

УДК 004.75
DOI: 10.15827/0236-235X.133.033-046

Дата подачи статьи: 28.01.21
2021. Т. 34. № 1. С. 033–046

Алгоритмы паттернов интеллектуального узла в составе беспроводной сенсорной сети

Г.П. Виноградов^{1,2}, д.т.н., профессор, зав. лабораторией, *wgp272ng@mail.ru*

*А.С. Емцев*², инженер-программист, *emcev@cps.tver.ru*

*И.С. Федотов*², инженер-программист, *twlucifer@gmail.com*

¹ *Тверской государственный технический университет, г. Тверь, 170026, Россия*

² *НИИ «Центрпрограммсистем», г. Тверь, 170024, Россия*

Одной из тенденций в области развития современных вооружений является «связывание» отдельных образцов с определенной степенью автономии в комплекс с использованием, как правило, беспроводных сенсорных сетей. Область применения таких комплексов – неопределенные и плохо формализуемые среды. Достичь их желаемой эффективности возможно, главным образом, путем совершенствования интеллектуальной составляющей системы управления комплексом в целом и отдельным узлом в частности. Однако следует отметить, что подавляющее число исследований в этой области остается лишь на теоретическом уровне. Существует разрыв между примитивными моделями поведения искусственных существ, например, в роевой робототехнике, моделями их взаимодействия и ожиданиями со стороны практики. Ситуация усугубляется требованиями скрытности, миниатюризации, малого энергопотребления.

Цель данного исследования – представить подход к разработке ПО интеллектуальных систем управления отдельным узлом сети, обладающим заданной степенью автономии при выполнении задач. Предложить в условиях ограничений по энергопотреблению и быстродействию сравнительно простые алгоритмы для придания узлу сети свойства интеллектуального поведения, обеспечить способность к изучению обстановки и принятию решений как самостоятельно с учетом получаемых от других устройств сети данных, так и в составе группы.

В работе используются методы теории нечетких множеств, теории построения нечетких моделей и сетей, а также подходы и алгоритмы построения бортовых интеллектуальных систем управления.

Показано, что требуемые алгоритмы можно разработать, если выявить классы типовых ситуаций и успешные способы действия в реальных условиях. На этой основе появляется возможность разработки формальных моделей поведения (паттернов) для реализации в системе управления узлом.

Предложена двухуровневая структура интеллектуальной системы управления сетью. Верхний уровень, реализуемый оператором, соответствует таким свойствам, как выживание, безопасность, выполнение обязательств согласно миссии, накопление и корректировка базы знаний в виде эффективных паттернов поведения. Объектом управления для нее является сеть, рассматриваемая как некая функциональная система. Она осуществляет расчет текущих показателей удельной ценности по результатам и эффективности в момент t , расчет и реализацию способа действия (поведения) в момент t согласно заданному паттерну поведения, мониторинг результатов реализации паттерна поведения.

Ключевые слова: *модель развития, беспроводная сеть, сенсор, искусственный интеллект.*

Современные методы ведения боевых действий предполагают активное использование *беспроводных сенсорных сетей* (БСС) с элементами искусственного интеллекта и достаточно высокой степенью автономии входящих в нее узлов. Например, каждая боевая единица может быть снабжена датчиками, которые фиксируют сведения о среде и о ее состоянии, а также средствами выполнения боевой задачи. Наличие сети позволяет собирать и передавать данные в пункты управления всех уровней. Данные могут собираться для разведки, изучения сил противника и местности, для наблюдения за боем, внесения корректировок в такти-

ческие задачи, для оценки ущерба после боя, обнаружения ядерных, биологических и химических атак, а также для использования их в системах наведения интеллектуальных боеприпасов.

Область применения БСС – это, как правило, неопределенные и плохо формализуемые среды [1], поэтому достичь желаемой эффективности подобных комплексов возможно, главным образом, путем совершенствования интеллектуальной составляющей их системы управления в целом и отдельным узлом в частности. Однако следует отметить, что подавляющее число исследований в этой области оста-

ется на теоретическом уровне [2]. Существует разрыв между примитивными моделями поведения искусственных существ, например, в роевой робототехнике, моделями их взаимодействия и ожиданиями со стороны практики [3, 4]. Ситуация усугубляется требованиями скрытности, миниатюризации, малого энергопотребления [5–7].

Результаты в области разработки MEMS-сенсоров, информационных технологий и систем связи с помощью радиоканалов и беспроводных протоколов позволили создать БСС. Типичная структура БСС включает многофункциональные моты (узлы) с малым уровнем энергопотребления, способные к коммуникации друг с другом, обеспечивающие сбор информации о состоянии предметной области и сетевой архитектуры для передачи данных в удаленные компьютерные сети [1].

Как правило, сенсорный узел в БСС выполняет только сбор данных и состоит из четырех основных компонентов: источник питания, датчик, блок обработки, система связи. Датчик аналогового типа фиксирует значение параметра предметной области, *аналого-цифровой преобразователь* (АЦП) переводит эти данные в цифровую форму. Микроконтроллер выполняет обработку данных и манипулирование ими. Система связи, как правило, состоит из радиостанций ближнего действия. Для питания системы используется небольшая батарея. Кроме того, сенсорный узел выполняет ряд важных функций, позволяющих решать задачи по передаче данных предметной области, в частности, анализ сети, объединение данных другого узла с собственными данными [1]. Для обеспечения беспроводной связи между устройствами с низким энергопотреблением, с возможностью выстраивания ячеистой топологии сети используется протокол обмена данными ZigBee. При сравнительно низкой скорости передачи данных он обеспечивает низкое энергопотребление, что имеет важное значение для живучести сети.

На практике в качестве мота сети требуется иметь автономный узел (агент), в котором реализуется набор функционала для выполнения требуемых задач в его зоне ответственности по информации, получаемой от собственных сенсоров и в результате коммуникации [8, 9]. Такой мот должен обладать свойством интеллектуального поведения, быть способным изучать обстановку и принимать решения как самостоятельно с учетом получаемых от других устройств сети данных, так и в составе группы.

Жесткие требования в ряде предметных областей к времени реагирования, объему памяти, быстродействию микрочипа и уровню энергопотребления предполагают использование сравнительно простых алгоритмов, построенных путем описания лучшего опыта человека при выполнении функциональных задач. В работах [10–12] показано, что такое описание можно реализовать, если выявить классы типовых ситуаций и успешные способы действия в реальных условиях для формирования моделей поведения (паттернов).

Паттерн – это результат активности естественной или искусственной сущности, связанный с действием, принятием решений, его реализацией и т.п., осуществленный в прошлом и рассматриваемый как шаблон (образец) для повторных действий или как обоснование действий по этому шаблону [13].

В работе [14] дана следующая обобщенная логическая схема описания паттерна:

Имя паттерна:

так как [мотивы M]
 поскольку [цели G]
 если [предусловия U']
 то способ действия $rq(t)$
 из-за чего [постусловие U'']
 ...
 есть альтернатива [$rp(t)$]

В этой модели все составляющие, кроме логических связей, могут представлять собой языковые конструкции на естественном или естественно-профессиональном языке. Следовательно, модель (1) – типовая естественно-языковая модель паттерна.

Формальная модель паттерна поведения в типовой ситуации приведена в [14, 15]. Там же показано, что модель предметной области и модель принятия решений человеком в условиях дефицита времени можно формализовать с помощью нечетких продукционных сетей.

Мотивы в модели (1) – это причины, определяемые потребностями и/или обязательствами. Они побуждают агента к активности, в рамках которой потребовался паттерн. Описание мотивов и целей в модели (1) представляет контекст повторного применения паттерна с позиции необходимости и возможности его применения (потребности → мотивы → цели). Контекст определяется ожидаемыми постусловиями. Постусловия связаны с целями, затребовавшими паттерн, но отражают их в форме возможного состояния, к которому приведет применение паттерна. Цели определяют желаемое состояние и могут быть причиной, опре-

деляющей необходимость и полезность стремления к их достижению. Способ действия представляет собой код как естественно-языковое описание схемы действия, реализация которой приводит к выполнению паттерна. Такое описание имеет вид методик, реализованных на языке программирования. Набор моделей или паттернов поведения образует опыт, или базу знаний его носителя. Существуют методики извлечения наиболее успешных моделей для их распространения.

Такая модель индивидуального поведения автономного узла предполагает формирование базы паттернов на основе экспериментального опыта, что позволяет реализовать эволюцию «кооперативного интеллекта» посредством искусственного когнитивного процесса, аналогичного имеющему место у естественных существ [13, 14]. Отметим, что этой возможности не имеют системы на основе знаний, так как в ней отсутствует компьютерная модель адаптивного поведения.

Общая архитектура интеллектуальной системы управления БСС имеет иерархическую структуру (рис. 1). Верхний уровень, реализуемый оператором, соответствует таким свойствам, как выживание, безопасность, выполнение обязательств согласно миссии, накопление и корректировка базы знаний в виде эффективных паттернов поведения. Объектом управления для нее является БСС, рассматриваемая как

некоторая функциональная система. Она осуществляет

- расчет текущих показателей удельной ценности по результатам и эффективности в момент t ;
- расчет и реализацию способа действия (поведения) в момент t согласно заданному паттерну поведения;
- мониторинг результатов реализации паттерна поведения.

Оператор обрабатывает паттерны поведения в ТС при выполнении миссии и анализирует выполнение миссии: $\mu A(\text{миссия выполнена}) \geq \alpha_V^*$. Узел рассчитывает последовательности состояний $y(t)$, реализует задачи миссии и выполняет расчет оценки удельной ценности и эффективности [16] отработки фактических ситуаций $V(t)$.

Такой подход к моделированию поведения узлов в сети на основе паттернов аналогичен подходу в прагматической эпистемологии, согласно которому знания рассматриваются как множество теорий или моделей, каждая из которых представляет собой описание поведения при решении определенного класса задач. Теории, описывающие класс явлений, могут перекрываться, и возникает проблема выбора способа решения задач в конкретной предметной области. Критерием выбора становится ожидаемая удельная ценность возможного результата [15].



Рис. 1. Схема системы управления узлом для двух тактов управления

Fig. 1. The node control system diagram for two control cycles

Процесс получения знания в предлагаемой схеме аналогичен процессу биологической эволюции. Следовательно, критерием истинности знания искусственной сущности является приспособленность знания для решения задач предметной области.

Модель данных

Реализация этого подхода связана с правильным соотношением вектора координат ситуации ($x_i, i = \overline{1, n}$) с тем или иным паттерном как по их количеству, так и по форме предъявления каждой координаты, а также с полной описания вектором ситуации (количество учитываемых параметров n). Связь конкретного класса и конкретного паттерна устанавливается при работе с лидером – действительным носителем знания. Соответствующая методика изложена в [14]. Так как человек описывает ситуацию средствами естественного языка, координаты ситуационного вектора являются лингвистическими переменными. Они используются для преобразования количественных шкал, используемых в сенсорах, в лингвистические шкалы, используемые человеком для описания значений наблюдаемого параметра-понятия в виде термов.

База знаний по применению паттернов поведения

Пусть состояние проблемной ситуации описывается ситуационным вектором $x_i, i = \overline{1, n}$, каждая координата которого является лингвистической переменной x_i с множеством термов $A_i = \{a_i^k, k = \overline{1, K^i}\}$. Определен набор классов реализаций ситуационного вектора, при котором есть паттерн поведения с успешным разрешением любой ситуации класса (кластера). При этом значения удельной ценности по результату превышают некий пороговый уровень. То же самое относится и к удельной ценности по эффективности при реализации паттерна.

Пусть накоплено множество $d_j, j = \overline{1, p}$, паттернов, каждый из которых связан с множеством кластеров ситуаций, для разрешения которых он был выбран. В соответствии с подходом, описанным в [9, 12], можно составить матрицу соответствия кластеров ситуаций располагаемым множествам паттернов поведения (табл. 1).

Таблица 1

Соотнесение класса значений ситуационного вектора в паттерном поведении

Table 1

Correlation of the class of the situational vector values in pattern behavior

№ п/п	Координаты ситуационного вектора					Паттерн
	x_1	...	x_j	...	x_n	
1.1 ...	a_1^{11}	a_i^{i1}	a_n^{11} ...	d_1
1. K^1	$a_1^{1K_1}$		$a_i^{iK_i}$		$a_n^{1K_1}$	
...
$j.1$...	a_1^{j1}	a_i^{ji}	a_n^{j1} ...	d_j
$j.K^j$	$a_1^{jK_1}$		$a_i^{jK_i}$		$a_n^{jK_j}$	
...
$p.1$...	a_1^{p1}	a_i^{pi}	a_n^{p1} ...	d_p
$p.K^p$	$a_1^{pK_1}$		$a_i^{pK_i}$		$a_n^{pK_p}$	

Если сгруппировать строки матрицы по паттернам, где каждая из них представляет собой конкретный ситуационный кластер, при котором успешно реализован соответствующий паттерн, то в результате можно получить кластер типовой ситуации. Для него паттерн поведения по структуре будет один и тот же, отличаться будут значения параметров.

Как правило, микрочипы узлов имеют операционную систему, подобную Tiny Os. Анализ возможных вариантов построения базы знаний в виде паттернов и их моделирование в среде Tiny Os показали, что наиболее приемлем подход, предложенный в [9, 12, 17], поэтому дальнейшее изложение материала будет соответствовать работам [9, 12].

Пронумеруем строки блока паттерна d_j двумя индексами: первый индекс – номер паттерна (он же номер блока), второй индекс – номер ситуации, входящий в j -й кластер.

Такая матрица определяет систему логических высказываний следующего вида:

«Если ... То, Иначе ...».

Например, строка матрицы $j.1$ – это высказывание

Если $x_1 = a_1^{j1}$ и $x_2 = a_2^{j2}$ и ... и $x_n = a_n^{jn}$
То d_j (2)

Иначе выражение для следующей строки.

Упорядоченную таким образом систему логических высказываний будем называть нечеткой матрицей знаний, или базой паттернов.

Алгоритм вычисления функции принадлежности паттерна

Каждое логическое высказывание типа (1) – это нечеткое отношение соответствующих нечетких множеств. Например, для строки j это будет $a_1^{j1} \times a_2^{j2} \times \dots \times a_n^{jn}$.

Функция принадлежности нечеткого множества, образованного этим нечетким отношением, будет $\mu_{a_1}^{j1}(x_1) \wedge \dots \wedge \mu_{a_i}^{ji}(x_i) \wedge \dots \wedge \mu_{a_n}^{jn}(x_n)$.

В данном случае вычисление этого выражения осуществляется с помощью операции \min .

Блок логических высказываний, относящийся к паттерну d_j , представляет собой объединение соответствующих нечетких множеств, образованных строками выделенного блока. Тогда функция принадлежности этого объединения будет определяться следующим образом:

$$\mu_{d_j}(x_i, i = \overline{1, n}) = \bigvee_{j=1}^p \bigwedge_{i=1}^n \mu_i^{jk_j}(x_i), \text{ где } \bigvee \text{ означает операцию max.}$$

Формально представленный алгоритм определения функции принадлежности паттерна d_j можно записать в виде:

- фиксируем произвольную точку

$$x_i^*, i = \overline{1, n} \in \times_{i=1}^n U_i;$$

- для каждого блока матрицы знаний, соответствующего d_j , определим $\mu_{d_j}(x_i, i = \overline{1, n})$ в этой точке согласно таблице 2.

Заметим, что для фиксированной точки x_i^* , $i = \overline{1, n} \in \times_{i=1}^n U_i$, блок матрицы, представленный в таблице 2, является числовым, так как вместо каждого термина a_i^{js} из этого блока поставлено

значение его функции принадлежности $(a_i^{js})^*$, вычисленное в точке x_i^* . Операция $\min_i (a_i^{is})^*$ производится над числами, стоящими в строках $i = \overline{1, n}$, и в столбец \min заносится минимальное число в соответствующей строке. Операция $\max_{js} \min_i (a_i^{js})^*$ определяет в полученном массиве строчных минимумов $j_s \in \overline{1, K_j}$ наибольшее значение. Проводя такие вычисления для каждой точки универсального множества, получим требуемые значения функции принадлежности.

Алгоритм выбора паттерна при наблюдении ситуационного вектора с количественными характеристиками

Бортовая интеллектуальная система с помощью своих измерительных средств формирует вектор $x_i^*, i = \overline{1, n}$, с количественными характеристиками координат. В этой точке, полученной на момент t , требуется определить значение функции принадлежности паттерна $d_j(x_i, i = \overline{1, n})$, $j \in \overline{1, p}$.

В этом случае функция принадлежности рассчитывается для каждого блока. Наиболее предпочтительным для разрешения проблемной ситуации будет паттерн d_j^* , вычисленный по правилу $d^*(x_i^*, i = \overline{1, n}) = \max_{j \in \overline{1, p}} d_j(x_i, i = \overline{1, n})$.

Алгоритмическая оболочка оперативного вывода по паттерну

Описанный подход реализован в форме программного продукта, который позволяет

Таблица 2

Расчет значений функций принадлежности блоков

Table 2

Calculation of the values of the blocks membership functions

№ п/п	Координаты ситуационного вектора					min	max	d_j
	x_1	...	x_i	...	x_n			
...
j_1	$(a_1^{j1})^*$...	$(a_i^{ji})^*$...	$(a_n^{jn})^*$	$\min_i (a_i^{i1})^*$	$\max_{js} \min_i (a_i^{jss})^*$	μ_{d_j}
...		
j_s	$(a_1^{js})^*$...	$(a_i^{js})^*$...	$(a_n^{js})^*$	$\min_i (a_i^{is})^*$		
...		
j_{K_j}	$(a_1^{jK_j})^*$...	$(a_i^{jK_j})^*$...	$(a_n^{jK_j})^*$	$\min_i (a_i^{iK_j})^*$		
...

смоделировать произвольную проблемную ситуацию (субситуацию), задавая ее ситуационным вектором с полным описанием его координат (для каждой координаты вводятся набор термов, относящаяся к каждому терму функция принадлежности (выбираемая из заданного списка) и универсальное множество, на котором она определена) и матрицей знаний.

Ниже в качестве примера рассматриваются алгоритмы некоторых паттернов.

Паттерн определения местоположения узла относительно пункта управления БСС

Одной из основных в интеллектуальной БСС является задача конфигурирования сети и определения зоны ответственности для каждого узла, для чего требуется знать координаты размещения узлов. Они рассчитываются относительно координат пункта управления БСС по степени удаленности от него.

Наиболее простым методом решения этой задачи является индикация уровня принятого сигнала. Любой беспроводной канал по стандарту IEEE 802.15.4 имеет протокольную функцию оценки качества связи, действие которой сводится к определению мощности принятого сигнала. Можно вывести результат этого измерения, откалибровать его по известному расстоянию, оценить дальность до источника и определить координаты нахождения узла.

Измерение расстояния производится следующим образом. Приемник с логарифмической амплитудной характеристикой принимает сигналы, по которым встроенный индикатор $RSSI$ формирует 8-разрядный код $RSSI_{VAL}$. Этот код получается в результате усреднения по восьми периодам (128 мкс) принятого сигнала и снабжается битом состояния, указывающим, когда $RSSI_{VAL}$ является валидным (то есть приемник имел возможность принять, по крайней мере, восемь периодов). Мощность принятого сигнала P (дБм) вычисляется по формуле

$$P = RSSI_{VAL} + RSSI_{OFFSET},$$

где $RSSI_{OFFSET}$ – эмпирически подбираемая постоянная (порядка 45 дБм).

Поскольку в идеальных условиях мощность обратно пропорциональна квадрату расстояния, логарифм мощности пропорционален расстоянию с некоторым коэффициентом, который устанавливается также эмпирически. Данный подход реализован в микроконтроллерах ZigBee фирмы TI серии CC2431 [18].

Паттерн самоорганизации БСС

Для динамической БСС с ячеистой топологией сети необходимо использовать алгоритм адаптивной кластеризации LEACH. Протокол LEACH с контролем топологии и кластеризацией с низким потреблением предназначен для сетей с иерархической структурой. Топология протокола LEACH показана на рисунке 2.

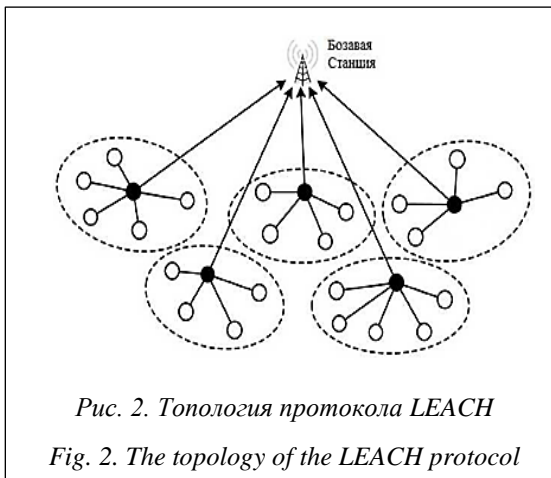


Рис. 2. Топология протокола LEACH

Fig. 2. The topology of the LEACH protocol

Узлы однородные, с ограниченным запасом энергии. В сети имеется базовая станция, и все узлы знают о ее физическом местоположении. Для предотвращения чрезмерного потребления энергии и придания БСС с изменяемой структурой свойств самоорганизации и адаптации под различные состояния сети по распределению энергии узлы динамически объединяются в кластеры, внутри которых происходит выбор головного узла [3, 4].

Головной узел выбирается для каждого кластера. Его функции – сбор данных с узлов (мотов) кластера, обработка и передача информации на шлюз или базовую станцию. Такое агрегирование данных в головных узлах значительно уменьшает энергопотребление в сети и увеличивает длительность жизненного цикла. Поэтому выбор головного кластера является основной задачей при разработке алгоритмов для иерархических БСС. Для этого работа LEACH разделена на раунды. Каждый из раундов состоит из фазы настройки и стационарной фазы. В фазе настройки осуществляются выбор головных узлов и объединение их с соседними устройствами. Решение о выборе принимается независимо всеми узлами в одно и то же время с помощью следующего алгоритма.

Каждому из n узлов присваивается случайное значение N . Значение порога $T(n)$ рассчитывается по формуле

$$T(n) = \frac{P}{1 - P * (r \bmod P^{-1})}, \forall n \in G; T(n) = 0, \forall n \notin G,$$

где r – номер текущего раунда; P – вероятность выбора узла головным; n – указатель на текущий узел; G – число узлов, которые не были головными в последних $1/P$ раундах [19].

Если $N < T(n)$, узел становится головным кластером. Такой алгоритм гарантирует, что каждый узел будет выбран головным узлом только один раз в течение $1/P$ раундов.

Узлы, которые были головными узлами в нулевом раунде, не могут быть таковыми снова в течение раундов $1/P$. После этого все определенные узлы могут становиться головным узлом.

После выбора головного узла другие сенсорные узлы определяют, в какой кластер они будут входить, на основе мощности получае-

мого сигнала RSS от головного узла. Когда все узлы организовались в кластеры, головной узел создает расписание передачи информации на основе метода TDMA, что гарантирует отсутствие коллизий при передаче сообщений.

Головной узел широкополосным способом рассылает расписание передачи и запрашивает своих членов кластера о передаче данных. Узлы передают данные в отведенные для этого интервалы TDMA. После получения сообщений от всех узлов головной узел формирует свои сообщения и передает их на шлюз или базовую станцию. Блок-схема протокола LEACH изображена на рисунке 3 [19].

В стационарной фазе раунда данные передаются базовой станцией. По сравнению с фазой настройки стационарная фаза более продолжительна для того, чтобы минимизировать расход энергии. Для этого используется протокол

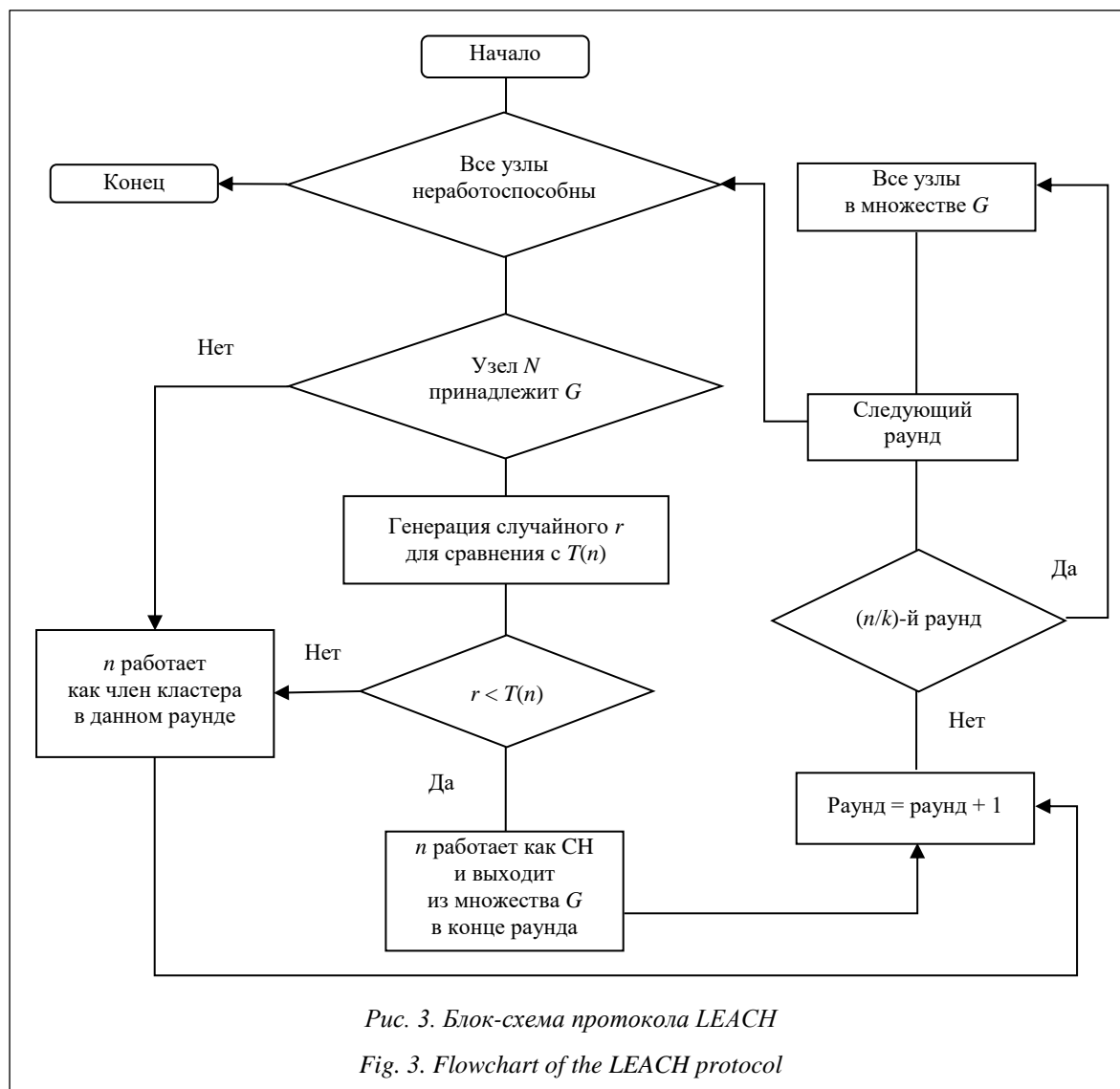
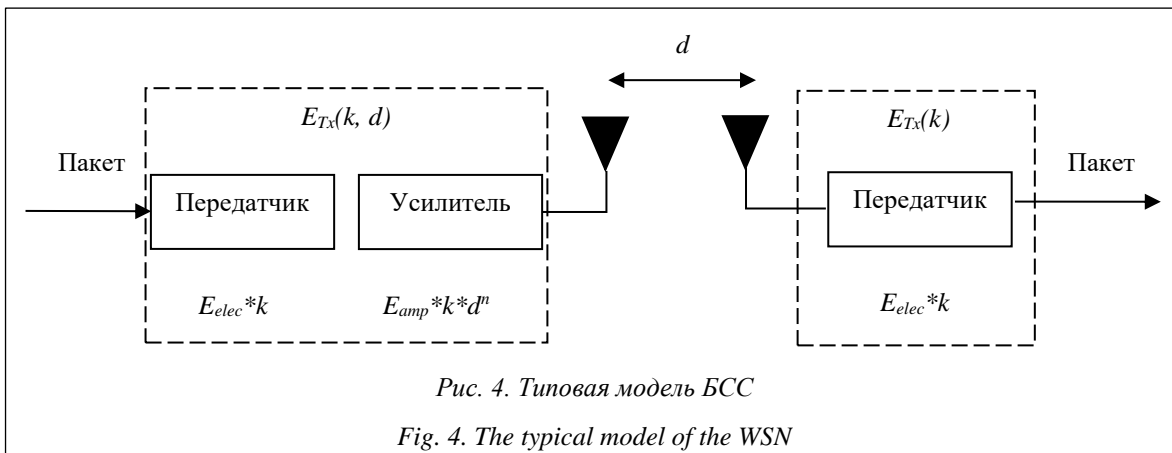


Рис. 3. Блок-схема протокола LEACH

Fig. 3. Flowchart of the LEACH protocol



MAC-уровня – CSMA, в котором каждый узел проверяет отсутствие передачи другими узлами до начала передачи на транспортном уровне. Критическим параметром сети является процент узлов, являющихся головным. В случае среднего и большого процентов и если расстояние до головного узла длиннее, чем между узлами и стоком, тратится больше энергии. Оптимальными считаются 5 % головных узлов от количества всех узлов сети. В таком случае протокол помогает достигнуть уменьшения потребления общей энергии сети в 7–8 раз в случае однонаправленной передачи, в 4–8 раз при передаче пакетов по нескольким узлам сразу.

Если требуется, то все узлы становятся неработоспособными (сеть переходит в спящий режим) благодаря последовательному разделению ролей головных узлов.

Общий расход энергии сети минимизируется с помощью разделения нагрузки между всеми узлами в разные моменты времени.

В качестве радиомодели используется типовая модель для БСС (рис. 4) [19].

Рассматриваемая радиомодель описывается следующими уравнениями:

$$E_{TX}(k, d) = E_{TX-elec} \cdot k + E_{amp} \cdot k \cdot d^2,$$

$$E_{RX}(k) = E_{TX-elec}(k) \cdot E_{RX-elec}(k),$$

где E_{TX} – энергия, затрачиваемая на передачу; E_{RX} – энергия, потребляемая при приеме; $E_{TX-elec}$, $E_{RX-elec}$ – энергии, необходимые для работы электронной схемы передатчика и приемника, соответственно; E_{amp} – энергия, необходимая для схемы усилителя; k – размер пакета; d – расстояние между передатчиком и приемником [19].

После установки, определения положения и кластеризации с определением головных узлов каждый узел БСС может приступать к монито-

рингу обстановки в зоне ответственности и выполнению задач.

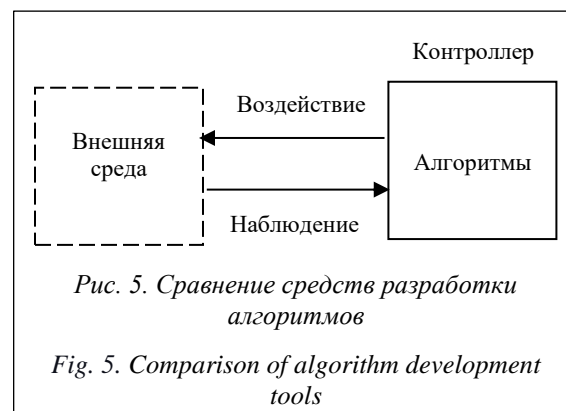
Паттерн построения рабочего канала

Для проектирования алгоритмов на уровне специализированной среды необходим канал передачи данных между контроллером и средой проектирования. Построение рабочего канала для наблюдения и обработки сигналов в реальном времени на предельной частоте преобразования АЦП показано на рисунке 5.

Использование двух буферов для накопления и передачи данных позволяет организовать без разрывов сбор, обработку и визуализацию данных (для избежания потери данных скорость последующего процесса должна быть не ниже скорости предыдущего).

Примером применения такой организации может быть модуль E14-440 [2] для многоканального ввода, вывода и обработки аналоговой и цифровой информации, подключаемый к компьютеру через шину USB.

Входные данные сначала заносятся в первую половину FIFO-буфера АЦП. После ее заполнения данные начинают передаваться



в регистр процессора, в то же время не прекращается сбор данных во вторую половинку FIFO-буфера. После накопления данных во второй половине FIFO-буфера опять начинается передача данных в регистр процессора и параллельно продолжается сбор данных уже в первую половину.

Паттерн определения направления движения

Далее приведена программа паттерна определения направления движения, написанная в среде разработки электронных устройств Arduino [20]:

```
bool flag=0;
bool flag2=0;
bool state=0;
bool state2=0;
int val=0;
void setup() {
  Serial.begin(9600); //задаём скорость передачи данных
  pinMode(A0,OUTPUT); //устанавливаем режим работы A0 на выход
  pinMode(A2,OUTPUT); //устанавливаем режим работы A2 на выход
  pinMode(A1,OUTPUT); //устанавливаем режим работы A1 на выход
}
void loop() {
  if(!digitalRead(9)&&!flag) { //определяем значение со входа 9
    delay(50);
    flag=1;
    if(flag2==0) {
      state=1;
    }
  }
  if(!digitalRead(12)&&!flag2) { //определяем значение со входа 12
    delay(50);
    flag2=1;
    if(flag==0) {
      state2=1;
    }
  }
  if(digitalRead(12)&&digitalRead(9)) { //определяем, является ли значение на обоих входах High
    if(!flag || !flag2) { // определяем, являются ли значения переменных flag и flag2 обратными
      state=0;
      state2=0;
      flag=0;
      flag2=0;
    }
    if(flag&&state2) { // определяем значения переменных flag и state2
      val++;
      digitalWrite(A0,HIGH);
```

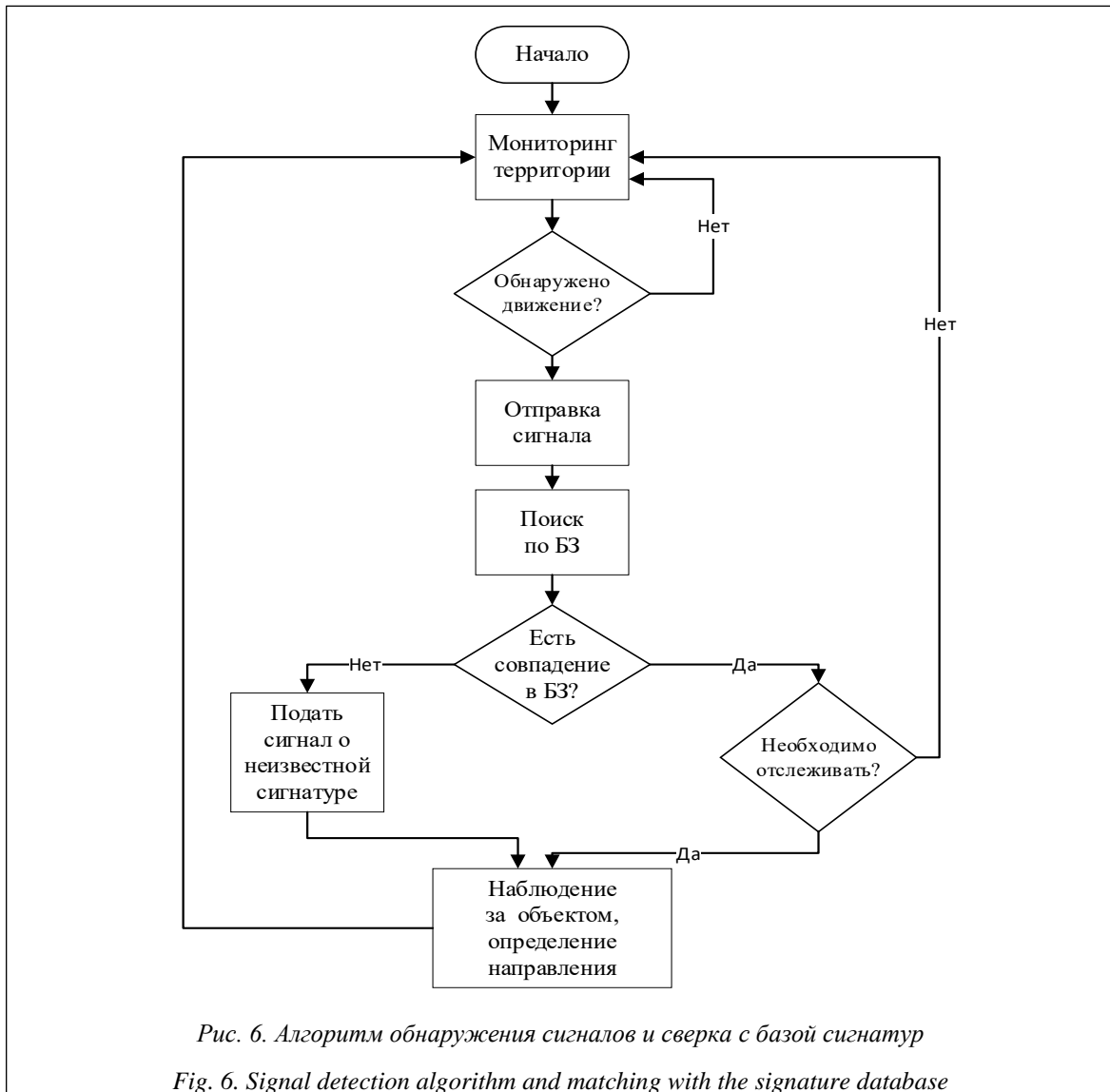
```
      digitalWrite(A2,LOW);
      delay(100);
      digitalWrite(A0,LOW);
      digitalWrite(A2,LOW);
      flag=0;
      flag2=0;
      state2=0;
    }
    if(flag2&&state) { // определяем значения переменных flag2 и state
      val--;
      digitalWrite(A2,HIGH);
      digitalWrite(A0,LOW);
      delay(100);
      digitalWrite(A2,LOW);
      digitalWrite(A0,LOW);
      flag=0;
      flag2=0;
      state=0;
    }
  }
  if(val<=0) {
    digitalWrite(A1,LOW);
  }
  else {
    digitalWrite(A1,HIGH);
  }
  Serial.println(val);
}
```

Когда на одном из датчиков появляется низкий уровень (датчики инверсные), переменной flag присваивается значение 1. В то же время и переменной state присваивается значение 1. Тем самым программа блокирует изменение второй переменной flag. Когда сигнала на датчиках нет, но есть изменения в значениях state и state2, программа совершает определенные действия, в данном случае мигает светодиодами (когда значение val > 0, включает светодиод), прибавляет/отнимает единицу к переменной val и выводит ее значение на монитор. Чтобы значение val изменилось, должны сработать оба датчика. Программа игнорирует срабатывание только одного датчика и сбрасывает переменные в ноль [21]. При обнаружении сигналов происходит сверка с базой сигнатур по алгоритму, представленному на рисунке 6.

Паттерн отправки сообщения о движении

Программа паттерна отправки сообщения о движении, разработанная в среде Arduino, имеет следующий вид:

```
int pirPin = 7; // Подключаем к входу 7
int minSecsBetweenMessage = 60; // Устанавливаем интервал сообщений
```



```

long lastSend = -minSecsBetweenMessage * 1000; //Отслеживаем, когда было отправлено последнее сообщение
void setup() {
  pinMode(pirPin, INPUT); //Устанавливаем на вход 7 режим вход
  Serial.begin(9600); //Задаём скорость передачи данных
}
void loop() {
  long now = millis(); //Получаем число миллисекунд с Arduino
  if (digitalRead(pirPin) == HIGH) { //Обнаруживаем движение и отправляем сообщение «Движение»
    if (now > (lastSend + minSecsBetweenMessage * 1000)) {
      Serial.println("MOVEMENT");
      lastSend = now;
    }
  } else { //т.к. прошло мало времени с отправки

```

```

    последнего сообщения, отправляем сообщение «Слишком рано»
    Serial.println("Too soon");
  }
}
delay(500);
}

```

Пример использования программной реализации алгоритма вывода по паттерну

Решается проблема тактики поведения узла – автономного необитаемого аппарата (АНПА) на основании приведенных далее данных.

Входная информация:

– ситуационный вектор $x_i, i = \overline{1,10}$ (Дальность до цели, Видимость, Глубина, Скорость, Запас хода, Количество боеприпасов, Состоя-

Таблица 3

Термы значений лингвистических переменных ситуационного вектора

Table 3

Terms of values of the situational vector linguistic variables

№ п/п	Координаты ситуационного вектора	Термы
1	Дальность до цели	Очень малая, Малая, Средняя, Большая, Очень большая
2	Видимость	Нулевая, Малая, Средняя, Большая, Отличная
3	Глубина	Минимальная, Малая, Средняя, Большая, Максимальная
4	Скорость	Минимальная, Малая, Средняя, Большая, Максимальная
5	Запас хода	Почти нет, Малый, Средний, Достаточный, Максимальный
6	Количество боеприпасов	Почти нет, Мало, Среднее, Достаточное, Полный боезапас
7	Состояние АНПА	Авария, Пока движется, Без крит. повреждений, Хорошее, Очень хорошее
8	Видимость противником	Оч. низкая, Низкое, Среднее, Высокое, Оч. хорошее
9	Метеорологические условия	Оч. плохие, Плохие, Нормальные, Хорошие, Отличные
10	Возможность атаки	Почти нет, Слабая, Средняя, Высокая, Оч. высокая

ние АНПА, Видимость противником, Метеорологические условия, Возможность атаки);

– термы (для каждой координаты ситуационного вектора определены термы, приведенные в таблице 3);

– функции принадлежности, в качестве которых выбраны трапециевидные функции принадлежности координат ситуационного вектора;

– паттерны: Залечь на дно, Вернуться на базу, Продолжать, но быть готовым вернуться, Продолжать, соблюдая осторожность, Продолжать выполнять задание в штатном режиме.

Заключение

На основе подходов нечетких рассуждений представлен алгоритм оперативного вывода по выбору паттерна поведения в ситуации, соответствующей типовой. Он ориентирован как на

автономное использование, так и на встраивание его в бортовую интеллектуальную систему. В статье рассмотрены принципы работы БСС, определены модели индивидуального поведения автономного узла БСС. Предложены алгоритмы паттернов автономного интеллектуального устройства в составе БСС для решения следующих задач: динамическое определение структуры сети в зависимости от ее текущего состояния; оперативное определение текущих координат узлов; определение местоположения узлов кластера с учетом топологии сети; установка узла в режим обнаружения сигнатур; определение при появлении сигнатур их типа путем сверки с базой паттернов, определение направления движения и местоположения; принятие решения о необходимости отслеживания или игнорирования сигнатуры; связь с другими узлами для выработки согласованных действий.

Литература

1. Akyildiz I.F., Su W., Sankarasubramaniam Y., Cayirci E. Wireless sensor networks: a survey. Computer Networks, vol. 37, no. 4, pp. 393–422. DOI: 10.1016/S1389-1286(01)00302-4.
2. Городецкий В.И., Самойлов В.В., Троцкий Д.В. Базовая онтология коллективного поведения автономных агентов и ее расширения // Изв. РАН. ТИСУ. 2015. № 5. С. 102–121. DOI: 10.7868/S0002338815030087.
3. Галкин П.В. Алгоритм управления и оптимизации информационных потоков в беспроводной сенсорной сети // ЕЕJET. 2014. Т. 6. № 3. С. 53–63. DOI: 10.15587/1729-4061.2014.30419.
4. Галкин П.В. Модель и алгоритм управления информационными потоками в беспроводной сенсорной сети. Наукові праці ДонНТУ. 2015. № 1. С. 17–32.
5. Муксунов Т.Р. Программно-аппаратные методы обработки сигналов для повышения эффективности радиоэлектронных датчиков: автореф. дис. Томск: ТУСУР РАН, 2015. 22 с.

6. Тараканов Е.В. Экспериментальные исследования протокола передачи данных с приоритетами в беспроводной сенсорной сети в системе Tossim // Изв. Томского политехнического университета. 2012. Т. 321. № 5. С. 223–227.
7. Цветков А.А. Алгоритмы эффективного кодирования информационных источников в сенсорных сетях // 64 науч. конф. студентов и аспирантов БГУ: сб. работ. 2007. Ч. 1. С. 58–60.
8. Баханов Л.Е., Давыдов А.Н., Корниенко В.Н., Слатин В.В., Федосеев Е.П., Федосов Е.А., Федун Б.Е. Системы управления вооружением истребителей: основы интеллекта многофункционального самолета. М.: Машиностроение, 2005. 399 с.
9. Fedunov В.Е. Problems of development of on-board real-time advisory expert systems for anthropocentric objects. J. Comput. Syst. Sci. Int., 1996, vol. 35, no. 5, pp. 816–827.
10. Fedunov В.Е. Constructive semantics of anthropocentric systems for development and analysis of specifications for onboard intelligent systems. J. Comput. Syst. Sci. Int., 1998, vol. 37, no. 5, pp. 796–806.
11. Федун Б.Е., Прохоров М.Д. Вывод по прецеденту в базах знаний бортовых интеллектуальных систем // Искусственный интеллект и принятие решений. 2010. № 3. С. 63–72.
12. Fedunov В.Е. Inference technique in knowledge bases of onboard operative advising expert systems. J. Comput. Syst. Sci. Int., 2002, vol. 41, no. 4, pp. 545–555.
13. Виноградов Г.П., Прохоров А.А., Шепелев Г.А. Паттерны в системах управления автономными системами // Информационные и математические технологии в науке и управлении. 2020. № 1. С. 40–54. DOI: 10.38028/ESI.2020.17.1.003.
14. Соснин П.И. Моделирование рассуждений. Ульяновск: Изд-во УлГТУ, 2000. 74 с.
15. Виноградов Г.П. Паттерны в автономных системах при выполнении миссий // НСМВИТ: тр. VIII междунар. науч.-практич. конф. 2020. Т. 2. С. 69–82.
16. Виноградов Г.П., Кузнецов В.Н. Моделирование поведения агента с учетом субъективных представлений о ситуации выбора // Искусственный интеллект и принятие решений. 2011. № 3. С. 58–72.
17. Федун Б.Е. Бортовые интеллектуальные системы тактического уровня для антропоцентрических объектов. М.: Де Либри, 2018. 246 с.
18. Еркин А. Расширение возможностей беспроводных сетей ZigBee: измерение координат узлов // Беспроводные технологии. 2011. № 1. С. 12–14.
19. Сафронов П.С. Анализ характеристик протокола функционирования беспроводных сенсорных сетей LEACH: автореф. дис. Челябинск: ЮУрГУ, 2017. 95 с.
20. Программирование Ардуино. URL: <http://arduino.ru/Reference> (дата обращения: 27.12.2020).
21. Обработка данных с датчика в режиме квазиреального времени. URL: <https://lhs-blog.info/programming/dlang/obrabotka-dannyih-s-datchika-v-rezhime-kvazirealnogo-vremeni/> (дата обращения: 29.12.2020).
22. Виноградов Г.П. Системы управления на основе паттернов // Software Journal: Theory and Applications, 2020, no. 2, pp. 10–19. URL: <http://swsys-web.ru/en/pattern-control-system.html> (дата обращения: 29.12.2020).

Algorithms for smart node patterns in a wireless sensor network

Vinogradov G.P.^{1,2}, *Dr.Sc. (Engineering), Professor, Head of Laboratory, wgp272ng@mail.ru*
Emtsev A.S.², *Engineer-Programmer, emcev@cps.tver.ru*
Fedotov I.S.², *Engineer-Programmer, twlucifer@gmail.com*

¹Tver State Technical University, Tver, 170026, Russian Federation

²R&D Institute Centerprogramsystem, Tver, 170024, Russian Federation

Abstract. One of the trends in the development of modern weapons is the "linking" of individual samples with a certain degree of autonomy into a complex using, as a rule, wireless sensor networks.

The scope of such complexes is uncertain and poorly formalized environments. It is possible to achieve their desired efficiency mainly by improving the intellectual component of the management system of the complex as a whole and the individual node in particular. However, it should be noted that the vast majority of research in this area remains only at the theoretical level. There is a gap between the primitive models of artificial entities' behavior, for example, in swarm robotics, models of their interaction, and expectations from the practice. The situation is aggravated by the secrecy requirements, miniaturization, and low energy consumption.

This paper presents an approach to the development of software for intelligent control systems for a separate network node that has a given degree of autonomy when performing problems. To offer relatively simple algorithms in the conditions of restrictions on power consumption and speed for giving the network node the properties of intelligent behavior, to provide the ability to study the situation and decide both independently, considering the data received from other network devices, and as part of a group.

There are methods of the fuzzy sets theory, the theory of building fuzzy models and networks, and approaches and algorithms for building on-board intelligent control systems in this work.

The work shows that the class identification of typical situations and successful action methods in actual conditions contribute to the development of the required algorithms. On this basis, it becomes possible to develop formal behavior models (patterns) for implementation in the node management system.

The authors propose a two-level structure of an intelligent network management system. The upper level, implemented by the operator, corresponds to such properties as survival, safety, the fulfillment of mission obligations, accumulation, and adjustment of the knowledge base as effective behavior patterns. The object of control for it is the network, considered as a functional system. It calculates the current indicators of specific value based on the results and effectiveness at time t , calculates and implements the method of action (behavior) at time t according to a given behavior pattern, and monitors the results of implementing the behavior pattern.

Keywords: development model, wireless network, sensor, artificial intelligence.

References

1. Akyildiz I.F., Su W., Sankarasubramanian Y., Cayirci E. Wireless sensor networks: a survey. *Computer Networks*, vol. 37, no. 4, pp. 393–422. DOI: 10.1016/S1389-1286(01)00302-4.
2. Gorodetsky V.I., Samoylov V.V., Trotskii D.V. The reference ontology of collective behavior of autonomous agents and its extensions. *J. Comput. Syst. Sci. Int.*, 2015, vol. 54, no. 5, pp. 765–782. DOI: 10.7868/S0002338815030087.
3. Galkin P.V. An algorithm for operating and optimizing information flows in wireless sensor networks. *EEJET*, 2014, vol. 6, no. 3, pp. 53–63. DOI: 10.15587/1729-4061.2014.30419 (in Russ.).
4. Galkin P.V. Model and algorithm information management in wireless sensor network. *Naukovi Praci DonNTU*, 2015, no. 1, pp. 17–32 (in Russ.).
5. Muksunov T.R. Hardware and software methods for signal processing to improve the efficiency of electronic sensors. *Ph.D. Thesis*, Tomsk, 2015, 22 p. (in Russ.).
6. Tarakanov E.V. Experimental researches of the priority data transmission protocol in the wireless sensor network in the Tossim system. *Bull. of the Tomsk Polytechnic University*, 2012, vol. 321, no. 5, pp. 223–227 (in Russ.).
7. Cvetkov A.A. Algorithms for efficient coding of information sources in sensor networks. *Proc. 64 Sci. Conf. BSU Students and Postgraduates*, 2007, no. 1, pp. 58–60 (in Russ.).
8. Bahanov L.E., Davydov A.N., Kornienko V.N., Slatin V.V., Fedoseev E.P., Fedosov E.A., Fedunov B.E. *Fighter Weapon Control Systems: The Basics of Multi-function Aircraft Intelligence*. Moscow, 2005, 399 p. (in Russ.).
9. Fedunov B.E. Problems of development of on-board real-time advisory expert systems for anthropocentric objects. *J. Comput. Syst. Sci. Int.*, 1996, vol. 35, no. 5, pp. 816–827.
10. Fedunov B.E. Constructive semantics of anthropocentric systems for development and analysis of specifications for onboard intelligent systems. *J. Comput. Syst. Sci. Int.*, 1998, vol. 37, no. 5, pp. 796–806.
11. Fedunov B.E., Prokhorov M.D. Case-based reasoning conclusion in the knowledge bases of onboard intelligent systems. *Artificial Intelligence and Decision Making*, 2010, no. 3, pp. 63–72 (in Russ.).
12. Fedunov B.E. Inference technique in knowledge bases of onboard operative advising expert systems. *J. Comput. Syst. Sci. Int.*, 2002, vol. 41, no. 4, pp. 545–555.
13. Vinogradov G.P., Prokhorov A.A., Shepelev G.A. Patterns in control systems by autonomous systems. *Information and Mathematical Technologies in Science and Management*, 2020, no. 1, pp. 40–54. DOI: 10.38028/ESI.2020.17.1.003 (in Russ.).

14. Sosnin P.I. *Modeling of Reasoning*. Ulyanovsk, 2000, 74 p. (in Russ.).
15. Vinogradov G.P. Patterns in autonomous systems when completing missions. *Proc. VIII Intern. Conf. FSSCIT-2020*, 2020, vol. 2, pp. 69–82 (in Russ.).
16. Vinogradov G.P., Kuznetsov V.N. Modeling agent's behavior based subjective perceptions on the situation of choice. *Artificial Intelligence and Decision Making*, 2011, no. 3, pp. 58–72 (in Russ.).
17. Fedunov B.E. *Tactical Onboard Intelligent Systems for Anthropocentric Objects*. Moscow, 2018, 246 p. (in Russ.).
18. Erkin A. Expanding the capabilities of wireless ZigBee networks: measuring the coordinates of nodes. *Wireless Technologies*, 2011, no. 1, pp. 12–14 (in Russ.).
19. Safronov P.S. Analysis of the characteristics of the protocol for functioning of LEACH wireless sensor networks. *Ph.D. Thesis*, Chelyabinsk, 2017 (in Russ.).
20. *Programmirovaniye Arduino*. Available at: <http://arduino.ru/Reference> (accessed December 27, 2020) (in Russ.).
21. *Data Processing from the Sensor in the Quasi-Real Time Mode*. Available at: <https://lhs-blog.info/programming/dlang/obrabotka-dannyih-s-datchika-v-rezhime-kvazirealnogo-vremeni/> (accessed December 29, 2020) (in Russ.).
22. Vinogradov G.P. Pattern control system. *Software Journal: Theory and Applications*, 2020, no. 2, pp. 10–19. Available at: <http://swsys-web.ru/en/pattern-control-system.html> (accessed December 29, 2020).

Для цитирования

Виноградов Г.П., Емцев А.С., Федотов И.С. Алгоритмы паттернов интеллектуального узла в составе беспроводной сенсорной сети // Программные продукты и системы. 2021. Т. 34. № 1. С. 033–046. DOI: 10.15827/0236-235X.133.033-046.

For citation

Vinogradov G.P., Emchev A.S., Fedotov I.S. Algorithms for smart node patterns in a wireless sensor network. *Software & Systems*, 2021, vol. 34, no. 1, pp. 033–046 (in Russ.). DOI: 10.15827/0236-235X.133.033-046.