

УДК 004.4

Е. А. МУРАВЬЕВА, Г. А. КАЯШЕВА

НЕЧЕТКИЕ РЕГУЛЯТОРЫ С ANY-TIME АЛГОРИТМОМ МИНИМИЗАЦИИ ВРЕМЕНИ ФАЗЗИФИКАЦИИ ПАРАМЕТРОВ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ В СОВОКУПНОСТЬ ЧЕТКИХ ТЕРМОВ

Предложено с помощью ANY-TIME алгоритма (постепенно улучшающий некоторое текущее приближенное решение, в данном случае продолжительность процесса фаззификации) минимизировать в режиме реального времени продолжительность цикла фаззификации нечетких регуляторов, в ходе которого четкий формат технологического параметра преобразуется в совокупность не перекрывающихся четких термов. Показано, что по быстродействию фаззификаторы с ANY-TIME алгоритмом на (40–50)% превосходят существующие.

Фаззификация ; нечеткие регуляторы ; ANY-TIME алгоритм

ВВЕДЕНИЕ

В существующих нечетких регуляторах (НР) при фаззификации параметров технологических процессов [1–3] из-за наложения друг на друга смежных термов приходится в каждом цикле сканирования обрабатывать весь алгоритм фаззификации независимо от текущего значения этих параметров. Однако в НР с четкими термами из-за равенства в любой момент времени логической единице только одного терма из множества термов, изображающих конкретный параметр [7], имеется возможность при каждом сканировании обрабатывать не весь алгоритм фаззификации, а только ту его часть, которая находится выше продукционного правила, антецедент которого в данный момент времени равен логической единице [8]. Поэтому в НР с четкими термами для снижения продолжительности фаззификации в начало базы продукций, реализующих фаззификацию, нужно располагать правила с наибольшей частотой срабатывания (сколько раз за заданное время антецедент продукционного правила принимает значение логической единицы). К сожалению, на этапе проектирования дискретно-логического регулятора определить достоверное значение частоты срабатывания упомянутых правил крайне сложно, что приводит к неоправданному увеличению времени сканирования алгоритма фаззификации, а значит, и к снижению времени отклика НР. Наиболее просто минимизировать это время с помощью ANY-TIME алгоритма.

1. ЛОГИЧЕСКИЕ ОСНОВЫ СНИЖЕНИЯ ВРЕМЕНИ ОТРАБОТКИ АЛГОРИТМА ФАЗЗИФИКАЦИИ НЕЧЕТКИХ РЕГУЛЯТОРОВ

В фаззификаторах применение ANY-TIME алгоритма стало возможным благодаря интерпретации параметров технологического процесса эквивалентной совокупностью четких термов, изображенной на рис. 1 и представляющей собой неперекрывающиеся друг друга прямоугольники. Внутри таких прямоугольников для любого четкого значения параметра p функция принадлежности равна единице.

Для придания процедуре фаззификации свойств ANY-TIME алгоритма предлагается следующая система продукционных правил:

$$\begin{aligned}
 &\text{Если } 0 \leq p < P_1, \\
 &\text{то } (p \equiv T_1) \& (C_{ч1} := C_{ч1} + 1); \\
 &\text{Если } p_1 \leq p < P_2, \\
 &\text{то } (p \equiv T_2) \& (C_{ч2} := C_{ч2} + 1); \\
 &\text{Если } p_2 \leq p < P_3, \\
 &\text{то } (p \equiv T_3) \& (C_{ч3} := C_{ч3} + 1); \quad (1) \\
 &\dots\dots\dots \\
 &\dots\dots\dots \\
 &\dots\dots\dots \\
 &\text{Если } p_{n-1} \leq p < P_n, \\
 &\text{то } (p \equiv T_n) \& (C_{чn} := C_{чn} + 1).
 \end{aligned}$$

Из системы (1) следует, что все обозначения в ней заимствованы из рис. 1 и цепочка отрезков $(0 \div P_1); (P_2 \div P_3); \dots; (P_{n-1} \div P_n)$, покрывая без наложения всю универсальную

числовую ось, преобразуется в термы $T_1 \div T_n$ технологического параметра p . Причем в любой момент времени только у одного продукционного правила антецедент равен логической единице, что свидетельствует о возможности применения в данном случае ANY-TIME алгоритма.

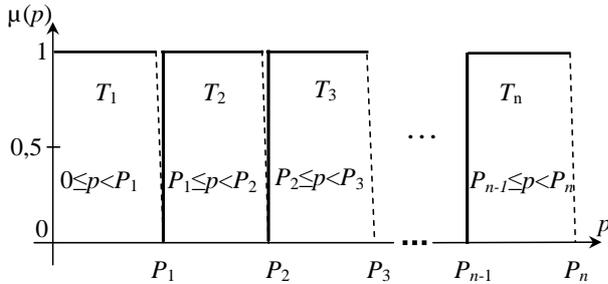


Рис. 1. Размещение четких термов параметра p на универсальной числовой оси

2. ANY-TIME АЛГОРИТМ МИНИМИЗАЦИИ ВРЕМЕНИ ФАЗЗИФИКАЦИИ

На рис. 2 изображена логическая схема алгоритма, соответствующая системе (1). Основу схемы составляют n соединенных в цепочку операторов условного перехода, в ветви «Да» которых включены операторы присваивания и счетчики, фиксирующие число инициализаций этих ветвей.

Система продукционных правил (1) отличается от типовых наличием в консеквентах продукций операторов $(S_{ч1} := S_{ч1} + 1) \div (S_{чn} := S_{чn} + 1)$, используемых для подсчета числа фактов истинности антецедента правил $(1 \div n)$. Такая информация необходима для перерасположения цепочки операторов $(p \equiv T_1 \div (p \equiv T_n), (0 \leq p < P_1) \div (P_{n-1} \leq p \leq P_n))$ и $(S_{ч1} := S_{ч1} + 1) \div (S_{чn} := S_{чn} + 1)$ с помощью программного блока «Модификация структуры алгоритма» (рис. 2) при срабатывании ветви «Да» оператора условного перехода « $T_{тек} \geq T_{зад}$ ». Здесь $T_{тек}$ — текущее время работы фаззификатора, а $T_{зад}$ — априорно заданное значение периода модификации структуры фаззификатора с целью уточнения порядка размещения цепочек операторов $(p \equiv T_1 \div (p \equiv T_n), (0 \leq p < P_1) \div (P_{n-1} \leq p \leq P_n))$ и $(S_{ч1} := S_{ч1} + 1) \div (S_{чn} := S_{чn} + 1)$ в порядке убывания их частоты срабатывания, поскольку именно такой порядок обеспечивает минимальное время фаззификации.

Рассмотрим принцип действия алгоритма, изображенного на рис. 2. В каждом цикле сканирования программно реализованных компонентов нечеткого регулятора инициирует-

ся поиск терма, соответствующего текущему четкому значению физической величины. С этой целью оператором условного перехода « $0 \leq p < P_1$ » определяется принадлежность текущего значения фаззифицируемой величины p отрезку $(0 \div P_1)$. Если $p \in (0 \div P_1)$, то по ветви «Да» управление передается оператору « $p \equiv T_1$ », который текущее четкое значение параметра p фаззифицирует в терм T_1 , а содержимое счетчика $S_{ч1}$ увеличивается на единицу. Если $T_{тек} \geq T_{зад}$, то с помощью блока «Модификация структуры алгоритма» в программе фаззификатора происходит перерасположение цепочки операторов $(0 \leq p < P_1) \div (P_{n-1} \leq p \leq P_n), (p \equiv T_1) \div (p \equiv T_n)$ и $(S_{ч1} := S_{ч1} + 1) \div (S_{чn} := S_{чn} + 1)$ в соответствии с частотой их исполнения. В противном случае цикл сканирования алгоритма фаззификации закончивается без модификации структуры программы фаззификатора. При срабатывании выхода «Да» операторов « $P_1 \leq p < P_2$ » \div « $P_{n-1} \leq p \leq P_n$ », описанная процедура повторяется, но уже при участии других ветвей алгоритма фаззификации.

С точки зрения законов реального мира, в любой момент времени параметр технологического процесса имеет только одно значение, вследствие чего в любой момент времени только у одного из операторов $(0 \leq p < P_1) \div (P_{n-1} \leq p \leq P_n)$ выход «Да» имеет истинное значение. В случае некорректного задания диапазонов существования термов $T_1 \div T_n$, когда ни одно из условий $(0 \leq p < P_1) \div (P_{n-1} \leq p \leq P_n)$ не выполняется, управление передается блоку «Аларм» для идентификации аварийной ситуации.

Блок «Модификация структуры алгоритма» реализует ANY-TIME алгоритм минимизации. При выполнении условия $T_{тек} \geq T_{зад}$ блок «Модификация структуры алгоритма» по информации, накопленной в счетчиках $(S_{ч1} := S_{ч1} + 1) \div (S_{чn} := S_{чn} + 1)$ представляет цепочки операторов $((0 \leq p < P_1) - (p \equiv T_1) - (S_{ч1} := S_{ч1} + 1) \div (P_{n-1} \leq p \leq P_n), (p \equiv T_n) \div (S_{чn} := S_{чn} + 1))$ на рис. 2 слева направо в порядке убывания их частоты срабатывания, после чего счетчики сбрасываются и начинается новый отсчет текущего времени $T_{тек}$. Тем самым на основе статистической информации, архивируемой в счетчиках $S_{ч1} \div S_{чn}$, через каждый промежуток времени, равный $T_{зад}$, происходит перестановка упомянутых выше цепочек операторов с целью минимизации времени отклика НР.

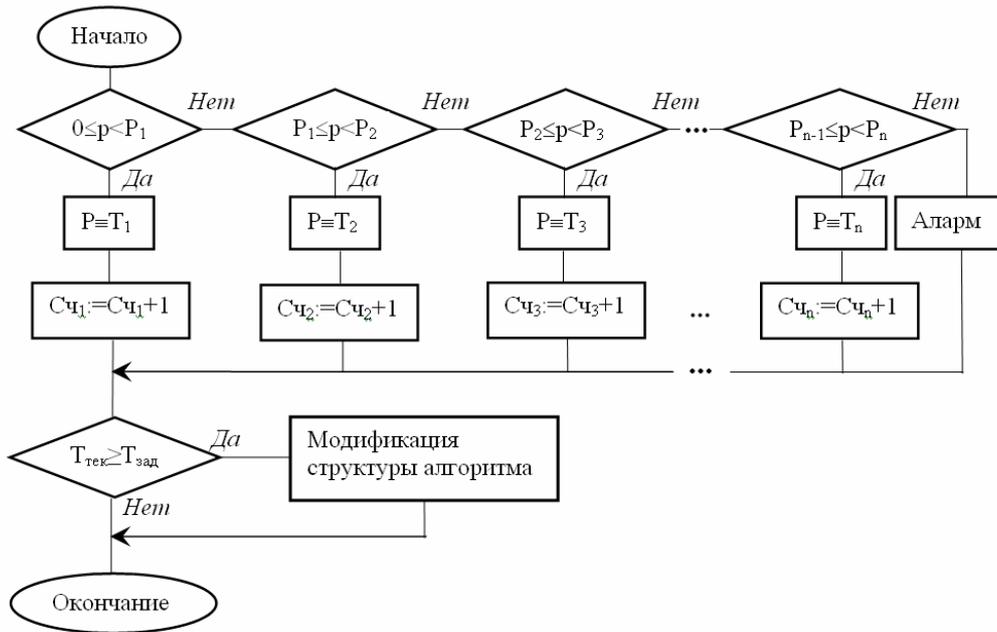


Рис. 2. Логическая схема ANY-TIME алгоритма минимизации продолжительности фаззификации параметра p в совокупность четких термов

3. КОЛИЧЕСТВЕННАЯ ОЦЕНКА МИНИМИЗАЦИИ ВРЕМЕНИ ФАЗЗИФИКАЦИИ С ПОМОЩЬЮ ANY-TIME АЛГОРИТМА

Пусть технологический параметр p в результате фаззификации эквивалентно преобразовывается в n четких термов $T_1 \div T_n$. Введем следующие обозначения: n_t — количество продукционных правил из системы (1), отработанных в текущем цикле сканирования алгоритма фаззификации; t_y и t_z — соответственно время отработки условной и заключительной частей одного продукционного правила; t_m — время проверки условия $T_{тек} \geq T_{зад}$.

С учетом принятых обозначений время цикла фаззификации можно выразить следующей формулой:

$$T_{ц} = n_t t_y + 2t_z + t_m. \quad (2)$$

При $n_t = 1$ из выражения (2) получается формула для минимального значения времени цикла:

$$T_{min} = t_y + 2t_z + t_m. \quad (3)$$

В этом случае в схеме на рис. 2 последовательно отработываются операторы $(0 \leq p < P_1) - (p \equiv T_1) - (S_{ч1} := S_{ч1} + 1) - (T_{тек} \geq T_{зад})$, а в системе продукционных правил (1) отработывается только первое правило, условная часть которого равна логической единице.

Аналогично из (2) при $n_t = n$ максимальное время цикла фаззификации

$$T_{1,0} = n t_y + 2t_z + t_m. \quad (4)$$

Применительно к схеме на рис. 2 выражение (4) соответствует ситуации, когда последовательно отработываются операторы $(0 \leq p < P_1); (P_1 \leq p < P_2); (P_2 \leq p < P_3); \dots; (P_{n-1} \leq p < P_n); (p \equiv T_n); (S_{чn} := S_{чn} + 1); (T_{тек} \geq T_{зад})$, а в системе продукционных правил (1) условная часть только последнего продукционного правила равна логической единице.

Семейство функций (2) $T_{ц} = f(n_t)$ для четырех наиболее известных производителей программируемых контроллеров приведено на рис. 3, из которого следует, что наибольшим быстродействием обладает контроллер фирмы YOKOGAWA, а наименьшим — российский контроллер КР-500. На этом же рисунке указаны минимальные $T_{min1} = 0,001 \text{ мс} \div T_{min4} = 0,004 \text{ мс}$ ($n_t = 1$) и максимальные $T_{max1} = 0,005 \text{ мс} \div T_{max4} = 26,6 \text{ мс}$ значения длительности сканирования алгоритма фаззификации при $n_t = 8$.

Используя те же обозначения, что и в (2), получим выражение для цикла сканирования $T_{тф}$ в типовом фаззификаторе, в котором, как известно [4; 5], при каждом цикле сканирования, безусловно, полностью и независимо от логического состояния antecedентов отработываются все продукционные правила.

Отсюда

$$T_{\text{тф}} = n(t_y + t_z). \quad (5)$$

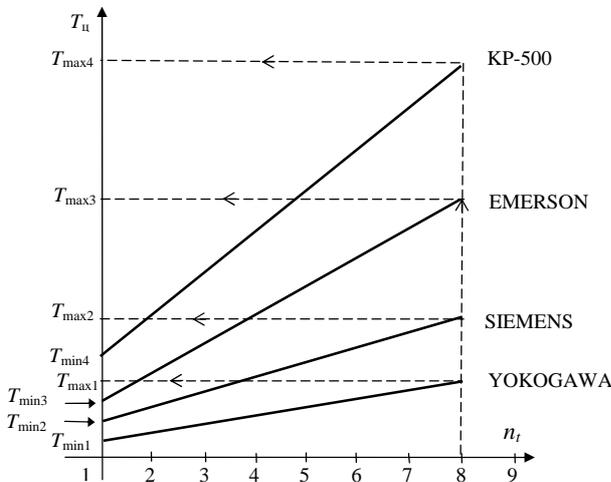


Рис. 3. Зависимость длительности цикла $T_{\text{ц}}$ сканирования алгоритма фаззификации от числа отработанных производственных правил n_t

Для упрощения выражений (3), (4) и (5) принимаем $t_y = t_z$, тогда они принимают следующий вид:

$$\begin{aligned} T_{\text{min}} &= 3t_z + t_m; \\ T_{1,0} &= (n + 2)t_y + t_m; \\ T_{\text{тф}} &= 2nt_y + t_m. \end{aligned} \quad (6)$$

Из выражений (6) разница между $T_{1,0}$ и T_{min} :

$$\Delta T_{1,0} = T_{1,0} - T_{\text{min}} = (n - 1)t_y. \quad (7)$$

В свою очередь, выражения (6) и (7) позволяют определить долю $\Delta T_{1,0}$ в продолжительности цикла сканирования $T_{\text{тф}}$ типового фаззификатора:

$$\frac{\Delta T_{1,0}}{T_{\text{тф}}} = 0,5 \left(1 - \frac{1}{n}\right). \quad (8)$$

Следует отметить, что даже при эффективной работе ANY-TIME алгоритма в каждом цикле сканирования обрабатываются (1÷15)% производственных правил, реализующих фаззификацию [8]. Поэтому реальное снижение продолжительности процедуры фаззификации в НР с четкими термами значительно меньше $\Delta T_{1,0}$, в связи с чем представляет практический интерес получение выражений, аналогичных выражениям (7) и (8) при (0,01; 0,1; 0,15)n:

$$\begin{aligned} \Delta T_{0,99} &= (0,99n - 1)t_y; \\ \frac{\Delta T_{0,99}}{T_{\text{тф}}} &= 0,495 - 0,5 \frac{1}{n}; \end{aligned} \quad (9)$$

$$\begin{aligned} \Delta T_{0,9} &= (0,9n - 1)t_y; \\ \frac{\Delta T_{0,9}}{T_{\text{тф}}} &= 0,45 - 0,5 \frac{1}{n}; \end{aligned} \quad (10)$$

$$\begin{aligned} \Delta T_{0,85} &= (0,85n - 1)t_y; \\ \frac{\Delta T_{0,85}}{T_{\text{тф}}} &= 0,425 - 0,5 \frac{1}{n}; \end{aligned} \quad (11)$$

Из рис. 4 следует, что наиболее интенсивное снижение продолжительности процедуры фаззификации имеет место при малых значениях n . По мере увеличения n эффективность работы ANY-TIME алгоритма вначале снижается, а затем стабилизируется на определенном уровне. Причем при $n_t = 1$ этот уровень составляет 0,5 (50%), при $n_t = 0,01n - 0,495$ (49,5%), при $n_t = 0,1n - 0,45$ (45%) и при $n_t = 0,15n - 0,425$ (42,5%).

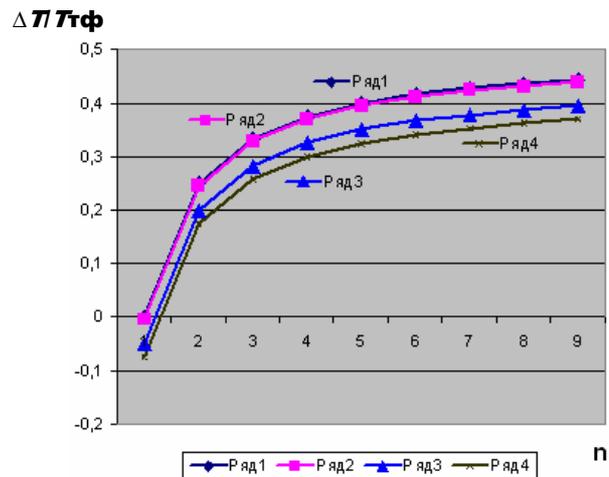


Рис. 4. Влияние ANY-TIME алгоритма на длительность процедуры. Ряд 1 — $\frac{\Delta T_{1,0}}{T_{\text{тф}}}$; ряд 2 — $\frac{\Delta T_{0,99}}{T_{\text{тф}}}$; ряд 3 — $\frac{\Delta T_{0,9}}{T_{\text{тф}}}$; ряд 4 — $\frac{\Delta T_{0,85}}{T_{\text{тф}}}$

Адекватность и достоверность полученных результатов подтверждена на программируемом контроллере Российского производства КР-500, на котором реализован НР с четкими термами, для регулирования температуры в стекловаренной печи. Длительность процедуры фаззификации НР, разработанного на упомянутом контроллере без ANY-TIME алгоритма составляет 0,073 мс, а при его использовании снизилась до 0,042 мс, то есть уменьшилась на 42%.

ВЫВОДЫ

1. Предложен алгоритм, позволяющий в режиме реального времени снизить длительность процедуры фаззификации нечетких регуляторов с четкими термами, а также получить аналитическое выражение, определяющее величины этого снижения.

2. По сравнению с типовыми нечеткими регуляторами, в нечетких регуляторах на основе четких термов продолжительность процедуры фаззификации на основе ANY-TIME алгоритма на (40÷50)% меньше, что позволяет применить их для управления быстродействующими технологическими процессами, представленными вербальной моделью.

3. Наиболее интенсивное снижение продолжительности процедуры фаззификации имеет место при количестве отработанных правил в текущем цикле сканирования $n_t = 1 \div 4$. При $n_t > 4$ рост эффективности работы ANY-TIME алгоритма вначале снижается, а при $n_t > 10$ стабилизируется на определенном уровне.

СПИСОК ЛИТЕРАТУРЫ

1. **Леоненков, А. В.** Нечеткое моделирование в среде MATLAB и fuzzy TECH / А. В. Леоненков. СПб. : БХВ-Петербург, 2005. 36 с.
2. **Деменков, Н. П.** Нечеткое управление в технических системах : учеб. пособие / Н. П. Деменков. М. : Изд-во МГТУ им. Н. Э. Баумана, 2005. 200 с.
3. **Штовба, С. Д.** Проектирование нечетких систем средствами MATLAB / С. Д. Штовба. М. : Горячая линия–Телеком, 2007. 288 с.

4. **Круглов, В. В.** Нечеткая логика и искусственные нейронные сети : учеб. пособие / В. В. Круглов, М. И. Длин, Р. Ю. Голунов. М. : Изд-во физ.-матем. лит-ры, 2001. 224 с.
5. **Усков, А. А.** Принципы построения систем управления с нечеткой логикой / А. А. Усков // Приборы и системы. Управление, контроль, диагностика. 2004. № 6. С. 7–13.
6. **Муравьева, Е. А.** Патент № 51242 на полезную модель, РФ / Е. А. Муравьева, Г. А. Каяшева, К. А. Колязов // Бюллетень изобретений. 2006. № 03.
7. **Муравьева, Е. А.** Патент № 2309443 РФ / Е. А. Муравьева, Г. А. Каяшева // Бюллетень изобретений. 2007. № 39.

ОБ АВТОРАХ



Муравьева Елена Александровна, доц. каф. автоматиз. технол. и инф. систем Филиала УГНТУ в Стерлитамаке. Дипл. инж. (УГНТУ, 1998). Канд. техн. наук по автоматиз. и упр. технол. проц. и произв. (УГАТУ, 2001). Иссл. в обл. интел. упр-я.



Каяшева Галина Александровна, асп. каф. автоматиз. хим.-технол. проц. УГНТУ. Дипл. инж. по автоматиз. (УГНТУ, 2005). Готовит дис. в обл. интел. упр. сл. техн. сист. с исп. нечеткой логики и дискр.-лог. моделей.