

Размита логика

Размитата логика е апарат, способен да решава сложни проблеми като управлението на сложни технологични обекти, определяне траекторията на движение на индустриални роботи, ABS спирачни системи и др. Въпреки името си, което оставя впечатление за нещо не съвсем точно и ясно, размитата логика в много от случаите се справя много по-добре от традиционните методи за управление. Апаратът и принципите на размитата логика са създадени от проф. Лотфи Задех (Lotfi Zadeh) от университета Бъркли, Калифорния. **Методологията и се базира на изразяване на управляващите въздействия на контролната система не чрез математически уравнения, а посредством лингвистични правила.** Много системи са прекалено сложни, за да се подават на управляващи въздействия, дори базирани на сложни математически уравнения. Тук на помощ идва размитата логика. Посредством набор от лингвистични правила, може акуратно да се опише поведението на системата и на тяхна база да бъдат изработени управляващи въздействия. Правилата се дефинират по следния начин.

АКО ТЕМПЕРАТУРАТА Е ТОПЛА ТОГАВА УСТАНОВИ СКОРОСТТА НА ВЕНТИЛАТОРА НА СРЕДНА

(IF THE TEMPERATURE IS WARM THEN SET FANSPEED TO MEDIUM)

Термините ТОПЛА и СРЕДНА са набори, които дефинират група от стойности, известни като **функции на принадлежност**. Посредством избор на набор от стойности вместо една точно определена стойност може да се контролира изходната променлива много по-прецизно.

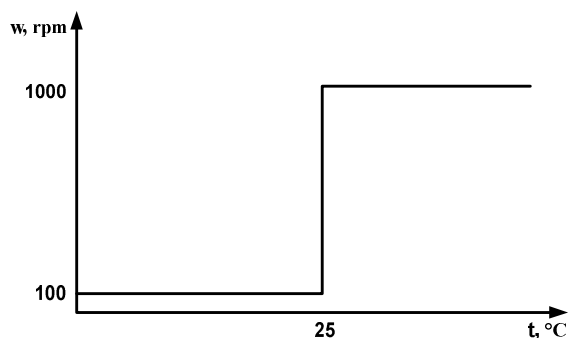
Сравнение с традиционния подход

Като пример може да се вземе управлението на вентилатор за регулиране на температурата в стая. При традиционния подход, с използването на релеен закон за управление, контрола на температурата се дефинира по следния начин:

If air_temperature $\geq 25^{\circ}\text{C}$ then set fan_speed to 1000 rpm.

If air_temperature $< 25^{\circ}\text{C}$ then set fan_speed to 100 rpm.

Традиционните контролни алгоритми се опират на вземане на решение при достигане определени точки от функцията на контролираната величина. На фиг. 1 е представена диаграмата на осъществяване на този начин на контрол.



Фиг. 1

Както се вижда от графиката, решението, което се реализира по този начин не е удачно за интервала обхващащ няколко градуса около точката на превключване. Там скоростта ще се променя често, понякога дължащо се и на грешки от измерването. Точно при решаването на такива проблеми размитата логика показва своето предимство.

Управление посредством размита логика

Прилагането на размитата логика може условно да се раздели на три части:

- „Фъзифициране” на данните (fuzzification) - преминаване на данните във вид удобен за използване от размитата логика.
- Създаване на лингвистични правила отнасящи се до решавания проблем.
- „Дефъзифициране” (defuzzification) – обратно преобразуване на резултатите

На фиг. 2 са представени основните етапи при работата с размита логика.



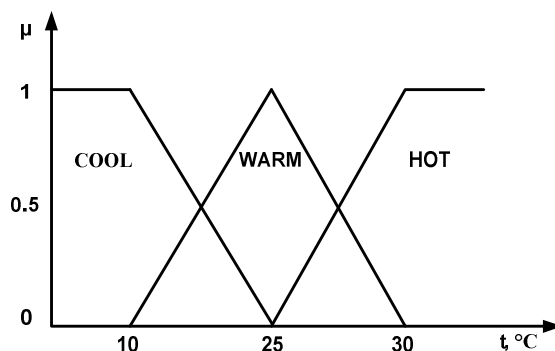
Фиг.2

„Фъзифициране”

В първата фаза измерените реални входни данни се преобразуват по такъв начин, че да се представят като стойности, показващи принадлежността им към някоя функция на принадлежност.

При нашия пример за контрол на температурата, първо трябва да се създадат функциите на принадлежност на входната променлива *air_temperature* (отнасяща се до температурата на въздуха). Тези функции се дефинират като набор от стойности на променливата и тяхната принадлежност към функцията.

При размитата логика е не само важно да се определи дали една входна променлива принадлежи на една функция на принадлежност, но и до каква степен и принадлежи. Една променлива може да принадлежи на различни функции на принадлежност в различна степен. На фиг. 3 са показани функции на принадлежност на променливата *air_temperature*.



Фиг.3

Както е показано на фиг.3, функциите на принадлежност се припокриват. Степента на принадлежност се получава като за съответната температура се изчисли размитата стойност на принадлежност към дадена функция.

Лингвистични правила

След като се дефинират функциите на принадлежност може да се премине към задаване на лингвистичните правила на системата. Броят на правилата не е ограничен, но трябва да се описват всички входни и изходни състояния на системата.

Пример:

If air_temperature is COOL then set fan_speed to SLOW.

If air_temperature is HOT then set fan_speed to FAST.

If air_temperature is WARM then set fan_speed to MEDIUM.

Тъй като правилата, с които се описва поведението на системата са лингвистични, това значи, че дори нелинейни системи могат лесно да се контролират посредством размита логика. След „фъзифицирането“ на входните променливи, те могат да се разглеждат като тегловни коефициенти, влизащи в изчисляването на всяко лингвистично зададено правило. Често е възможно да се премахнат правила без това да навреди на крайния резултат при управление на системата.

„Дефъзифициране“

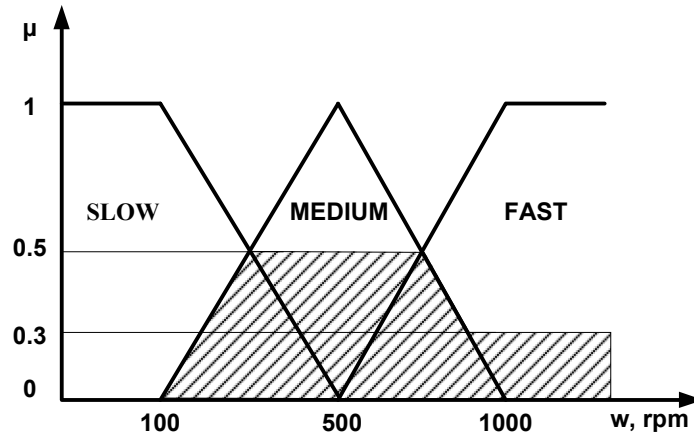
След като входните променливи се преобразуват към изискванията на размитата логика и се обработят от лингвистичните правила, то трябва да се генерира изходен сигнал към системата, която се контролира. Този изходен сигнал трябва да има реална управляваща стойност. **Преминаването от размита в реална стойност се нарича „дефъзифициране“**. Има няколко метода да се извърши това, като най-разпространените са:

- чрез „дефъзифициране“ спрямо максимума – взема се максималната стойност от всички правила
- чрез изчисляване на центроид – изчислява се тегловната сума от всички правила. Формулата за изчисляване на центроид е следната:

$$Z = \frac{\sum_{i=1}^n F_i S_i}{\sum_{i=1}^n F_i} ,$$

където: Z – изходната променлива в реални стойности, F_i – стойността на принадлежност към съответната изходна променлива, S_i – максималната стойността за съответната размита изходна променлива.

На фиг.4 са показани размитите стойности на изчислената изходна величина.



Фиг.4

От примера може да се изчисли центроида както следва:

$$Z = \frac{\sum_{i=1}^3 F_i S_i}{\sum_{i=1}^3 F_i} = \frac{100 \cdot 0 + 500 \cdot 0.5 + 1000 \cdot 0.3}{0 + 0.5 + 0.3} = 687,5 \text{rpm}$$

Математически операции при размитата логика

При прилагането на лингвистичните правила се използват следните операции – **AND (И), OR (ИЛИ) и NOT (НЕ)**.

AND -> осъществява се ст.нар. min функция

Пример:

$$x_1 = 0.3, \quad x_2 = 0.6$$

$$\min(x_1, x_2) = 0.3$$

OR -> осъществява се с т.нар. max функция

Пример:

$$x_1 = 0.3, \quad x_2 = 0.6$$

$$\max(x_1, x_2) = 0.6$$

NOT -> осъществява се с т.нар. not функция

Пример:

$$x = 0.2$$

$$\text{not}(x) = 1 - 0.2 = 0.8$$