

Хрульков В. Н., Черный С. П.
V. N. Khrulkov, S. P. Cherny

ОСОБЕННОСТИ РЕАЛИЗАЦИИ НЕЧЁТКИХ АЛГОРИТМОВ УПРАВЛЕНИЯ НА БАЗЕ ПРОГРАММИРУЕМЫХ ЛОГИЧЕСКИХ КОНТРОЛЛЕРОВ

FEATURES OF FUZZY CONTROL ALGORITHM IMPLEMENTATION BASED ON PROGRAMMABLE LOGIC CONTROLLERS

Хрульков Владимир Николаевич – аспирант Комсомольского-на-Амуре государственного университета (Россия, Комсомольск-на-Амуре); 681013, Хабаровский край, г. Комсомольск-на-Амуре, пр. Ленина, 27. E-mail: kepapu@knastu.ru.

Vladimir N. Khrulkov – Post-graduate Student, Komsomolsk-na-Amure State University (Russia, Komsomolsk-on-Amur); 681013, Khabarovsk territory, Komsomolsk-on-Amur, Lenin Ave., 27. E-mail: kepapu@knastu.ru.

Черный Сергей Петрович – кандидат технических наук, доцент кафедры электропривода и автоматизации промышленных установок Комсомольского-на-Амуре государственного университета (Россия, Комсомольск-на-Амуре); 681013, Хабаровский край, г. Комсомольск-на-Амуре, пр. Ленина, 27. E-mail: kepapu@knastu.ru.

Sergey P. Cherny – PhD in Engineering, Associate Professor, Electric Drive and Automation of Industrial Installations Department, Komsomolsk-na-Amure State University (Russia, Komsomolsk-on-Amur); 681013, Khabarovsk territory, Komsomolsk-on-Amur, 27 Lenin str. E-mail: kepapu@knastu.ru.

Аннотация. В представленной работе рассматривается один из подходов к аппаратной реализации интеллектуальных систем управления на основе теории нечётких множеств. Синтез нечёткого логического регулятора выполнен на базе универсальных программируемых логических контроллеров (ПЛК) и реализует процедуры управления маломощным электроприводом. Выбор объекта регулирования и аппаратных средств, реализующих интеллектуальную систему управления, был произведён по результатам анализа доступности на рынке и технических характеристик, учитывающих особенности программирования таких систем с использованием универсальных ПЛК. В качестве алгоритма нечёткого логического вывода был выбран алгоритм Мамдани как наиболее гибкий, универсальный и обладающий большим потенциалом по перенастройке блоков дефаззификации. Представлены и проанализированы результаты работы нечёткого регулятора, управляющего скоростью вращения вала двигателя, проведён сравнительный анализ данных, полученных опытным путём на ПЛК, с результатами моделирования, полученными в среде инженерных вычислений MATLAB.

Summary. In the present paper, one of the approaches to the hardware implementation of intelligent control systems based on the theory of fuzzy sets is considered. The synthesis of a fuzzy logic controller is based on universal programmable logic controllers (PLCs) and implements control procedures for a low-power electric drive. The choice of the object of regulation and hardware implementing an intelligent control system was made based on the results of an analysis of market availability and technical characteristics that take into account the programming features of such systems using universal PLCs. The Mamdani algorithm was chosen as the fuzzy inference algorithm, as the most flexible, versatile and with great potential for reconfiguring defuzzification blocks. The results of the operation of a fuzzy controller regulating the speed of motor shaft rotation are presented and analyzed, a comparative analysis of the data obtained experimentally on a PLC with the simulation results in the Matlab engineering computing environment is carried out.

Ключевые слова: нечёткий логический регулятор, ПЛК, электродвигатель постоянного тока, система управления.

Key words: fuzzy current controller, PLC, DC motor, control system.

Исследования, изложенные в материале статьи, проводились в рамках гранта № ВНО10-2021 «Разработка принципов построения интеллектуальных систем управления сложными техническими объектами на основе критериев энергоэффективности».

Введение

Технологии управления, основанные на теории нечётких множеств, в настоящий момент нашли своё применение при формализации процессов, протекающих в сложных технологических объектах, и активно применяются для реализации сложных законов регулирования таких объектов. Нечёткие алгоритмы управления в совокупности с использованием классических способов и методик регулирования позволяют существенно улучшить ряд динамических характеристик развитых систем автоматизации, упростить синтез процедур управления такими системами, а также расширить возможности учёта и компенсации детерминированных и недетерминированных возмущающих воздействий.

Главной проблемой, связанной с аппаратной реализацией интеллектуальных систем управления, основанных на мягких вычислениях, является отсутствие специализированных средств промышленного исполнения или их недоступность вследствие введённых ограничительных мер. В качестве примера можно отметить ряд продуктов компании Siemens, которые на текущий момент либо сняты с производства, либо исключены из поддержки, среди них микроконтроллер Siemens SAE81C99A и программное обеспечение FuzzyControl++. Для других известных производителей подобных технических средств, таких как Arduino, Schneider Electric, Raspberry Pi, характерны сложности, обусловленные интеграцией и существенной ограниченностью специализированных библиотек, позволяющих синтезировать рассматриваемые нечёткие логические контроллеры.

Выполнение подобного рода систем управления с применением универсальных промышленных контроллеров представляет собой достаточно трудоёмкую задачу по причине необходимости программной реализации механизмов нечёткого вывода с использованием специализированных средств проектирования.

Для аппаратной реализации нечёткого регулятора, имеющего в своей основе ПИД-закон управления, предлагается использовать аппаратно-программную платформу Arduino DUE, имеющую 32-битное ядро, позволяющее выполнять операции с данными шириной 4 байта за 1 такт (частота процессора (CPU) 84 МГц), и библиотеку Embedded Fuzzy Logic Library (eFLL) для реализации систем, основанных на мягких вычислениях.

В качестве объекта управления используется электродвигатель постоянного тока серии Tetrix MAX, на валу которого установлен квадратурный энкодер US Digital для реализации обратной связи по скорости. Анализ технических характеристик показал, что быстродействие контроллера и датчика в обратной связи достаточно для реализации нечёткого ПИД-регулятора на предлагаемой аппаратной платформе. Пропорциональный канал регулирования скорости вращения электродвигателя постоянного тока представляет собой ШИМ в совокупности с нечётким корректирующим модулем, в основе которого заложен алгоритм нечёткого логического вывода Мамдани. Предварительная настройка регулятора осуществлялась с применением языка инженерных вычислений MATLAB, т. к. это позволяет оценить адекватность реализованных моделей. Использование функций принадлежности треугольного вида при формализации входных и выходных лингвистических переменных в модели системы и интеллектуальном регуляторе, реализованном на микроконтроллере, приводит к некоторому рассогласованию между выходными сигналами, описывающими переходные характеристики по скорости.

Преимущество использования аппаратной платформы Arduino заключается в её доступности на рынке, простоте языка программирования, наличии и удобстве использования стандартного набора библиотек, а также возможности подключения большого количества разных типов датчиков и исполнительных механизмов. Вместе с тем необходимо отметить, что существенными недостатками библиотеки eFLL для дальнейших исследований при построении интеллектуальных си-

стем являются отсутствие нечёткого логического вывода Сугено, использование центроидного метода приведения к чёткости, а также применение функций принадлежности исключительно аппроксимированного вида.

Методика реализации нечёткого логического регулятора аппаратными средствами

Основой для формирования интеллектуальной системы управления с использованием нечёткого логического регулятора (НЛР) на основе ПЛК будет сравнительный анализ функционирования блоков дефаззификации на имитационной модели, реализованной средствами языка инженерных вычислений.

На начальном этапе синтез основных блоков НЛР был проведён в программной среде MATLAB с использованием модуля Fuzzy Logic Toolbox. Для проверки работоспособности НЛР на ПЛК была использована аппаратно-программная платформа Arduino DUE, имеющая 32-битное ядро и библиотеку Embedded Fuzzy Logic Library (eFLL) для реализации систем, основанных на мягких вычислениях (см. рис. 1).

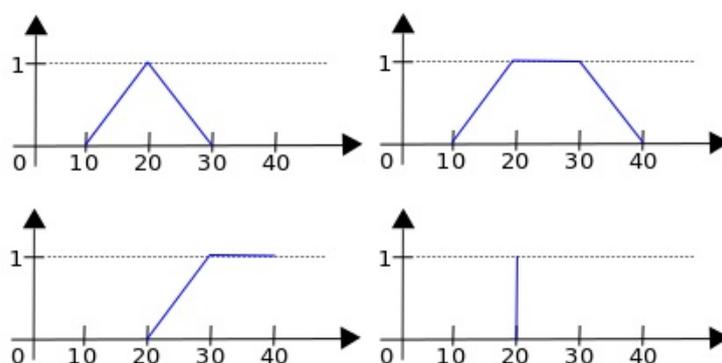


Рис. 1. Функции принадлежности, предоставляемые библиотекой eFLL

На рис. 2 приведена функциональная схема нечёткого модуля, используемого для предварительного тестирования НЛР. При построении был использован алгоритм нечёткого логического вывода Мамдани с центроидным методом приведения к чёткости, а также одной входной лингвистической переменной и одной выходной. Выбранная библиотека eFLL существенно ограничена и настроена на реализацию алгоритма вывода Мамдани. Кроме того, необходимо отметить, что ограничения также распространяются на форму функций принадлежности (сигмоидальные, гауссовы, z- и s-образные), методы приведения к чёткости (первого и среднего максимумов), а также алгоритм вывода Такаги-Сугено.

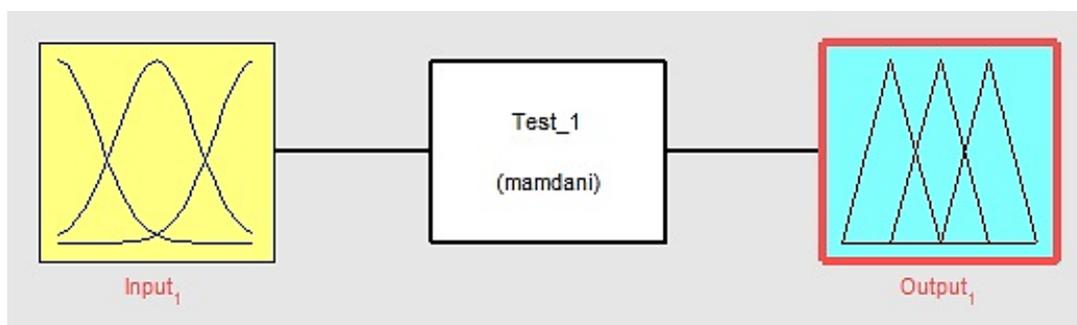


Рис. 2. Нечёткий модуль

На рис. 3 и 4 представлены входная и выходная лингвистические переменные НЛР с тремя элементами в базовом терм-множестве и функциями принадлежности треугольной формы. Необходимо отметить, что для проверки качества функционирования блоков дефаззификации данных НЛР для ПЛК используются функции принадлежности аппроксимированного вида.

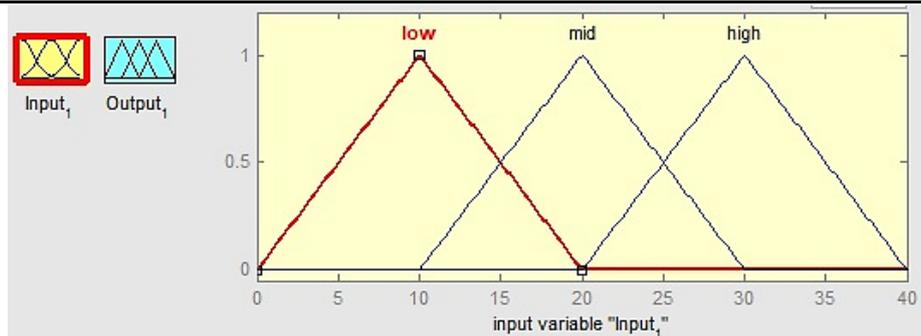


Рис. 3. Входная лингвистическая переменная НЛР

База нечётких продукционных правил имеет следующий вид:

1. If (input_1 is low) then (Output_1 is low);
2. If (input_1 is mid) then (Output_1 is mid);
3. If (input_1 is high) then (Output_1 is high);

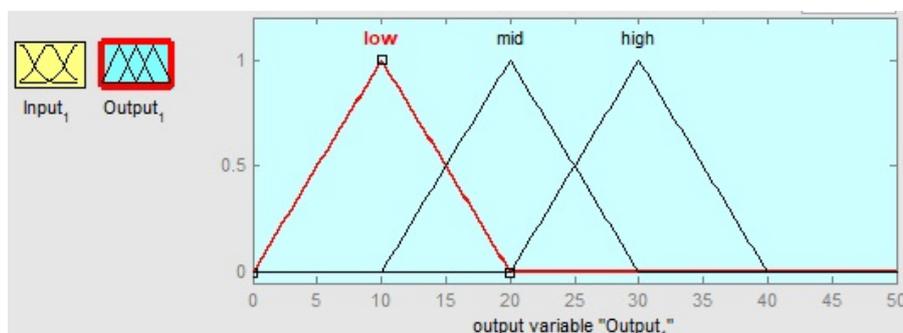


Рис. 4. Выходная лингвистическая переменная НЛР

В табл. 1 представлены результаты дефаззификации регуляторами, реализованными средствами имитационного моделирования и ПЛК. Данные для входных параметров в интервалах от 1 до 10 и от 30 до 40 были исключены в связи с полным совпадением результатов нечёткого вывода у ПЛК и модели. Из таблицы видно, что в зоне пересечения функций принадлежности имеется повторяющаяся ошибка при генерации результата НЛР, которую необходимо нивелировать в дальнейшем при построении системы с использованием ПЛК. Наличие данной ошибки связано с тем, что алгоритм нечёткого вывода Мамдани, осуществлённый в библиотеке eFLL для Arduino, реализован не в полной мере и имеет ряд существенных ограничений.

Далее проведём моделирование работы нечёткого регулятора с применением языка инженерных вычислений. Основные параметры модели были получены непосредственно с объекта регулирования, а также выведена зависимость изменения скорости вращения вала двигателя от заполнения ШИМ с учётом ограничений обеспечения пуска двигателя минимальным значением заполнения.

База нечётких продукционных правил имеет следующий вид:

1. If (RPS is Low) then (PWM is LowP);
2. If (RPS is LowMid) then (PWM is LowMidP);
3. If (RPS is Mid) then (PWM is MidP);
4. If (RPS is MidHigh) then (PWM is MidHighP);
5. If (RPS is High) then (PWM is HighP);

После настройки и тестирования нечёткого регулятора на имитационной модели проведём анализ его работоспособности непосредственно с применением ПЛК. На рис. 5 представлена схема подключения основных элементов к плате контроллера.

Результаты дефазификации средствами ПЛК и MATLAB

Входные параметры	Вывод MATLAB (x1)	Вывод ПЛК (x2)	Ошибка (x1-x2)
11	11,3	11,61	-0,31
12	12,4	12,73	-0,33
13	13,3	13,59	-0,29
14	14,2	14,32	-0,12
15	15	15	0
16	15,8	15,8	0
17	16,7	16,61	0,09
18	17,6	17,43	0,17
19	18,7	18,28	0,42
20	20	20	0
21	21,3	21,61	-0,31
22	22,4	22,73	-0,33
23	23,3	23,59	-0,29
24	24,2	24,32	-0,12
25	25	25	0
26	25,8	25,8	0
27	26,7	26,61	0,09
28	27,6	27,43	0,17
29	28,7	28,28	0,42

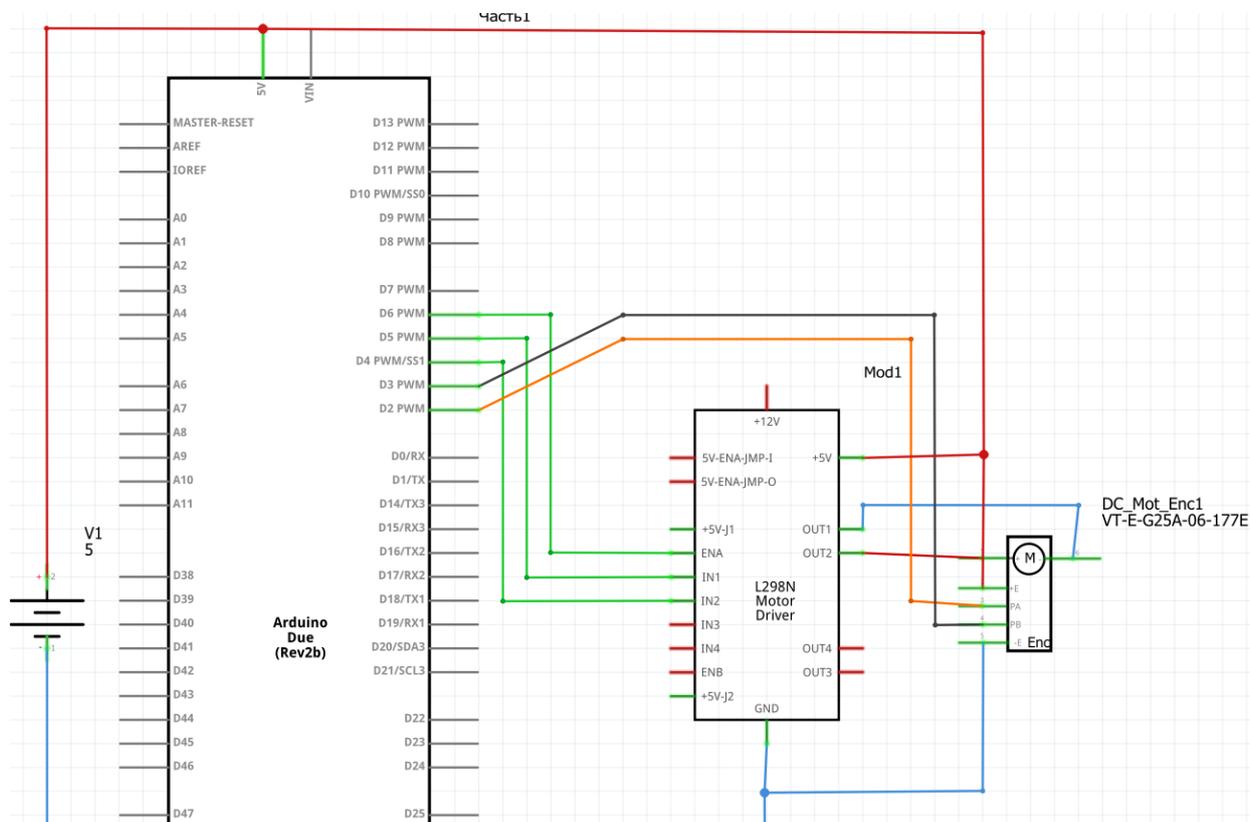


Рис. 5. Функциональная схема подключения ПЛК

Алгоритм подключения электропривода реализован по следующей схеме: фазы двигателя подключены к драйверу двигателя L298N в разъёмы OUT1 и OUT2; питание энкодера 5В и Gnd берётся непосредственно с портов контроллера; для обработки показаний с датчика разъёмы энкодера А и В подключаются к цифровым портам 2 и 3 ПЛК соответственно, а драйверы ena, in1 и in2 соединены с портами контроллера 6, 5 и 4, функции которых состоят в задании уровня ШИМ и направления вращения вала двигателя.

Ниже представлен синтаксис настройки входных и выходных термов нечёткого регулятора треугольной формы, которые были отстроены с помощью функции MATLAB Fuzzy Logic Toolbox.

Настройка входных термов скорости вращения вала электродвигателя на Arduino:

```
FuzzyInput *RPS = new FuzzyInput(1); // создание первого нечёткого входа
```

```
FuzzySet *LowR = new FuzzySet(83.36, 87.74, 87.74, 101); // создание термов  
RPS->addFuzzySet(LowR);
```

```
FuzzySet *LowMidR = new FuzzySet(91.4, 100, 100, 111); // создание термов  
RPS->addFuzzySet(LowMidR);
```

```
FuzzySet *MidR = new FuzzySet(101, 111, 111, 121);  
RPS->addFuzzySet(MidR);
```

```
FuzzySet *MidHighR = new FuzzySet(111, 121, 121, 130);  
RPS->addFuzzySet(MidHighR);
```

```
FuzzySet *HighR = new FuzzySet(121, 134.4, 134.4, 186);  
RPS->addFuzzySet(HighR);
```

```
fuzzy->addFuzzyInput(RPS);
```

Настройка выходных термов уровня ШИМ Arduino:

```
FuzzyOutput *PWM_1 = new FuzzyOutput(1); // создание первого нечёткого выхода
```

```
FuzzySet *LowP = new FuzzySet(60, 80, 80, 100);  
PWM_1->addFuzzySet(LowP);
```

```
FuzzySet *LowMidP = new FuzzySet(80.1, 102, 102, 124.9);  
PWM_1->addFuzzySet(LowMidP);
```

```
FuzzySet *MidP = new FuzzySet(105, 137, 137, 160);  
PWM_1->addFuzzySet(MidP);
```

```
FuzzySet *MidHighP = new FuzzySet(140, 157.5, 157.5, 240);  
PWM_1->addFuzzySet(MidHighP);
```

```
FuzzySet *HighP = new FuzzySet(230, 255, 255, 280);  
PWM_1->addFuzzySet(HighP);
```

```
fuzzy->addFuzzyOutput(PWM_1);
```

Далее представлен фрагмент описания базы знаний НЛР, который соответствует данным, полученным с учётом имитационного моделирования (см. рис. 6 и 7).

Объявление правил нечёткого регулятора:

```

//настройка первого правила
FuzzyRuleAntecedent *ifRPSLow = new FuzzyRuleAntecedent();
ifRPSLow->joinSingle(LowR);

FuzzyRuleConsequent *thenPWM_1_Low = new FuzzyRuleConsequent();
thenPWM_1_Low->addOutput(LowP);

FuzzyRule *fuzzyRule01 = new FuzzyRule(1, ifRPSLow, thenPWM_1_Low);
fuzzy->addFuzzyRule(fuzzyRule01);

// настройка второго правила
FuzzyRuleAntecedent *ifRPSLowMid = new FuzzyRuleAntecedent();
ifRPSLowMid->joinSingle(LowMidR);

FuzzyRuleConsequent *thenPWM_1_LowMid = new FuzzyRuleConsequent();
thenPWM_1_LowMid->addOutput(LowMidP);

FuzzyRule *fuzzyRule02 = new FuzzyRule(2, ifRPSLowMid, thenPWM_1_LowMid);
fuzzy->addFuzzyRule(fuzzyRule02);

```

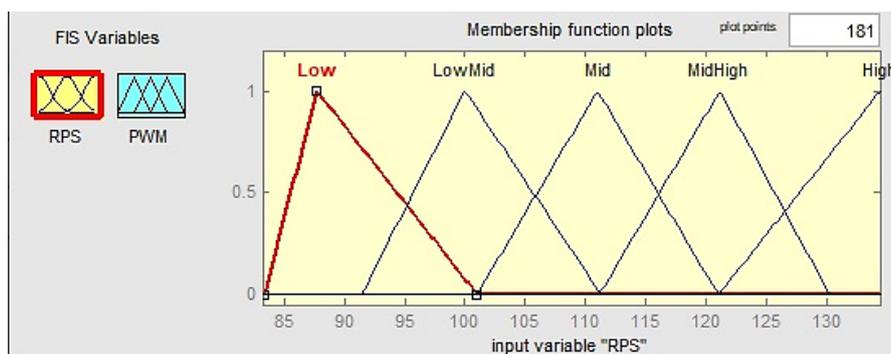


Рис. 6. Базовое терм-множество лингвистической переменной «скорость вращения вала»

Фрагмент программного кода считывания входного значения скорости вращения с монитора порта:

```

void loop ()
{
//ввод данных через монитор порта

input_m = Serial.readString();
irm = input_m.toInt();
if (0 < input_m.length() && irm >= 0 && irm <= 25500)
{
input = float(irm) / 100.00;
if (input<=90) {input=90;}
if (input>=130) {input=130;}
}
}

```

С помощью монитора порта задаётся входное значение необходимой скорости вращения вала двигателя с учётом следующих ограничений: выбранный двигатель может развивать минимальную скорость в 90 об/мин и максимальную в 130 об/мин.

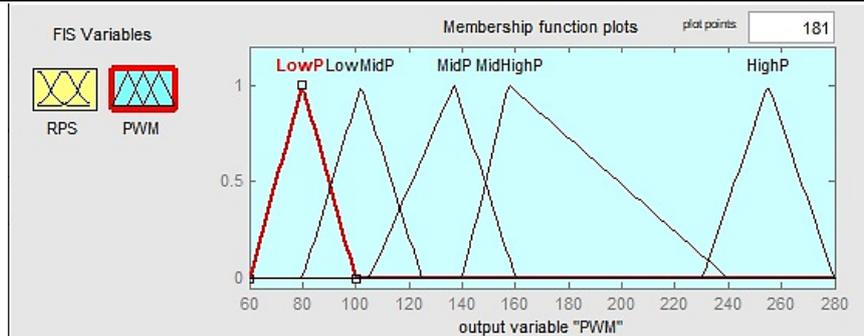


Рис. 7. Базовое терм-множество лингвистической переменной «уровень ШИМ»

Код запуска электродвигателя:

```
if ((input > 90) && (P_start == HIGH)) //запуск двигателя параметром с монитора порта с последующей остановкой данной команды
```

```
{
  PWM_S = PWM_S + 50;
  if (PWM_S >= 255) { PWM_S = 255; }
  analogWrite(ena, PWM_S);
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
}
```

```
if (velRPS >= input)
{P_start = LOW;}
}
```

Алгоритм запуска двигателя осуществляется по следующей схеме:

- проверяется входной параметр скорости и сравнивается с пороговым значением;
- логическая переменная, отвечающая за первичный пуск двигателя, проверяется на истинность ($P_Start = 1$);
- увеличивается уровень ШИМ (переменная PWM_S) до тех пор, пока значение показаний энкодера не станет равным заданному значению;
- управление двигателем переходит к нечёткому регулятору.

Программный код управления скоростью вращения двигателя с использованием НЛР имеет следующую нотацию:

```
if (P_start == LOW)
{
  // регулятор
  Error = input - velRPS;
  U = k * Error;
  RPS = input + U;
  Serial.print (" // ");
  Serial.print ("RPS = "); Serial.print (RPS);
  Serial.print (" // ");
  Serial.print ("Error = "); Serial.print (Error);
  Serial.println("\n\n\nEntrance: ");
  Serial.print("\t\t\tRPS: ");
  Serial.println(RPS);
  fuzzy->setInput(1, RPS);
  fuzzy->fuzzify();
  float output = fuzzy->defuzzify(1);
  Serial.println("Result: ");
}
```

```

Serial.print("\t\tPWM_1: ");
Serial.println(output);

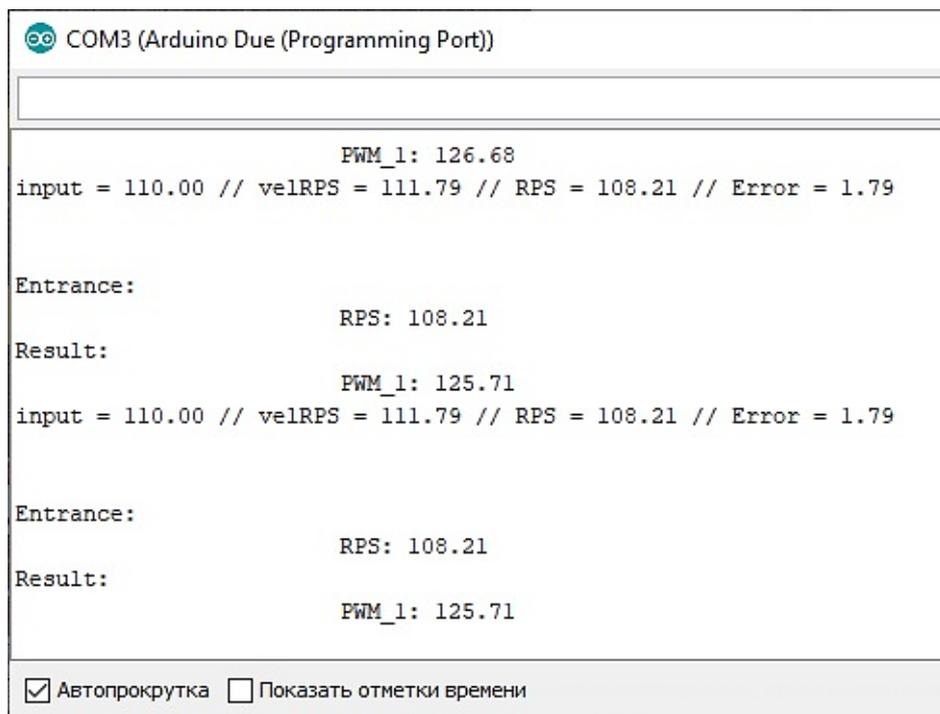
analogWrite( ena, output );
digitalWrite( in1, HIGH );
digitalWrite( in2, LOW );

}
}

```

В базовом варианте пропорциональная составляющая закона управления формируется с использованием стандартных классических подходов, а затем передаётся в качестве уставки на интеллектуальное управляющее устройство (переменная первичного запуска $P_Start = 0$). Рассчитанное значение отклонения передаётся на вход нечёткого регулятора, и после дефаззификации полученный результат записывается в переменную *output*, которая отвечает за уровень сигнала ШИМ, подаваемый на установленный порт контроллера.

На рис. 8 – 11 представлены результаты работы регулятора при различных входных задающих воздействиях. Из полученных результатов можно сделать вывод, что между интеллектуальной системой управления, реализованной с использованием ПЛК, и имитационной моделью существует небольшое отклонение. Подобного рода расхождения обусловлены особенностями реализации алгоритмов нечёткого вывода с применением языка инженерных вычислений и весьма ограниченными возможностями специализированного программного обеспечения ПЛК.



```

COM3 (Arduino Due (Programming Port))

PWM_1: 126.68
input = 110.00 // ve1RPS = 111.79 // RPS = 108.21 // Error = 1.79

Entrance:
RPS: 108.21
Result:
PWM_1: 125.71
input = 110.00 // ve1RPS = 111.79 // RPS = 108.21 // Error = 1.79

Entrance:
RPS: 108.21
Result:
PWM_1: 125.71

 Автопрокрутка  Показать отметки времени

```

Рис. 8. Результат работы нечёткого регулятора на Arduino при входном значении 110

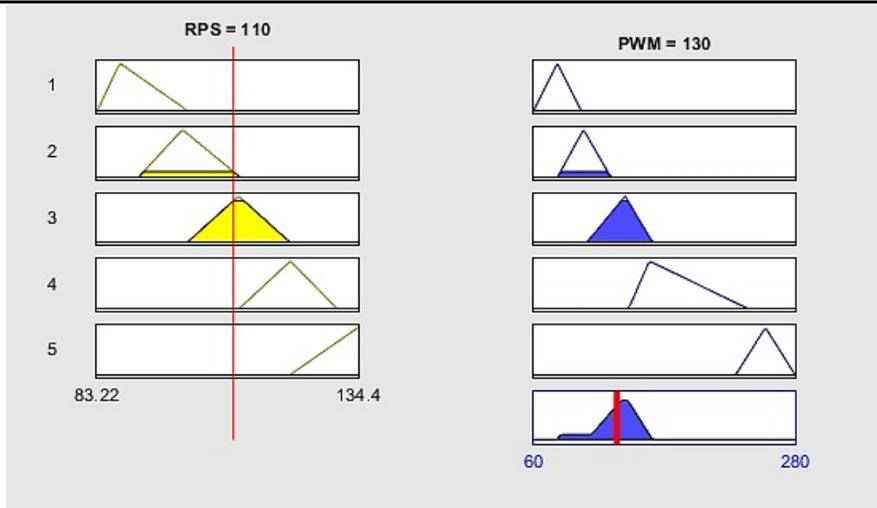


Рис. 9. Результат работы смоделированного нечёткого регулятора при входном значении 110

```

COM3 (Arduino Due (Programming Port))

PWM_1: 201.23
input = 125.00 // velRPS = 125.12 // RPS = 124.88 // Error = 0.12

Entrance:
RPS: 124.88
Result:
PWM_1: 199.83
input = 125.00 // velRPS = 124.68 // RPS = 125.32 // Error = -0.32

Entrance:
RPS: 125.32
Result:
PWM_1: 201.95

 Автопрокрутка  Показать отметки времени

```

Рис. 10. Результат работы нечёткого регулятора на Arduino при входном значении 125

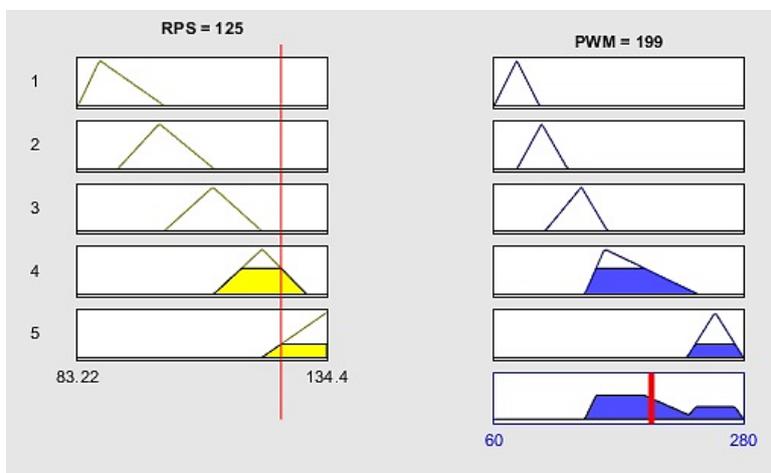


Рис. 11. Результат работы смоделированного нечёткого регулятора при входном значении 125

Заключение

Анализ результатов, полученных в ходе имитационного моделирования с применением стандартных программных платформ и экспериментальных данных, формируемых на основе программной реализации механизма нечёткого логического вывода, с использованием аппаратных средств показал минимальную разницу при дефазификации выходной лингвистической переменной. Наличие ошибки обусловлено рядом весьма существенных условий, заложенных в библиотеку eFLL, среди которых необходимо отметить ограничения, связанные с количеством и видом функций принадлежности, а также с отсутствием алгоритма нечёткого вывода Сугено. Также возможности библиотеки ограничивают реализацию более сложных структур и форм нечётких систем управления. Однако необходимо отметить, что применение такого подхода позволит выполнить элементарные интеллектуальные модули, формирующие процедуры управления для систем регулирования нижнего уровня автоматизации, с учётом отсутствия у основных производителей специализированных аппаратных средств.

ЛИТЕРАТУРА

1. Cherniy S. P., Solovyev V. A. General Principals of Building the Fuzzy Multi-Cascade Tracking Control Systems / 2018 International Multi-Conference on Industrial Engineering and Modern Technologies, FarEastCon 2018. 2019. С. 8602646. DOI: 10.1109/FarEastCon.2018.8602646.
2. S. P. Cherniy, V. I. Susdorf, A. V. Buzikaeva, V. N. Khrukov Analysis of approaches to modelling the fuzzy control systems with extension of their functional capabilities / EAI Endorsed Transactions on Energy Web, 2020, ew 18: e19, DOI:10.4108/eai.13-7-2018.165496.
3. V. V. Trotsenko, «Technological process control Systems and information technologies» / V. V. Trotsenko [et al.]. – Moscow: Yurayt, 2017. – 160 p.
4. R. Goswami, D. Joshi, «Performance Review of Fuzzy Logic Based Controllers Employed in Brushless DC Motor», Proce-dia Computer Science, vol. 132, pp. 623-631, 2018.
5. A. M. Zaki, M. El-Bardini, F. A. S. Soliman, and M. M. Sharaf, «Embedded two level direct adaptive fuzzy controller for DC motor speed control», Ain Shams Engineering Journal, vol. 9, pp. 65-75, March 2018.
6. B. N. Kommula, and V. R. Kota, «Direct instantaneous torque control of Brushless DC motor using firefly Algorithm based fractional order PID controller», Journal of King Saud University – Engineering Sciences, vol. 32, issue 2, pp. 133-140, February 2020.
7. A. R. Ofoli, «Fuzzy-Logic Applications in Electric Drives and Power Electronics», Power Electronics Handbook (Fourth Edition), 2018, pp. 1221-1243.
8. A. de la Guerra, L. Alvarez-Icaza, Lizeth Torres, «Brushless DC motor control with unknown and variable torque load», IFAC-PapersOnLine, vol. 5, Issue 13, 2018, pp. 644-649.
9. A. Rodriguez-Molina, M. G. Villarreal-Cervantes, J. Alvarez-Gallegos, Mario Aldape-Perez, «Bio-inspired adaptive control strategy for the highly efficient speed regulation of the DC motor under parametric uncertainty», Applied Soft Computing, vol. 75, pp. 29-45, February 2019.
10. Solovyev V. A., Deryuzhkova N. E., Htet K. A. Fuzzy-Based Multi-Point Interconnected Control System of Electrical Drives for Sheet Met-Al Forming / 2018 International Multi-Conference on Industrial Engineering and Modern Technologies, FarEastCon 2018. 2019. С. 8602706. DOI: 10.1109/FarEastCon.2018.8602706.
11. Пивоваров, А. И. Адаптивное управление по состоянию контуром скорости с настройкой на модульный оптимум / А. И. Пивоваров, С. В. Стельмашук // Научно-техническое творчество аспирантов и студентов. – Комсомольск-на-Амуре: ФГБОУ ВО «КНАГУ», 2017. – С. 861-864.
12. Стельмашук, С. В. Синтез следящего электропривода с контуром скорости, настроенного на модульный оптимум / С. В. Стельмашук, В. А. Бунаков // Учёные записки Комсомольского-на-Амуре государственного технического университета. Науки о природе и технике. – 2016. – № IV-1 (28). – С. 54-62.