

Практика. Управление на основе нечеткой логики

Мишуров С.С.

05.12.2024

Содержание

1	Общее описание задания	1
1.1	Описание системы управления температурой в помещении	1
1.2	Алгоритм управления на основе нечеткой логики	2
1.2.1	Фаззификация	2
1.2.2	Применение правил логического вывода над нечеткими множествами	2
1.2.3	Дефаззификация	3
2	Выполнение работы	3
2.1	Определение элементов системы управления	3
2.2	Реализация вычислений	3
	Приложение 1. Представление лингвистических переменных	5
	Приложение 2. Представление правил логического вывода	5

1 Общее описание задания

Цель работы: освоение методов управления на основе нечеткой логики.

Необходимо реализовать на языке программирования (например, python) алгоритм, который для некоторой системы управления температурой в помещении будет вычислять параметр управления для поддержания комфортной температуры.

1.1 Описание системы управления температурой в помещении

Система управления температурой в помещении предназначена для автоматического регулирования температуры в закрытом пространстве, таком как комната или здание, для обеспечения комфорта. В общем случае такая система включает в себя следующие компоненты:

- **Датчик температуры** передает информацию о температуре в систему управления (на практике обычно устанавливается несколько датчиков, показания которых агрегируют).
- **Вычислитель** принимает входные данные от датчика температуры и применяет *алгоритм* нечеткой логики для вычисления необходимых параметров управления.
- **Исполнительные механизмы** получают сигналы от вычислителя и осуществляют необходимые действия для регулирования температуры. Они могут включать в себя термостаты, кондиционеры, системы отопления или вентиляторы.
- **Интерфейс пользователя** позволяет устанавливать желаемый режимы работы, что осуществляется путем выбора правила логического вывода, каждый из которых соответствует определенному режиму, например, экономичный или комфортный.

В настоящей работе необходимо реализовать алгоритм управления, который будет реализовываться вычислителем.

1.2 Алгоритм управления на основе нечеткой логики

В общем виде алгоритм управления на основе нечеткой логики выглядит следующим образом:

- **Фаззификация** - преобразование точечных выходных данных, полученных от датчика, в нечеткие значения, то есть вычисление функции принадлежности нечеткому множеству для конкретного значения температуры, заданной лингвистической переменной "температура", значения которой соответствуют термам "жарко", "комфортно", "холодно".
- **Применение правила логического вывода** - получение функции принадлежности для нечеткого множества "оптимальное управление" на основе логического правила "отображающего" значения лингвистической переменной "температура" во множество значений лингвистической переменной "уровень нагрева" значения которой соответствуют термам "слабый нагрев", "умеренный нагрев", "интенсивный нагрев":
 - если "жарко", то осуществлять "слабый нагрев".
 - если "комфортно", то осуществлять "умеренный нагрев".
 - если "холодно", то осуществлять "интенсивный нагрев".
- **Дефаззификация** - определение степени принадлежности к нечетким множествам и выбора наиболее подходящего действия - конкретного значения на шкале управления нагревателем.

1.2.1 Фаззификация

В вашем примере с управлением температурой в помещении фаззификация включает определение того, насколько текущая температура соответствует нечетким множествам "жарко", "комфортно" и "холодно".

1.2.2 Применение правил логического вывода над нечеткими множествами

При реализации правил логического вывода (конструкций вида "если ..., то ...") будем руководствоваться методом Мамдани¹, которое включает следующие шаги:

- **Активация нечетких правил.** Определите уровень активации каждого нечеткого правила как минимальное значение между уровнями принадлежности входных данных к соответствующим нечетким множествам в условной части правила.
- **Нечеткий вывод.** Для каждого активированного правила определите нечеткий вывод, который представляет собой нечеткое множество, определяемое функцией принадлежности вывода. Уровень принадлежности вывода определяется уровнем активации правила. Для приведенного выше примера получение нечеткого множества, носителем которого будут элементы из нечеткого множества "умеренный нагрев" со значениями функций принадлежности соответствующей уровню активации. Например, для правила "если "жарко", то "слабый нагрев"" при значении температуры равном t^* уровень активации будет:

$$\mu_{C_1}(s) = \min[\mu_{\text{комфортно}}(t = t^*), \mu_{\text{умеренный_нагрев}}(s)]$$

где $t \in T$, T - множество возможных значений температуры в помещении, а $s \in S$, S - множество возможных значений параметра управления (положений регулятора)

В результате мы получим набор (в нашем случае три по количеству правил вывода) нечетких множеств: C_1, C_2, C_3 , таких что $C_i = \{s, \mu_{C_i}(s)\}$.

- **Объединение нечетких выводов.** Объедините нечеткие выводы всех активированных правил в единое нечеткое множество. Это делается с помощью операции объединения нечетких множеств. Для нашего примера это будет нечеткое множество, полученное объединением трех нечетких множеств, которые образованы соответствующими правилами вывода:

$$\mu_{\text{оптимальное_управление}}(s) = \cup \mu_{C_i}(s) = \max[\mu_{C_1}(s), \mu_{C_2}(s), \mu_{C_3}(s)]$$

¹Мамдани, Э. Х. (1975). «Анализ сложных систем и управление на основе лингво-нечетких моделей». IEEE Transactions on Systems, Man and Cybernetics, 5(3), 431-435.

1.2.3 Дефаззификация

Дефаззификация — это процесс преобразования нечеткого вывода в четкое значение или решение. После применения нечетких правил и получения нечеткого вывода, дефаззификация заключается в определении конкретного значения на основе нечеткого вывода. В нашем примере с управлением температурой дефаззификация предполагает получение конкретного значения s^* , которое является представителем нечеткого множества оптимального управления:

$$C_{\text{оптимальное_управление}} : \{s, \mu_{\text{оптимальное_управление}}(s)\}$$

Для дефаззификации мы будем использовать **метод первого максимума**, который заключается в выборе первого локального максимума функции принадлежности нечеткого вывода. Иными словами, мы выбираем первое наиболее достоверное значение в нечетком множестве, которое имеет максимальный уровень принадлежности. То есть двигаясь по значению s "слева направо" (от меньшего к большему) мы примем в качестве s^* такое значение s , при котором функция принадлежности оптимального управления $\mu_{\text{оптимальное_управление}}(s)$ впервые достигает максимального значения.

2 Выполнение работы

2.1 Определение элементов системы управления

Характеристика температуры в помещении задается значениями лингвистической переменной "температура", которые определены как термы, обозначающие соответствующие нечеткие множества "жарко", "комфортно", "холодно". Функции принадлежности этих нечетких множеств имеют форму трапеций, который заданы линейно-кусочными функциями. Например, такая

$$\mu_{\text{комфортно}}(t) = \begin{cases} 1, & \text{если } t \text{ между } 20^\circ C \text{ и } 22^\circ C, \\ 0, & \text{если } t \leq 18^\circ C \text{ или } t \geq 24^\circ C, \\ \text{линейно увеличивается/уменьшается между этими значениями.} \end{cases}$$

где t - значение на шкале возможных температур в помещении.

Значение для управления задаются аналогичным образом, например, так:

$$\mu_{\text{слабый_нагрев}}(s) = \begin{cases} 1, & \text{если } s \text{ между } 4 \text{ и } 6, \\ 0, & \text{если } s \leq 0 \text{ или } s \geq 10, \\ \text{линейно увеличивается/уменьшается между этими значениями.} \end{cases}$$

где s - значение на шкале возможных положений регулятора (нагревателя).

Такие функции можно задать четырьмя точками, значения которых могут быть записаны в виде списка. Таким образом, термы одной лингвистической переменной можно сериализовать в виде json-строки, содержащей список объектов, соответствующих термам (пример см. в [приложении](#)).

Правила логики управления ("отображения") задаются набором импликаций вида:

$$\mu(\text{"управление"}) = \begin{cases} \text{если "жарко", то "слабый нагрев"}, \\ \text{если "комфортно", то "умеренный нагрев"}, \\ \text{если "холодно", то "интенсивный нагрев"}. \end{cases}$$

Соответственно правила данного вида можно также представить в виде json-строки (пример см. в [приложении](#)).

2.2 Реализация вычислений

Реализуйте алгоритмы (методы), которые для:

- вычисления значений функций принадлежности для заданных элементов множества (фаззификации);
- вычисления уровней активации для каждого логического правила;
- значений функций принадлежности активированного правила;
- аккумулярования (объединения) нечетких множеств, полученных при применении уровней активации;

- вычисления (дефаззификация) итогового значения управления.

Реализуйте общий алгоритм вычисления оптимального управления для фактической температуры в помещении.

ПРИЛОЖЕНИЕ 1

Представление лингвистических переменных

```
{
  "температура": [
    {
      "id": "холодно{}",
      "points": [
        [0,1],
        [18,1],
        [22,0],
        [50,0]
      ]
    },
    {
      "id": "комфортно{}",
      "points": [
        [18,0],
        [22,1],
        [24,1],
        [26,0]
      ]
    },
    {
      "id": "жарко{}",
      "points": [
        [0,0],
        [24,0],
        [26,1],
        [50,1]
      ]
    }
  ]
}
```

ПРИЛОЖЕНИЕ 2

Представление правил логического вывода

```
[
  ["холодно{}", "интенсивно"],
  ["нормально{}", "умеренно"],
  ["жарко{}", "слабо"]
]
```