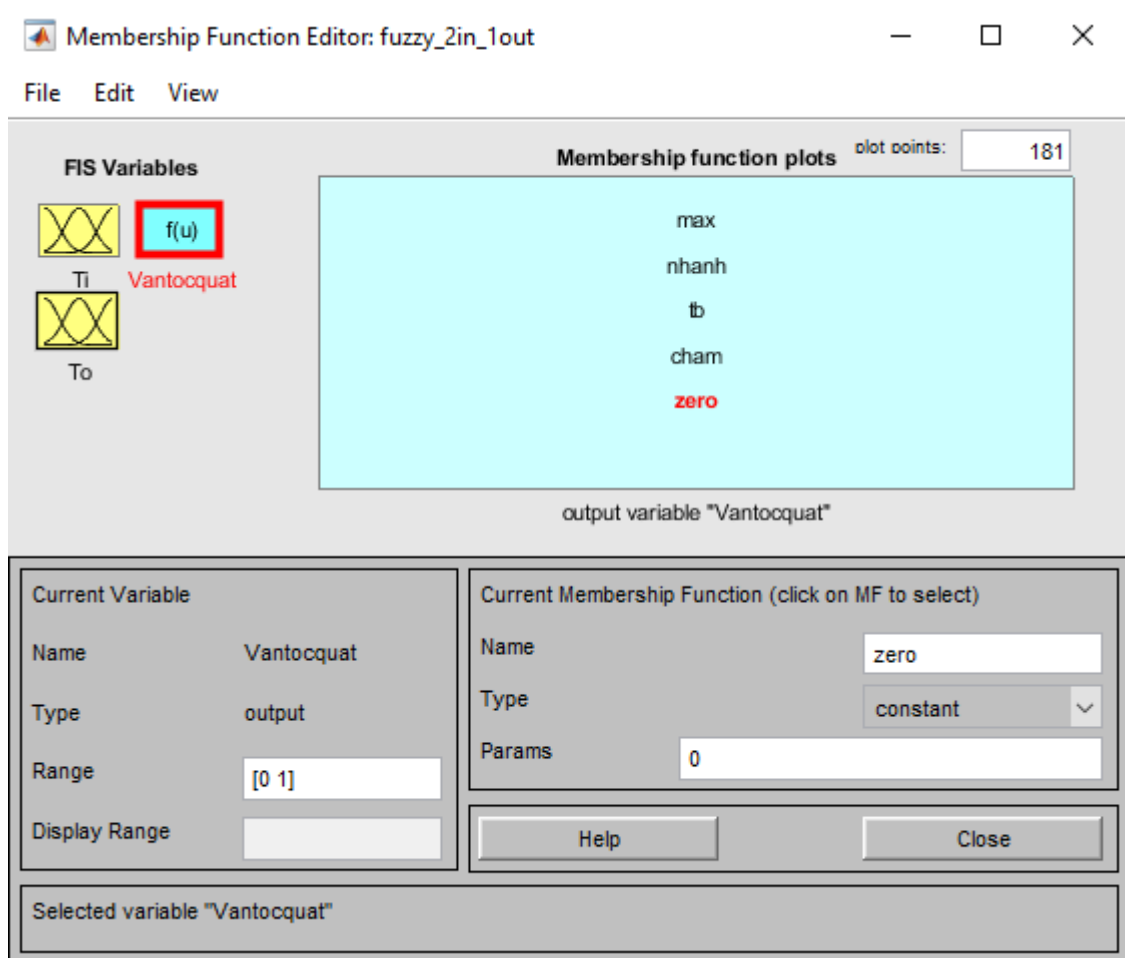


Hình 19: Mờ hóa các ngõ vào Fuzzy Logic (Tin)



Hình 20: Mờ hóa các ngõ vào Fuzzy Logic (Tout)

Ngoài ra ở ngõ ra cần phải xác định các mức tốc độ quạt bao gồm: [Đứng yên, chậm, Trung bình, nhanh và tốc độ tối đa] tương ứng với các giá trị [0, 150, 300, 450, 600].

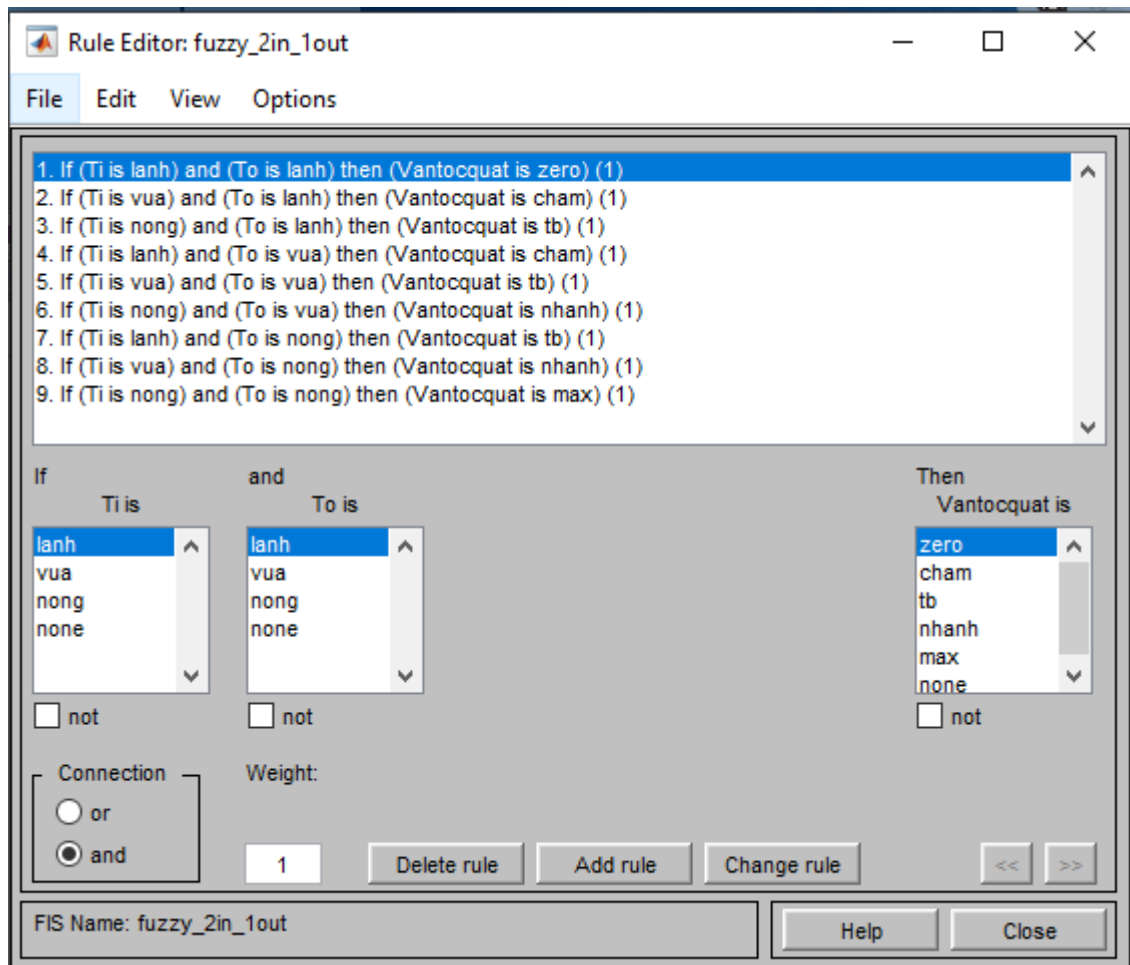


Hình 21: Mờ hóa ngõ ra

Sau khi đã thực hiện được các bước cơ bản ban đầu ta tiến hành tạo các “luật mờ” cho khối Fuzzy Logic, thông qua giá trị mà các chuyên gia điều khiển đã đúc kết được sau quá trình điều khiển tốc độ quạt như sau:

Tout \ Tin	Lạnh	Vừa	Nóng
Lạnh	Zero	Chậm	Trung Bình
Vừa	Chậm	Trung Bình	Nhanh
Nóng	Trung Bình	Nhanh	Tốc Độ Tối Đa

Bảng 1 Các Giá Trị Tốc Độ



Hình 22: Tạo luật mờ cho khối Fuzzy Logic

Sau khi tạo các luật mờ thành công, để có sự liên kết giữa hai khối xử lý trung tâm và giao diện người dùng thì ta tiến hành tạo khối S-function. Trong phần khởi tạo ta khai báo hai trường hợp khi nhấn “SET” đó là nhấn và thả.

```

function [sys,x0,str,ts] = inpgui_sf(t,x,u,flag,Ts)
switch flag,
case 0,
    [sys,x0,str,ts]=mdlInitializeSizes(Ts);
case 2,
    sys=mdlUpdate(t,x,u,Ts);
case 3,
    sys = mdlOutputs(t,x,u); % Calculate outputs
case { 1, 4, 9 },
    sys = [0 0 0 0];
case 'pushBtnDwn', %Call coming from input gui when button pressed
    ud = get(gcf,'UserData');
    set(gcf,'BackgroundColor','Yellow'); %change color
    set(gcf,'WindowButtonUpFcn','inpgui_sf([],[],[],'pushBtnUp');'); % specify the button release function
    ud.pushbutton = 1; %update gui config
    set(gcf,'Userdata',ud); %update gui config data struct
case 'pushBtnUp',
    ud = get(gcf,'UserData');
    chnd = findobj(gcf,'Tag','pushbutton');
    set(chnd,'BackgroundColor',[0.831 0.816 0.784]);
    set(gcf,'WindowButtonUpFcn','');
    ud.pushbutton = 0;
    set(gcf,'Userdata',ud);
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end

```

Hình 23: Khai báo chương trình liên kết

Khi nhấn nút “SET” thì chương trình sẽ tiến hành lấy giá trị ở biến editbox làm đầu vào cho khối s\_function. Sau khi đã có giá trị đầu vào thì khối s-function cũng sẽ đưa giá trị này thành đầu vào của khối Fuzzy\_Logic. Từ hai giá trị này sẽ tính toán ra được giá trị cho quạt, giá trị này cũng sẽ trở thành đầu vào của khối s\_function. Hiện thị giá trị trên giao diện giúp ta dễ dàng nhận biết được giá trị đang chạy.

```

function sys=mdlUpdate(t,x,u,Ts)
fig = get_param(gcf,'UserData');
sys=x;
if ishandle(fig), %
    if strcmp(get(fig,'Visible'),'on'),
        chnd = findobj(fig,'Tag','editbox'); % get handle to editbox
        str = floor(u);
        if num2str(str)
            set(chnd, 'String', str) ; %set value of editbox
        else
            set(chnd,'String','NA') ; %set value of editbox
        end
    end
end
end

```

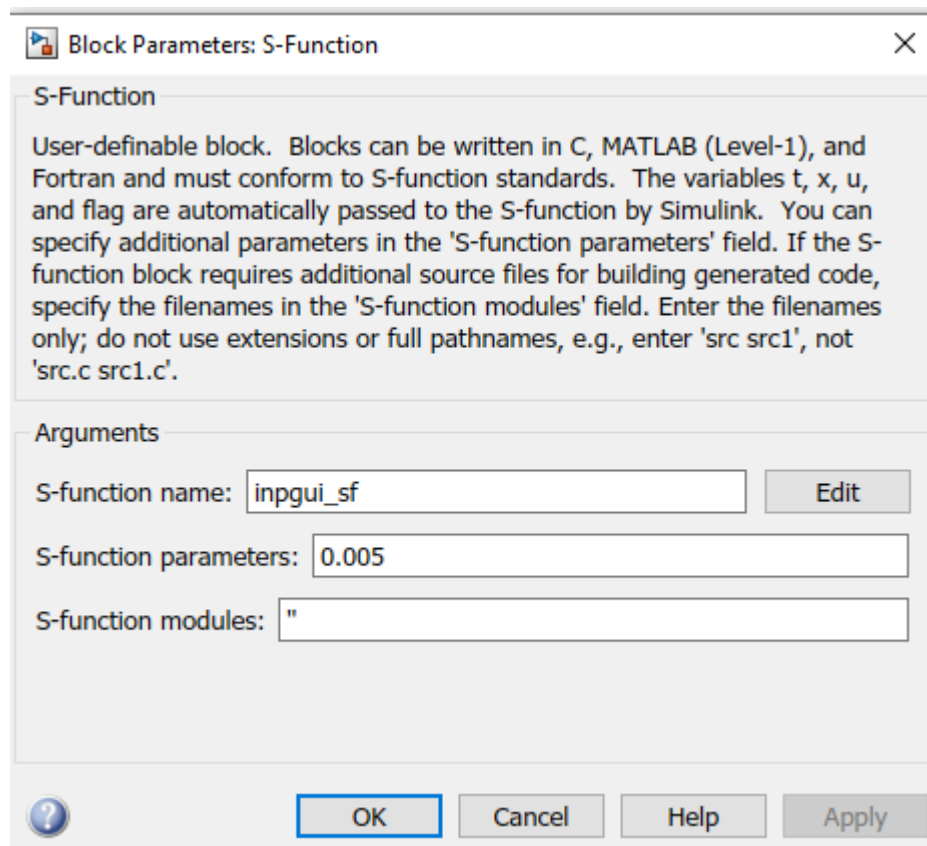
Hình 24: Lấy giá trị cho "editbox"

```

function sys = mdlOutputs(t,x,u)
    fig = get_param(gcf,'UserData') ;
    if ishandle(fig),
        %     chnd = findobj(fig,'Tag','spd_edt') ;
        ud = get(fig,'UserData') ;
        %     set(chnd,'String', num2str(ud.VehSpd)) ;
        sys = [ud.editfield ud.edit2field];
    else
        sys = [0 0];
    end
end

% end mdlOutputs

```

Hình 25: Cập nhật giá trị cho hàm `input_sf`

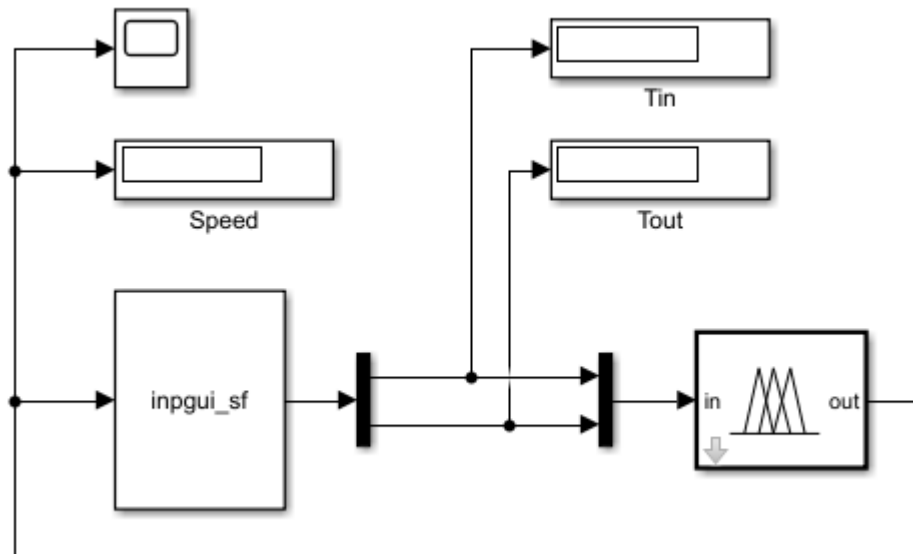
Hình 26: Gán giá trị cho khối S-function

## Chương 3

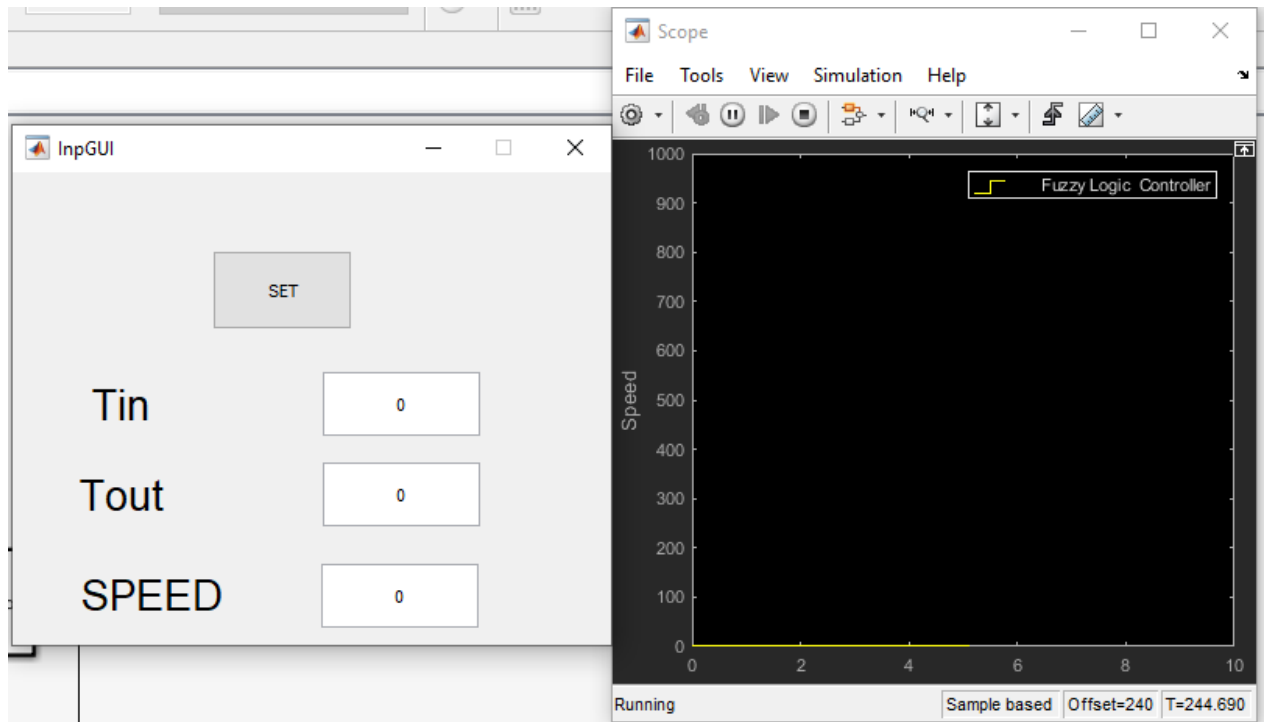
### MÔ PHỎNG THỰC TẾ

#### 3.1 Mô phỏng thực tế của hệ thống trên phần mềm MATLAB

Dưới đây là hình ảnh hoàn thiện của giao diện điều khiển tốc độ quạt thông qua giá trị 2 cảm biến vào tháng 12/2020.

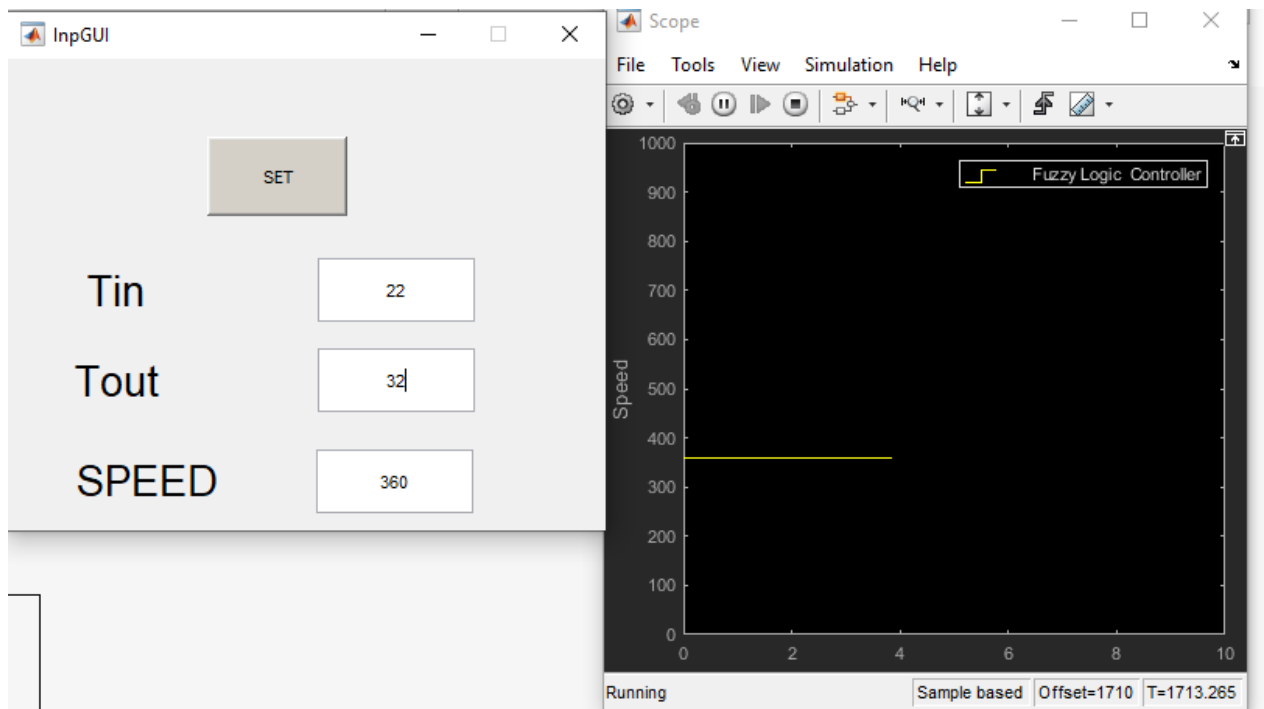


Hình 27: Các thành phần của giao diện



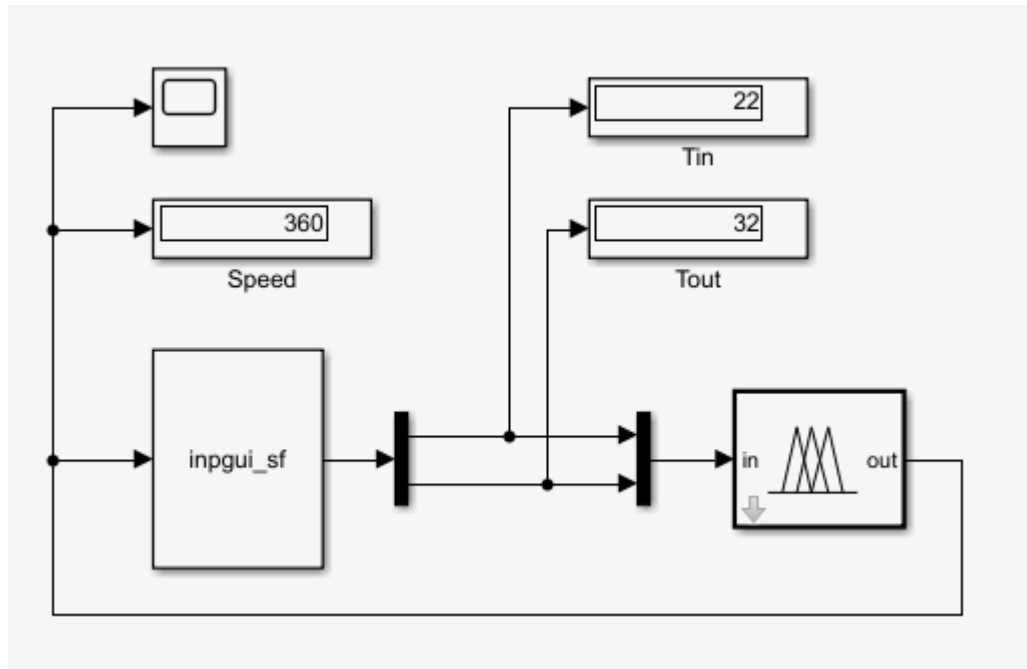
Hình 28: Giao diện khi chạy mô phỏng MATLAB

Khi giả sử nhiệt độ cảm biến trong phòng đo được: 22 độ C, nhiệt độ cảm biến bên ngoài đo được: 32 độ C. Khối Fuzzy Logic sẽ tự động tính toán được tốc độ quạt hợp lý.



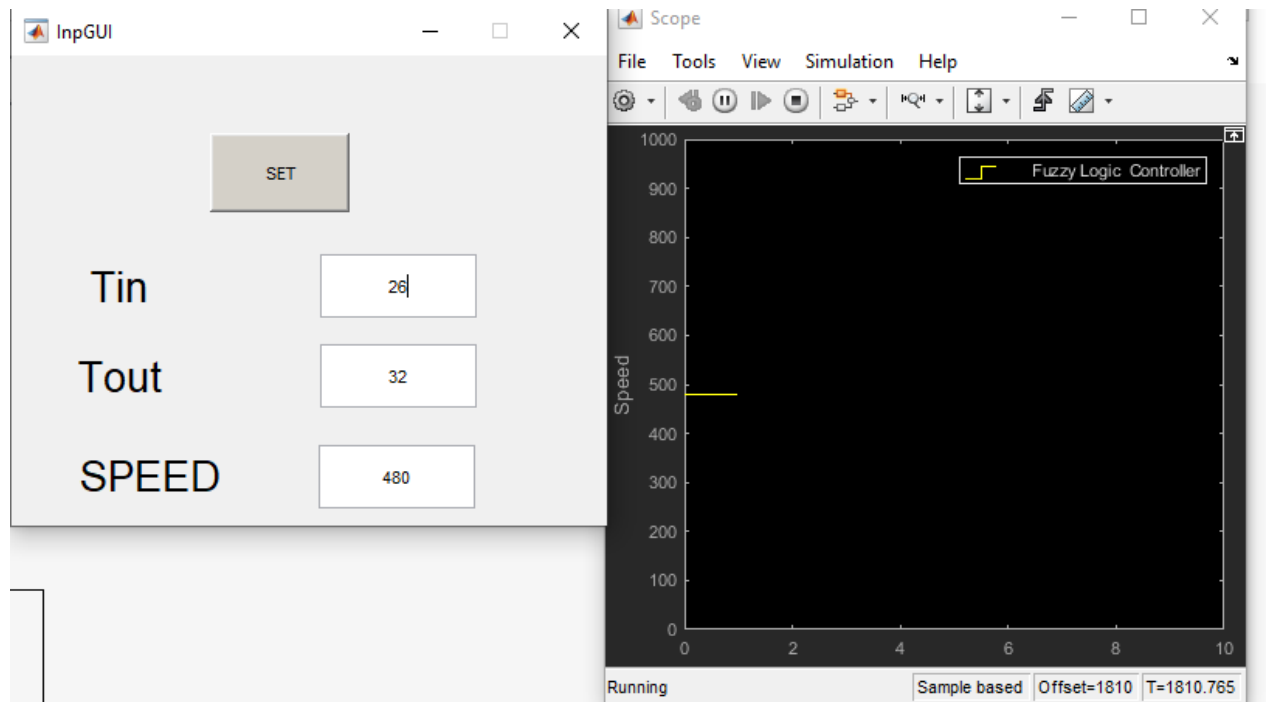
Hình 29: Chạy thử nghiệm lần 1



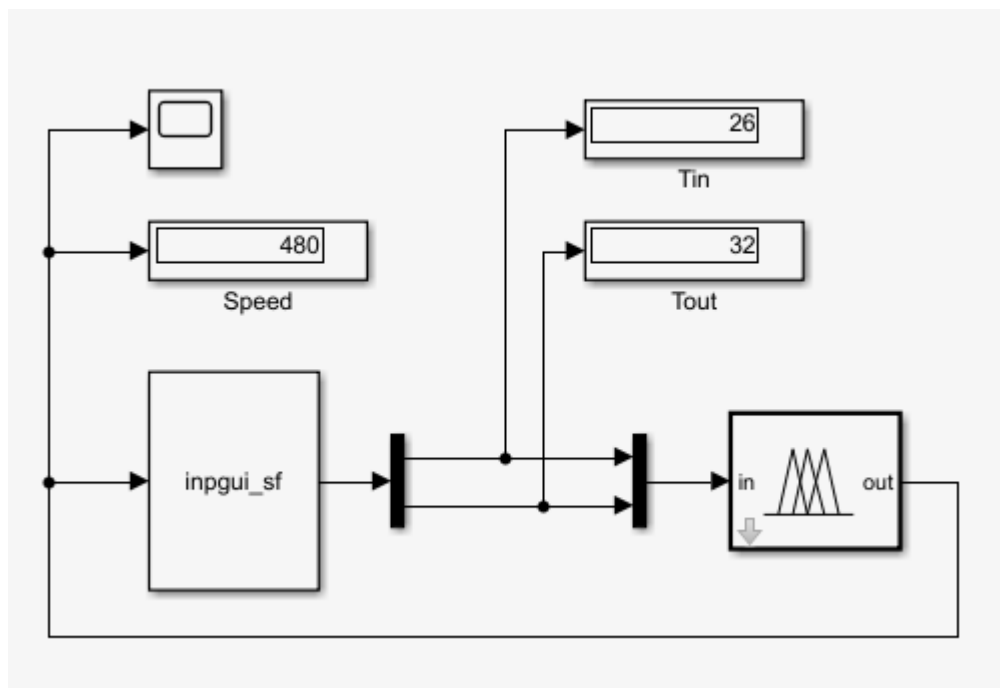


Hình 30: Chạy thử nghiệm lần 1 khối xử lý trung tâm

Khi thay đổi nhiệt độ cảm biến trong phòng từ 22 độ C lên 26 độ C và giữ nguyên nhiệt độ bên ngoài phòng: 32 độ C. Khối Fuzzy Logic sẽ tự động tính toán lại tốc độ quay của quạt.



Hình 31: Chạy thử nghiệm lần 2



Hình 32: Chạy thử nghiệm lần 2 khối xử lý trung tâm

## Chương 4

# KẾT LUẬN VÀ HƯỚNG NGHIÊN CỨU PHÁT TRIỂN

Chủ đề “Điều khiển tốc độ động cơ bằng Fuzzy Logic” vẫn đang là chủ đề được quan tâm nhiều so với phương pháp điều khiển truyền thống, bởi vì có thể tổng hợp được bộ điều khiển mà không cần biết chính xác thành phần và các tham số đầu vào của hệ thống điều khiển một cách chính xác.

Cùng với sự phát triển của khoa học công nghệ, cách mạng công nghiệp 4.0 dự đoán sẽ chuyển hóa toàn bộ thế giới thực của chúng ta sang thế giới số, thay đổi cách tiếp cận của con người ở tất cả các lĩnh vực. Cũng như các nước trên thế giới, Việt Nam có rất nhiều lĩnh vực hiện đang hoạt động trên nền cách mạng 4.0 như: công nghiệp, nông nghiệp... Việc ứng dụng kỹ thuật Fuzzy Logic trong thiết kế bộ điều khiển cho bộ truyền động có tham số biến đổi đang còn khá hấp dẫn, còn nhiều tiềm năng để phát triển. Bởi vì nó khắc phục được nhược điểm của những hướng nghiên cứu khác.

Việc điều khiển tốc độ quạt bằng 2 giá trị cảm biến đầu vào đã thực hiện tốt những yêu cầu như mục tiêu ban đầu đề ra, trong giai đoạn hiện tại và tương lai, bản thân em đã và đang nghiên cứu tiếp những bộ điều khiển với nhiều tham số đầu vào và những tham số này sẽ thay đổi liên tục.

## TÀI LIỆU THAM KHẢO

### Tiếng Việt:

- [1]. Sử dụng Logic Mờ điều khiển động cơ không đồng bộ, Huỳnh Thanh Tuấn, Dương Hoài Nghĩa.
- [2]. Logic Mờ trong hệ thống điều khiển tốc độ động cơ không đồng bộ ba pha, Nguyễn Thị Hiên.

### Tiếng Anh:

- [3]. Simulation of fuzzy logic control for DC servo motor using Arduino based on MATLAB/Simulink Analysis of Technical Loss in Distribution Line System, Narong Mungkung, Nittaya Gomurut, Tanes Tanitteerapan, Somchai Arunrungrusmi, Weerachai Chaokumnerd and Toshifumi Yuji, Proceedings of the 8th WSEAS International Conference on Telecommunications and Informatics, 2009.
- [4]. Implementation of a Fuzzy Logic Speed Controller for a Permanent Magnet DC Motor Using a LowCost Arduino Platform.
- [5]. Caldo, R. B., Seranilla, J. T., Castillo, D. J., Diocales, K. S., Gulle, W. D., Nunez, B. L., & Parreno, C. T. (2015). Design and development of fuzzy logic-controlled dimming lighting system using Arduino microcontroller. 2015 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM).
- [6]. Using fuzzy logic and light-sensor for automatic adjustment of backlight brightness in a mobile computer, Cuing-Hung Cheng; Ying-Wen Bai.