Кукса П.П.

**Московский Государственный Технический Университет
им. Н.Э. Баумана**

**E-mail: kouxa@online.ru
WWW: http://www.geocities.com/pkouxa**

Программная реализация нечеткого логического контроллера

Выбор структур данных во многом определяется конечным приложением, для которого разрабатывается нечеткий логический контроллер (НЛК) и возможностями компилятора или интерпретатора, что особенно характерно для реализации НЛК на основе микроконтроллеров (МК). В этом случае используются достаточно простые компиляторы (подмножество ANSI C) или интерпретаторы (MCS BASIC). Популярным решением для МК является использование табличной реализации НЛК – нечеткий вывод осуществляется в режиме off-line, т.к. таблица содержит все возможные комбинации входных значений и соответствующих им выходных. Такой подход обуславливается низкой вычислительной сложностью табличной реализации (в [1] показывается, что табличная реализация (количество входов, термов и уровней дискретизации $p = 2, n_1 = n_2 = 5, q_{od} = 32$) для MC68000 на порядок быстрее стандартного нечеткого вывода (он-лайн варианта)), но имеет ряд существенных недостатков:

- высокая емкостная сложность ($O(s \cdot q^p), q = \max\{n_1,..,n_p\}$, где s - число выходов);
- фиксирование параметров нечеткого вывода на ранних стадиях разработки НЛК;
- невозможность адаптации параметров НЛК (использование весовых коэффициентов $w_1,..,w_p$ и w_o, масштабирующих входные и выходные переменные, дает возможность в некоторой степени оказывать влияние на результат нечеткого вывода);
- количество уровней квантования прямо влияет на точность НЛК (повышение точности НЛК ведет к экспоненциальному повышению емкостных затрат при увеличении уровней квантования, либо к увеличению вычислительной сложности при использовании интерполяции);
- применимость только при небольшом числе переменных (до 3-х). Для большего числа переменных возможно использование иерархической табличной структуры (рис. 1).

Иерархическая структура имеет меньшую емкостную сложность $s(p-1)q^2$. При использовании иерархической структуры требуется разработка промежуточных таблиц, реализующих вывод на основе промежуточных правил и промежуточных переменных, что представляет собой достаточно сложную задачу (для промежуточных переменных возникает проблема назначения базовых терм-множеств, кроме того, общее количество промежуточных правил может превышать количество исходных). Результат вывода в общем случае чувствителен к распределению входных переменных по уровням иерархии, что является существенным недостатком и требует детального анализа при проектировании НЛК. Доминирующие входы должны располагаться на верхних уровнях иерархии [2].

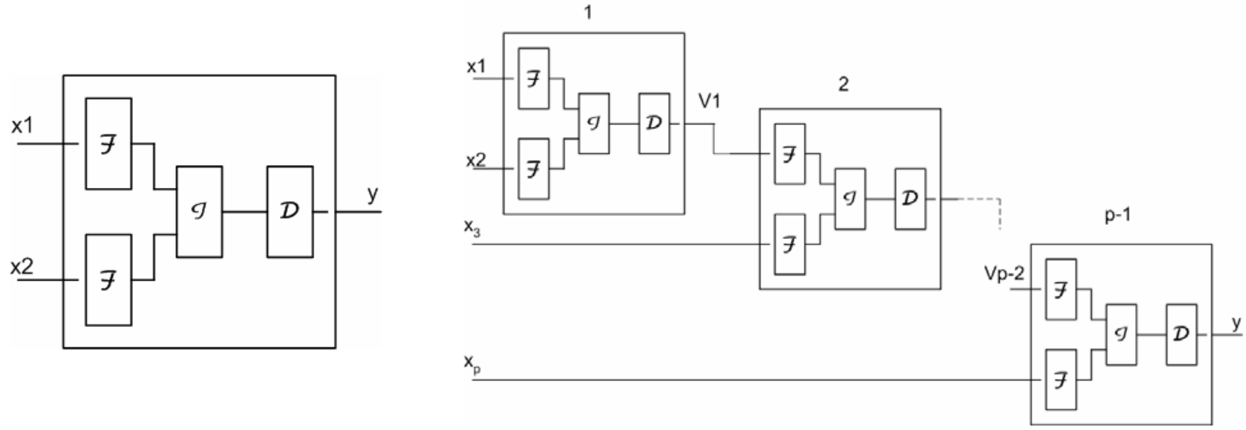

Рис. 1

Для неадаптивных НЛК база знаний (набор правил + набор лингвистических определений) имеет статическую форму, при программной реализации могут использоваться простейшие статические структуры данных: массив записей, массив статических векторов (рис. 2).

Для адаптивных НЛК база знаний имеет динамический характер: параметры нечетких множеств (НМ) входных и выходных переменных, состав правил и их количество могут изменяться. Превалирующей операцией является модификация параметров функций принадлежности (ФП) НМ. Матричное представление НЛК приводит к простым реализациям алгоритмов нечеткого вывода и процедур обучения НЛК.

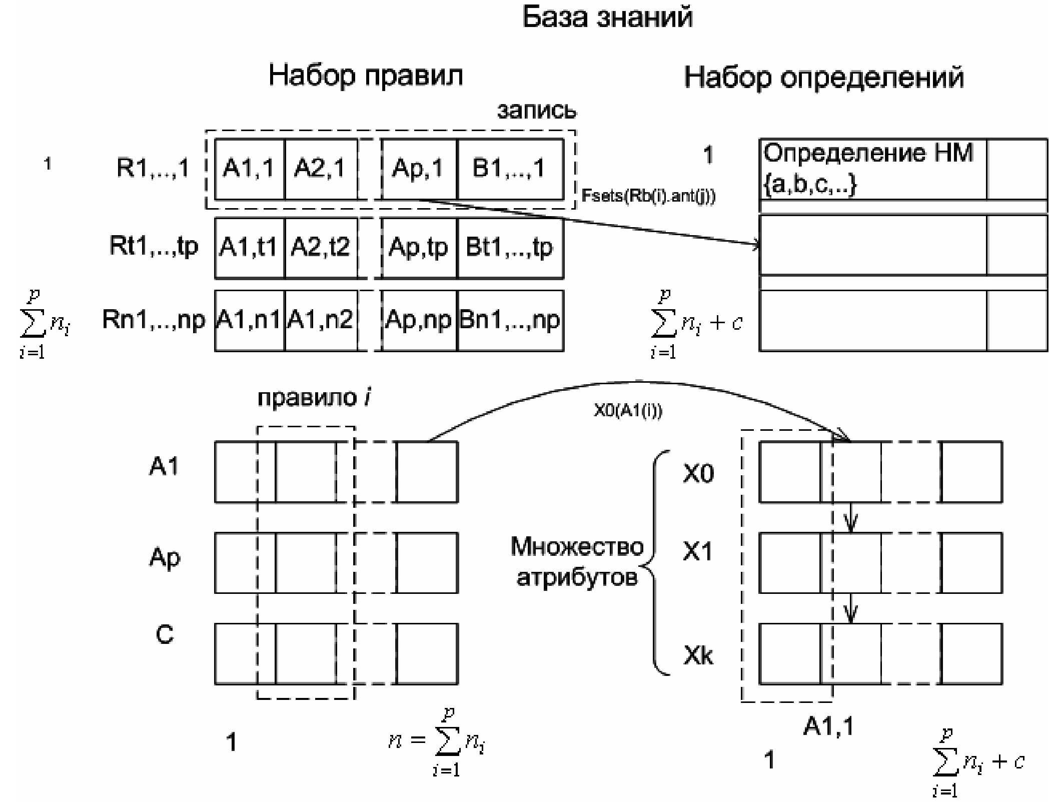

Рис. 2

Рассмотрим матричное представление НЛК для нечеткой TSK-системы 0-го порядка, в которой i-тое правило имеет вид: $R^{(i)}$: ЕСЛИ (x_1 есть $A_{i,1}$ И … И x_p есть $A_{i,p}$) ТО y есть C_i.

Параметры ФП НМ $A_{i,j}$ и заключений правил представляются в виде матриц:

матрицы параметров входных ФП столбец заключений

$$M_{n\times p}=\begin{bmatrix} m_{1,1} \; \dots \; m_{1,p} \\ m_{n,1} \; \dots \; m_{n,p} \end{bmatrix} \quad S_{n\times p}=\begin{bmatrix} s_{1,1} \; \dots \; s_{1,p} \\ s_{n,1} \; \dots \; s_{n,p} \end{bmatrix} \qquad C_{n\times 1}=\begin{bmatrix} c_1 \\ c_n \end{bmatrix}$$

матрица входных данных

$$X_{N\times p}=\begin{bmatrix} x_{1,1} & \dots & x_{1,p} \\ x_{N,1} & \dots & x_{N,p} \end{bmatrix}$$

Нечеткий вывод осуществляется циклической обработкой матриц M, S и матрицы данных X .

```
/*Нечеткий логический вывод*/
for i=1:N
/*определение для каждого правила степени его истинности (веса)*/
for j=1:n
  u1=1
  for t=1:p
    u1=u1*MF(X(i,t),M(j,t),S(j,t))
  end
  /*u1 = степень истинности нечеткой посылки правила j*/
  U = [U u1]
end
nu=U/sum(U) /*нормализация веса каждого правила*/
f=nu*C /*выход НЛК*/
end
```

Алгоритм обучения TSK-модели:

$[X(i,j)T(i)], i=1..N, j=1..p$ - набор данных “вход-выход”.

$$\varepsilon(k)=(T(k)-f)^2/2;\quad e(k)=T(k)-f;\quad c_i=c_i-\alpha\frac{\partial\varepsilon}{\partial c_i};\quad \frac{\partial\varepsilon}{\partial c_i}=\frac{\partial\varepsilon}{\partial y}\frac{\partial y}{\partial c_i}=-1\cdot e\cdot\frac{\tau_i}{\sum_{j=1}^{p}\tau_j}$$

$\Rightarrow c_i=c_i+\alpha\cdot e\dfrac{\tau_i}{\sum_{j=1}^{p}\tau_j}$ - правило адаптации для *i*-го заключения.

Правила адаптации для нелинейных параметров модели получаются аналогично из соотношения: $\dfrac{\partial\varepsilon}{\partial s_{ij}(m_{ij})}=\dfrac{\partial\varepsilon}{\partial y}\dfrac{\partial y}{\partial\tau_i}\dfrac{\partial\tau_i}{\partial\mu_{ij}}\dfrac{\partial\mu_{ij}}{\partial s_{ij}(m_{ij})}$, где s и m – параметры гауссовых ФП. Алгоритм адаптации при этом может выглядеть следующим образом:

```
for i=1:N
….
e = T(i) – f /*ошибка на i-том наборе*/
for l=1:n
  for k=1:p
    /*модификация входных ФП*/
   M(l,k)=M(l,k)+alpha*e*((X(i,k)-M(l,k))/(sigma(l,k)^2))*(c0(l)-f)*U(l)/sum(U)
   S(l,k)=S(l,k)+alpha*e*(((X(i,k)-M0(l,k))^2)/(sigma(l,k)^3))*(c0(l)-f)*U(l)/sum(U)
  end
end
c0=c0+alpha*e*fa^T /*вектор-столбец обновленных значений заключений правил*/
end
```

Вычислительная сложность матричной реализации (нечеткого вывода) $O(\prod_{i=1}^{p} n_i p)$.

Преимущество матричной реализации – прямой доступ ко всем параметрам нечеткой системы.

Реляционные нечеткие модели также могут использоваться в адаптивных НЛК. Реляционная модель опирается на представление нечетких правил в виде нечетких отношений $R_i = A_i \times B_i$; $\mu_{R_i}(x, y) = \min\{\mu_{A_i}(x), \mu_{B_i}(y)\}$. Общее отношение получается объединением локальных нечетких отношений $R = \bigcup_{i=1}^{n} R_i$. Нечеткое отношение может быть представлено в матричном виде. Нечеткий вывод получается в результате композиции $B = A' \circ R$ (например, максминной (max-min), эквивалентной произведению матриц с заменой операции суммирования на max, произведения на min) общего нечеткого отношения и входного нечеткого множества.

Литература

1. Y.H. Kim, Computational complexity of general fuzzy logic control and its simplification for a loop controller //Fuzzy Sets and Systems 111 (2000) 215-224.
2. S. Bolognani, M.Ziglotto Hardware and Software Configurations for Multi-Input Fuzzy Logic Controllers //IEEE Tran. on Fuzzy Systems, Vol. 6, No 1, 1998.