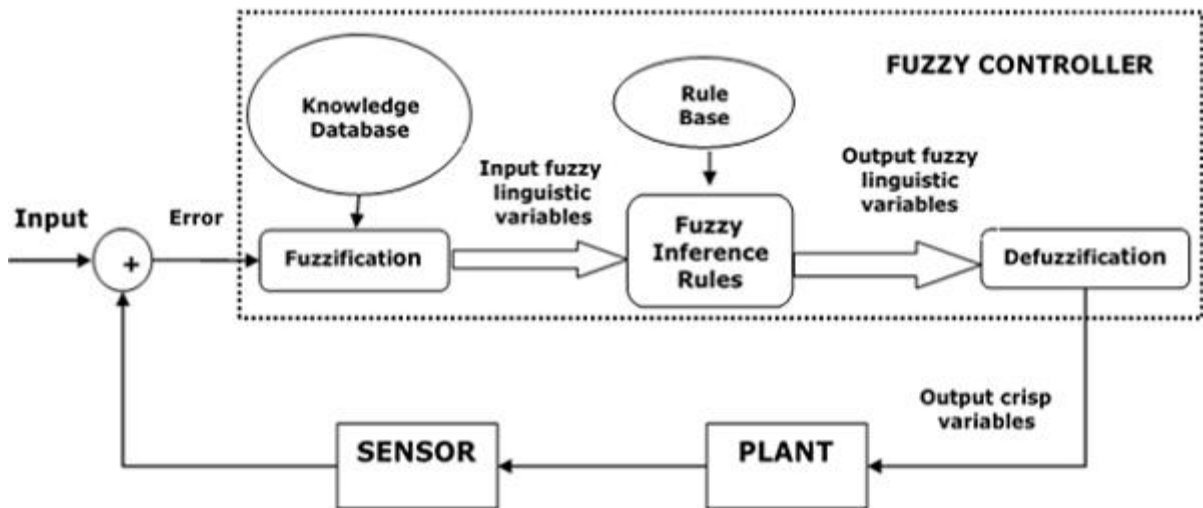


FUZZY LOGIC CONTROLLER

Câu 1:

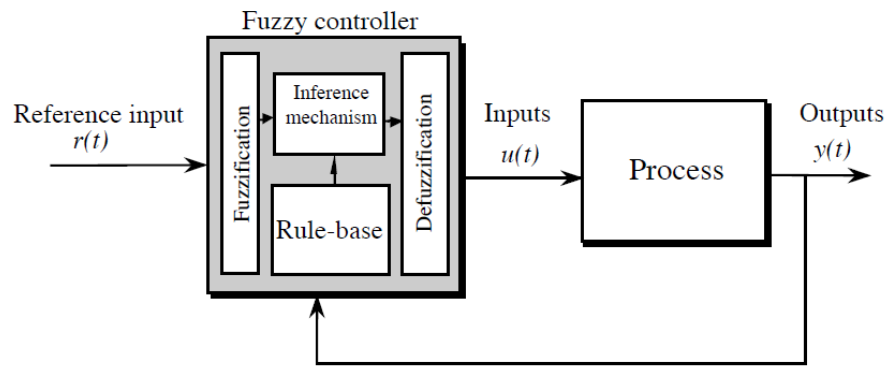
a) Hãy cho biết yếu tố nào quan trọng nhất khi thiết kế bộ điều khiển Fuzzy. Hãy trình bày các bước khi thiết kế bộ điều khiển Fuzzy.



Yếu tố quan trọng khi thiết kế bộ điều khiển Fuzzy:

- Hiểu được đặc tính của cơ hệ đang xét.(Thực ra so với những bộ điều khiển thông thường khác thì bộ điều khiển fuzzy sẽ không chú trọng quá nhiều vào tính chất và đặc tính của cơ hệ. Nhưng để thiết kế một bộ điều khiển Fuzzy cũng như những bộ khác thì cũng cần quan trọng vào đặc tính của cơ hệ).
- Lựa chọn số biến ngôn ngữ cho input và output và *khoảng range* cho từng biến ngôn ngữ đấy.
- Bộ luật (rule-base) cho tập fuzzy đang khảo sát.

Các bước khi thiết kế Fuzzy Controller



Bước 1: Xác định tập hợp các quy tắc mờ (các thông số bao gồm biến đầu vào/ đầu ra, biến trạng thái của plant, Membership Function (MFs), miền giá trị, linguistic variable). Cấu hình cho tập con Fuzzy (hay Fuzzy sets) – Chia số tập con fuzzy và mỗi tập con sẽ gán với một tên biến ngôn ngữ. Luôn đảm bảo rằng các tập fuzzy này bao gồm các thành phần trong thông tin mà chúng ta xét. Nói cách khác ở bước này, ngoài việc xác định tập mờ, chúng ta tiến hành lựa chọn biến ngôn ngữ phù hợp cho plant đang khảo sát, MFs, số lượng biến ngôn ngữ nhiều hay ít tùy vào mức độ đáp ứng của phần cứng (ví dụ như vi điều khiển).

Bước 2: Làm mờ (Fuzzification) là việc chuyển đổi giá trị input để bộ suy luận mờ (Inference Mechanism) có thể dễ dàng hoạt động và áp dụng rule-base. Cụ thể là Fuzzyfication process sẽ lấy giá trị input và tìm ra giá trị số của MFs để xác minh đó là biến nào.

Bước 3: Đặt ra các bộ luật (rule-base) để chỉ định mức độ quan hệ giữa input và output là như thế nào. Bước này khá quan trọng, là điểm mấu chốt của Fuzzy Controller, bộ luật sẽ dựa trên If [condition] Then [action] (nếu [điều kiện] thì [hành động]). Cùng với đó việc lựa chọn phép toán so sánh AND/OR sẽ quyết định cho bước tiếp theo.

Bước 4: Tổng hợp các giá trị sau khi qua bộ suy luận mờ (Inference Mechanism or Inference Engine) gồm giá trị MFs sau suy luận với output MFs. Nói cách khác, sau khi thực hiện xét rule-base thông qua các toán tử AND/OR sẽ cho ra các giá trị MIN/MAX tương ứng với giá trị input MFs xét trong rule gọi là μ (đại diện cho giá trị MFs tương ứng). Từ μ này kết hợp với output MFs sẽ cho ra miền diện tích với chiều cao là μ .

Bước 5: Kết hợp tất cả các kết quả của **Bước 4** để cho ra phân phối đầu ra (output distribution).

Bước 6: Giải mờ (Defuzzification) sẽ đưa ra giá trị output để làm đầu vào cho plant (tín hiệu điều khiển/ tín hiệu đầu vào) của cơ hệ khảo sát. Nói chung thì Defuzzification sẽ áp dụng các phương pháp như COG hay COA để đưa giá trị mà ta đã tổng hợp các bộ fuzzy output ở **Bước 5** về lại giá trị đại số tương ứng với các giá trị input đưa vào bộ điều khiển Fuzzy.

b) Đặc tính chính của bộ điều khiển Fuzzy là gì? Tại sao có sự sai lệch kết quả đầu ra khi giải mờ bằng phương pháp COG (Centre of Gravity) và phương pháp CA (Centre Average) ?

Đặc tính chính của bộ điều khiển Fuzzy có thể nói tới về tính ứng dụng của nó, áp dụng hầu hết các cơ hệ mà không cần quan tâm đến việc *cơ hệ có tuyến tính hay phi tuyến tính* và tính áp dụng rộng rãi vào công nghệ đời sống. Bởi chính việc áp dụng hầu hết mọi cơ hệ nên các hệ phi tuyến có trong các ứng dụng đời sống như máy giặt, tủ lạnh, hàng không vũ trụ...

Phương pháp COG và phương pháp CA đều là phương pháp dùng để giải mờ. Trong khi đó COG thì phụ thuộc vào dạng của MFs nếu là Triangular/ Trapezoidal thì công thức tính diện tích hình sẽ dễ dàng hơn các dạng còn lại như Sigmoid/ Bell/ Gaussian. Ngược lại thì phương pháp CA sẽ không quan tâm nhiều đến hình dạng của MFs vì chúng chỉ sử dụng giá trị tâm của MFs trong output fuzzy. Với sự khác nhau cơ bản này đã cho ra được kết quả của hai phương pháp này khác nhau hoàn toàn.

c) Hãy giải thích tại sao biến đầu vào của bộ điều khiển Fuzzy thông thường sử dụng 02 thống số là sai số e và đạo hàm \dot{e} ? Trong trường hợp sử dụng duy nhất chỉ 01 thống số đầu vào là e có đảm bảo kết quả đầu ra mong muốn hay không?

Thông thường những bộ điều khiển thông thường luôn quan tâm đến sai số của hệ thống, thực tế thì bộ điều khiển Fuzzy cũng vậy cũng có thể sử dụng 01 input đầu vào là sai số e hay một số sai số khác liên quan đến cơ hệ khảo sát. Nhưng vì đáp ứng của

cơ hệ mà chúng ta thường sử dụng thêm đạo hàm e' để bộ điều khiển thông minh Fuzzy có thể tối ưu được việc điều khiển hệ thống đấy.

Nếu sử dụng nhiều hơn một thông số đầu vào thì việc chúng ta lựa chọn biến ngôn ngữ, membership function và các khoảng giá trị (range) nhiều hơn, kèm với đó và bộ quy tắc (rule-base) sẽ trở nên phức tạp (ví dụ 02 input(e, e') và 1 output chẳng hạn, nếu lựa chọn số biến ngôn ngữ cho mỗi thông số vào/ra là 3 thì tổng cộng sẽ có thông thường là 9 rule-base). Nhưng nếu ta chỉ sử dụng đầu vào chỉ sai số e , bộ điều khiển Fuzzy sẽ đơn giản hơn, hoạt động gần như 02 input bởi lúc này rule-base giảm xuống (có thể nói rằng giảm từ 2-D xuống còn 1-D), tất nhiên số lượng biến sẽ ít hơn thông thường. Hệ 01 input 1 output hay gọi là SISO (Single Input Single Output).

d) Khi thiết kế bộ điều khiển Fuzzy, có thể nói hàm membership dạng tam giác cho kết quả chính xác hơn hàm membership dạng Gaussian hay không? Tại sao?

Khi thực hiện thiết kế bộ điều khiển Fuzzy, điều quan trọng không kém bộ quy tắc (rule-base) mà còn là dạng Membership function vì nó ảnh hưởng trực tiếp đến việc làm mờ cũng như giải mờ cho bộ điều khiển. Đối với một số cơ hệ có đáp ứng đơn giản, cơ bản phổ thông và cơ hệ có thể tuyến tính hoá được thì các dạng MFs như tam giác (Triangular) hay hình thang (Trapezoidal) sẽ phù hợp và đủ đáp ứng cho hệ thống cũng như phần cứng ứng dụng xử lý cho hệ thống đấy. Mức độ phức tạp của dạng hàm sẽ giúp cho chúng ta giải quyết những bài toán khác, những cơ hệ phi tuyến khác không còn ở mức đơn giản và tín hiệu mong muốn thay đổi bất thường theo thời gian, cũng có thể nói rằng nếu áp dụng các dạng hàm khác như Gaussian/Bell/Sigmoidal vào các bài phổ thông vẫn sẽ cho kết quả chính xác hơn nhưng đôi khi vẫn không thể cho đáp ứng được đặc trưng của bài toán đang xét. Vì thế còn tùy thuộc vào cơ hệ như thế nào ta sẽ lựa chọn dạng hàm MFs để ứng dụng tối ưu nhất.

e) Khi thiết kế miền giá trị cho các biến ngôn ngữ, miền giá trị $[min, max]$ của biến đầu vào/ đầu ra của Fuzzy đóng vai trò như thế nào?

Là một trong những đặc trưng quan trọng của bộ điều khiển, ảnh hưởng trực tiếp đến đáp ứng đầu ra cũng như đầu vào. Miền giá trị $[min, max]$ của từng biến ngôn ngữ

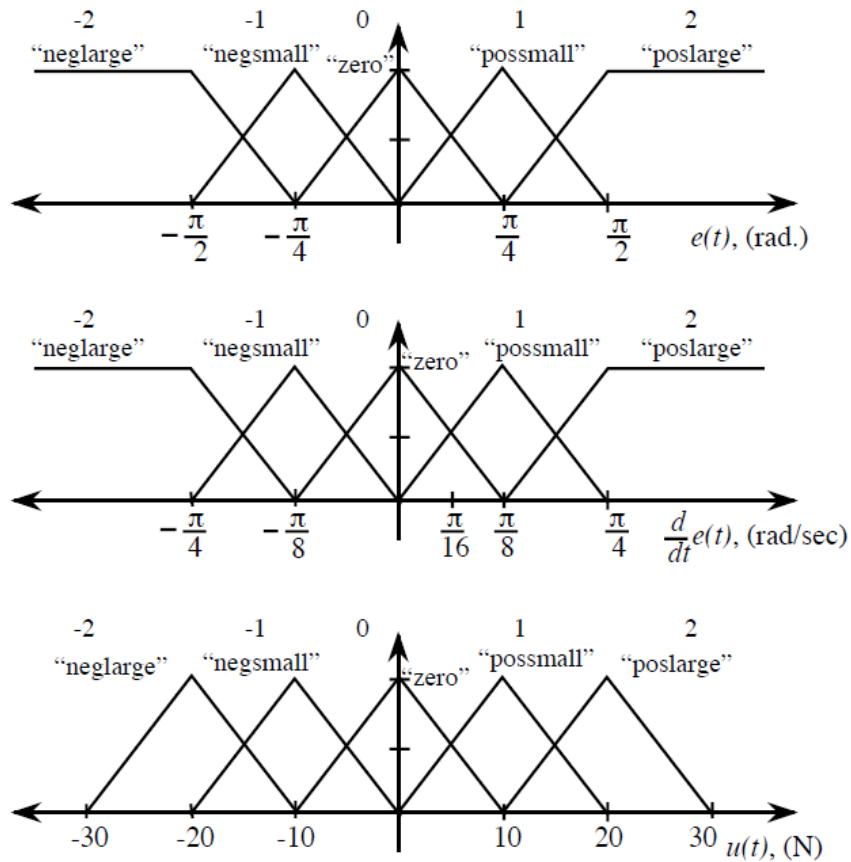
có vai trò lớn nhất là sau khi giải mờ đến bước mà bộ suy luận kết hợp với rule-base để cho ra các giá trị output fuzzy set, thì nếu như miền giá trị đặt ra [min, max] nhỏ, không chứa các giá trị input lấy từ cơ hệ trong quá trình bộ điều khiển hoạt động. Mà tập input sau khi giải mờ sẽ liên quan đến việc đặt rule-base vì thế nếu miền giá trị không chứa giá trị mà bộ suy luận tìm ra, kết quả một số luật sẽ không được thực hiện và cho ra đáp ứng hệ thống không còn đúng. Để khắc phục điều này ta cần hiểu rõ được behavior của cơ hệ để từ đó có những điều chỉnh phù hợp hoặc là điều chỉnh rule-base hoặc là điều chỉnh miền giá trị [min, max].

f. Nếu nói số lượng biến ngôn ngữ của một biến vào/ra nào đó càng nhiều thì sẽ cho kết quả đầu ra càng chính xác là đúng hay sai? Tại sao?

Một cơ hệ trong thực tế cũng như trải qua vài bước để mô phỏng lại đi chẳng nữa cũng khó có thể nhận được kết quả đúng chính xác do còn phải xử lý các loại nhiễu ảnh hưởng đến. Bởi một cơ hệ hoàn hảo nhất là không sử dụng bất kì một bộ điều khiển nào. Đối với bộ điều khiển thông minh Fuzzy cũng không có ngoại lệ, biến ngôn ngữ có vai trò trong việc chia miền giá trị thành các “biến tên” cụ thể với một miền giá trị con xác định trong đây. Vì là điều khiển Fuzzy nên những gì con người không biết thì chúng sẽ là mờ và tìm cách để điều khiển chúng. Việc thêm nhiều biến ngôn ngữ cũng vậy giúp chúng ta lấy được nhiều thông tin hơn, đáp ứng cho cơ hệ ở những điểm giá trị còn chưa xác định hay nói cách khác ta vẫn chưa hiểu cụ thể tại một số vị trí cơ hệ đáp ứng linh hoạt khác nhau không dự đoán được. Vì vậy số lượng biến ngôn ngữ càng nhiều sẽ giúp kết quả đầu ra chính xác hơn so với ít biến ngôn ngữ hơn, nhưng cũng mắc phải những khó khăn trong việc đặt rule-base và áp dụng vào phần cứng để điều khiển cơ hệ. Mặt khác, còn tùy vào cơ hệ không nhất thiết nhiều sẽ chính xác bởi chuyện ta hiểu được cơ hệ như thế nào đã cho ta được số lượng biến cần phải có nếu hơn số lượng đấy thì chỉ nhận thêm một số đặc trưng khác của cơ hệ mà ta dự đoán là sẽ có mà thôi. Tóm lại số lượng biến ngôn ngữ của một biến vào/ra nào đó càng nhiều sẽ cho kết quả đầu ra càng chính xác và phụ thuộc vào cơ hệ khảo sát.

Câu 2:

Cho các dạng MFs của đầu vào, đầu ra và bảng luật như sau:



"force" u		"change-in-error" \dot{e}				
		-2	-1	0	1	2
"error" e	-2	2	2	2	1	0
	-1	2	2	1	0	-1
	0	2	1	0	-1	-2
	1	1	0	-1	-2	-2
	2	0	-1	-2	-2	-2

Hãy tính giá trị đầu ra u của bộ điều khiển khi $e = 3\pi/8$ và $\dot{e} = \pi/16$

Làm mờ (Fuzzification or Fuzzifier)

Nếu $e = 3\pi/8$ thì $\mu(3\pi/8) = 0,5$, điều này cho thấy có thể kết luận rằng nó chắc chắn $e = 3\pi/8$ là "possmall" và cũng có thể kết luận chắc chắn nó là "poslarge".

Hay $\mu_{possmall}(e(t)) = \mu_{poslarge}(e(t)) = 0,5$

Tương tự, $\mu_{zero}\left(\frac{d}{dt}e(t)\right) = \mu_{possmall}\left(\frac{d}{dt}e(t)\right) = 0,5$

Xác định các luật để sử dụng (Determining which rules to use)

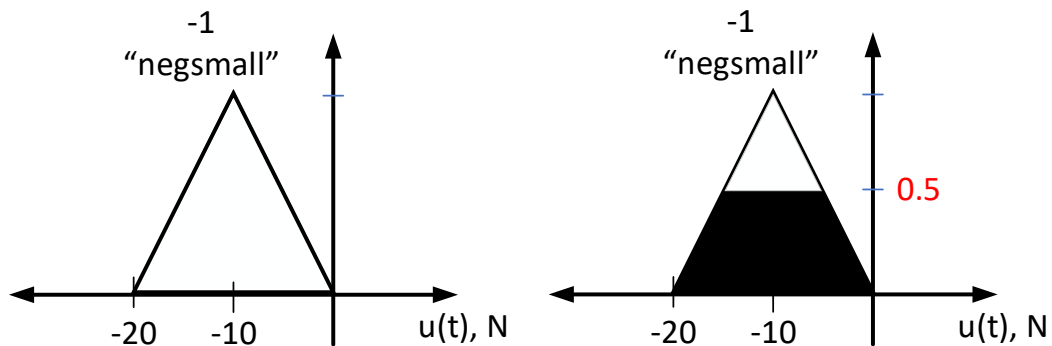
{rule1} Nếu *error* là *possmall* và *change-in-error* là *zero* thì *lực (u)* là *negsmall*

Khi sử dụng toán tử “*Và*”, thì phương pháp để lựa chọn chắc chắn một cách đúng đắn mệnh đề này và mệnh đề khác gồm sử dụng Minimum, Maximum hoặc Product giữa hai giá trị μ của MFs.

Khi sử dụng Product chúng ta sẽ tính $\mu_{premise} = \mu_{possmall} * \mu_{zero}$ điều này sẽ làm cho giá trị $\mu_{premise}$ phụ thuộc vào cả 2 MFs này.

Sử dụng Minimum ta có: $\mu_{premise} = \min\left\{\mu_{possmall}(e(t)), \mu_{zero}\left(\frac{d}{dt}e(t)\right)\right\}$

$$\mu_{(premise(1))} = \min\{0.5, 0.5\} = 0.5$$

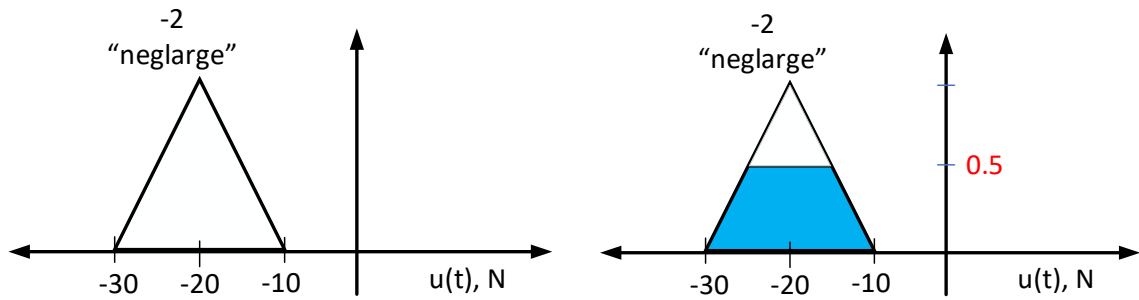


{rule2} Nếu *error* là *possmall* và *change-in-error* là *possmall* thì *lực (u)* là *neglarge*

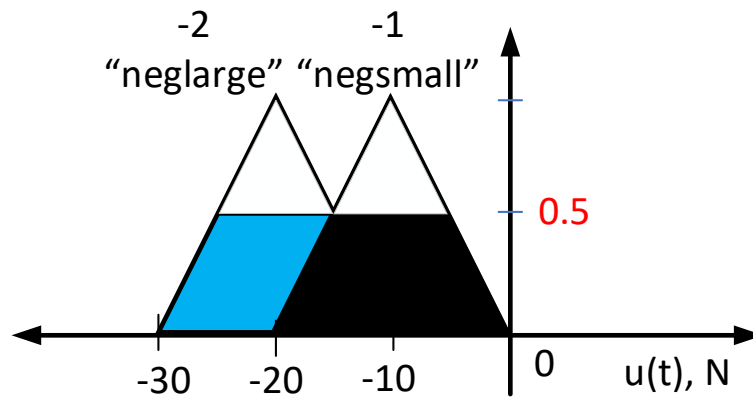
{rule3} Nếu *error* là *poslarge* và *change-in-error* là *zero* thì *lực (u)* là *neglarge*

{rule4} Nếu *error* là *poslarge* và *change-in-error* là *possmall* thì *lực (u)* là *neglarge*

$$\Rightarrow \mu_{(premise(2))} = \min\{0.5, 0.5\} = 0.5$$



Giải mờ (Defuzzification)



Phương pháp tọa độ trọng tâm COG: $u^{crisp} = \frac{\sum_i b_i \int u_{(i)}}{\sum_i \int u_{(i)}}$

$\int u_{(i)}$ được tính như sau $\int u_{(i)} = \sum w \left(h - \frac{h^2}{2} \right)$. Với w là chiều rộng của MFs và h là chiều cao tương ứng là $\mu_{(premise(i))}$.

Suy ra ta có $\int u_{(i)} = 20 * \left(0.5 - \frac{0.5^2}{2} \right) + 20 * \left(0.5 - \frac{0.5^2}{2} \right) = 7.5 + 7.5 = 15$

Vậy $u^{crisp} = \frac{(-10)(7.5) + (-20)(7.5)}{7.5 + 7.5} = -15N$

Câu 3:

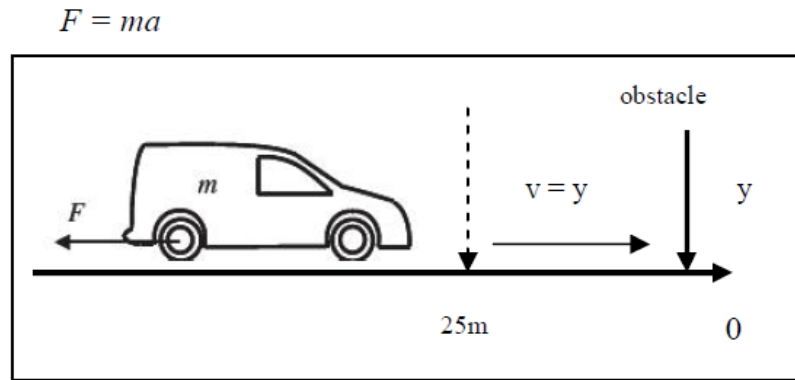


Fig. 1: Model of a Car

Các thông số của hệ thống trên:

Khối lượng xe: 2000 kg

Vị trí xe bắt đầu phanh: -25 m

Vận tốc ban đầu $s(0) = 20\text{ km/h}$

Lực thắng (phanh) $b = 13600\text{ N}$

6 bước thực hiện thiết kế FLC (mamdani)

1. determining a set of fuzzy rules
2. fuzzifying the inputs using the input membership functions,
3. combining the fuzzified inputs according to the fuzzy rules to establish a rule strength,
4. finding the consequence of the rule by combining the rule strength and the output membership function,
5. combining the consequences to get an output distribution, and
6. defuzzifying the output distribution (this step is only if a crisp output (class) is needed).

The following is a more detailed description of this process.

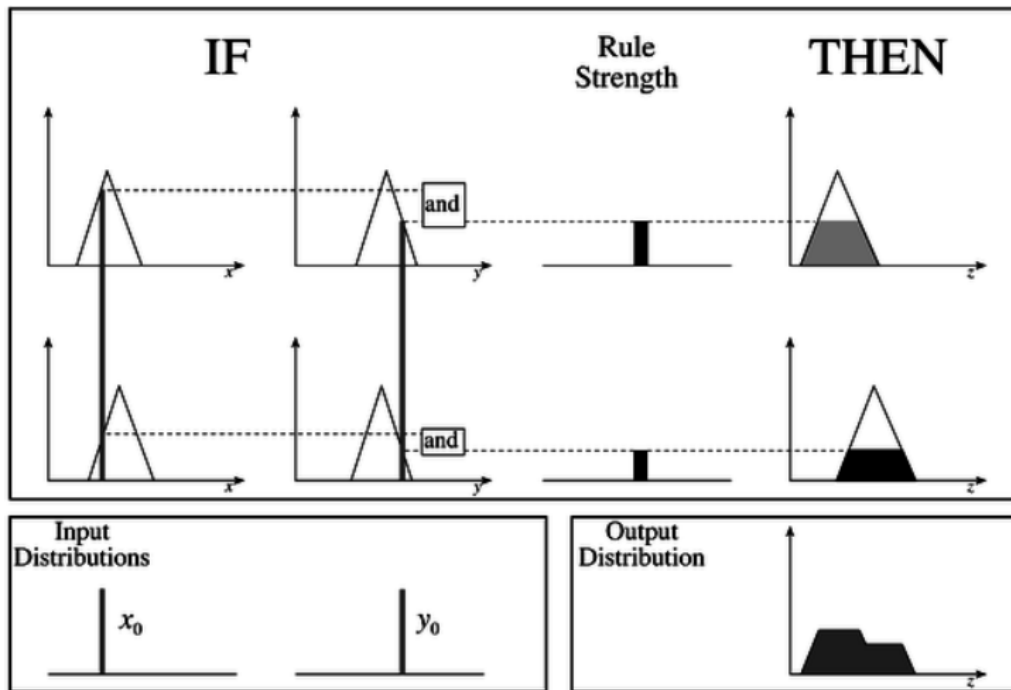
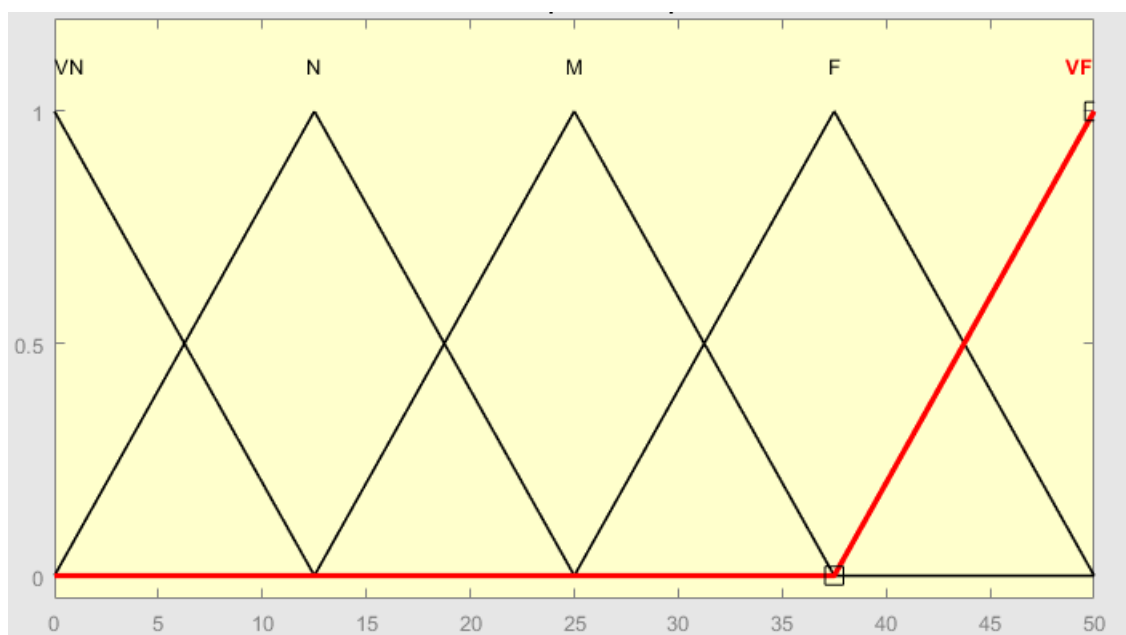


Figure 4-1: A two input, two rule Mamdani FIS with crisp inputs

Membership function (MFs), lựa chọn là Triangular MFs cho các biến ngôn ngữ đầu vào/ra. Việc lựa chọn Gaussian MFs hay các dạng MFs khác là tùy thuộc vào hệ thống điều khiển và áp dụng vào phần cứng để điều khiển chúng. Và MFs cho các biến ngôn ngữ của tốc độ xe cũng như khoảng cách của xe tới cột đèn giao thông gồm 5 biến:

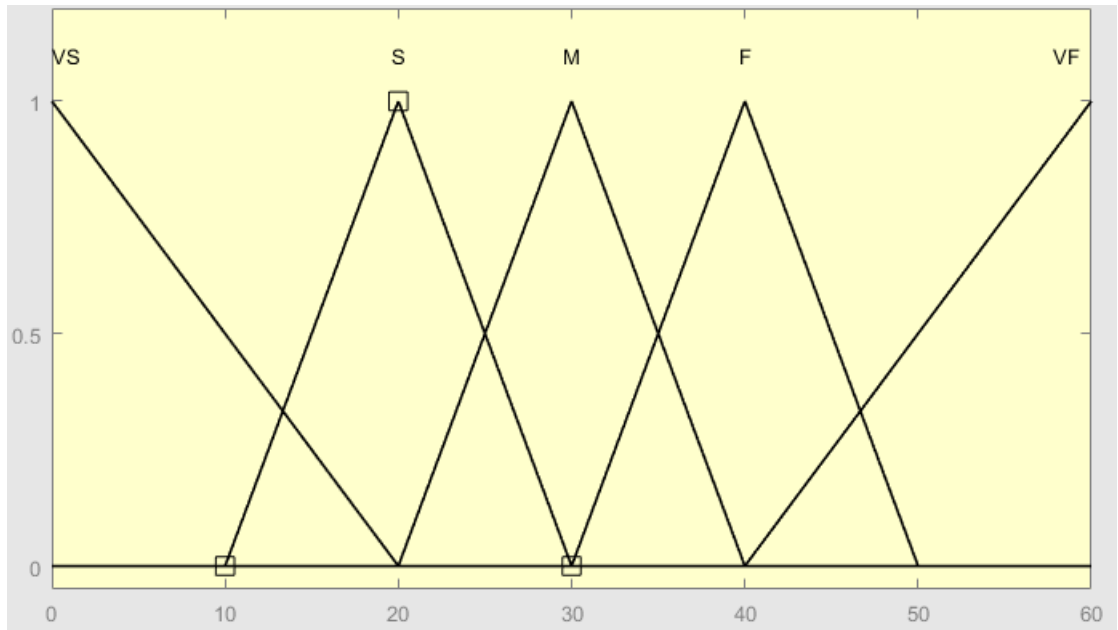
Với vị trí xe $d = \{Very\ Near\ (VN),\ Near\ (N),\ Medium\ (M),\ Far\ (F),\ Very\ Far\ (VF)\}$

Miền giá trị của d từ 0m đến 50 m. Khoảng cách an toàn xe so với cột đèn giao thông là 10 m.



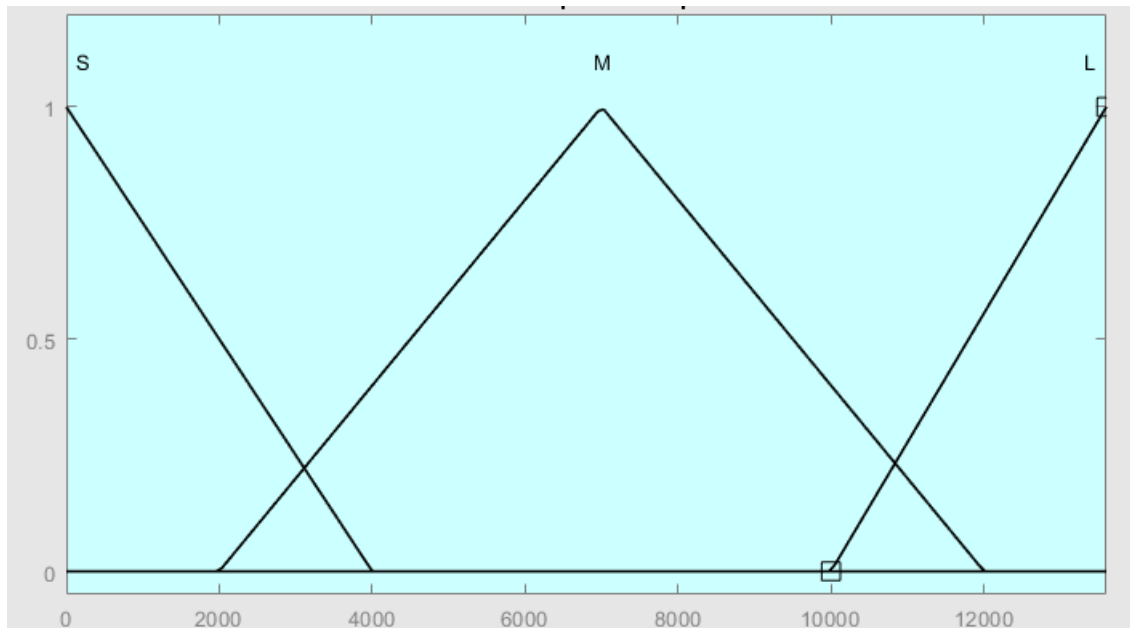
Với tốc độ xe $s = \{Very\ Slow\ (VS),\ Slow\ (S),\ Medium\ (M),\ Fast\ (F),\ Very\ Fast\ (VF)\}$

Miền giá trị của s từ 0km/h đến 60km/h ở khu vực đường đông dân cư. Vận tốc an toàn của xe là 40km/h.



Với lực thắng (phanh) $b = \{Small (S), Medium (M), Large (L)\}$

Miền giá trị của b từ 0N đến 13600N.



Bộ luật được phát triển dựa vào tập mờ input và output. 25 rule-base của bài toán này sẽ dựa trên rule-base này: **Nếu** vị trí xe so với cột đèn là *Near* **VÀ** vận tốc xe là *Fast* **thì** lực thắng là *Large*.

$d \backslash s$	VS	S	M	F	VF
VN	S	S	S	M	M
N	S	S	S	M	M
M	S	S	M	L	L
F	M	M	L	L	L
VF	M	M	L	L	L

Giải mờ bằng phương pháp COA hoặc COG.

Câu 4: Một mobile robot (robot gồm 2 động cơ dẫn động cho 2 bánh xe chủ động và 1 bánh tự lựa dẫn hướng) cần di chuyển từ vị trí bắt đầu (*Start Position* – $(x_s; y_s)$) đến vị trí đích (*Goal Position* – $(x_g; y_g)$). Mobile này có khả năng tránh các vật cản khi di chuyển nhờ vào các cảm biến phát hiện vật cản được gắn trên robot (ví dụ cảm biến hồng ngoại, siêu âm,...)

Lưu ý: Tọa độ các vật cản đã biết trước, tọa độ vị trí *Start* và *Goal* đã biết trước.

Hãy thiết kế bộ điều khiển Fuzzy để giúp robot có thể thực hiện được công việc trên ?

Giải

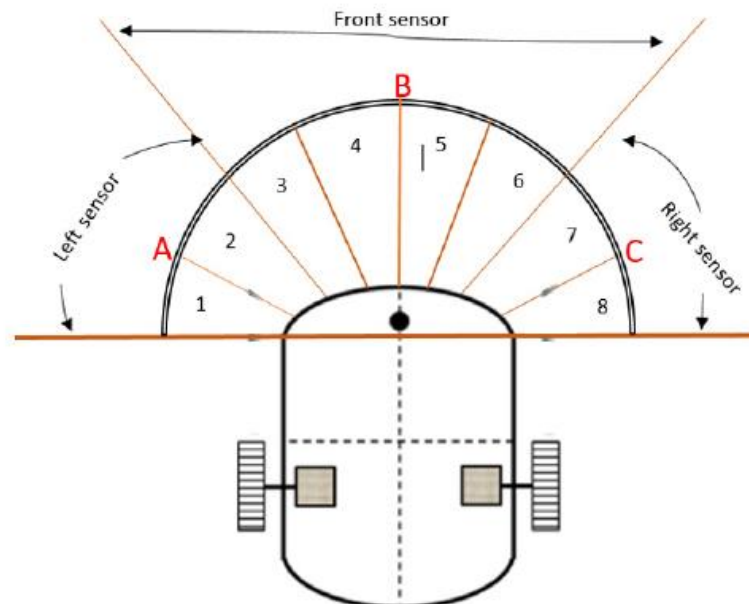


Figure .3: Ultrasonic sensor arrangement

Cảm biến tránh vật cản : Cảm biến siêu âm (sử dụng 8 cảm biến chia thành 3 hướng cũng là 3 input đầu vào)

Bước 1: Xác định các biến ngôn ngữ vào/ ra (Input/ Output)

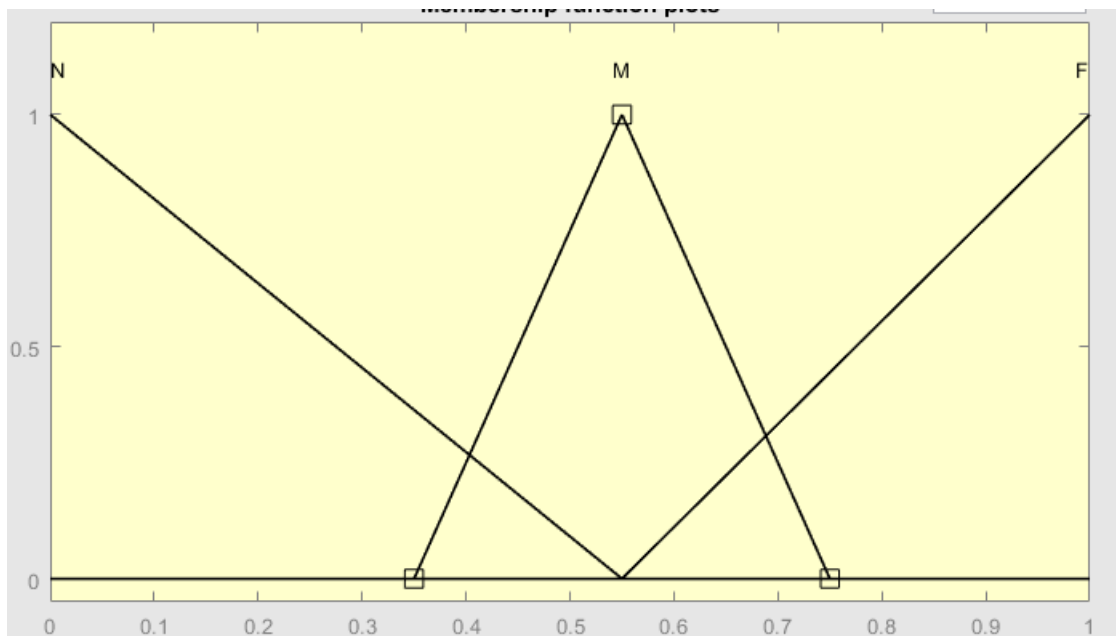
Input đầu vào gồm dL, dF, dR lần lượt là khoảng cách trả về từ các vùng cảm biến trái, trước và phải. Mỗi input gồm 3 biến ngôn ngữ:

$dL = \{ \text{NEAR (N), MEDIUM (M), FAR (F)} \}$

$dF = \{ \text{NEAR (N), MEDIUM (M), FAR (F)} \}$

$dR = \{ \text{NEAR (N), MEDIUM (M), FAR (F)} \}$

Với miền giá trị nằm trong đoạn $[0;1]$ (m)

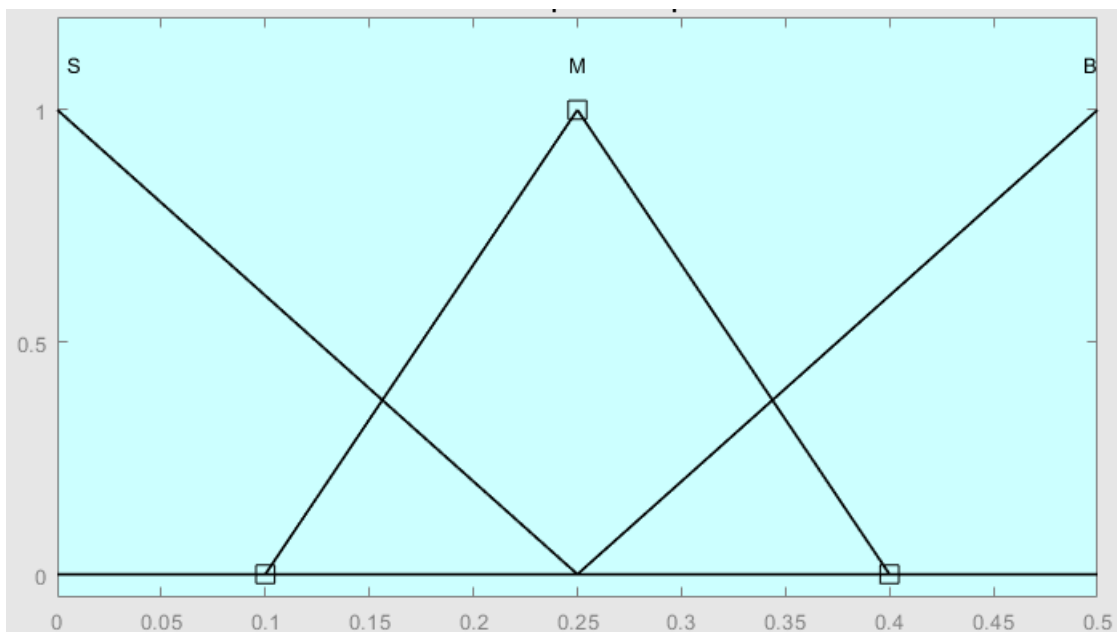


Output là tốc độ của motor trái và motor phải của robot. Gồm 3 biến ngôn ngữ:

$vL = \{ \text{SMALL (S), MEDIUM (M), BIG (B)} \}$

$vR = \{ \text{SMALL (S), MEDIUM (M), BIG (B)} \}$

Với miền giá trị nằm trong đoạn $[0; 0.5]$ (m/s)



Bước 2: Tiến hành giải mờ

Xét trường hợp $x_{10} = 0.275m, x_{20} = 0.55m, x_{30} = 0.275m$.

Ta có giá trị $\mu(x_{10}) = 0.5, \mu(x_{20}) = 1, \mu(x_{30}) = 0.5$

Với $\mu(x_{10}) = 0.5$ kết luận rằng chắc chắn là “NEAR”. $\mu(x_{20}) = 1$ chắc chắn là “MEDIUM” và $\mu(x_{30}) = 0.5$ chắc chắn là “FAR”

Bước 3: Xây dựng bộ luật (rule-base).

Rule	dL	dF	dR	vL	vR	Rule	dL	dF	dR	vL	vR
1	N	N	N	S	S	14	M	M	M	M	M
2	N	N	M	M	S	15	M	M	F	M	S
3	N	N	F	B	M	16	M	F	N	S	B
4	N	M	N	S	S	17	M	F	M	M	M
5	N	M	M	M	S	18	M	F	F	B	M
6	N	M	F	B	M	19	F	N	N	S	B
7	N	F	N	M	M	20	F	N	M	S	M
8	N	F	M	B	S	21	F	N	F	S	B
9	N	F	F	B	M	22	F	M	N	N	M
10	M	N	N	S	B	23	F	M	M	M	B
11	M	N	M	S	S	24	F	M	F	M	B
12	M	N	F	B	S	25	F	F	N	S	M
13	M	M	N	S	B	26	F	F	M	M	B
						27	F	F	F	B	B

Bước 4: Giải mờ và tối ưu hóa

Giải mờ bằng phương pháp COA