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

1. **Giới thiệu tiểu luận**
   1. **Đặt vấn đề**

Kính thưa PGS.TS Trần Thu Hà, sau khi được học môn xử lí 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. Chính vì thế em quyết định trình bài bài tiểu luận này. Đề tài có ý tưởng của chính bản thân em và tham khảo những dự án có sẵn trên internet.

Việc điều khiển tốc độ quạt trong nhà xưởng hiện nay chủ yếu thực hiện bằng tay chứ chưa có thề điều chỉnh tốc độ hợp lý bằng việc đọc các giá trị cảm biến nhiệt độ. Do đó yêu cầu phải tạo ra đươc 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 liên tục. 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.

Vì vậy, đây là một vấn đề cần có sự đầu tư nghiêm túc để tìm ra một hướng giải pháp tối ưu, hợp lý. Như những gì mà yêu cầu đã đề ra và quyết định tạo ra một giao diện giúp mô phỏng điều khiển tốc độ quạt trên phần mềm MATLAB sử dụng Fuzzy Logic.

## **1.2.Mục tiêu tiểu luận**

Mục tiêu của tiểu luận là thiết kế hệ thống có thể tương tác và điều khiển được tốc độ quạt trên phần mềm MATLAB thông qua Fuzzy Logic.

Fuzzy Logic dựa trên thông tin không đầy đủ của hệ thống, hoạt động giống như cách suy luận của con người để đưa xa cách xử lý thích hợp nhất. Có thể áp dụng với những hệ thống phức tạp mà trước đây chưa xử lý ổn định được. Fuzzy Logic sử dụng kinh nghiệm thực tế của người vận hành, ứng dụng cách xử lý của những chuyên gia. Gần gũi với tư duy suy nghĩ của con người trong việc vận hành.

Ngày nay, bộ điều khiển Fuzzy Logic đã được ứng dụng trong nhiều sản phẩm phục vụ nhu cầu hằng ngày của con người: “Nồi cơm điện, Máy điều hòa không khí, máy giặt”. Ngoài ra, còn được ứng dụng trong những lĩnh vực liên quan đến công nghiệp như: điều khiển máy tàu thủy, hệ lò nhiệt… Nói cách khác Fuzzy logic có thể điều khiển các hệ thống phi tuyến, hệ thống không hình dung được các tham số và mô hình.

Ở nội dung tiểu luận em sẽ tạo ra một giao diện giúp người dùng nhập hai giá trị đầu vào: Tin và Tout để tính toán giá trị tốc độ quạt sử dụng Fuzzy Logic.

Graphical user interface

Description automatically generated

Hình 1: Giao diện mô phỏng ban đầu

Diagram

Description automatically generated

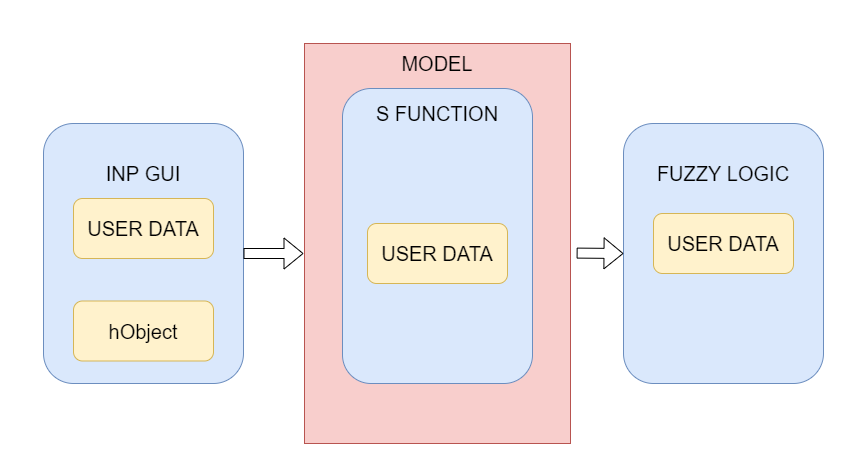
Hình 2: Giao diện ban đầu khối simulink

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 kĩ thuật so với các ngôn ngữ lập trình truyền thống như C, C++, và Fortran.[[1]](#footnote-1) 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 …

Graphical user interface, text, application

Description automatically generated

Hình 3: Giao diện MatLab



Hình 4: Mô hình giao tiếp

Để tiến xa hơn sẽ áp dụng việc điều khiển với nhiều thông số đầu vào và đầu ra hơn. Ở đề tài này hướng nghiên cứu là tạo ra ứng dụng được nền tảng và hệ sinh thái mà phần MatLab đã có sẵn giúp liên kết hai thanh phần: “Giao diện người dùng” và “phần xử lý trung tâm”, từ đó dễ dàng thay đổi các giá trị giả lập. Để việc điều chỉnh tốc độ quạt trở nên gọn gàng không tốn quá nhiều code.

Kết quả đạt được với ý tưởng này đã thực hiện được việc mô phỏng, hệ thống giao diện người dùng và câu lệnh hoàn chỉnh. Hướng phát triển đề xuất sẽ thêm nhiều giá trị đầu vào để việc điều khiển sát với thực tế.

1. **Nội dung tiểu luận**
   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 nghi 4 gải quyết một vấn đề đó chính là: “Nếu … Thì …”. Bản chất của điều khiển fuzzy logic chính là tổng quát hóa các điểu 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 tuyền 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.

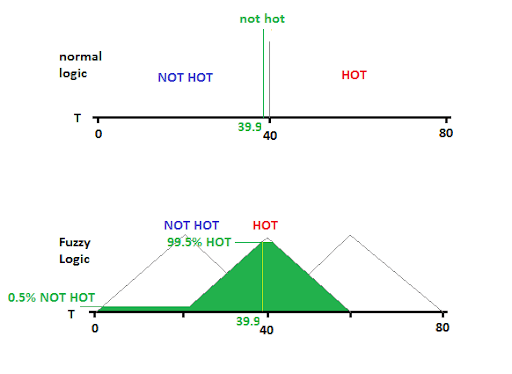
+ Gần gũi với suy nghĩ tư duy của người vận hành.

+ Dễ thay đổi khi hệ thống cần sự nâng cấp.

* 1. **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 rang. 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:

Như vậy, ta có thể thấy giá trị của 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:



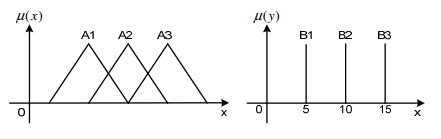
Hình 5: Mô hình hóa ngõ vào

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 95% 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 luật 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. **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 Mamdami và mô hình mờ Sugeno.

Mô hình mờ Mamdami là bộ điều khiển trong dó 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[[2]](#footnote-2). 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 Mamdami 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 Mamdami và Sugeno cho ra hai giá trị tương đương nhau.



Hình 6: Mô hình mờ Mamdami 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:

+ Nếu x là A1 thì y là B1.

+ Nếu x là A2 thì y là B2.

+ Nếu x là A3 thì y là B3.

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

+ Nếu x là A1 thì y là 5.

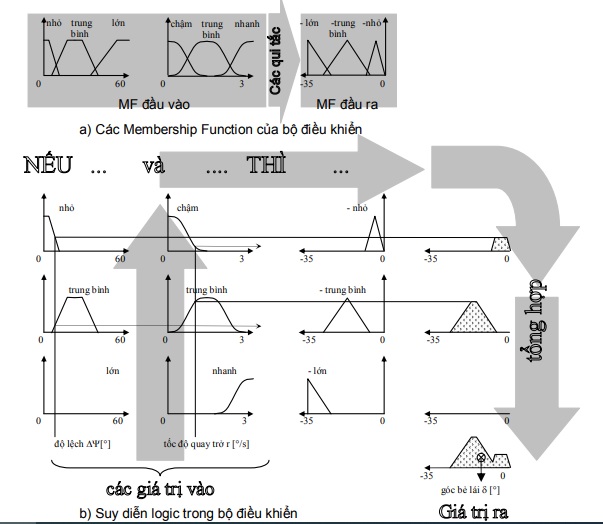
+ Nếu x là A2 thì y là 10.

+ Nếu x là A3 thì y là 15.

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.

* 1. **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.



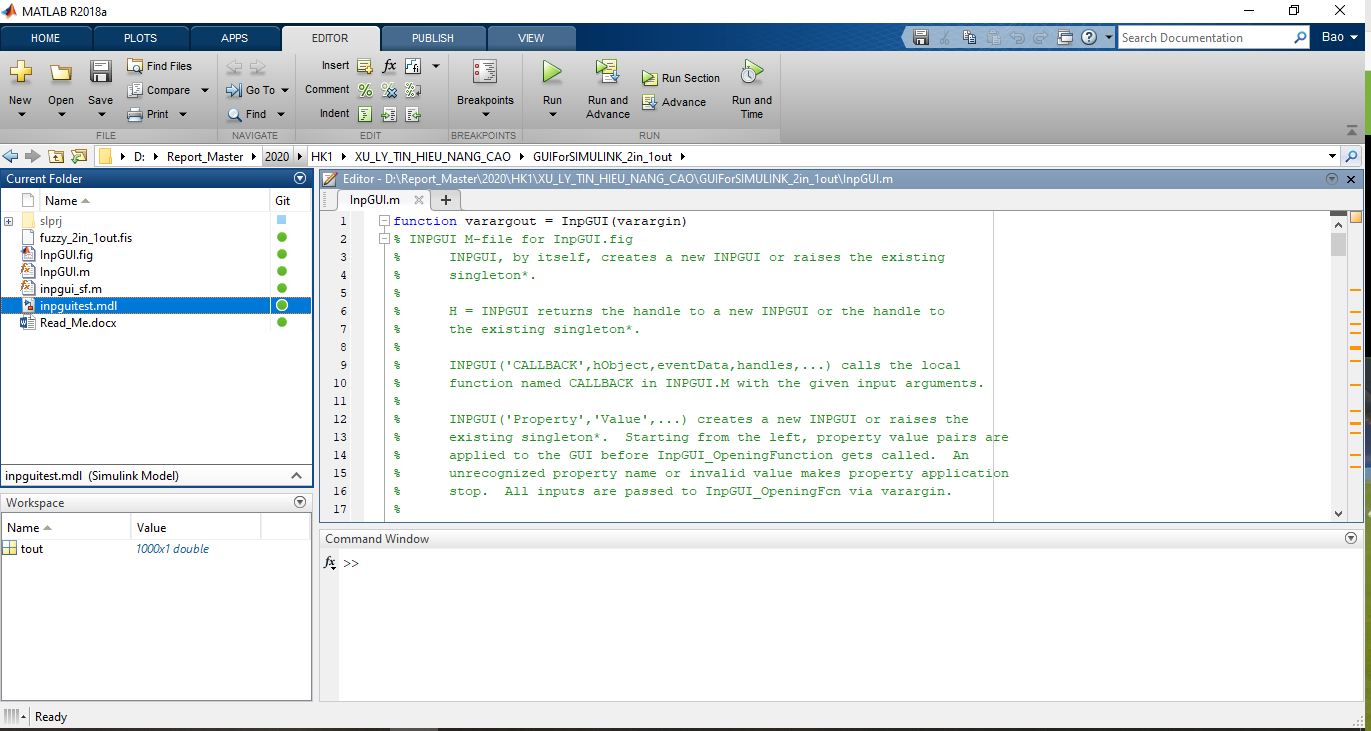
Hình 7: Nguyên lý hoạt động khối Fuzzy Logic

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. **Cơ sở ứng dụng cho mô hình**

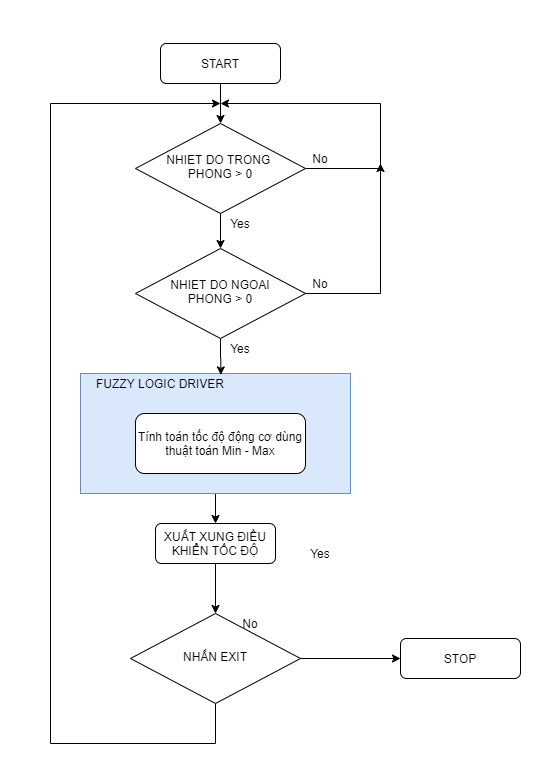
Cơ sở ứng dụng cho mô hình:

* + 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 8: Giao diện MatLab

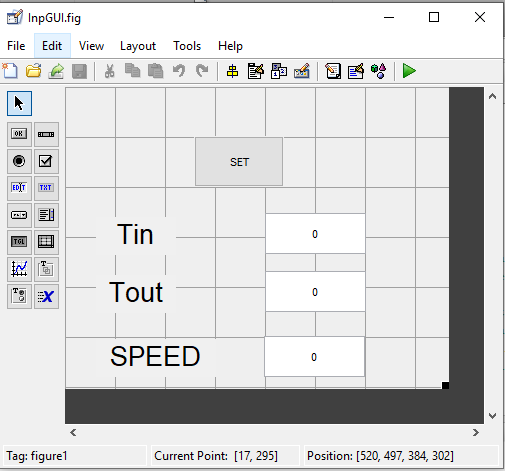
1. **Thiết kế lập trình mô hình**
   1. **Sơ Đồ Khối**

****

Hình 9: Lưu đồ giải thuật hệ thống

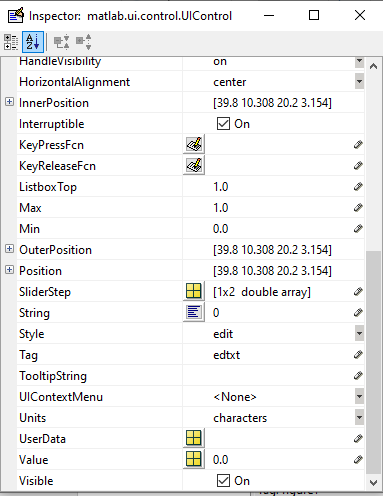
Hệ thống gồm 2 cảm biến dùng để xác định giá trị nhiệt độ bên trong phòng và bên ngoài phòng. Nếu nhiệt độ bên trong phòng và bên ngoài phòng đều lớn hơn 0 thì hai giá trị này sẽ được đưa vào bộ Fuzzy Logic Driver, sử dụng thuật toán tìm “Max – Min”, từ đó tính toán ra được tốc độ cần thiết cho quạt. 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.

* 1. **Tạo Giao Diện**

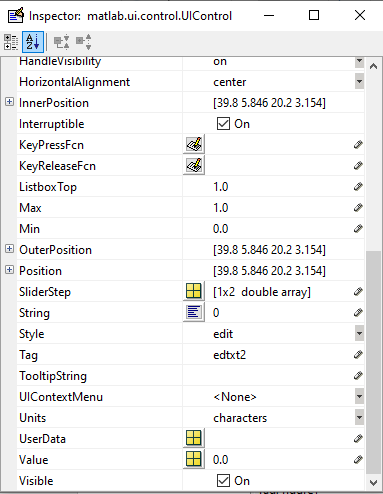


Hình 10: Giao diện người dùng

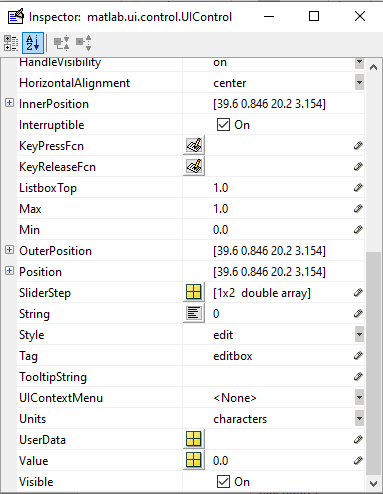
Để tiến hành tạo giao diện tương tác giữa người dùng và MatLab trước hết ta chọn “guide” trên cửa sổ gõ lệnh của MatLab, sau đó lần lượt sử dụng các khối: Text để hiển thị văn bản: “Tin, Tout, SPEED”, Button để tạo nút “SET” và EditText để hiển thị các giá trị mà mình muốn thể hiện.



Hình 11: Thay đổi tên tag edtxt



Hình 12: Thay đổi tên tag edtxt2



Hình 13: Thay đổi tên tag editbox

Sau khi đã tạo được giao diện như hình ta tiến hình 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.

Text

Description automatically generated

Hình 14: 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 cài giá trị vào biến “hObject” trong Matlab.

Text

Description automatically generated

Hình 15: Lấy giá Tin trị từ giao diện

Text

Description automatically generated

Hình 16: 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: “editfiled, edit2filed”. 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.

Text

Description automatically generated

Hình 17: Lấy giá trị Tout từ giao diện

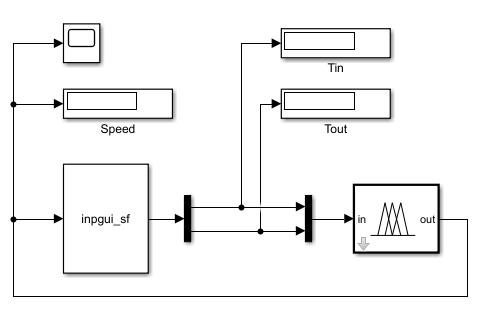
Text

Description automatically generated

Hình 18: Hiển thị giá trị Tout trên giao diện

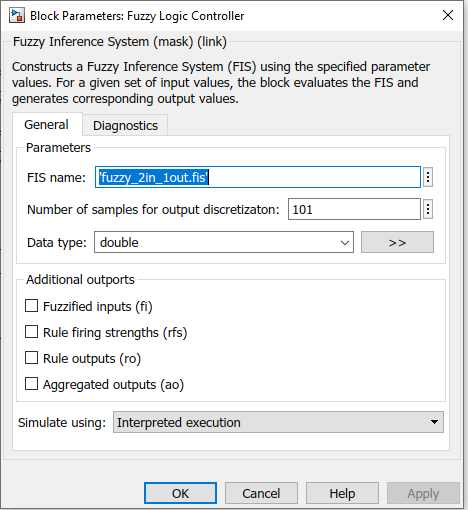
* 1. **Tạo các khối liên kết**

Tạo các khối Xử lý trung tâm sử dụng MatLab



Hình 19: Tạo khối liên kết

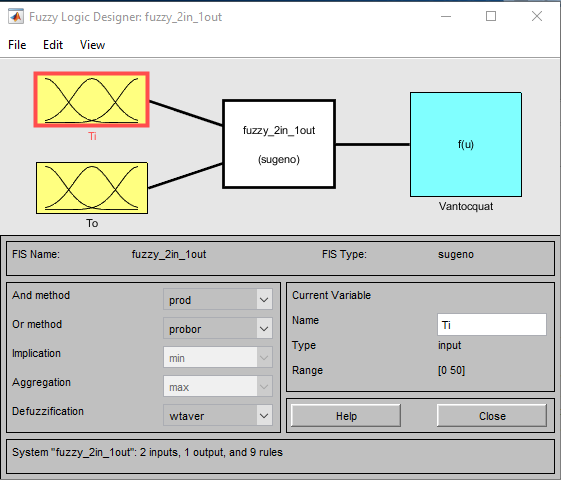
Ta tiếp tục tạo phần Simulink cho hệ thống bằng cách gõ: “Simulink” trên thanh gõ lệnh của MatLab. Gọi các khối: “Display” để hiển thị các giá trị Nhiệt độ bên trong phòng và ngoài phòng và Tốc độ hiện tại của quạt giống như các khối Textedit đã trình bày ở phần: “tạo giao diện tương tác người dùng”.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



Hình 20: Gọi chương trình fuzzy logic

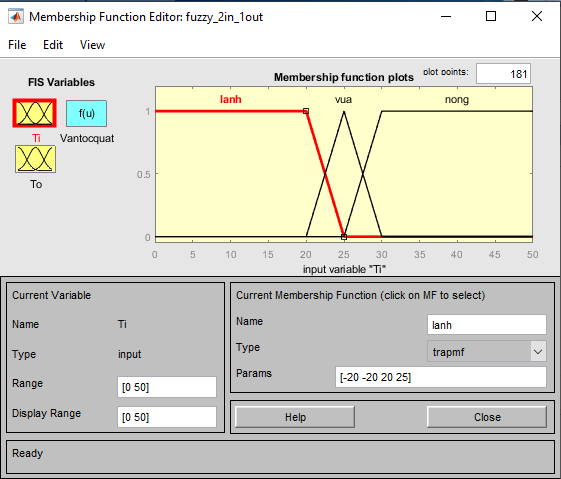
Gọi khối Fuzzy-Logic và tiến hành gán chương trình fuzzy\_logic: “fuzzy\_2in\_1out.fis”. Chương trình này giúp chúng ta tính toán được giá trị quạt bằng hai giá trị ở ngõ vào với các điều kiện được khai báo như hình bên dưới.

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.

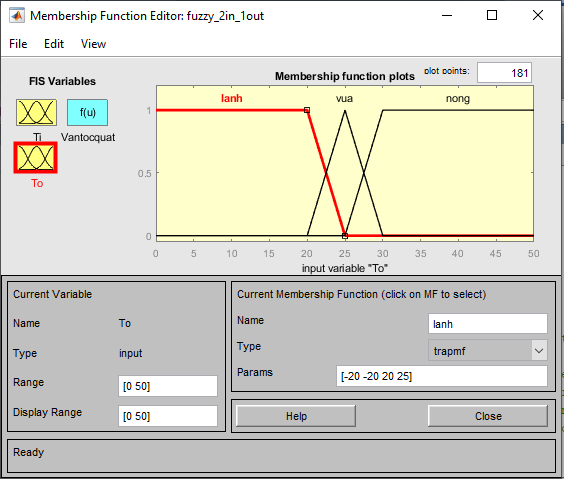


Hình 21: Tạo khối fuzzy logic

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 thang. Trường hợp của Tout khi mờ hóa ngõ vào các điều kiện giống như với Tin.

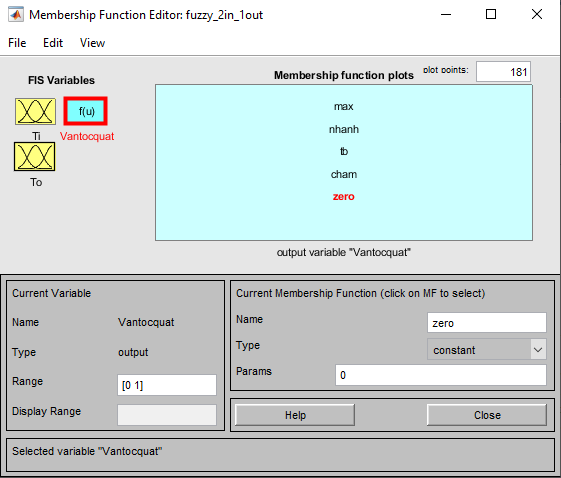


Hình 22: Mờ hóa các ngõ vào fuzzy logic(Tin)



Hình 23: 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].

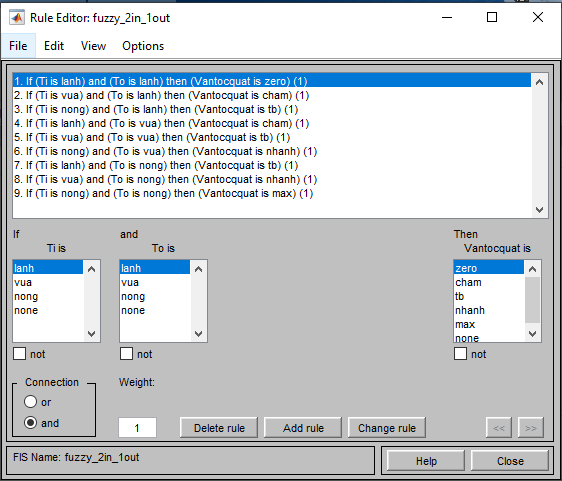


Hình 24: Mờ hóa ngõ ra

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 đút 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 25: Tạo luật mờ cho khối fuzzy logic

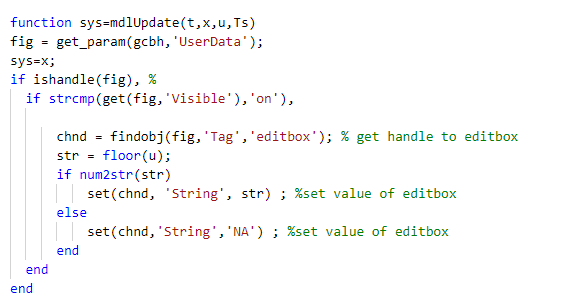
Sau khi tạo các luật mờ thành công, để có sự liên kết giữa hai khối xử lý trung tâm và giao diện người dùng thì ta tiến hành tạo khối S-function. Trong phần khởi tạo ta khai báo hai trường hợp khi nhấn “SET” đó là nhấn và nhả.

Text

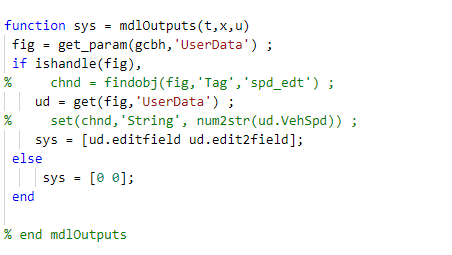
Description automatically generated

Hình 26: Khai báo chương trình liên kết

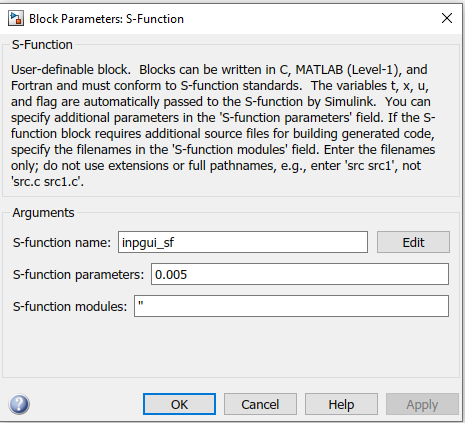
Khi nhấn nút “SET” thì chương trình sẽ tiến hành lấy giá trị ở biến editbox làm đầu vào cho khối s\_function. Sau khi đã có giá trị đầu vào thì khối s-function cũng sẽ đưa giá trị này thành đầu vào của khối Fuzzy\_Logic.Từ hai giá trị này sẽ tính toán ra được giá trị cho quạt, giá trị này cũng sẽ trở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



Hình 27: Lấy Giá Trị cho editbox



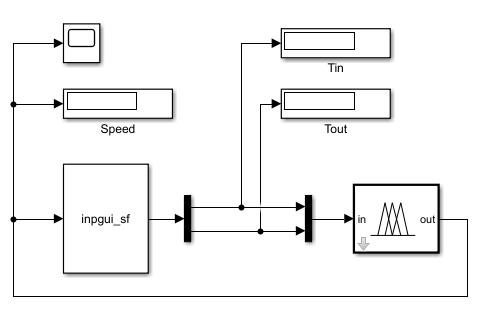
Hình 28: Cập nhật Giá Trị cho Hàm input\_sf



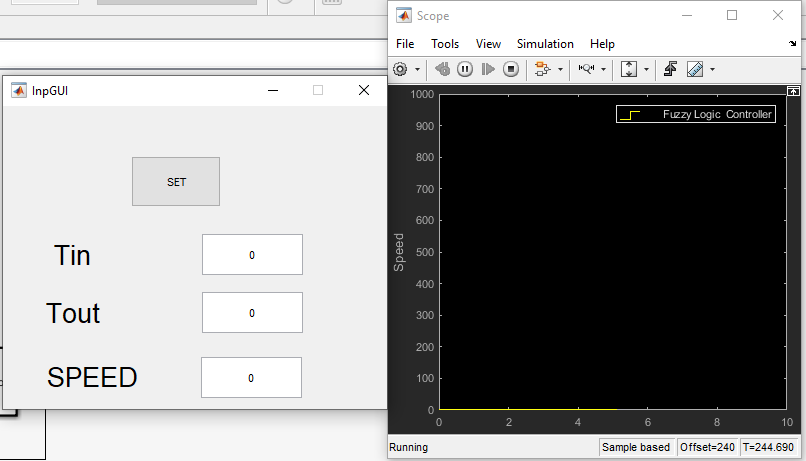
Hình 29: Gán hàm cho khối S-function

1. **Mô phỏng thực tế**

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.

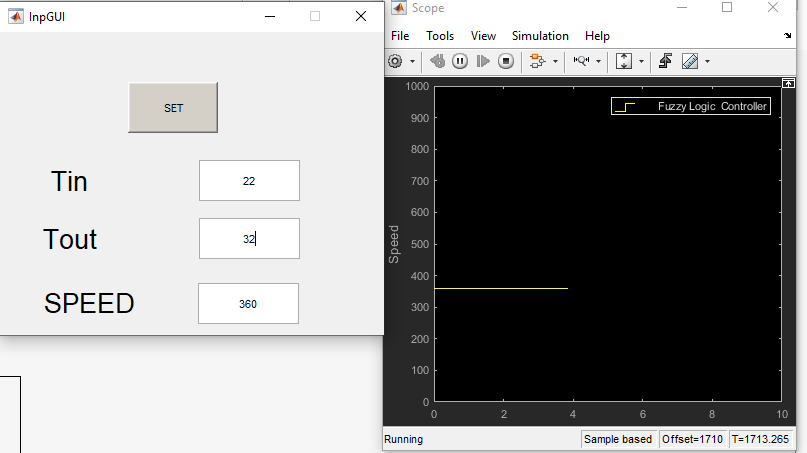
****

Hình 30: Các thành phần của giao diện

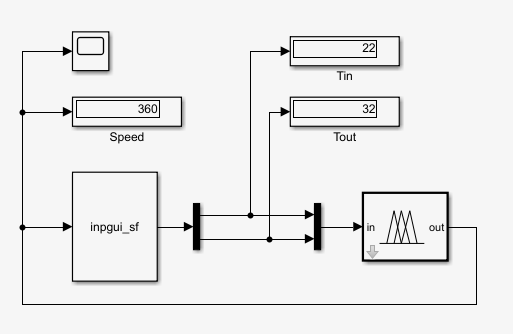
****

Hình 31: 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.

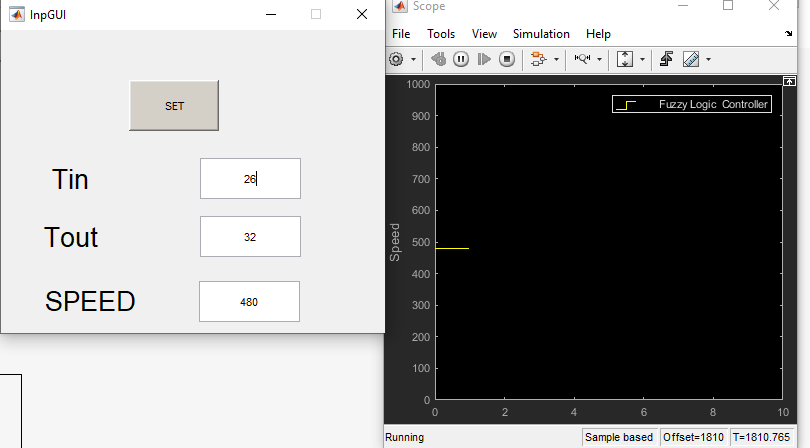
**

Hình 32: Thử nghiệm Lần 1

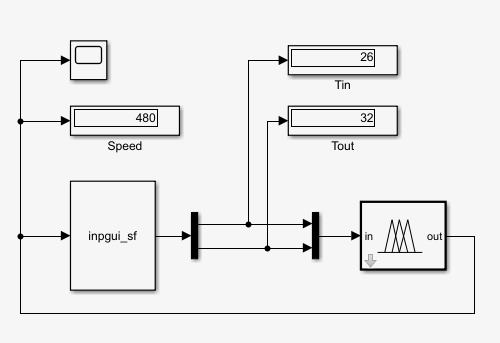
**

Hình 33: 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 34: Thử nghiệm lần 2

**

Hình 35: Thử nghiệm lần 2 khối xử lý trung tâm

1. **Kết luận**

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

**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 lí 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

Danh sách tài liệu tham khảo:

+ [ĐIỀU KHIỂN THÔNG MINH.pdf (hcmute.edu.vn)](http://cyu.hcmute.edu.vn/Resources/Docs/SubDomain/cyu/%C4%90I%E1%BB%80U%20KHI%E1%BB%82N%20TH%C3%94NG%20MINH.pdf)

+ [Logic mờ – Wikipedia tiếng Việt](https://vi.wikipedia.org/wiki/Logic_m%E1%BB%9D#:~:text=L%C3%B4gic%20m%E1%BB%9D%20(ti%E1%BA%BFng%20Anh%3A%20Fuzzy,l%C3%B4gic%20v%E1%BB%8B%20t%E1%BB%AB%20c%E1%BB%95%20%C4%91i%E1%BB%83n.)

+ [Điều khiển Logic mờ (Fuzzy Logic) (baoanjsc.com.vn)](https://baoanjsc.com.vn/tin-hang/dieu-khien-logic-mo--fuzzy-logic-_2_1_41712_vn.aspx)

Mục Lục

[**1.** **Giới thiệu tiểu luận** 1](#_Toc57847804)

[**1.1.** **Đặt vấn đề** 1](#_Toc57847805)

[**1.2.Mục tiêu tiểu luận** 1](#_Toc57847806)

[**2.** **Nội dung tiểu luận** 5](#_Toc57847807)

[**2.1.** **Tổng Quan Fuzzy Logic** 5](#_Toc57847808)

[**2.2.** **Định Nghĩa Tập Mờ** 5](#_Toc57847809)

[**2.3.** **Xây dựng mô hình mờ** 7](#_Toc57847810)

[**2.4.** **Nguyên lý hoạt động của Fuzzy Logic** 8](#_Toc57847811)

[**2.5.** **Cơ sở ứng dụng cho mô hình** 9](#_Toc57847812)

[**3.** **Thiết kế lập trình mô hình** 10](#_Toc57847813)

[**3.1.** **Sơ Đồ Khối** 10](#_Toc57847814)

[**3.2.** **Tạo Giao Diện** 11](#_Toc57847815)

[**3.3.** **Tạo các khối liên kết** 17](#_Toc57847816)

[**4.** **Mô phỏng thực tế** 26](#_Toc57847817)

[**5.** **Kết luận** 30](#_Toc57847818)

Mục Lục Hình Ảnh

[Hình 1: Giao diện mô phỏng ban đầu 2](#_Toc57849011)

[Hình 2: Giao diện ban đầu khối simulink 3](#_Toc57849012)

[Hình 3: Giao diện MatLab 4](#_Toc57849013)

[Hình 4: Mô hình giao tiếp 4](#_Toc57849014)

[Hình 5: Mô hình hóa ngõ vào 6](#_Toc57849015)

[Hình 6: Mô hình mờ Mamdami và Sugeno 7](#_Toc57849016)

[Hình 7: Nguyên lý hoạt động khối Fuzzy Logic 8](#_Toc57849017)

[Hình 8: Giao diện MatLab 9](#_Toc57849018)

[Hình 9: Lưu đồ giải thuật hệ thống 10](#_Toc57849019)

[Hình 10: Giao diện người dùng 11](#_Toc57849020)

[Hình 11: Thay đổi tên tag edtxt 12](#_Toc57849021)

[Hình 12: Thay đổi tên tag edtxt2 13](#_Toc57849022)

[Hình 13: Thay đổi tên tag editbox 14](#_Toc57849023)

[Hình 14: Khai báo khối kết nối 14](#_Toc57849024)

[Hình 15: Lấy giá Tin trị từ giao diện 15](#_Toc57849025)

[Hình 16: Hiển thị giá trị Tin lên giao diện 15](#_Toc57849026)

[Hình 17: Lấy giá trị Tout từ giao diện 16](#_Toc57849027)

[Hình 18: Hiển thị giá trị Tout trên giao diện 16](#_Toc57849028)

[Hình 19: Tạo khối liên kết 17](#_Toc57849029)

[Hình 20: Gọi chương trình fuzzy logic 18](#_Toc57849030)

[Hình 21: Tạo khối fuzzy logic 19](#_Toc57849031)

[Hình 22: Mờ hóa các ngõ vào fuzzy logic(Tin) 20](#_Toc57849032)

[Hình 23: Mờ hóa các ngõ vào fuzzy logic(Tout) 21](#_Toc57849033)

[Hình 24: Mờ hóa ngõ ra 22](#_Toc57849034)

[Hình 25: Tạo luật mờ cho khối fuzzy logic 23](#_Toc57849035)

[Hình 26: Khai báo chương trình liên kết 24](#_Toc57849036)

[Hình 27: Lấy Giá Trị cho editbox 24](#_Toc57849037)

[Hình 28: Cập nhật Giá Trị cho Hàm input\_sf 25](#_Toc57849038)

[Hình 29: Gán hàm cho khối S-function 25](#_Toc57849039)

[Hình 30: Các thành phần của giao diện 26](#_Toc57849040)

[Hình 31: Giao diện khi chạy mô phỏng MatLab 26](#_Toc57849041)

[Hình 32: Thí nghiệm 1 27](#_Toc57849042)

[Hình 33: Thí nghiệm 2 27](#_Toc57849043)

[Hình 34: Thí nghiệm 3 28](#_Toc57849044)

[Hình 35: Thí nghiệm 4 28](#_Toc57849045)

1. [Giới thiệu ngắn gọn về Matlab - Matlab4vn (google.com)](https://sites.google.com/site/matlab4vn/basic/gioithieu) [↑](#footnote-ref-1)
2. [Microsoft Word - Giao trinh Dieu khien mo va mang noron\_DaSua\_080519 (uneti.edu.vn)](https://khoadien.uneti.edu.vn/wp-content/uploads/2019/12/TLHT_-Dieu-khien-mo-va-mang-noron.pdf) [↑](#footnote-ref-2)