

СИНТЕЗ НЕЙРО-НЕЧЕТКИХ СЕТЕЙ С РАНЖИРОВАНИЕМ И СПЕЦИФИЧЕСКИМ КОДИРОВАНИЕМ ПРИЗНАКОВ ДЛЯ ДИАГНОСТИКИ И АВТОМАТИЧЕСКОЙ КЛАССИФИКАЦИИ ПО ПРЕЦЕДЕНТАМ

Решена задача автоматизации синтеза нейро-нечетких сетей для диагностирования и автоматической классификации по признакам. Предложен метод синтеза нейро-нечетких моделей по прецедентам, который определяет оценки взаимосвязи входных признаков, выделяет взаимосвязанные признаки, оценивает влияние признаков на выходной признак, а также выявляет взаимосвязанные термы, включает в модель наиболее важные признаки и термы, устраняет дуближ термов и признаков, использует различные варианты кодирования сигналов, а также переупорядочивает признаки, обеспечивая группировку признаков, формирует правила для прецедентов ранее не встречавшихся классов или существенно отличающихся от имеющихся правил своего класса, а уже имеющиеся правила корректирует на основе поступающих прецедентов с учетом числа ранее рассмотренных наблюдений. Предложенный метод позволяет существенно ускорить синтез нейро-нечетких моделей, обеспечивая приемлемую точность и более высокий уровень обобщения данных, снизить сложность и избыточность, а также повысить интерпретабельность нейромодели. Проведены эксперименты по решению практических задач диагностирования и автоматической классификации, подтвердившие работоспособность и применимость предложенного метода. Получена зависимость ошибки модели, синтезированной предложенным методом, от заданной разрядности признаков. Использование полученной зависимости позволяет на практике более рационально выбирать значение числа интервалов разбиения диапазонов значений признаков, обеспечивая приемлемую точность нейромодели.

Ключевые слова: выборка, нейро-нечеткая сеть, нечеткий вывод, обучение по прецедентам, диагностирование.

НОМЕНКЛАТУРА

ЭВМ – электронная вычислительная машина;
 δ_j – длина интервала, на которые разбивается диапазон значений j -го признака;
 μ_q – принадлежность к q -му классу;
 $\mu_{q,k}$ – принадлежность экземпляра к k -му эталону (правилу) q -го класса;
 $\mu_{q,k,i}$ – принадлежность экземпляра к k -му эталону (правилу) q -го класса по i -му признаку;
 $\mu_{q,k,i,l}$ – принадлежность экземпляра к k -му эталону (правилу) q -го класса по l -му интервалу i -го признака;
 $\mu_{i,l}(x^s)$ – функции принадлежности к l -му терму i -го признака;
 $\omega_{q,k}$ – нормированный вес k -го правила q -го класса;
 a, b, c, d – параметры функции принадлежности;
 C – набор правил-эталонов;
 $C^{(q,k)}$ – k -й эталон (правило) q -го класса;
 \bar{E} – ошибка модели;
 E – ошибка сети для задач с дискретным выходом;
 $E_{об.}$ – ошибка модели для обучающей выборки;
 $E_{тест.}$ – ошибка модели для тестовой выборки;
 f – критерий качества модели;
 $F()$ – структура нейромодели;
 $I^*(q,i,l)$ – значимость l -го интервала i -го признака относительно q -го класса;
 i, j – номера признаков;
 I_G – коэффициент обобщения нейромодели;

I_j – информативность j -го признака;
 $I_{i,l}$ – информативность l -го интервала i -го признака;
 I_w – число настраиваемых параметров нейромодели;
 k – номер правила (эталона);
 K – число классов;
 l – номер разряда (интервала);
 L – размерность кодирующего бинарного вектора – элемента сенсорной матрицы;
 N – число входных признаков;
 N' – число входных признаков, после отбора признаков;
 opt – условное обозначение оптимума;
 q – номер класса;
 Q^q – число правил (эталонов) q -го класса в наборе C ;
 $R(x^s, C^{(q,k)})$ – расстояние от экземпляра x^s до k -го эталона q -го класса $C^{(q,k)}$;
 $r_{i,j}$ – мера взаимосвязи входных признаков x_i и x_j ;
 $r_{i,y}$ – мера важности для каждого входного признака относительно выходного признака;
 S – объем выборки;
 $S_{об.}$ – число экземпляров в обучающей выборке;
 $S_{тест.}$ – число экземпляров в тестовой выборке;
 $t_{об.}$ – время, затраченное на обучение нейро-нечеткой модели;
 w – параметры нейромодели;
 $w^{(q,k)}$ – вес k -го правила q -го класса;
 x – набор входных векторов прецедентов;
 X – набор прецедентов;
 x_j – j -й входной признак;
 x^s – набор входных признаков s -го прецедента;
 x_j^s – значение j -го входного s -го прецедента;
 y – выходной признак;
 x_j^{\min}, x_j^{\max} – минимальное и максимальное значения признака x_j ;

$x_{j,l}^s$ – значение l -го разряда j -го признака s -го экземпляра в специфическом кодовом представлении;
 y^s – значение выходного признака для s -го прецедента.

ВВЕДЕНИЕ

Для обеспечения и поддержания высокого уровня качества, надежности и долговечности выпускаемых и эксплуатируемых изделий промышленного производства необходимо своевременно осуществлять их диагностирование [1].

Помимо методов физических измерений [1, 2], специфических для каждой конкретной задачи, ключевым элементом процесса диагностирования является диагностическая модель, построение которой требует либо полного знания физики диагностируемого объекта, либо привлечение человека-эксперта, либо использование методов машинного обучения [3].

Построение физических моделей [2], как правило, возможно только для хорошо изученных объектов диагностирования и специфически зависит от конкретной задачи. Такие методы непригодны для более широко встречающихся случаев, когда физика диагностируемого объекта является не полностью изученной, а сам объект подвергается воздействию внешних факторов [3].

Очевидными недостатками экспертных методов [4] является их чрезвычайная зависимость от субъективных знаний эксперта, его компетентности и ответственности, сложность извлечения и формализации знаний, а также сложность модификации и адаптации полученной модели принятия решений с учетом возможных изменений диагностируемого объекта, внешней среды.

Методы машинного обучения [5, 6], в отличие от методов построения физических моделей, не требуют знания физики диагностируемого объекта, а, в отличие от экспертных методов [4], не требуют привлечения человека-эксперта и способны автоматически извлекать знания из данных в процессе построения модели [3]. Поэтому данные методы целесообразно выбрать в качестве базиса для построения диагностических моделей.

Среди методов машинного обучения [5, 6] широкое распространение на практике в последнее время получили нейро-нечеткие сети [4, 7], которые способны обучаясь по прецедентам извлекать знания из данных, а также могут интегрировать в свою структуру экспертные знания и являются преобразуемыми в понятные для человека продукционные правила вида «если, то» [4].

Однако большинство известных методов построения нейро-нечетких моделей [4, 7] предполагают вовлечение пользователя в процесс создания сети и решение задач многомерной нелинейной оптимизации, характеризующихся неопределенностью выбора начальной точки поиска. Полученные диагностические модели в результате оказываются избыточными [3].

Объектом исследования данной работы является процесс построения диагностических моделей по прецедентам.

Предметом исследования данной работы являлись методы построения нейро-нечетких сетей для решения задач диагностирования.

Целью работы являлось создание метода синтеза диагностических моделей на основе нейро-нечетких сетей, свободного от указанных выше недостатков.

1 ПОСТАНОВКА ЗАДАЧИ

Пусть мы имеем исходную выборку $X = \langle x, y \rangle$ – набор S прецедентов о зависимости $y(x)$, $x = \{x^s\}$, $y = \{y^s\}$, $s = 1, 2, \dots, S$, характеризующихся набором N входных признаков $\{x_j\}$, $j = 1, 2, \dots, N$, и выходным признаком y . Каждый s -й прецедент представим как $\langle x^s, y^s \rangle$, $x^s = \{x_j^s\}$, $y^s \in \{1, 2, \dots, K\}$, $K > 1$.

Тогда задача синтеза нейромодели зависимости $y(x)$ будет заключаться в определении такой структуры $F()$ и значений параметров w нейромодели, при которых будет удовлетворен критерий качества модели $f(F)$, $w, \langle x, y \rangle \rightarrow opt$ [3]. Обычно критерий качества обучения нейромодели определяют как функцию ошибки модели:

$$\bar{E} = \frac{1}{2} \sum_{s=1}^S (y^s - F(w, x^s))^2 \rightarrow \min.$$

Для задач с дискретным выходом ошибку обученной сети можно характеризовать также формулой:

$$E = \frac{100\%}{S} \sum_{s=1}^S |y^s - F(w, x^s)| \rightarrow \min.$$

2 ЛИТЕРАТУРНЫЙ ОБЗОР

В настоящее время известно большое число различных архитектур нейро-нечетких сетей [4, 7]:

- нечеткие нейронные системы (fuzzy neural systems): в нейронных сетях применяются принципы нечеткой логики для ускорения процесса настройки или улучшения других параметров; нечеткая логика является лишь инструментом нейронных сетей и такая система не может быть интерпретирована в нечеткие правила, поскольку представляет собой «черный ящик»;

- конкурирующие нейро-нечеткие системы (concurrent neuro-fuzzy systems): нечеткая система и нейронная сеть работают над одной задачей, не влияя на параметры друг друга;

- параллельные нейро-нечеткие системы (cooperative neuro-fuzzy systems): настройка параметров выполняется с помощью нейронных сетей, после чего нечеткая система функционирует самостоятельно;

- интегрированные (гибридные) нейро-нечеткие системы (integrated neuro-fuzzy systems) – системы с тесным взаимодействием нечеткой логики и нейронных сетей. Данный тип нейро-нечетких сетей является наиболее подходящим для решения задач диагностирования.

Наиболее широко используемой на практике архитектурой интегрированных нейро-нечетких сетей являются сети Мамдани [8, 9], представляющие собой нейро-нечеткий классификатор. Однако методы их построения предполагают вовлечение пользователя в процесс создания сети (задание числа термов, видов их функций принадлежности, начальных значений параметров функций принадлежности термов, правил принятия решений), либо требуют решения задач кластер-анализа [10] набора прецедентов с последующей настройкой весов на основе градиентных методов оптимизации [11] с расчетом частных производных целевой функции на основе техники обратного распространения ошибки [4, 5].

Полученные диагностические модели в результате оказываются неоптимальными по структуре (зачастую избыточными), и, как следствие, оказываются менее точными и более сложными для восприятия и анализа человеком, что несколько осложняет их практическое применение [3].

Поэтому необходимо разработать метод, позволяющий получать диагностические модели, не требующие участия человека в процессе синтеза структуры и настройки параметров нейромодели, а также позволяющий снизить сложность получаемых моделей и повысить их интерпретируемость и обобщающие свойства.

3 МАТЕРИАЛЫ И МЕТОДЫ

Избыточность нейро-нечетких моделей, как правило, обусловлена тем, что набор используемых признаков является избыточным, а нечеткие термы формируются субъективно пользователем и могут излишне детализировать разбиение пространства признаков. Поэтому для снижения избыточности нейро-нечетких моделей предлагается в процессе синтеза сети определять оценки взаимосвязи входных признаков между собой, что позволит выделить взаимосвязанные признаки, оценивать влияние признаков на выходной признак (класс), а также выявлять взаимосвязанные термы, что позволит включать в модель наиболее важные признаки и термы, устранить дублирование термов и признаков путем оставления лишь части из них.

Для повышения вариативности (потенциала получения возможных решений) синтезируемых моделей предлагается использовать различные варианты кодирования сигналов, а также переупорядочивать признаки, обеспечивая тем самым группировку признаков по подобию с целью получения более легко воспринимаемых человеком диагностических моделей.

Для автоматизации формирования термов и выделения правил на основе заданного набора прецедентов предлагается реализовать процедуру адаптивного четкого кластер-анализа прецедентов, в процессе реализации которой формировать новые эталоны для прецедентов ранее не встречавшихся классов или существенно отличающихся от имеющихся эталонов своего класса, а уже имеющиеся эталоны корректировать на основе поступающих прецедентов с учетом числа ранее рассмотренных наблюдений.

На основе приведенных идей более формально изложим предлагаемый метод.

Этап инициализации. Задать обучающую выборку $\langle x, y \rangle$. Определить минимальное и максимальное значения признаков:

$$x_j^{\min} = \min_{s=1,2,\dots,S} \{x_j^s\}, x_j^{\max} = \max_{s=1,2,\dots,S} \{x_j^s\}.$$

Задать размерность кодирующего бинарного вектора – элемента сенсорной матрицы L . Очевидно, что должно быть $L > 1$.

Этап кодирования сигналов. Заменить значение каждого признака в выборке бинарным кодом. Для этого возможно использовать интервальное или колоночное кодирование.

Вначале на основе заданного L определим длину интервала каждого j -го признака:

$$\delta_j = \frac{x_j^{\max} - x_j^{\min}}{L}, j = 1, 2, \dots, N.$$

При интервальном кодировании диапазон значений каждого признака разбивается на заданное число интервалов равной длины, а значение признака, заменяется на бинарный вектор, элементы которого (разряды) соответствуют интервалам значений признака. В процессе кодирования определяется номер интервала, в который попало значение признака для конкретного экземпляра, после чего соответствующий элемент кодирующего вектора устанавливается равным единице, а остальные элементы – равными нулю:

$$x_{j,l}^s = \begin{cases} 0, l\delta_j < x_j^s - x_j^{\min}; \\ 1, (l-1)\delta_j \leq x_j^s - x_j^{\min} \leq l\delta_j; \\ 0, (l-1)\delta_j > x_j^s - x_j^{\min}, \end{cases}$$

$$j = 1, 2, \dots, N, l = 1, 2, \dots, L.$$

При колоночном кодировании диапазон значений каждого признака также разбивается на заданное число интервалов равной длины, значение признака, заменяется на бинарный вектор, элементы которого (разряды) соответствуют интервалам значений признака и определяется номер интервала, в который попало значение признака для конкретного экземпляра, однако все элементы кодирующего вектора, соответствующие данному интервалу и предыдущим ему интервалам устанавливаются равными единице, а остальные элементы, соответствующие интервалам больших значений признака – равными нулю:

$$x_{j,l}^s = \begin{cases} 1, l\delta_j \leq x_j^s - x_j^{\min}; \\ 0, l\delta_j > x_j^s - x_j^{\min}, \end{cases}$$

$$j = 1, 2, \dots, N, l = 1, 2, \dots, L.$$

Этап выделения правил. Каждый экземпляр выборки x^s в выбранном кодовом представлении $x^s = \{x_{j,l}^s\}$ представить в виде модифицированного продукционного правила $\{x_{i,l}^s\} \rightarrow y^s$.

Этап обобщения правил. Выполнить динамическое формирование правил-эталонов кластеров.

Для этого в начале задать пустой набор правил-эталонов $C = \emptyset$.

Затем просматривая экземпляры выборки ($s=1,2,\dots,S$) для текущего s -го экземпляра определить, имеется ли эталон его класса в наборе эталонов.

Если эталон для его класса отсутствует в C , то принять $q=y^s$, добавить эталон и занести в него s -й экземпляр: $C = C \cup C^{(q,1)}$, $C^{(q,1)} = x^s$: $C_{j,l}^{(q,1)} = \{x_{j,l}^s\}, j=1, 2, \dots, N; l=1, 2, \dots, L, w^{(q,1)}=1$.

В противном случае, если эталон класса для текущего s -го экземпляра отсутствует в C , то определить расстояние от экземпляра x^s до всех эталонов $R(x^s, C^{(q,k)})$, $q=1, 2, \dots, Q$; $k=1, 2, \dots, Q^q$:

$$R(x^s, C^{(q,k)}) = \sqrt{\sum_{j=1}^N \sum_{l=1}^L (x_{j,l}^s - C_{j,l}^{(q,k)})^2}.$$

Если расстояние от экземпляра x^s до ближайшего к нему эталона чужого класса меньше, чем расстояние от экземпляра до ближайшего эталона своего класса, т.е.

$$\min_{\substack{y^s \neq q, \\ k=1,2,\dots,Q^q}} \{R(x^s, C^{(q,k)})\} < \min_{\substack{y^s = q, \\ k=1,2,\dots,Q^q}} \{R(x^s, C^{(q,k)})\},$$

тогда сформировать новый эталон и занести в него экземпляр: $q=y^s$, $k=Q^q+1$, $C = C \cup C^{(q,k)}$, $C^{(q,k)} = \{x_{j,l}^s\}$, $w^{(q,k)}=1$; в противном случае – новый эталон получить как среднее между эталоном и экземпляром $C^{(q,k)} = (w^{(q,k)}C^{(q,k)} + x^s) / (w^{(q,k)} + 1)$:

$$C_{j,l}^{(q,k)} = \frac{w^{(q,k)}C_{j,l}^{(q,k)} + x_{j,l}^s}{w^{(q,k)} + 1},$$

$j=1, 2, \dots, N$; $l=1, 2, \dots, L$,
после чего принять: $w^{(q,k)} = w^{(q,k)} + 1$.

Этап ранжирования кодовых представлений сигналов. При построении и анализе диагностических и распознающих моделей желательно выявить и понимать взаимосвязь признаков, а также зависимость принимаемых решений от описательных признаков. Для этого предлагается определить ряд показателей.

Определим меру взаимосвязи входных признаков относительно интервалов (сенсоров) одного уровня, используя формулу:

$$r_{i,j} = r_{j,i} = \frac{\sum_{s=1}^S \sum_{l=1}^L \sum_{t=1}^L \{1 | x_{j,l}^s = x_{i,t}^s\}}{SL^2}, \quad i, j = 1, 2, \dots, N.$$

Признаки будут тем сильнее взаимосвязанными, чем больше экземпляров попадает в и интервалы (сенсоры) одного уровня.

Определим меру важности для каждого входного признака относительно выходного признака как

$$r_{i,y} = \frac{\sum_{s=1}^S \sum_{l=1}^L \sum_{t=1}^K \{1 | x_{j,l}^s = 1, y^s = t\}}{SLK}, \quad i = 1, 2, \dots, N.$$

Входной и выходной признаки будут тем сильнее взаимосвязанными, чем больше экземпляров попадает в и интервалы (сенсоры) одного класса.

После определения мер близости признаков ранжируем и перенумеруем признаки. В качестве первого признака целесообразно выбрать признак с наибольшим значением $r_{i,y}$. Затем в новый набор включим последовательно признаки, наиболее тесно связанные с предыдущими признаками, в порядке убывания значений $r_{i,y}$.

Данный этап можно пропустить для стационарных сигналов, распределенных во времени.

Этап оценки значимости интервалов и признаков. Признаки и их интервалы значений могут иметь различную ценность при принятии решений.

При этом для каждого класса значимость конкретного интервала признака определяется тем, сколько экземпляров, попавших в этот интервал будет принадлежать к данному классу и тем, сколько экземпляров, попавших в этот интервал не будет принадлежать к данному классу. Для определения значимости интервалов относительно классов предлагается использовать показатель, выражаемый формулой:

$$I^*(q, i, l) = 2I(q, i, l) - \sum_{p=1}^K I(p, i, l),$$

где $I(q, i, l) = \sum_{s=1}^S \sum_{l=1}^L \sum_{t=1}^L \{x_{i,l}^s | y^s = q\}$, $q=1, 2, \dots, K$; $i=1, 2, \dots, N$;
 $l=1, 2, \dots, L$.

На основе введенных показателей можно определить – показатель информативности l -го интервала i -го признака:

$$I_{i,l} = \frac{1}{K} \sum_{l=1}^K I^*(q, i, l), \quad i = 1, 2, \dots, N; \quad l = 1, 2, \dots, L;$$

– показатель информативности j -го признака:

$$I_i = I_{i,l} = \frac{1}{L} \sum_{l=1}^L I_{i,l} = \frac{1}{KL} \sum_{l=1}^L \sum_{q=1}^K I^*(q, i, l), \quad i = 1, 2, \dots, N.$$

Этап удаления малозначимых интервалов значений признаков. Все интервалы признаков, у которых значение $I_{i,l} = 0$, можно считать малозначимыми и удалить из дальнейшего рассмотрения. Среди оставшихся интервалов можно рассматривать как малозначимые те, для которых

$$I_{i,l} \leq \left[\frac{ES}{L} \right],$$

где E – заданное заранее значение максимально допустимой ошибки.

Этап удаления малозначимых признаков. Если среди признаков имеется такой, что все его интервалы признака малозначимыми, то данный признак следует исключить из рассмотрения.

Этап формирования термов признаков. На основе выделенных интервалов значений признаков возможно определить функции принадлежности к термам признаков на основе трапецевидной функции ($a \leq b \leq c \leq d$):

$$\mu_{i,l}(x^s) = \begin{cases} 0, & x \leq a; \\ \frac{x-a}{b-a}, & a \leq x \leq b; \\ 1, & b \leq x \leq c; \\ \frac{d-x}{d-c}, & c \leq x \leq d; \\ 0, & d \leq x. \end{cases}$$

При інтервальному кодированні параметри функції приналежності к l -му терму i -го признака будуть ініціалізуватися по формулам:

$$a = (l-1)\delta + x_i^{\min}, b = (l-1)\delta + x_i^{\min}, c = l\delta + x_i^{\min}, \\ d = l\delta + x_i^{\min}.$$

При колоночному кодированні параметри функції приналежності к l -му терму i -го признака будуть ініціалізуватися по формулам:

$$a = x_i^{\min}, b = x_i^{\min}, c = l\delta + x_i^{\min}, d = l\delta + x_i^{\min}.$$

Етап формироваання системи нечеткого вывода. На данном етапе задаються принципи преобразования значений функций принадлежности к термам в принадлежности к кластерам и классам, а также способ дефаззификации результата.

Принадлежности к сформированным эталонам определим по формулам:

$$\mu^{q,k} = \max_{i=1,2,\dots,N} \{\mu_{q,k,i}\} \text{ либо } \mu^{q,k} = \max_{i=1,2,\dots,N} \left\{ \frac{I_i \mu_{q,k,i}}{\sum_{j=1}^N I_j} \right\},$$

где

$$\mu_{q,k,i} = \max_{l=1,2,\dots,L} \{\mu_{q,k,i,l}\} \text{ либо}$$

$$\mu_{q,k,i} = \max_{l=1,2,\dots,L} \left\{ \frac{I_{i,l} \mu_{q,k,i,l}}{\sum_{p=1}^L I_{i,p}} \right\},$$

$$\mu_{q,k,i,l} = \mu_{i,l}(x^s) C_{i,l}^{(q,k)},$$

$$q=1, 2, \dots, K, k=1, 2, \dots, Q^q, i=1, 2, \dots, N; l=1, 2, \dots, L.$$

Принадлежности к классам определим формулами:

$$\mu_q = \max_{k=1,2,\dots,Q^q} \{\omega_{q,k} \mu^{q,k}\},$$

$$\text{где } \omega_{q,k} = \frac{w^{(q,k)}}{\sum_{p=1}^{Q^q} w^{(q,p)}}.$$

Номер класса, соответственно, будет определяться по формуле:

$$y^s = \arg \max_{q=1,2,\dots,K} \{\mu_q\} \text{ или } y^s = \left[\frac{\sum_{q=1}^K q \mu_q}{\sum_{q=1}^K \mu_q} \right].$$

Этап синтеза нейро-нечеткой сети. На основе выделенных термов и определенных функций системы нечеткого вывода можно сформировать структуру нейро-нечеткой сети в соответствии со схемой, приведенной на рис. 1.

На входной (первый) слой сети поступают входные сигналы – значения признаков распознаваемого экземпляра, которые далее поступают на второй слой, выполняющий фаззификацию. Выходы нейронов второго слоя представляют собой значения функций принадлежности распознаваемого экземпляра к термам признаков. Нейроэлементы третьего слоя сети комбинируют принадлежности к термам в принадлежности к эталонам кластеров, на основе которых нейроны четвертого слоя определяют принадлежности к классам. Единственный нейрон пятого (выходного) слоя осуществляет дефаззификацию и выдает на выходе расчетное значение номера класса распознаваемого экземпляра.

Этап настройки весов. На основе обучающей выборки оценить ошибку для построенной сети. Если ошибка является приемлемой, то завершить синтез сети, в противном случае выполнить процедуру обучения посредством решения задачи многомерной оптимизации параметров нечетких термов второго слоя, используя метод обратного распространения ошибки [4, 11] или эволюционные алгоритмы [3, 5].

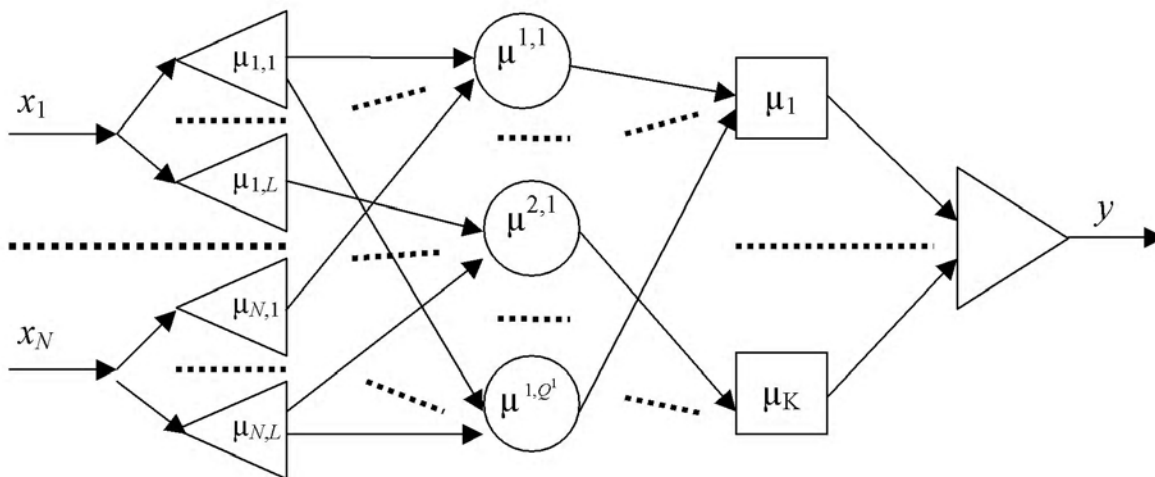


Рисунок 1 – Схема нейро-нечеткой сети

Если в результате обучения достигнут требуемый уровень точности (приемлемая ошибка), то завершить синтез сети, в противном случае – увеличить число интервалов L и перейти к этапу кодирования. Поскольку число интервалов L не должно превышать числа экземпляров в выборке S , то при получении на данном этапе $L \geq S$ следует прекратить дальнейший поиск и вернуть полученную модель с наилучшей точностью и наименьшим L .

4 ЭКСПЕРИМЕНТЫ

Для исследования практической применимости предложенного метода он был программно реализован как дополнение к Автоматизированной системе синтеза нейросетевых и нейро-нечетких моделей для неразрушающей диагностики и классификации образов по признакам [12].

Разработанное программное обеспечение использовалось для решения задач неразрушающего диагностирования и автоматической классификации, характеристики которых представлены в табл. 1.

Для каждой задачи на основе кластер анализа человеком выделялись нечеткие термы и правила, по которым определялась структура нейро-нечетких моделей, после чего на основе обратного распространения ошибки выполнялась настройка параметров моделей. Также для каждой задачи на основе выборки данных в автоматическом режиме на основе предложенного метода синтезировались нейро-нечеткие модели.

5 РЕЗУЛЬТАТЫ

Результаты проведенных вычислительных экспериментов представлены в табл. 2.

Здесь коэффициент обобщения определялся по формуле:

$$I_G = \frac{NS}{I_w}, I_w \geq 1.$$

Чем больше значение коэффициента обобщения, тем выше уровень обобщения обучающей выборки данных нейро-нечеткой моделью.

С целью изучения зависимости ошибки модели E от заданной разрядности признаков L для различных задач (выборок данных различной размерности и природы [3]) получен график, приведенный на рис. 2.

На рис. 3 представлен график усредненной зависимости ошибки модели E от доли отобранных признаков в первоначальном наборе признаков N'/N для различных задач (выборок данных различной размерности и природы [3]).

На рис. 4 представлен график усредненной зависимости коэффициента обобщения модели I_G от доли отобранных признаков в первоначальном наборе признаков N'/N для различных задач (выборок данных различной размерности и природы [3]).

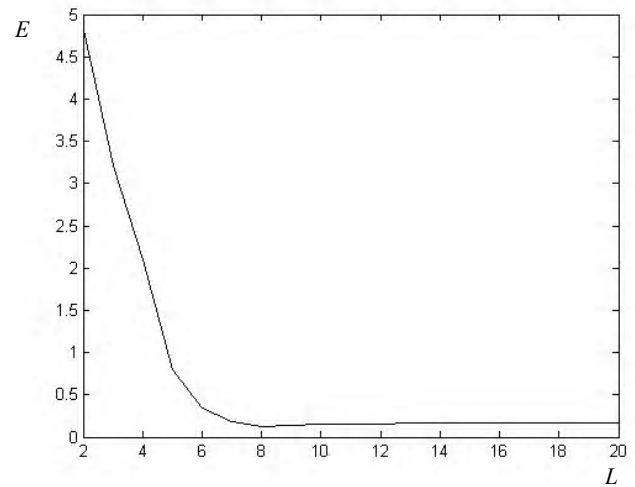


Рисунок 2 – График усредненной зависимости E от L

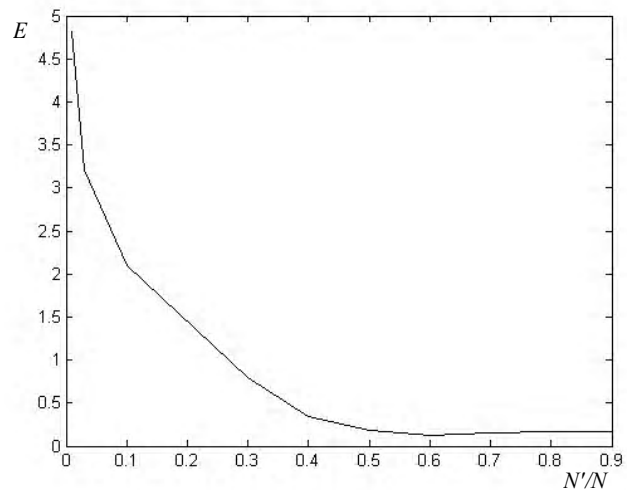


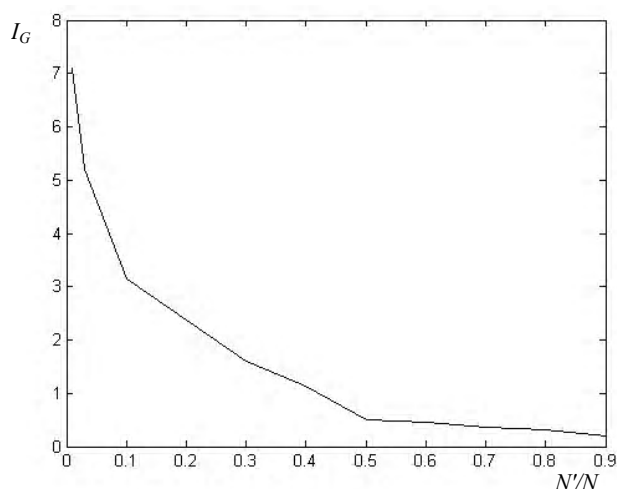
Рисунок 3 – График усредненной зависимости E от N'/N

Таблица 1 – Характеристики практических задач диагностирования и автоматической классификации

Название задачи	N	$S_{об.}$	$S_{тест.}$	K
Диагностирование лопаток газотурбинных авиадвигателей по спектрам свободных затухающих колебаний после ударного возбуждения [3]	100	16	16	2
Автоматическая классификация растительных объектов по коэффициентам спектральной яркости [13]	55	120	128	2
Диагностирование хронического обструктивного бронхита [14]	28	102	103	2

Таблица 2 – Результаты вычислительных экспериментов

Название задачи	Сеть Мамдани, заданная человеком и обученная методом обратного распространения ошибки				Сеть Мамдани, синтезированная на основе предложенного метода			
	$E_{об.}, \%$	$E_{тест.}, \%$	$t_{об.}, с$	I_G	E	$E_{тест.}$	$t_{об.}, с$	I_G
Диагностирование лопаток газотурбинных авиадвигателей [3]	0	0	13,92	0,8	0	0	12,75	1,33
Автоматическая классификация растительных объектов [13]	1,7	2,3	23,14	4,29	0	0,8	21,89	5
Диагностирование хронического обструктивного бронхита [14]	4,9	2,9	18,85	2,83	0	0,9	17,14	3,19

Рисунок 4 – График усредненной зависимости I_G от N'/N

6 ОБСУЖДЕНИЕ

Как видно из табл. 2, предложенный метод позволяет существенно ускорить синтез нейро-нечетких моделей, обеспечивая приемлемую точность и более высокий уровень обобщения данных моделью. Это позволяет рекомендовать предложенный метод для использования на практике.

Полученная зависимость ошибки модели E от заданной разрядности признаков L для выборок данных различной размерности и природы, график которой приведен на рис. 2, показывает, что при начальном значении $L=2$ ошибка также имеет наибольшее значение, которое постепенно снижается до некоторого значения, после чего начинается постепенный рост ошибки. Изначально высокое значение ошибки объясняется недостаточной разрядностью кодирования вследствие большой потери информации при разбиении на интервалы значений признаков из-за малого числа интервалов. Постепенное убывание ошибки по мере роста значения L объясняется увеличением объема информации, доступного для отображения в модели. Дальнейший рост ошибки объясняется ухудшением обобщающих свойств модели вследствие избыточности ее структуры и параметров, а также излишней детализации разбиения признакового пространства.

Представленный на рис. 3 график показывает, что при небольшом числе используемых признаков относительно размера исходного набора признаков ошибка построенной модели является большей, чем при использовании большего числа признаков. Однако при приближении числа используемых в модели признаков к числу признаков исходного набора убывание ошибки становится незначительным. Таким образом, отбор признаков должен быть направлен на поиск оптимального соотношения N'/N , обеспечивающего приемлемый уровень ошибки.

График, изображенный на рис. 4, показывает, что при большом числе используемых признаков относительно размера исходного набора признаков коэффициент обобщения построенной модели является большим, чем при использовании большего числа признаков. Таким образом, отбор признаков должен быть направлен также на поиск оптимального соотношения N'/N , обеспечивающего не только приемлемый уровень ошибки, но и высокий уровень обобщения модели.

ЗАКЛЮЧЕНИЕ

В работе решена актуальная задача автоматизации синтеза нейро-нечетких сетей для диагностирования и автоматической классификации по признакам.

Научная новизна работы состоит в том, что предложен метод синтеза нейро-нечетких моделей по прецедентам, который в

процессе синтеза сети определяет оценки взаимосвязи входных признаков между собой, выделяет взаимосвязанные признаки, оценивает влияние признаков на выходной признак, а также выявляет взаимосвязанные термы, включает в модель наиболее важные признаки и термы, устраняет дуближ термов и признаков путем оставления лишь части из них, использует различные варианты кодирования сигналов, а также переупорядочивает признаки, обеспечивая группировку признаков, формирует правила для прецедентов ранее не встречавшихся классов или существенно отличающихся от имеющихся правил своего класса, а уже имеющиеся правила корректирует на основе поступающих прецедентов с учетом числа ранее рассмотренных наблюдений.

Предложенный метод позволяет существенно ускорить синтез нейро-нечетких моделей, обеспечивая приемлемую точность и более высокий уровень обобщения данных, снизить сложность и избыточность, а также повысить интерпретируемость нейромодели.

Практическая значимость результатов работы состоит в том, что проведены эксперименты по решению практических задач диагностирования и автоматической классификации, подтвердившие работоспособность и применимость предложенного метода. Получена зависимость ошибки модели, синтезированной предложенным методом, от заданной разрядности признаков. Использование полученной зависимости позволяет на практике более рационально выбирать значение числа интервалов разбиения диапазонов значений признаков, обеспечивая приемлемую точность нейромодели. Получены зависимости ошибки синтезированной модели и коэффициента обобщения от доли используемых в модели признаков в исходном наборе признаков.

Перспективы дальнейших исследований заключаются в том, чтобы изучить зависимость оптимального значения числа интервалов разбиения диапазонов значений признаков от информативности набора признаков и размера набора признаков.

БЛАГОДАРНОСТИ

Работа выполнена при поддержке международного проекта «Centers of Excellence for young REsearchers» программы «Темпус» Европейской Комиссии (№ 544137-TEMPUS-1-2013-1-SK-TEMPUS-JPHES) в рамках госбюджетных научно-исследовательских тем Запорожского национального технического университета «Интеллектуальные информационные технологии диагностирования и автоматической классификации» и «Интеллектуальные методы диагностирования систем управления удаленными техническими объектами».

СПИСОК ЛИТЕРАТУРЫ

1. Биргер И. А. Техническая диагностика / И. А. Биргер. – М. : Машиностроение, 1978. – 240 с.
2. Ding S. X. Model-based fault diagnosis techniques: design schemes, algorithms, and tools / S. X. Ding. – Berlin: Springer, 2008. – 473 p.
3. Интеллектуальные информационные технологии проектирования автоматизированных систем диагностирования и распознавания образов : монография / [С. А. Субботин, Ан. А. Олейник, Е. А. Гофман, С. А. Зайцев, Ал. А. Олейник] ; под ред. С. А. Субботина. – Харьков : Компания СМІТ, 2012. – 318 с.
4. Субботин С.О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень : навч. посібник / С. О. Субботин. – Запоріжжя : ЗНТУ, 2008. – 341 с.
5. Computational intelligence: a methodological introduction / [R. Kruse, C. Borgelt, F.Klawonn et. al.]. – London: Springer-Verlag, 2013. – 488 p. DOI: 10.1007/978-1-4471-5013-8_1
6. Bishop C. M. Pattern recognition and machine learning / C. M. Bishop. – New York : Springer, 2011. – 740 p.
7. Buckleya J. J. Fuzzy neural networks: a survey / J. J. Buckleya, Y. Hayashi // Fuzzy sets and systems. – 1994. –Vol. 66, Issue 1. – P. 1–13.

8. Mamdani E. H. An experiment in linguistic synthesis with fuzzy logic controller / E. H. Mamdani, S. Assilian // *International journal of man-machine studies*. – 1975. – Vol. 7, № 1. – P. 1–13.
9. Chai Y. Mamdani Model based Adaptive Neural Fuzzy Inference System and its Application / Y. Chai, L. Jia, Z. Zhang // *International Journal of Computer, Electrical, Automation, Control and Information Engineering* Vol:3, No:3, 2009. –P. 663–670.
10. Леоненков А. В. Нечеткое моделирование в среде MATLAB и fuzzyTECH / А. В. Леоненков. – СПб.: БХВ-Петербург, 2003. – 736 с.
11. Химмельблау Д. Прикладное нелинейное программирование / Д. Химмельблау. – М.: Мир, 1974. – 534 с.
12. Комп'ютерна програма «Автоматизована система синтезу нейромережних та нейро-нечітких моделей для неруйнівної

Субботін С. О.

Д-р техн. наук, професор, професор кафедри програмних засобів Запорізького національного технічного університету, Запоріжжя, Україна

СИНТЕЗ НЕЙРО-НЕЧІТКИХ МЕРЕЖ З РАНЖИРУВАННЯМ І СПЕЦИФІЧНИМ КОДУВАННЯМ ОЗНАК ДЛЯ ДІАГНОСТИКИ Й АВТОМАТИЧНОЇ КЛАСИФІКАЦІЇ ЗА ПРЕЦЕДЕНТАМИ

Вирішено задачу автоматизації синтезу нейро-нечітких мереж для діагностування й автоматичної класифікації за ознаками. Запропоновано метод синтезу нейро-нечітких моделей за прецедентами, що визначає оцінки взаємозв'язку вхідних ознак, виділяє взаємозалежні ознаки, оцінює вплив ознак на вихідну ознаку, а також виявляє взаємозалежні терми, включає в модель найбільш важливі ознаки і терми, усуває дубляж термів і ознак, використовує різні варіанти кодування сигналів, а також переупорядковує ознаки, забезпечуючи групування ознак, формує правила для прецедентів класів, що раніше не зустрічалися, або таких, що істотно відрізняються від наявних правил свого класу, а уже наявні правила коректує на основі прецедентів, що надходять, з урахуванням числа раніше розглянутих спостережень. Запропонований метод дозволяє істотно прискорити синтез нейро-нечітких моделей, забезпечуючи прийнятну точність і більш високий рівень узагальнення даних, знизити складність і надмірність, а також підвищити інтерпретабельність нейромоделі. Проведено експерименти з вирішення практичних задач діагностування й автоматичної класифікації, що підтвердили працездатність і застосовність запропонованого методу. Отримано залежність помилки моделі, синтезованої запропонованим методом, від заданої розрядності ознак. Використання отриманої залежності дозволяє на практиці більш раціонально вибирати значення кількості інтервалів розбиття діапазонів значень ознак, забезпечуючи прийнятну точність нейромоделі.

Ключові слова: вибірка, нейро-нечітка мережа, нечітке виведення, навчання за прецедентами, діагностування.

Subbotin S. A.

Dr.Sc., Professor, Professor of department of software tools, Zaporizhzhya National Technical University, Zaporizhzhya, Ukraine

THE NEURO-FUZZY NETWORK SYNTHESIS WITH THE RANKING AND SPECIFIC ENCODING OF FEATURES FOR THE DIAGNOSIS AND AUTOMATIC CLASSIFICATION ON PRECEDENTS

The problem of automation synthesis of neuro-fuzzy networks for diagnostics and automatic classification on features is solved. The method of neuro-fuzzy model synthesis on precedents is proposed.

It evaluates the relationship of input features, extracts related features, evaluates the impact of features on the output feature and identify related terms, includes the most important features and terms in the model, eliminates dubbing of the terms and features, uses different types of signal encoding, and also arranges features providing their grouping, forms a rules for precedents of previously non-experienced classes or significantly different from the existing rules of their class, adjust the existing rules based on incoming precedents, given the number previously considered observations. The proposed method allows to significantly accelerate the synthesis of neuro-fuzzy models, providing acceptable accuracy and a higher level of generalization of data, reduce complexity and redundancy, as well as increase of interpretability of neural model. The experiments on solution of practical problems of diagnosis and automatic classification are conducted. They confirmed the efficiency and applicability of the proposed method. The dependence of error of the model synthesized by the proposed method from the specified feature bitness is obtained. Using the obtained dependence allows to more rational choose the value of the number of divisions of the range of feature values in practice providing a reasonably accurate neural model.

Keywords: sample, neuro-fuzzy network, fuzzy inference, training on precedents.

REFERENCES

1. Birger I. A. *Tekhnicheskaya diagnostika*. Moscow. Mashinostroyeniye, 1978, 240 p.
2. Ding S. X. *Model-based fault diagnosis techniques: design schemes, algorithms, and tools*. Berlin, Springer, 2008, 473 p.
3. Subbotin S. A., Oleynik An. A., Gofman Ye. A., Zaytsev S. A., Oleynik Al. A. ; pod red. S. A. Subbotina. *Intellektual'nyye informatsionnyye tekhnologii proyektirovaniya avtomatizirovannykh sistem diagnostirovaniya i raspoznavaniya obrazov* : monografiya. Khar'kov, Kompaniya SMIT, 2012, 318 p.
4. Subbotin S. O. *Podannya u obrobka znan' u sistemakh shuchnoho intelektu ta pidtrymky pryynyattya rishen'* : navch. posibnyk. Zaporizhzhya, ZNTU, 2008, 341 p.
5. Kruse R., Borgelt C., Klawonn F. et. al. *Computational intelligence: a methodological introduction*. London: Springer-Verlag, 2013, 488 p. DOI: 10.1007/978-1-4471-5013-8_1
6. Bishop C. M. *Pattern recognition and machine learning*. New York, Springer, 2011, 740 p.
7. Buckleya J. J., Hayashi Y. *Fuzzy neural networks: a survey*, *Fuzzy sets and systems*, 1994, Vol. 66, Issue 1, pp. 1–13.
8. Mamdani E. H. Assilian S. An experiment in linguistic synthesis with fuzzy logic controller, *International journal of man-machine studies*, 1975, Vol. 7, No. 1, pp. 1–13.
9. Chai Y., Jia L., Zhang Z. Mamdani Model based Adaptive Neural Fuzzy Inference System and its Application, *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, Vol:3, No:3, 2009, pp. 663–670.
10. Leonenkov A. V. *Nechotkoye modelirovaniye v srede MATLAB i fuzzyTECH*. Sankt-Peterburg, BKHV-Peterburg, 2003, 736 p.
11. Khimmel'blau D. *Prikladnoye nelineynoye programmirovaniye*. Moscow, Mir, 1974, 534 p.
12. Subbotin S. O. *Komp'yuterna prohrama «Avtomatyzovana sistema syntezu neyromerezhevykh ta neyro-nechitkykh modeley dlya neruynivnoyi diahnostryky ta klasyfikatsiyi obraziv za oznakamy»* : cvidotstvo pro reyestratsiyu avtors'koho prava na tvir № 35431. Derzh. departament intellektual'noyi vlasnosti. № 34011 ; zayavl. 21.04.10 ; zareyestr. 21.10.10.
13. Shama Ye. O., Subbotin S. O., Morshevavka S. V. *Pobudova klasyfikatora roslennykh ob'yektiv za dopomohoyu neyronnykh merezh*, *Radioelektronika, informatyka, upravlinnya*, 2013, No. 1, pp. 55–61.
14. Kolisnyk N. V., Subotyn S. O. *Modelyuvannya imunopatohezezu khronichnoho obstruktyvnoho bronkhitu za dopomohoyu neyromerezh*. Suchasni problemy biolohiyi, ekolohiyi ta khimiyi : II Mizhnarodna konferentsiya, Zaporizhzhya, 1–3 zhovtnya 2009 r. : zbirka materialiv. Zaporizhzhya, ZNU, pp. 124–125.

Статья поступила в редакцию 17.01.2016.

После доработки 10.02.2016.

¹Д-р техн. наук, профессор, заведующий лабораторией Центра разработки программных продуктов и аппаратно-программных комплексов при Ташкентском Университете Информационных Технологий, Ташкент, Узбекистан

²Канд. техн. наук, старший научный сотрудник Центра разработки программных продуктов и аппаратно-программных комплексов при Ташкентском Университете Информационных Технологий, Ташкент, Узбекистан

³Младший научный сотрудник Центра разработки программных продуктов и аппаратно-программных комплексов при Ташкентском Университете Информационных Технологий, Ташкент, Узбекистан

ПОСТРОЕНИЕ РАСПОЗНАЮЩИХ ОПЕРАТОРОВ В УСЛОВИЯХ ВЗАИМОСВЯЗАННОСТИ ПРИЗНАКОВ

Рассмотрена задача распознавания образов, заданных в пространстве взаимосвязанных признаков. Предложен новый подход к построению модели распознающих операторов, учитывающий взаимосвязанность заданных признаков. При этом построение модели осуществлено для распознающих операторов типа потенциальных функций. Основная идея предлагаемого подхода заключается в формировании независимых подмножеств взаимосвязанных признаков и выделении предпочтительной модели зависимости для каждого подмножества сильносвязанных признаков. Анализ полученных результатов показывает, что рассмотренные распознающие операторы используются в тех случаях, когда между признаками объектов, принадлежащих к одному и тому же классу, существует некоторая зависимость. При слабом выражении этой зависимости используется классическая модель распознающих операторов. Основным преимуществом предложенных распознающих операторов является улучшение точности и значительное сокращение объема вычислительных операций при распознавании неизвестных объектов, что позволяет применить их при построении распознающих систем, работающих в режиме реального времени.

Ключевые слова: распознавание образов, модель распознающих операторов, потенциальная функция, зависимость признаков, подмножество сильносвязанных признаков, репрезентативный признак, предпочтительная модель зависимости.

НОМЕНКЛАТУРА

\mathfrak{S} – множество допустимых объектов;

\tilde{S}^m – множество выделенных допустимых объектов

$(\tilde{S}^m \subset \mathfrak{S}, \tilde{S}^m = \{S_1, \dots, S_i, \dots, S_m\})$;

\tilde{S}^q – множество контрольных объектов

$(\tilde{S}^q \subset \mathfrak{S}, \tilde{S}^q = \{S'_1, \dots, S'_q\})$;

\tilde{K}_j – множество эталонных объектов, принадлежа-

щих к классу K_j ($\tilde{K}_j \subset \tilde{S}^m$);

$|\tilde{K}_j|$ – мощность множества объектов \tilde{K}_j ;

V_o – обучающая выборка

$(V_o = \{< S_i; \chi(S_i) > : S_i \in \tilde{S}^m\})$;

V_k – контрольная выборка

$(V_k = \{< S_u; \chi(S_u) > : S_u \in \tilde{S}^q\})$;

$\chi(S_i)$ – индикатор объекта S_i

$(\chi(S_i) = -1, \text{ при } S_i \in K_1, \chi(S_i) = +1, \text{ при } S_i \in K_2)$;

X – пространства n -мерных признаков

$(X = (x_1, \dots, x_i, \dots, x_n))$;

A – произвольный алгоритм распознавания

$(A = B \circ C)$;

B – распознающий оператор,

$B(I_o, \tilde{S}^q) = (b_1, b_2, \dots, b_q)$;

C – решающее правило $C(b_i) = \lambda_i (\lambda_i \in \{-1, 0, +1\})$;

$\tilde{\pi}$ – вектор параметров распознающего оператора;

$\tilde{\pi}_0$ – вектор параметров распознающего оператора,

вычисленных по выборке V_o ;

$\varphi(\tilde{\pi})$ – функционал качества распознающего опера-

тора $B(\tilde{\pi})$;

n' – число подмножеств сильносвязанных признаков;

Ω_q – q -е подмножество сильносвязанных признаков;

Y – пространство n' -мерных репрезентативных признаков ($Y = (y_1, \dots, y_{n'})$).

ВВЕДЕНИЕ

В настоящее время все более широкий круг специалистов уделяет внимание проблеме распознавания образов, и число научных публикаций по данной тематике постоянно растет. Это связано с тем, что в последние годы распознавание образов находит все большее применение в науке, технике, производстве и повседневной жизни.

В проблеме распознавания образов центральное место занимает задача построения распознающих алгоритмов. Для рассмотрения разнообразных постановок этой задачи можно выделить три параметра, от способа задания которых во многом определяется метод решения. К ним относятся, во-первых, вид описания распознаваемых объектов (независимые и взаимосвязанные признаки), во-вторых, способ предъявления обучающей информации (единственная выборка и последовательность выборок) и, в-третьих, тип используемой модели, на основе которой формируется алгоритм распознавания (см. раздел 2).

Такой подход дает возможность выделить отдельные подклассы задач распознавания образов, описанных в пространстве взаимосвязанных признаков. Однако решение этих задач исследовано недостаточно (см. раздел 2).

Целью данной работы является разработка модифицированной модели распознающих операторов типа потенциальных функций [1–3], направленная на приспособление данной модели к решению задач распознавания образов, описанных в пространстве взаимосвязанных признаков.

Для достижения этой цели необходимо решить следующие задачи:

- проанализировать существующие модели распознающих операторов и определить круг решаемых задач;
- разработать модель алгоритмов распознавания, основанных на оценке взаимосвязанности признаков;
- провести экспериментальные исследования для оценки эффективности разработанных распознающих операторов.

Объектом исследования являются распознающие операторы типа потенциальных функций. Предмет исследования – модифицированные распознающие операторы, основанные на оценке взаимосвязанности признаков.

Основная идея предлагаемой модели распознающих операторов заключается в поиске некоторых зависимостей между признаками, характеризующими объекты, принадлежащие к одному и тому же классу.

В научном плане результаты данной работы представляют собой новое решение научной задачи, связанной с построением распознающих алгоритмов при условии большой размерности пространства признаков. Практическая значимость полученных результатов заключается в том, что разработанные алгоритмы и программы могут быть применены при решении прикладных задач в условиях большой размерности пространства признаков (например, при идентификации человека по фото-портрету).

1 ПОСТАНОВКА ЗАДАЧИ

Для простоты, но без ограничения общности, рассмотрим задачу распознавания образов в случае двух классов. Пусть набор объектов $\tilde{S}^q = \{S'_1, \dots, S'_q\}$ ($\tilde{S}^q \in \{S\}$) задан в пространстве признаков X большой размерности ($X = (x_1, \dots, x_i, \dots, x_n)$, $n > 200$).

Основная задача заключается в построении такого распознающего оператора B , который с применением решающего правила C (см. формула 2 в разделе 2) позволяет вычислить значения характеристической функции $\tilde{\lambda}_i = \chi(S'_i)$, $i = \overline{1, q}$ по выборке V_o :

$$B(\tilde{S}^q) = \|b_i\|_q, \quad \|b_i\|_q = (b_1, \dots, b_i, \dots, b_q)^T,$$

$$C(\|b_i\|_q) = (\tilde{\lambda}_1, \dots, \tilde{\lambda}_i, \dots, \tilde{\lambda}_q)^T, \quad \tilde{\lambda}_i \in \{-1, 0, 1\}.$$

Здесь $\tilde{\lambda}_i$ интерпретируется следующим образом: если $\tilde{\lambda}_i \in \{-1, +1\}$ ($\tilde{\lambda}_i = -1$ – объект S'_i входит в класс K_1 , $\tilde{\lambda}_i = +1$ – объект S'_i входит в класс K_2), то $\tilde{\lambda}_i$ есть значение характеристической функции $\lambda(S'_i)$ на допустимом объекте S'_i , вычисленное оператором B ; если $\tilde{\lambda}_i = 0$, то считается, что оператор B не смог определить значение характеристической функции $\lambda(S'_i)$.

2 ОБЗОР ЛИТЕРАТУРЫ

В течение длительного времени подавляющее большинство приложений теории распознавания образов

было связано с плохо формализованными областями – медициной, геологией, социологией, химией и т. д. Поэтому на первом этапе развития распознавания появилось множество алгоритмов, которые носили характер проектов различных технических устройств или алгоритмов для решения конкретных прикладных задач. Ценность разработанных алгоритмов определялась, прежде всего, достигнутыми экспериментальными результатами [2, 4].

Второй этап развития теории распознавания образов характеризуется переходом от отдельных алгоритмов к построению моделей – семейства алгоритмов для единого описания методов решения классификационных задач. Потребность в синтезе моделей алгоритмов распознавания образов определялась необходимостью фиксировать класс алгоритмов при выборе оптимальной или хотя бы приемлемой процедуры решения конкретной задачи.

На данном этапе развития Ю. И. Журавлев показал [2], что произвольный алгоритм распознавания можно представить как последовательное выполнение операторов B (распознающий оператор) и C (решающее правило):

$$A = B \circ C. \quad (1)$$

Из (1) следует, что каждый алгоритм A можно разделить на два последовательных этапа. На первом этапе распознающий оператор B осуществляет перевод допустимого объекта S_i в числовую оценку b_i : $B(S_i) = b_i$.

На втором этапе по числовой оценке b_i решающее правило C определяет принадлежность объекта S_i к классам K_1 и K_2 :

$$C(\mu(S_i)) = C(b_i), \quad C(b_i) = \tilde{\lambda}_i,$$

$$\tilde{\lambda}_i = \begin{cases} -1, & \text{если } \mu(S_i) < -c; \\ 0, & \text{если } c \leq \mu(S_i) \leq c; \\ +1, & \text{если } c < \mu(S_i), \end{cases} \quad (2)$$

где c – параметр решающего правила.

К настоящему времени построено и изучено несколько типов моделей. К их числу относятся [1–12]:

- модели, основанные на использовании разделяющих функций;
- модели, основанные на использовании математической статистики и теории вероятности;
- модели, основанные на использовании аппарата математической логики;
- модели, основанные на использовании принципа потенциалов;
- модели, основанные на использовании алгоритмов вычисления оценок.

Анализ литературных источников, в частности [1–12], показывает, что разработанные модели алгоритмов распознавания, в основном, ориентированы на решение задач, где объекты описаны в пространстве независимых признаков (или зависимость между признаками достаточно слабая).

На практике часто встречаются прикладные задачи распознавания образов, заданных в пространстве признаков большой размерности. При решении подобных

задач предположение о независимости признаков достаточно часто не выполняется [4, 13]. Следовательно, остается недостаточно решенным вопрос по созданию распознающих алгоритмов, которые могут быть применены для решения прикладных задач распознавания при больших размерностях признакового пространства и наличии взаимосвязанности признаков.

3 МАТЕРИАЛЫ И МЕТОДЫ

Для решения задачи, сформулированной в разделе 1, предлагается подход, который является логическим продолжением работ академика Ю. И. Журавлева и его учеников. На базе этого подхода разработана модель модифицированных распознающих операторов, основанных на выявлении независимых подмножеств взаимосвязанных признаков и выделении предпочтительной модели зависимости для каждого подмножества сильносвязанных признаков. В качестве исходной модели рассматриваются распознающие операторы типа потенциальных функций [1–3].

Предлагаемая модель распознающих операторов включает следующие основные этапы.

1. *Выделение подмножеств сильносвязанных признаков.* На этом этапе определяется система «независимых» подмножеств признаков, состав которой будет зависеть от параметра n' . В зависимости от способа задания меры близости между подмножествами сильносвязанных признаков (Ω_p и Ω_q) и функционала качества классификационного анализа можно получить разнообразные процедуры выделения независимых множеств сильносвязанных признаков. Процедура выделения подмножеств сильносвязанных признаков более подробно рассмотрена в [14].

2. *Формирование набора репрезентативных признаков.* В результате выполнения данного этапа получаем сокращенное пространство признаков, размерность которого намного меньше исходного ($n' < n$). Далее сформированное пространство признаков обозначим через Y ($Y = (y_1, \dots, y_{n'})$).

Процедура выделения набора репрезентативных признаков более подробно рассмотрена в [15].

3. *Определение моделей зависимости в каждом подмножестве признаков для класса K_j ($j = \overline{1, l}$).* Пусть x_i – произвольный признак, принадлежащий подмножеству Ω_q . Предполагается, что элементы Ω_q линейно упорядочены по индексу признаков (т. е. $x_i < x_j$, если $i < j$).

Далее, нулевым элементом (x_0) подмножества Ω_q считается y_q , остальные элементы обозначаются через x_i

$$(N_q = |\Omega_q|; i = 1, \dots, N_q - 1).$$

Пусть модель зависимости в Ω_q имеет вид

$$y_q = F(\bar{c}, x_i), \quad x_i \in \Omega_q \setminus y_q,$$

где \bar{c} – вектор неизвестных параметров, x_i – произвольный признак, принадлежащий подмножеству Ω_q , F – функция из некоторого заданного класса $\{F\}$.

Вычисленные значения вектора неизвестных параметров \bar{c} определяют модель зависимости в подмножестве признаков Ω_q для класса K_j ($j = \overline{1, l}$). В зависимости от задания параметрического вида $F(\bar{c}, x)$ и метода определения \bar{c} получаем разнообразные модели зависимости во множестве признаков Ω_q ($q = \overline{1, n'}$).

4. *Выделение предпочтительных моделей зависимости.* Поиск предпочтительной модели зависимости в Ω_q осуществляется на основе оценки доминированности рассматриваемой модели, которая разделяет объекты, принадлежащие множеству \tilde{S}^m , на два подмножества K_1 и K_2 [16]:

$$T_{qi} = \frac{L_1 \sum_{S \in \tilde{K}_2} |y_q - F_q(\bar{c}, x_i)|}{L_2 \sum_{S \in \tilde{K}_1} |y_q - F_q(\bar{c}, x_i)|}, \quad L_1 = |\tilde{K}_1|, \quad L_2 = |\tilde{K}_2|.$$

Чем больше величина T_{qi} , тем больше отдается предпочтение i -ой модели зависимости в подмножестве Ω_q . Если несколько моделей получают одинаковое предпочтение, то выбирается любая из них.

В результате выполнения данного этапа определяется предпочтительная модель зависимости для подмножества признаков Ω_q . Далее рассматриваются только эти модели зависимости.

5. *Определение функции различия между объектами S_u и S .* Пусть заданы два объекта S_u и S в пространстве $X = (x_1, \dots, x_i, \dots, x_n)$:

$$S_u = (a_{1u}, \dots, a_{nu}) \quad \text{и} \quad S = (b_1, \dots, b_n).$$

На данном этапе определяется расстояние между этими объектами по всем подмножествам Ω_q ($q = \overline{1, n'}$):

$$d(S_u, S) = \sum_{q=1}^{n'} \tau_q d_q(S_u, S),$$

$$d_q(S_u, S) = |\phi_q - \psi_q|, \quad \phi_q = (a_{ui} + b_i)/2,$$

$$\psi_q = (F_q(\bar{c}, a_{ui}) + F_q(\bar{c}, b_i))/2,$$

где τ_q – параметр распознающего оператора.

6. *Определение функции близости между объектами S_u и S .* На этом этапе функция близости между объектами S_u и S определяется как потенциальная функция второго типа, например [1, 3],

$$U(S_u, S) = \frac{1}{1 + \xi d(S_u, S)},$$

где ξ – параметр распознающего оператора.

7. *Вычисление оценки принадлежности к классу K_j .* Оценка принадлежности объекта S к классу K_j ($j = 1, 2$) определяется как функция оценки близости:

$$\mu(S) = \sum_{S_u \in \tilde{K}_2} \gamma_u U(S_u, S) - \sum_{S_u \in \tilde{K}_1} \gamma_u U(S_u, S),$$

где γ_u – параметр распознающего оператора.

Таким образом, определена модель распознающих операторов типа потенциальных функций, основанных на оценке взаимосвязанности признаков. Произвольный оператор B из этой модели полностью определяется заданием набора параметров $\tilde{\pi}$ [17]. Совокупность всех распознающих операторов из предлагаемой модели обозначаем через $B(\tilde{\pi}, S)$. Определение наилучшего распознающего оператора в рамках рассмотренной модели осуществляется в пространстве параметров $\tilde{\pi}$. Наилучший оператор $B(\tilde{\pi}_o, S)$ выбирается на основе поиска минимального значения функционала качества распознающего оператора:

$$\varphi(\tilde{\pi}) = \frac{1}{m} \sum_{S \in V_o} \theta \| \chi(S) - C(B(\tilde{\pi}, S)) \|, \quad \theta(x) = \begin{cases} 1, & \text{при } x = 0; \\ 0, & \text{при } x \neq 0, \end{cases} \quad (3)$$

где $\| \cdot \|$ – норма булевого вектора.

4 ЭКСПЕРИМЕНТЫ

Экспериментальное исследование работоспособности предложенной модели распознающих операторов осуществлено на примере решения модельной задачи.

Исходные данные распознаваемых объектов для модельной задачи сгенерированы в пространстве зависимых признаков. Количество классов в данном эксперименте равно двум. Объем исходной выборки – 1100 реализаций (по 550 реализаций для объектов каждого класса). Количество признаков в модельном примере равно 20. Число подмножеств сильносвязанных признаков – 5.

В качестве испытываемых моделей распознающих операторов были выбраны: классическая модель распознающих операторов типа потенциальных функций (B_1), модель распознающих операторов, основанных на вычислении оценок (B_2), и модель (B_3), предлагаемая в настоящей работе. Сравнительный анализ перечисленных моделей распознающих операторов при решении рассмотренной задачи проведен по следующим критериям:

- точность распознавания объектов контрольной выборки;
- время, затраченное на обучение;
- время, затраченное на распознавание объектов из контрольной выборки.

Для вычисления указанных критериев при решении рассматриваемой задачи, в целях исключения удачного (или неудачного) разбиения исходной выборки V на две равные части V_o и V_k ($V = V_o \cup V_k$, V_o – выборка для обучения, V_k – выборка для контроля), используется метод скользящего контроля [18]. В этом методе исходная выборка объектов случайным образом разбивается на 110 непересекающихся блоков, включающих по 10 объектов каждый. При этом требуется, чтобы во всех блоках сохранилась пропорция по количеству объектов, принадлежащих к разным классам. В результате получается, что в каждом блоке по 5 объектов каждого класса. Процесс скользящего контроля по этим блокам включает несколько шагов. На каждом шаге выбирают 10 из 110 блоков в качестве обучающей выборки, и на этой выборке обучаются распознающие операторы с заданными парамет-

рами. Обученный таким образом распознающий оператор проверяется на остальных 100 блоках (контрольной выборке). В результате каждой проверки определяется и фиксируется оценка качества распознающих операторов по указанным критериям. При выполнении каждого очередного шага для оценки качества распознающих операторов из контрольной и обучающей выборок выбирают по одному блоку и меняют их местами. В целях исключения повторного использования объектов обучающей выборки соответствующие блоки маркируются, и при выборе кандидатов для включения в обучающую выборку они не участвуют. После завершения процедуры скользящего экзамена точность распознавания и временные показатели определялись как средние. Эксперименты проводились на компьютере Pentium IV Dual Core 2,2 GHz с объемом оперативной памяти 1 Gb.

5 РЕЗУЛЬТАТЫ

Основным результатом данной статьи является модель распознающих операторов, основанных на оценке взаимосвязанности признаков (см. раздел 3). Построение данной модели выполнено в рамках модели алгоритмов типа потенциальных функций и заключалось в выявлении независимых подмножеств взаимосвязанных признаков с последующим выделением предпочтительной модели зависимости для каждого подмножества сильносвязанных признаков. Данная модель распознающих операторов имеет большое значение при решении задачи распознавания образов в условиях взаимосвязанности признаков и позволяет расширить область применения метода потенциальных функций.

В целях проверки работоспособности предложенной модели операторов проведено экспериментальное исследование при решении модельной задачи. Как уже отмечалось ранее, модельная задача решалась с применением классической модели распознающих операторов типа потенциальных функций (B_1), модели распознающих операторов, основанных на вычислениях оценок, (B_2) и предлагаемой модели (B_3). Точность распознавания в процессе обучения (см. формула 3 в разделе 3) для B_1 равна 94,8%, для B_2 – 97,6%, и, наконец, для B_3 – 99,1%. Результаты решения рассматриваемой задачи с применением B_1 , B_2 и B_3 в процессе контроля приведены в табл. 1.

Сравнение этих результатов показывает (см. табл. 1), что предложенная модель распознающих операторов $|B_3|$ позволила повысить точность распознавания объектов, описанных в пространстве взаимосвязанных признаков (более чем на 10% выше, чем по B_1 и B_2). Это объясняется тем, что в моделях B_1 и B_2 не учитывается взаимосвязанность признаков. Однако для модели B_3

Таблица 1 – Результаты решения задачи с помощью различных распознающих операторов

Распознающий оператор	Время (в сек.)		Точность распознавания (в %)
	обучения	распознавания	
B_1	4,141	0,014	80,5
B_2	6,127	0,012	82,7
B_3	9,041	0,002	93,4

имеет место некоторое увеличение времени обучения за счет реализации дополнительной процедуры формирования независимых подмножеств взаимосвязанных признаков.

6 ОБСУЖДЕНИЕ

Разработанные операторы отличаются от традиционных распознающих операторов типа потенциальных функций тем, что они основаны на оценке взаимосвязанности признаков. Поэтому эти алгоритмы целесообразно использовать в тех случаях, когда между признаками обнаруживается какая-нибудь зависимость. Несомненно, что эта зависимость должна отличаться для объектов каждого класса. Это позволяет описать объекты каждого класса индивидуальной моделью. Если зависимость между признаками слаба, то используется классическая модель распознающих операторов (например, модель, рассмотренная в [2, 3]). Следовательно, предложенная модель распознающих операторов не является альтернативной моделям распознающих операторов типа потенциальных функций, а только дополняет их.

В случае, когда между признаками всех рассматриваемых объектов обнаруживается достаточно сильная зависимость, то в процессе формирования набора репрезентативных признаков (описанных на первом и втором этапах) исключаются признаки, повторяющие одну и ту же информацию, что обеспечивает выбор признаков, достаточно хорошо представляющих все те признаки, которые не содержатся в данном наборе [4].

Результаты приведенного экспериментального исследования показывают, что предложенная модель распознающих операторов позволяет решать задачу распознавания образов более точно в условиях взаимосвязанности признаков. При этом показатель времени, затраченного на распознавание объектов, намного меньше, чем тот же показатель B_1 и B_2 (см. табл. 1). Это связано с тем, что в предложенных распознающих операторах для распознавания объекта используются только предпочтительные модели зависимости, что обусловило повышение скорости распознавания объектов. Поэтому эту модель можно использовать при разработке систем распознавания, функционирующих в режиме реального времени. Вместе с тем необходимо отметить тот факт, что время, израсходованное на обучение алгоритма, увеличилось, т. к. для построения оптимального распознающего оператора требуется оптимизировать большее число параметров, чем при использовании традиционной модели распознающих операторов, в частности B_1 и B_2 .

ВЫВОДЫ

1. Анализ существующих моделей распознающих операторов показал, что в основном они ориентированы на решение прикладных задач распознавания объектов с независимыми признаками. В большинстве прикладных задач распознавания, встречающихся в науке, технике и производстве, рассматриваемые образы характеризуются взаимосвязанными признаками. Хотя разработаны различные алгоритмы решения задач распознавания для таких образов, они оказались малоэффективными с точки зрения точности и сложности вычисления.

2. Предложен новый подход для построения модели распознающих операторов и на основе этого подхода построена модифицированная модель распознающих операторов типа потенциальных функций. Предложенная модель операторов опирается на оценку взаимосвязанности признаков. Она позволяет расширить модели распознающих операторов типа потенциальных функций и увеличивает область применения метода потенциальных функций. Данная модель ориентирована на решение задачи распознавания объектов, заданных в пространстве взаимосвязанных признаков.

3. Результаты решения модельной задачи показали, что предложенная модель распознающих операторов улучшает точность и значительно сокращает число вычислительных операций посредством распознавания неизвестного объекта, заданного в пространстве взаимосвязанных признаков. Она может быть использована при составлении различных программных комплексов, ориентированных на решение прикладных задач распознавания.

БЛАГОДАРНОСТИ

Работа выполнена в рамках гранта А5-ФА-Ф022 Государственного комитета по координации развития науки и технологий Республики Узбекистан.

СПИСОК ЛИТЕРАТУРЫ

1. Айзерман М. А. Метод потенциальных функций в теории обучения машин / М. А. Айзерман, Э. М. Браверманн, Л. И. Розоноэр. – М.: Наука, 1970. – 348 с.
2. Журавлев Ю. И. Избранные научные труды / Ю. И. Журавлев. – М.: Магистр, 1998. – 420 с.
3. Журавлев Ю. И. Распознавание. Математические методы. Программная система. Практические применения / Ю. И. Журавлев, В. В. Рязанов, О. В. Сенько. – М.: Фазис, 2006. – 159 с.
4. Камиллов М. М. Современное состояние вопросов построения моделей алгоритмов распознавания / М. М. Камиллов, Н. М. Мирзаев, С. С. Раджабов. // Научный журнал: Химическая технология. Контроль и управление. – 2009. – № 2. – С. 21–27.
5. Загоруйко Н. Г. Прикладные методы анализа данных и знаний / Н. Г. Загоруйко. – Новосибирск: ИМ СО РАН, 1999.
6. Лбов Г. С. Логические решающие функции и вопросы статистической устойчивости решений / Г. С. Лбов, Н. Г. Старцева. – Новосибирск: Изд-во ИМ СО РАН, 1999. – 211 с.
7. Шлезингер М. Десять лекций по статистическому и структурному распознаванию / М. Шлезингер, В. Главач. – К.: Наукова думка, 2004. – 545 с.
8. Потапов А. С. Распознавание образов и машинное восприятие / А. С. Потапов. – СПб.: Политехника, 2007. – 548 с.
9. Duda R. O. Pattern Classification, Second Edition / R. O. Duda, P. E. Hart, D. G. Stork. – New York: John Wiley, Inc., 2001. – 680 p.
10. Vapnik V. Statistical Learning Theory / V. Vapnik. – New York: John-Wiley Sons, Inc., 1998. – 732 p.
11. Theodoridis S. Pattern Recognition: Theory and Applications, 4th edition / S. Theodoridis, K. Koutroumbas. – New York: Academic Press, 2009. – 957 p.
12. Dougherty G. Pattern Recognition and Classification: An Introduction / G. Dougherty. – New York: Springer, 2013. – 196 p.
13. Фазылов Ш. Х. Построение модели алгоритмов вычисления оценок с учетом большой размерности признакового пространства / Ш. Х. Фазылов, Н. М. Мирзаев, С. С. Раджабов // Вестник СГТУ. – Саратов, 2012. – № 1 (64), Вып. 2. – С. 274–279.
14. Мирзаев Н. М. Алгоритмы выделения подмножеств сильно связанных признаков / Н. М. Мирзаев // Вопросы кибернетики: сб. науч. тр. – Ташкент: ИМИТ АН РУз, 2008. – Вып. 177. – С. 99–104.
15. Мирзаев О. Н. Выделение репрезентативных признаков при построении алгоритмов распознавания / О. Н. Мирзаев // Проблемы информатики и энергетики. – Ташкент, 2008. – № 6. – С. 23–27.

16. Модель распознающих операторов, основанных на принципе ближайшего соседа, в условиях взаимосвязанности признаков / Ш. Х. Фазылов, Н. М. Мирзаев, С. С. Раджабов, И. К. Каримов // Информатика и системы управления. – Благовещенск, 2012. – № 4. – С. 34–42.
17. Мирзаев Н. М. О параметризации моделей алгоритмов распознавания, основанных на оценке взаимосвязанности признаков / Фазылов Ш. Х.¹, Мирзаев Н. М.², Мирзаев О. Н.³
18. Vorontsov K. V. Computable Combinatorial Overfitting Bounds / K. V. Vorontsov, A. I. Frey, E. A. Sokolov // Journal of Machine Learning and Data Analysis. – 2013. – Т. 1, № 6. – С. 734–743. Статья поступила в редакцию 09.11.2015. После доработки 02.12.2015.

¹Д-р техн. наук, профессор, завідувач лабораторії Центру розробки програмних продуктів і апаратно-програмних комплексів при Ташкентському Університеті Інформаційних Технологій, Ташкент, Узбекистан

²Канд. техн. наук, старший науковий співробітник Центру розробки програмних продуктів і апаратно-програмних комплексів при Ташкентському Університеті Інформаційних Технологій, Ташкент, Узбекистан

³Молодший науковий співробітник Центру розробки програмних продуктів і апаратно-програмних комплексів при Ташкентському Університеті Інформаційних Технологій, Ташкент, Узбекистан

ПОБУДОВА РОЗПІЗНАЮЧИХ ОПЕРАТОРІВ В УМОВАХ ВЗАЄМОП'ЯЗАНОСТІ ОЗНАК

Розглянуто задачу розпізнавання образів, заданих у просторі взаємозалежних ознак. Запропоновано новий підхід до побудови моделі розпізнавальних операторів, що враховує взаємозв'язок заданих ознак. При цьому побудова моделі здійснена для розпізнавальних операторів типу потенційних функцій. Основна ідея запропонованого підходу полягає у формуванні незалежних підмножин взаємозалежних ознак і виділенні кращої моделі залежності для кожної підмножини сильнозв'язаних ознак. Аналіз отриманих результатів показує, що розглянуті розпізнавальні оператори використовуються в тих випадках, коли між ознаками об'єктів, що належать до доного класу, існує деяка залежність. При слабкому вираженні цієї залежності використовується класична модель розпізнавальних операторів. Основною перевагою запропонованих розпізнавальних операторів є поліпшення точності і значне скорочення обсягу обчислювальних операцій при розпізнаванні невідомих об'єктів, що дозволяє застосувати їх при побудові розпізнавальних систем, які працюють у режимі реального часу.

Ключові слова: розпізнавання образів, модель розпізнавальних операторів, потенційна функція, залежність ознак, підмножина сильнозв'язаних ознак, репрезентативна ознака, краща модель залежності.

Fazilov Sh. Kh.¹, Mizraev N. M.², Mizraev O. N.³

¹D. Sc., Professor, Head of the Laboratory of the Centre for development of software and hardware-program complexes at Tashkent University of information technologies, Tashkent, Uzbekistan

²PhD, Senior Researcher at the Centre for Development of Software and Hardware-program Complexes at Tashkent University of information technologies, Tashkent, Uzbekistan

³Junior researcher at the Centre for Development of Software and Hardware-program Complexes at Tashkent University of information technologies, Tashkent, Uzbekistan

BUILDING OF RECOGNITION OPERATORS IN CONDITION OF FEATURES' CORRELATIONS

The problem of recognizing of patterns given in the space of correlated features is considered. The new approach to the building of model of recognition operators, which considers the correlation of given features, is proposed. The building of the model is carried out for potential function type recognition operators. The main idea of the proposed approach is formation of uncorrelated subsets of strongly correlated features and extracting preferred correlation model for each of subsets of strongly correlated features. Analysis of the results shows that the considered recognition operators are used in cases when there is a certain correlation between objects belonging to the same class. When the expression of this relationship is weak, classical model of recognition operators is used. The main advantage of the proposed recognition operators is to improve the accuracy and the significant reduction in the volume of computational operations in recognition of unknown objects, which allows them to use when building recognition systems working in real time.

Keywords: pattern recognition, model of recognition operators, potential function, features' correlations, subset of strongly correlated features, representative feature, preferred correlation model.

REFERENCES

1. Ayzeman M. A., Braverman E. M., Ayzeman M. A., Rozonoer L. I. Metod potentsialnyh funktsiy v teorii obucheniya mashin. Moscow, Nauka, 1970, 348 p.
2. Zhuravlev Yu. I. Izbrannyye nauchnyye trudy. Moscow, Magistr, 1998, 420 p.
3. Zhuravlev Yu. I., Ryazanov V. V., Senko O. V. Raspoznavaniya. Matematicheskiye metody. Programmnaya sistema. Prakticheskiye primeneniya. Moscow, Fazis, 2006, 159 p.
4. Kamilov M. M., Mirzayev N. M., Radjabov S. S. Sovremennoye sostoyaniye voprosov postroyeniya modeley algoritmov raspoznavaniya, *Nauchnyy zhurnal: Himicheskaya tehnologiya. Kontrol i upravleniye*, 2009, No. 2, pp. 21–27.
5. Zagoruyko N. G. Prikladniye metody analiza dannyh i znaniy. Novosibirsk, IM SO RAN, 1999.
6. Lbov G. S., Startseva N. G. Logicheskiye reshayushiy funktsii i voprosy statisticheskoy ustoychivosti resheniy. Novosibirsk, Izd-vo IM SO RAN, 1999, 211 p.
7. Shlezinger M., Glavach V. Desyat lektsiy po statisticheskomu i strukturnomu raspoznavaniyu. Kiev, Naukova dumka, 2004, 545 p.
8. Potapov A. S. Raspoznavaniye obrazov i mashinnoye vospriyatiye. Sankt-Peterburg, Politehnika, 2007, 548 p.
9. Duda R. O. Hart P. E. Stork D. G. Pattern Classification, Second Edition. New York, John Wiley, Inc., 2001, 680 p.
10. Vapnik V. Statistical Learning Theory. New York, John-Wiley Sons, Inc., 1998, 732 p.
11. Theodoridis S., Koutroumbas K. Pattern Recognition: Theory and Applications, 4th edition. New York, Academic Press, 2009, 957 p.
12. Dougherty G. Pattern Recognition and Classification: An Introduction. New York, Springer, 2013, 196 p.
13. Fazylov Sh. H., Mirzayev N. M., Radjabov S. S. Postroyeniye modeli algoritmov vychisleniya otsenok s uchytom bolshoy razmernosti priznakovogo prostranstva, *Vestnik SGTU*, Saratov, 2012. – No. 1 (64), Vyp. 2, pp. 274–279.
14. Mirzaev N. M. Algoritmy vydeleniya podmnozhestv silnosvyazannyh priznakov, *Voprosy kibernetiki: Sb. nauch. tr.* Tashkent, IMIT AN RUz, 2008, Vyp. 177, pp. 99–104.
15. Mirzaev O. N. Vydeleniye perpezentativnyh priznakov pri postroyenii algoritmov raspoznavaniya, *Problemy informatiki i energetiki*. Tashkent, 2008, No. 6, pp. 23–27.
16. Fazylov Sh. H., Mirzayev N. M., Radjabov S. S., Karimov I. K. Model raspoznavayushih operatorov, osnovannyh na printsipe bliyayshego sosedya, v usloviyah vzaimosvyazannosti priznakov, *Informatika i sistemy upravleniya*. Blagoveshensk, 2012, No. 4, pp. 34–42.
17. Mirzayev N. M., Radjabov S. S., Zhumayev T. S. O parametrizatsii modeley algoritmov raspoznavaniya, osnovannyh na otsenke vzaimosvyazannosti priznakov, *Problemy informatiki i energetiki*, Tashkent, 2008, No. 2–3, pp. 23–27.
18. Vorontsov K. V., Frey A. I., Sokolov E. A. Computable Combinatorial Overfitting Bounds, *Journal of Machine Learning and Data Analysis*, 2013, Vol. 1, No. 6, pp. 734–743.

ПРОГРЕСИВНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

ПРОГРЕССИВНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

PROGRESSIVE INFORMATION TECHNOLOGIES

УДК 681.324

Бабаков Р. М.

Канд. техн. наук, доцент, доцент кафедры компьютерных наук Донецкого национального технического университета,
г. Красноармейск, Украина

ПРОМЕЖУТОЧНАЯ АЛГЕБРА ПЕРЕХОДОВ В МИКРОПРОГРАММНОМ АВТОМАТЕ

Решена задача формализации задания микропрограммного автомата, в структуре которого часть автоматных переходов реализуется неканоническим способом. Предложен новый подход к организации функции переходов микропрограммного автомата, в соответствии с которым функция переходов представляется в виде семейства частичных функций, каждая из которых определена лишь на части области определения функции переходов автомата и соответствует некоторому подмножеству автоматных переходов.

С учетом предложенного подхода традиционное представление автомата в виде многоосновной алгебры претерпевает ряд изменений. Во-первых, взаимная независимость функций переходов и выходов, образующих сигнатуру алгебры, позволяет рассматривать их отдельно друг от друга, что приводит к представлению автомата в виде двух алгебр: алгебры переходов, сигнатура которой содержит только функцию переходов, и алгебры выходов, сигнатура которой содержит только функцию выходов. Во-вторых, представление функции переходов в виде множества частичных функций приводит к замене алгебры переходов множеством подалгебр переходов, в каждой из которых сигнатура образована частичной функцией переходов.

На примере микропрограммного автомата со счетчиком показано, что закон преобразования кодов состояний в рамках некоторого подмножества переходов может быть задан некоторой алгебраической функцией (операцией переходов), использующей скалярную интерпретацию кодов состояний структурного автомата. Скалярную интерпретацию кодов состояний совместно с операцией переходов предлагается представлять в виде т.н. промежуточной алгебры переходов, изоморфной соответствующим подалгебрам переходов абстрактного и эквивалентного ему структурного автоматов.

Ключевые слова: микропрограммный автомат, частичная функция переходов, промежуточная алгебра переходов, автомат на счетчике.

НОМЕНКЛАТУРА

МПА – микропрограммный автомат;
 S_A – абстрактный автомат;
 A – множество состояний абстрактного автомата;
 Z – входной алфавит абстрактного автомата;
 W – выходной алфавит абстрактного автомата;
 G_A – абстрактная алгебра, задающая абстрактный автомат;
 \mathcal{A}_A – носитель абстрактной алгебры;
 \mathcal{F}_A – сигнатура абстрактной алгебры;
 δ – функция переходов абстрактного автомата;
 λ – функция выходов абстрактного автомата;
 G_δ – абстрактная алгебра переходов;
 \mathcal{A}_δ – носитель абстрактной алгебры переходов;
 \mathcal{F}_δ – сигнатура абстрактной алгебры переходов;

G_λ – абстрактная алгебра выходов;
 \mathcal{A}_λ – носитель абстрактной алгебры выходов;
 \mathcal{F}_λ – сигнатура абстрактной алгебры выходов;
 a_n – начальное состояние автомата;
 D_δ – область определения функции переходов абстрактного автомата;
 δ_i – частичная функция переходов абстрактного автомата;
 N_δ – количество частичных функций переходов δ_i ;
 B – количество переходов автомата;
 a_i – текущее состояние абстрактного автомата;
 z_j – входной сигнал абстрактного автомата;
 a_k – состояние перехода абстрактного автомата;
 D_{δ_i} – область определения функции δ_i ;
 G_{δ_i} – частичная абстрактная подалгебра переходов;

© Бабаков Р. М., 2016

DOI 10.15588/1607-3274-2016-1-8

\mathcal{A}_{δ_i} – носитель частичной абстрактной подалгебры переходов G_{δ_i} ;

\mathcal{F}_{δ_i} – сигнатура частичной абстрактной подалгебры переходов G_{δ_i} ;

A_{δ_i} – подмножество состояний, встречающихся в кортежах функции δ_i ;

Z_{δ_i} – подмножество входных сигналов, встречающихся в кортежах функции δ_i ;

S_S – структурный автомат;

G_S – структурная алгебра переходов;

\mathcal{A}_S – носитель структурной алгебры переходов;

\mathcal{F}_S – сигнатура структурной алгебры переходов;

d – структурная функция переходов;

l – структурная функция выходов;

X – множество структурных входных сигналов;

Y – множество структурных выходных сигналов;

G_d – структурная алгебра переходов;

\mathcal{A}_d – носитель структурной алгебры переходов;

\mathcal{F}_d – сигнатура структурной алгебры переходов;

G_l – структурная алгебра выходов;

\mathcal{A}_l – носитель структурной алгебры выходов;

\mathcal{F}_l – сигнатура структурной алгебры выходов;

D_d – область определения структурной функции переходов;

d_i – частичная структурная функция переходов;

N_d – количество частичных структурных функций переходов;

D_{d_i} – область определения функции d_i ;

F – мощность множества Z ;

L – мощность множества X ;

G – мощность множества W ;

N – мощность множества Y ;

$K_S(a)$ – структурный код состояния автомата;

$K_S(z)$ – структурный код входного сигнала;

T – структурный код текущего состояния автомата;

R – разрядность кода T ;

$КС_{d_i}$ – комбинационная схема, реализующая структурную функцию переходов d_i ;

MX – мультиплексор;

$КС_{Inc}$ – схема управления мультиплексором MX ;

$РП$ – регистр памяти;

$СФМО$ – схема формирования микроопераций;

$СТ$ – инкрементный счетчик состояний;

$K_I(a_i)$ – промежуточный код состояния a_i ;

G_I – промежуточная алгебра переходов;

\mathcal{A}_I – носитель промежуточной алгебры переходов;

\mathcal{F}_I – сигнатура промежуточной алгебры переходов;

O_{Inc} – операция инкремента;

$K_I(A_{\delta_i})$ – множество кодов $K_I(a_i)$, используемых в функции δ_i .

ВВЕДЕНИЕ

Важным компонентом современных вычислительных систем является устройство управления, одним из способов реализации которого является микропрограммный автомат [1]. Наблюдаемый сегодня рост сложности алгоритмов, интерпретируемых МПА, способствует увеличению аппаратных затрат в логической схеме автомата. В данном аспекте актуальной является задача снижения аппаратных затрат в схеме МПА, одно из решений которой состоит в разработке новых структур МПА и их формальных математических моделей. Классические математические модели абстрактного и структурного автоматов подходят для описания предложенной В. М. Глушковым канонической структуры МПА, однако при использовании иных структур данные модели в некоторых случаях не отражают внутренние информационные процессы, ограничивая возможности проектировщика цифровых устройств в вопросах оптимизации логической схемы синтезируемого устройства.

Объектом исследования в работе выступает микропрограммный автомат, предметом исследования – математическая модель МПА, предполагающая специальное представление функции переходов автомата. Цель работы состоит в разработке и формализации такой модели и достигается путем анализа и обобщения свойств одной из известных структур МПА – автомата на счетчике.

1 ПОСТАНОВКА ЗАДАЧИ

Основной научной задачей, решаемой в данной работе, является разработка формальных математических моделей абстрактного и структурного автоматов, использующих представление функции переходов автомата в виде множества частичных функций. Модели должны обеспечивать выражение закономерностей преобразования информации в части функции переходов структурного автомата, подчеркивая эквивалентность данной функции с функцией переходов тождественного абстрактного автомата. Подтверждение состоятельности предлагаемых моделей должно быть продемонстрировано на примере одной из известных структур МПА.

2 ОБЗОР ЛИТЕРАТУРЫ

Современная теория автоматов имеет тесную связь с теорией универсальных алгебр. Алгебраический подход находит применение в теории автоматов, способствуя установлению связи теории автоматов с другими областями математики. Математический аппарат универсальных алгебр лежит в основе алгебраической теории автоматов, изучающей дискретные автоматы с абстрактно-алгебраической точки зрения [2, 3].

Базовым объектом изучения в теории универсальных алгебр является алгебраическая система, определяемая как абстрактный объект

$$G = \langle \mathcal{A}, \mathcal{F}, \mathcal{P} \rangle, \quad (1)$$

состоящий из трех множеств: непустого множества \mathcal{A} , множества операций $\mathcal{F} = \{\mathcal{F}_0, \dots, \mathcal{F}_\xi, \dots\}$ определенных на \mathcal{A} , и множества предикатов (отношений) $\mathcal{P} = \{\mathcal{P}_0, \dots, \mathcal{P}_\eta, \dots\}$ заданных на \mathcal{A} [4]. Под операцией

здесь понимается функциональное отношение любой арности, заданное на множестве \mathcal{A} . Под предикатом понимается функция на множестве \mathcal{A} , значения которой, называемые модальностями или степенями истинности, лежат в особом множестве, состоящем из двух элементов – «истина» и «ложь» [4, 5].

Множество \mathcal{A} называют носителем системы G . Набор множеств $\langle \mathcal{F}, \mathcal{P} \rangle$ образует сигнатуру или структуру алгебраической системы. В случае конечности носителя данные множества также конечны. Сигнатуру с пустым множеством отношений называют функциональной, с пустым множеством операций – предикатной [6].

Алгебраическая система (1) называется *универсальной алгеброй* или просто *алгеброй*, если ее сигнатура функциональна, и *моделью* (реляционной системой), если ее сигнатура предикатна [4]. Таким образом, алгеброй является следующий двухэлементный объект [4, 5]:

$$G = \langle \mathcal{A}, \mathcal{F} \rangle. \quad (2)$$

Иногда носитель алгебры представляется несколькими совокупностями разнородных объектов, объединение которых в единое множество является неестественным [7]. В этом случае оказывается удобным представлять носитель в виде семейства непересекающихся множеств. Пусть $\mathcal{A} = \{\mathcal{A}_1, \dots, \mathcal{A}_k\}$ – множество основ (носителей), $\mathcal{F} = \{f_1, \dots, f_m\}$ – сигнатура, причем $f_i: \mathcal{A}_1^1 \times \mathcal{A}_1^2 \times \dots \times \mathcal{A}_i^{k_i} \rightarrow \mathcal{A}_j$. Тогда система (2) называется *многоосновой* (k -основой) алгеброй [5, 7]. Многоосновная алгебра имеет несколько носителей, а каждая операция сигнатуры является соответствием, действующим из прямого произведения некоторых носителей в некоторый носитель [8].

В теории автоматов абстрактный и структурный автоматы иногда рассматриваются как алгебры, поскольку их структура как математических объектов подобна структуре алгебраической системы [9]. Представление автомата в виде алгебры дает возможность использовать алгебраические приемы при решении таких задач, как декомпозиция автоматов, их классификация, описание поведения и другие [3].

3 МАТЕРИАЛЫ И МЕТОДЫ

Дадим собственную алгебраическую интерпретацию конечных автоматов.

Абстрактный автомат.

Как известно, конечный абстрактный автомат S_A представляется совокупностью трех конечных множеств – состояний, входных и выходных слов, и двух детерминированных функций – переходов и выходов: $S_A = \langle A, Z, W, \delta, \lambda \rangle$ [10]. В таком виде автомат может рассматриваться как трехосновная алгебра с двумя операциями [7]. Будем называть алгебру, отождествляемую с абстрактным автоматом, *абстрактной алгеброй* G_A :

$$G_A = \langle \mathcal{A}_A, \mathcal{F}_A \rangle, \quad (3)$$

$$\mathcal{A}_A = \{A, Z, W\}, \quad (4)$$

$$\mathcal{F}_A = \{\delta, \lambda\}. \quad (5)$$

Фактически, выражения (3)–(5) перечисляют объекты, которые должны быть определены для задания абстрактного автомата. При этом способ задания объектов не имеет значения и может быть различным [2].

Функции δ и λ реализуют различные соответствия и могут рассматриваться отдельно друг от друга. Это оказывается полезным, например, в тех случаях, когда выходные реакции автомата неизвестны или не представляют интереса, а объектом исследования является доступная наблюдению эволюция его внутренних состояний под воздействием входных сигналов. В работах [3, 7, 11] упоминаются т.н. *автоматы без выходов* (полуавтоматы, X-автоматы, редукты) – объекты, задаваемые только с помощью множества состояний, входного алфавита и функции переходов.

Абстрактный автомат без функции выходов есть двухосновная алгебра:

$$G_\delta = \langle \mathcal{A}_\delta, \mathcal{F}_\delta \rangle = \langle \{A, Z\}, \{\delta\} \rangle, \quad (6)$$

без функции переходов – трехосновная алгебра

$$G_\lambda = \langle \mathcal{A}_\lambda, \mathcal{F}_\lambda \rangle = \langle \{A, Z, W\}, \{\lambda\} \rangle. \quad (7)$$

Выражение (7) справедливо для автомата Мили. Для автомата Мура необходимо исключить компонент Z :

$$G_\lambda = \langle \mathcal{A}_\lambda, \mathcal{F}_\lambda \rangle = \langle \{A, W\}, \{\lambda\} \rangle. \quad (8)$$

Будем называть алгебру (6) *абстрактной алгеброй переходов*, алгебры (7) и (8) – *абстрактной алгеброй выходов*. Тогда абстрактный автомат есть двойка

$$S_A = \langle G_\delta, G_\lambda \rangle. \quad (9)$$

В случае выделения на множестве состояний специального начального состояния (инициальный автомат) в выражение (9) может быть добавлен компонент $a_n \in A$:

$$S_A = \langle G_\delta, G_\lambda, a_n \rangle. \quad (10)$$

Как известно, в абстрактном автомате функция d есть соответствие

$$\delta: (D_\delta \subseteq A \times Z) \rightarrow A. \quad (11)$$

Будучи представленной в виде (11), функция δ считается, в общем случае, частично определенной на множестве $A \times Z$. Исходя из математического определения соответствия, даваемого в [12], функция (11) есть множество кортежей

$$\langle a_i, z_j, a_k \rangle \in \delta, \quad (12)$$

каждый из которых соответствует одному автоматному переходу из состояния $a_i \in A$ под воздействием входного сигнала $z_j \in Z$ в состояние $a_k \in A$.

Разобьем некоторым образом множество кортежей вида (12) на несколько непустых, попарно непересекающихся подмножеств, максимально возможное число которых равно числу переходов автомата. Каждое такое подмножество переходов автомата, будем называть *частичной функцией переходов* δ_i , понимая здесь под термином «частичная» ее неполное определение на области определения D_δ функции переходов δ :

$$\delta_i: (D_{\delta_i} \subseteq D_\delta) \rightarrow A. \quad (13)$$

В этом случае функция переходов δ есть множество подмножеств кортежей (12), то есть множество частичных функций переходов:

$$\delta = \{\delta_1, \dots, \delta_{N_\delta}\}, \quad (14)$$

где $1 \leq N_\delta \leq B$. При этом $\bigcup_{i=1}^{N_\delta} \delta_i = \delta$.

Подобное разбиение может быть выполнено и для функции выходов.

Свяжем с каждой частичной функцией переходов d_i собственную алгебру вида

$$G_{\delta_i} = \langle \mathcal{A}_{\delta_i}, \mathcal{F}_{\delta_i} \rangle = \langle \{A_{\delta_i}, Z_{\delta_i}\}, \{\delta_i\} \rangle. \quad (15)$$

В данном выражении $A_{\delta_i} \subseteq A$ есть множество всех состояний, присутствующих в кортежах вида (12) функции δ_i . Иными словами, в A_{δ_i} входят те и только те состояния, переходы из которых или переходы в которые реализуются функцией δ_i . Множество $Z_{\delta_i} \subseteq Z$ есть множество входных сигналов, присутствующих в кортежах вида (12) функции δ_i . Поскольку $A_{\delta_i} \subseteq A$, $Z_{\delta_i} \subseteq Z$ и для любых $a \in A_{\delta_i}$ и $z \in Z_{\delta_i}$ выполняется равенство $\delta_i(a, z) = \delta(a, z)$, выражение (15) есть *подалгебра* алгебры (6) [4]. Будем называть алгебру (15) *абстрактной подалгеброй переходов*. В алгебраической теории автоматов понятию подалгебры соответствует понятие *подавтомата* [2, 3].

Множество частичных функций переходов приводит ко множеству абстрактных подалгебр переходов, отождествляемому с абстрактной алгеброй переходов (6):

$$G_\delta = \{G_{\delta_1}, \dots, G_{\delta_{N_\delta}}\}. \quad (16)$$

Компоненты \mathcal{A}_δ и \mathcal{F}_δ алгебры G_δ формируются путем объединения соответствующих компонентов всех подалгебр:

$$\mathcal{A}_\delta = \bigcup_{i=1}^{N_\delta} \mathcal{A}_{\delta_i} = \left\{ \bigcup_{i=1}^{N_\delta} A_{\delta_i}, \bigcup_{i=1}^{N_\delta} Z_{\delta_i} \right\}, \quad \mathcal{F}_\delta = \bigcup_{i=1}^{N_\delta} \mathcal{F}_{\delta_i} = \left\{ \bigcup_{i=1}^{N_\delta} \delta_i \right\}.$$

Отметим, что в то время как частичные функции δ_i образуют *разбиение*, т.е. $\forall(i, j \in [1, N_\delta], i \neq j) : \delta_i \cap \delta_j = \emptyset$, семейства множеств A_{δ_i} и Z_{δ_i} образуют на множествах A и Z соответствующие *покрытия*, что в общем случае допускает $\forall(i, j \in [1, N_\delta]) : A_{\delta_i} \cap A_{\delta_j} \neq \emptyset$ и $Z_{\delta_i} \cap Z_{\delta_j} \neq \emptyset$. Присутствие одного и того же состояния $a \in A$ в носителях нескольких подалгебр означает, что переходы в данное состояние и/или из него выполняются с помощью разных частичных функций δ_i .

По аналогии с (16), в виде множества подалгебр представляются абстрактная алгебра выходов (7) автомата Мили и абстрактная алгебра выходов (8) автомата Мура.

Структурный автомат.

Модель структурного автомата $S_S = \langle X, Y, A, d, l \rangle$ также является трехосновной алгеброй, носитель \mathcal{A}_S которой образован конечными множествами структур-

ного автомата, а сигнатура \mathcal{F}_S – структурной функцией переходов d и структурной функцией выходов l :

$$\mathcal{A}_S = \{X, A, Y\}, \quad (17)$$

$$\mathcal{F}_S = \{d, l\}. \quad (18)$$

Отметим, что в выражении (17) множество состояний A отличается от одноименного множества в выражении (4) тем, что содержит не абстрактные, а структурные состояния, представленные своими структурными кодами [10].

Назовем алгебру, отождествляемую со структурным автоматом, *структурной алгеброй* G_S :

$$G_S = \langle \mathcal{A}_S, \mathcal{F}_S \rangle = \langle \{X, A, Y\}, \{d, l\} \rangle. \quad (19)$$

Разделение функций переходов и выходов дает две отдельные алгебры: *структурную алгебру переходов*:

$$G_d = \langle \mathcal{A}_d, \mathcal{F}_d \rangle = \langle \{A, X\}, \{d\} \rangle \quad (20)$$

и *структурную алгебру выходов* (выражение (21) – для автомата Мили, (22) – для автомата Мура):

$$G_l = \langle \mathcal{A}_l, \mathcal{F}_l \rangle = \langle \{A, X, Y\}, \{l\} \rangle, \quad (21)$$

$$G_l = \langle \mathcal{A}_l, \mathcal{F}_l \rangle = \langle \{A, Y\}, \{l\} \rangle. \quad (22)$$

При этом структурный автомат может рассматриваться как двойка

$$S_S = \langle G_d, G_l \rangle. \quad (23)$$

Согласно модели $S_S = \langle X, Y, A, d, l \rangle$, функция d есть соответствие

$$d : (D_d \subseteq X \times A) \rightarrow A. \quad (24)$$

В структурном автомате все элементы множеств абстрактного автомата представляются в виде слов в структурном (двоичном) алфавите. Предположение о том, что память структурного автомата строится в базисе триггеров D-типа, дает возможность отождествлять результат структурной функции переходов (структурный код состояния перехода) со множеством функций возбуждения элементов памяти структурного автомата.

При $|X| = L$ и $\lceil \log_2(|A|) \rceil = R$ выражение (24) принимает вид:

$$d : (D_d \subseteq x_1 \times x_2 \times \dots \times x_L \times e_1 \times e_2 \times \dots \times e_R) \rightarrow e_1 \times e_2 \times \dots \times e_R, \quad (25)$$

где каждый из элементов x и e есть множество букв структурного (двоичного) алфавита, т.е. множество $\{0, 1\}$ [9]. Фактически, (25) есть соответствие:

$$d : (D_d \subseteq \{0, 1\}^{L+R}) \rightarrow \{0, 1\}^R, \quad (26)$$

элементами которого являются кортежи, состоящие из $(L+2R)$ букв $q \in \{0, 1\}$:

$$\langle q_1, \dots, q_{L+2R} \rangle \in D_d. \quad (27)$$

Структурная функция переходов (24), являясь множеством кортежей (27), может быть представлена в виде множества *частичных структурных функций* переходов:

$$d = \{d_1, \dots, d_{N_d}\}, \quad (28)$$

$$d_i : (D_{d_i} \subseteq D_d) \rightarrow A. \quad (29)$$

Отметим, что в выражении (28) максимально возможное количество частичных функций N_d определяется числом кортежей (27). Если в случае абстрактного автомата имеем $1 \leq N_d \leq B$, то для значения N_d дело обстоит иначе.

В процессе кодирования каждому символу z_f входного алфавита абстрактного автомата ставится в соответствие вектор вида $\langle x_1, \dots, x_L \rangle$, образованный значениями переменных x_1, \dots, x_L , соответствующих одноименным входным структурным сигналам. Поскольку способ кодирования элементов множества Z может быть произвольным, возможны ситуации, когда в векторе $\langle x_1, \dots, x_L \rangle$, соответствующем входному символу $z_f \in Z$, некоторая переменная x_l может принимать произвольное значение [13]. В этом случае каждому переходу, выполняющемуся под воздействием входного сигнала z_f , в структурной функции d соответствуют два вектора вида (27), в первом из которых компонент, соответствующий x_l , равен нулю, во втором – единице.

В общем случае количество фиктивных переменных в коде символа z_f может быть больше единицы. Такие ситуации часто возникают в процессе структурного синтеза МПА, заданных в виде ГСА, когда переход из состояния в состояние зависит лишь от части логических условий, отождествляемых со структурными входными сигналами. В работе [13] предлагается каждый структурный входной сигнал условно представлять заданным на трехзначном множестве $\{0, 1, -\}$, где элемент «-» соответствует произвольному значению из $\{0, 1\}$. Будем полагать, что каждая из функций d_i есть множество кортежей вида (27) с трехзначным представлением структурных входных сигналов. Тогда, как и для абстрактных частичных функций, имеем $1 \leq N_d \leq B$. Понятно, что при $N_d = B$ каждому переходу автомата будет сопоставлена отдельная функция вида (29), в то время как при $N_d = 1$ все переходы автомата реализуются одной общей структурной функцией переходов.

Установим теперь формальную связь между абстрактной и структурной алгебрами переходов. Однотипность алгебраических систем характеризуется возможностью установления между ними различных отображений, таких как *гомоморфизм*, *изоморфизм* и др. [4]. Воспользуемся понятием *гомоморфизма автоматов*, которое В. М. Глушков определяет следующим образом [11]. Гомоморфизм Ψ автомата $A(Z_A, A_A, W_A, \delta_A, \lambda_A)$ в автомат $B(Z_B, A_B, W_B, \delta_B, \lambda_B)$ есть совокупность трех однозначных отображений: $\psi_1 : Z_A \rightarrow Z_B$, $\psi_2 : A_A \rightarrow A_B$, $\psi_3 : W_A \rightarrow W_B$, удовлетворяющих для любых элементов $a \in A_A$ и $z \in Z_A$ соотношениям:

$$\psi_1(\delta_A(a, z)) = \delta_B(\psi_1(a), \psi_2(z)), \quad (30)$$

$$\psi_3(\lambda_A(a, z)) = \lambda_B(\psi_1(a), \psi_2(z)). \quad (31)$$

Хотя МПА в части функции переходов может быть задан как абстрактной алгеброй (3), так и структурной алгеброй (19), установление гомоморфизма между данными алгебрами, с формальной точки зрения, невозможно. Причиной является несоответствие носителей (4) и

(17), а точнее – различные мощности множеств, образующих носители. Тот факт, что в общем случае $(|Z| = F) \neq (|X| = L)$ и $(|W| = G) \neq (|Y| = N)$, не позволяет установить между элементами соответствующих множеств взаимно однозначное соответствие, обязательное для гомоморфизма.

Преобразуем структурную алгебру (19) следующим образом.

1. Пусть носитель \mathcal{A}_S содержит следующие три множества

– множество $K_S(Z) = \{K_S(z_1), \dots, K_S(z_F)\}$ структурных кодов входных сигналов абстрактного автомата, где каждый элемент $K_S(z_f)$ есть вектор $\langle x_1, \dots, x_L \rangle$ значений структурных входных сигналов, определенных на трехэлементном алфавите $\{0, 1, -\}$;

– множество $K_S(A) = \{K_S(a_1), \dots, K_S(a_M)\}$ структурных кодов состояний абстрактного автомата, заданных векторами $\langle e_1, \dots, e_R \rangle$, $e_r \in \{0, 1\}$;

– множество $K_S(W) = \{K_S(w_1), \dots, K_S(w_G)\}$ структурных кодов выходных сигналов абстрактного автомата, заданных векторами $\langle y_1, \dots, y_N \rangle$, $y_n \in \{0, 1\}$.

2. Структурные функции переходов d и выходов l , образующие сигнатуру \mathcal{F}_S , определим выражениями (32) и (33), (34) соответственно:

$$d : (D_d \subseteq K_S(Z) \times K_S(A)) \rightarrow K_S(A). \quad (32)$$

$$l : (D_l \subseteq K_S(Z) \times K_S(A)) \rightarrow K_S(W). \quad (33)$$

$$l : (D_l \subseteq K_S(A)) \rightarrow K_S(W). \quad (34)$$

Выражение (33) справедливо для автомата Мили, (34) – для автомата Мура.

Тогда структурная алгебра (19) принимает следующий вид:

$$G_S = \langle \mathcal{A}_S, \mathcal{F}_S \rangle = \langle \{K_S(Z), K_S(A), K_S(W)\}, \{d, l\} \rangle. \quad (35)$$

Структурные алгебры переходов и выходов определяются аналогично (20)–(22) следующим образом:

$$G_d = \langle \mathcal{A}_d, \mathcal{F}_d \rangle = \langle \{K_S(A), K_S(Z)\}, \{d\} \rangle, \quad (36)$$

$$G_l = \langle \mathcal{A}_l, \mathcal{F}_l \rangle = \langle \{K_S(A), K_S(Z), K_S(W)\}, \{l\} \rangle, \quad (37)$$

$$G_l = \langle \mathcal{A}_l, \mathcal{F}_l \rangle = \langle \{K_S(A), K_S(W)\}, \{l\} \rangle. \quad (38)$$

Структурная алгебра (35) отличается от (19) тем, что формальными аргументами и значениями ее операций являются не отдельные структурные сигналы, а элементы носителей абстрактной алгебры, представленные в виде векторов элементарных структурных сигналов. Это позволяет установить гомоморфизм абстрактной алгебры (3) в структурную алгебру (35). Поскольку между носителями структурной алгебры (35) и носителями абстрактной алгебры (3) могут быть установлены однозначные соответствия $K_S(A) \rightarrow A$, $K_S(Z) \rightarrow Z$, $K_S(W) \rightarrow W$, гомоморфизм алгебр (3) и (35) является *изоморфизмом* (взаимно однозначным гомоморфизмом, [11]). Очевидно, что существуют также изоморфизмы между алгебрами переходов (6) и (36), а также между алгебрами выходов: (7) и (37) для автомата Мили, (8) и (38) для автомата Мура.

Функция переходов (32) есть множества кортежей следующего вида:

$$\langle K_S(a_i), K_S(z_j), K_S(a_k) \rangle \in D_d \quad (39)$$

и может быть представлена в виде семейства частичных функций переходов (28), причем

$$d_i : (D_{d_i} \subseteq D_d) \rightarrow K_S(A). \quad (40)$$

Изоморфизм алгебр (3) и (35) устанавливает, в том числе, отображение между элементами соответствующих носителей данных алгебр. Сопоставляя каждому компоненту любого кортежа (12) соответствующий ему элемент из носителей структурной алгебры, можно получить тождественный кортеж (39). Следовательно, каждой частичной функции переходов δ_i абстрактного автомата, определяемой выражением (13) и являющейся некоторым подмножеством множества кортежей (12), может быть поставлена в соответствие частичная функция переходов d_i , определяемая выражением (40) и являющаяся подмножеством множества кортежей (39). При этом изоморфизм абстрактной и структурной алгебр возможен при любом разбиении функций абстрактного автомата на множества частичных функций. Данное утверждение также справедливо как алгебр переходов (6) и (36) и для алгебр выходов (7)/(8) и (37)/(38).

По аналогии с (15), *структурной подалгеброй переходов* будем называть алгебру вида:

$$G_{d_i} = \langle \mathcal{A}_{d_i}, \mathcal{F}_{d_i} \rangle = \langle \{K_S(A_{d_i}), K_S(Z_{d_i})\}, \{d_i\} \rangle. \quad (41)$$

Здесь $K_S(A_{d_i}) \subseteq K_S(A)$ – множество структурных кодов состояний, присутствующих в кортежах (39) частичной структурной функции d_i ; $K_S(Z_{d_i}) \subseteq K_S(Z)$ – множество структурных кодов входных сигналов, используемых для кодирования входных сигналов в кортежах (39) функции d_i . Структурную алгебру переходов (35), по аналогии с (16), представим в виде множества структурных подалгебр переходов:

$$G_d = \{G_{d_1}, \dots, G_{d_{N_d}}\}. \quad (42)$$

Как и в случае абстрактного автомата, компоненты структурной алгебры переходов (42) формируются объединением элементов соответствующих компонентов своих подалгебр. Аналогичным образом в виде множества своих подалгебр могут быть представлены структурные алгебры выходов (37) и (38).

4 ЭКСПЕРИМЕНТЫ

Рассмотрим отдельно функцию переходов автомата. В соответствии с предложенным В.М. Глушковым каноническим методом структурного синтеза функция переходов представляется в векторном виде и задается системой канонических скалярных уравнений [10]. Система скалярных уравнений строится известным способом [10, 13], при котором для каждой пары $\langle K_S(a_i), K_S(z_j) \rangle$ кортежа (39), соответствующего одному переходу автомата, формируется терм, присутствие которого в том или ином уравнении системы определяется элементом

В структурной теории автоматов известен класс устройств, называемых автоматами на счетчике или С-автоматами [14]. В их основе лежит принцип специального кодирования состояний, при котором на множестве состояний заданной ГСА выделяются т.н. линейные последовательности состояний, внутри которых коды состояний задаются в естественном (последовательном) порядке.

Применим принцип представления функции переходов автомата в виде множества частичных функций к автомату на счетчике. Пусть произвольный абстрактный автомат задан алгеброй (3), а реализующий его автомат на счетчике – алгеброй (35). Функции переходов данных автоматов образуют сигнатуры абстрактной алгебры переходов (6) и структурной алгебры переходов (36) соответственно. В силу изоморфизма алгебр (3) и (35), алгебры переходов (6) и (36) также изоморфны:

$$G_\delta \leftrightarrow G_S. \quad (43)$$

Тот факт, что в С-автомате часть переходов реализуется с использованием инкрементного счетчика, часть – в виде системы канонических уравнений, позволяет представить функцию d в виде двух частичных функций d_1 и d_2 , образующих сигнатуры двух подалгебр переходов:

$$G_{d_1} = \langle \mathcal{A}_{d_1}, \mathcal{F}_{d_1} \rangle = \langle \{K_S(A_{d_1}), K_S(Z_{d_1})\}, \{d_1\} \rangle; \quad (44)$$

$$G_{d_2} = \langle \mathcal{A}_{d_2}, \mathcal{F}_{d_2} \rangle = \langle \{K_S(A_{d_2}), K_S(Z_{d_2})\}, \{d_2\} \rangle. \quad (45)$$

Представление функции d в виде двух частичных функций предполагает аналогичное представление абстрактной функции переходов δ . Частичные абстрактные функции переходов δ_1 и δ_2 образуют сигнатуры двух абстрактных подалгебр переходов:

$$G_{\delta_1} = \langle \mathcal{A}_{\delta_1}, \mathcal{F}_{\delta_1} \rangle = \langle \{A_{\delta_1}, Z_{\delta_1}\}, \{\delta_1\} \rangle; \quad (46)$$

$$G_{\delta_2} = \langle \mathcal{A}_{\delta_2}, \mathcal{F}_{\delta_2} \rangle = \langle \{A_{\delta_2}, Z_{\delta_2}\}, \{\delta_2\} \rangle. \quad (47)$$

Из изоморфизма (43) вытекают изоморфизмы соответствующих подалгебр:

$$G_{\delta_1} \leftrightarrow G_{d_1}; \quad (48)$$

$$G_{\delta_2} \leftrightarrow G_{d_2}. \quad (49)$$

Структурная схема С-автомата, отражающая представление структурной функции переходов в виде двух частичных функций, приведена на рис. 1.

Здесь схема KC_{d_1} реализует частичную структурную функцию переходов d_1 и представляет собой R -разрядный инкрементор. Структурный код T текущего состояния, представленный набором структурных переменных $\langle T_1, \dots, T_R \rangle$, в котором $T_r \in \{0, 1\}$, интерпретируется на входе KC_{d_1} как R -разрядное беззнаковое целое, представленное в структурном (двоичном) алфавите. На выходе инкрементора формируется структурный код состояния перехода, обозначенный символом d_1 и отождествляемый со значением одноименной функции переходов.

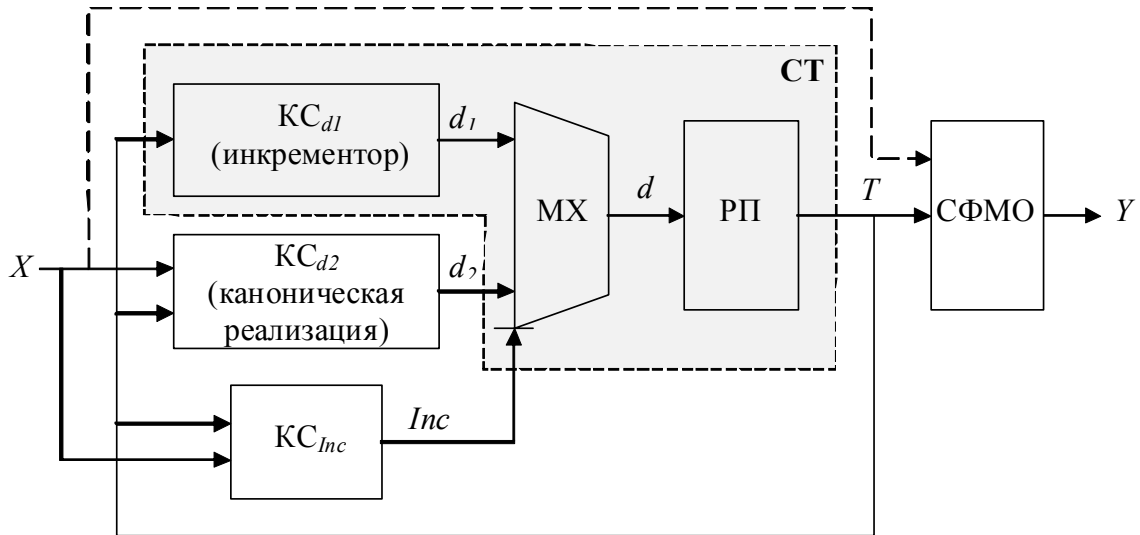


Рисунок 1 – Структурная схема С-автомата

Частичная функция d_2 представлена комбинационной схемой $КС_{d_2}$. Данная схема строится по известной методике [14] и реализует систему канонических уравнений, соответствующую функции d_2 . На вход $КС_{d_2}$ поступают структурный код текущего состояния $T = \langle T_1, \dots, T_R \rangle$ и структурный код входного слова, представленный набором $X = \langle x_1, \dots, x_L \rangle$, в котором $x_l \in \{0, 1, -\}$. Код, формируемый на выходе $КС_{d_2}$, обозначен символом d_2 и отождествляется с значением одноименной функции переходов.

Поскольку каждая функция d_i , представленная в виде логической схемы, определена на всех наборах аргументов, но при этом отдельный автоматный переход реализуется всегда какой-то одной частичной функцией, на этапе структурного синтеза возникает задача организации выбора нужной частичной функции при каждом переходе автомата. В С-автомате данная задача решается мультиплексированием одновременно формируемых значений двух частичных функций. На рис. 1 схема $КС_{Inc}$ вырабатывает сигнал Inc , управляющий мультиплексором MX . Формируемый на выходе мультиплексора структурный код d состояния перехода отождествляется со значением структурной функции переходов автомата и поступает в регистр памяти $РП$. Отметим, что в структуре С-автомата, приведенной в [14], блокам $КС_{d_1}$, MX и $РП$ соответствует инкрементный счетчик $СТ$.

Увеличение числа переходов, реализуемых функцией d_1 , приводит, в общем случае, к снижению аппаратных затрат в схеме формирования переходов С-автомата по сравнению с канонической структурой МПА. В связи с этим С-автоматы рекомендуется использовать для линейных ГСА, в которых не менее 75% вершин являются операторными [14].

Не смотря на то, что каждая частичная функция реализует собственное подмножество переходов, одно и то же состояние автомата может встречаться в кортежах нескольких частичных функций. Например, переход в

состояние a_k может выполняться с помощью частичной функции δ_1 , переход из этого состояния – с помощью частичной функции δ_2 . Если изоморфизмы (48) и (49) задавать независимо друг от друга, возможны ситуации, когда один и тот же структурный код присвоен двум различным состояниям, либо одному и тому же состоянию присвоено два различных структурных кода. Сказанное относится также и к структурным кодам входных сигналов. Подобные неоднозначности делают невозможным последующий синтез логической схемы автомата.

Решение данной проблемы заключается во *взаимосвязанном задании изоморфизмов* (48) и (49), при котором:

- каждое состояние $a_i \in A$ получает единственный и уникальный структурный код $K_S(a_i)$ в соответствующих носителях всех структурных подалгебр, в которых оно встречается;

- каждый входной сигнал $z_j \in Z$ получает единственный и уникальный структурный код $K_S(z_j)$ в соответствующих носителях всех структурных подалгебр, в которых он встречается.

В алгебраической форме данные требования могут быть выражены требованием существования изоморфизма (43). Данный изоморфизм устанавливает, в том числе, взаимно однозначные соответствия между каждым состоянием $a_i \in A$ и его структурным кодом $K_S(a_i) \in K_S(A)$, а также между каждым входным сигналом $z_j \in Z$ и его структурным кодом $K_S(z_j) \in K_S(Z)$. В результате за каждым состоянием и входным сигналом закрепляется единственный структурный код независимо от количества используемых подалгебр.

Применяемая в С-автомате операция инкремента является арифметической операцией и определена для скалярных величин. В процессе синтеза С-автомата на этапе формирования линейных последовательностей состояний каждое состояние $a_i \in A_{\delta_1}$ получает уникальный «скалярный» код $K_I(a_i)$ из множества натуральных чисел. «Скалярные» коды назначаются всем состояниям, переходы в которые или переходы из которых

выполняются с помощью операции инкремента. Таким образом, результатом кодирования всех состояний множества A_{δ_1} является множество скалярных кодов $K_I(A_{\delta_1})$. В то же время состояния, отсутствующие в кортежах функции δ_1 , «скалярных» кодов не получают.

Обозначим операцию инкремента символом O_{Inc} . Заметим, что множество $K_I(A_{\delta_1})$ совместно с операцией O_{Inc} можно рассматривать как алгебру

$$G_I = \langle \mathcal{A}_I, \mathcal{F}_I \rangle = \langle K_I(A_{\delta_1}), O_{Inc} \rangle. \quad (50)$$

Здесь операция O_{Inc} формально представляется множеством кортежей вида $\langle K_I(a_i), K_I(a_j) \rangle \in O_{Inc}$. Компонент $K_I(a_i) \in K_I(A_{\delta_1})$ есть «скалярный» код исходного состояния, компонент $K_I(a_j) \in K_I(A_{\delta_1})$ – «скалярный» код состояния перехода. Очевидно, что подобный кортеж не содержит информации о входном сигнале, под воздействием которого выполняется переход, и в таком виде не может быть поставлен в соответствие кортежам частичных функций δ_1 и d_1 .

Введем в алгебру (50) дополнительный носитель Z_{δ_1} , идентичный одноименному носителю из абстрактной подалгебры G_{δ_1} . В результате получим двухосновную алгебру:

$$G_I = \langle \mathcal{A}_I, \mathcal{F}_I \rangle = \langle \{K_I(A_{\delta_1}), Z_{\delta_1}\}, \{O_{Inc}\} \rangle, \quad (51)$$

в которой операция O_{Inc} задается множеством кортежей вида:

$$\langle K_I(a_i), z_j, K_I(a_k) \rangle \in O_{Inc}, \quad (52)$$

где $K_I(a_i), K_I(a_j) \in K_I(A_{\delta_1})$, $z_j \in Z_{\delta_1}$.

Теперь, учитывая возможность установления попарных взаимно однозначных соответствий между носителями A_{δ_1} , $K_I(A_{\delta_1})$ и $K_S(A_{d_1})$, а также между носителями Z_{δ_1} и $K_S(Z_{d_1})$, каждому кортежу вида (12) в функции δ_1 и соответствующему ему кортежу вида (39) в функции d_1 может быть поставлен в соответствие единственный кортеж вида (52). Это позволяет задать операцию O_{Inc} таким множеством кортежей (52), чтобы в рамках алгебры (51) она выполняла преобразования «скалярных» кодов состояний по тому же закону, что и функции δ_1 и d_1 в своих подалгебрах, реализуя, таким образом, те же автоматные переходы, что и функции δ_1 и d_1 .

Все это позволяет говорить о взаимном изоморфизме трех алгебр: алгебры (51), абстрактной подалгебры G_{δ_1} и структурной подалгебры G_{d_1} :

$$G_{\delta_1} \leftrightarrow G_I \leftrightarrow G_{d_1}. \quad (53)$$

Данное выражение содержит три изоморфизма: $G_{\delta_1} \leftrightarrow G_{d_1}$, $G_{\delta_1} \leftrightarrow G_I$ и $G_I \leftrightarrow G_{d_1}$. Учитывая требование их одновременного существования, будем для краткости рассматривать данные изоморфизмы как единый изоморфизм (53).

Если структурные коды состояний, встречающиеся в кортежах частичной функции d_2 , могут задаваться произвольно (из допустимого множества структурных кодов), то в случае функции d_1 структурные коды должны задаваться таким образом, чтобы сделать возможным их преобразование с помощью операции инкремента. Задание «скалярных» кодов состояний в естественном (инкрементном) порядке есть задание транзитивного замыкания для отношения инкремента на некотором множестве скалярных чисел. На произвольном множестве структурных кодов, для которых неизвестна их «скалярная» интерпретация, отношение инкремента не может быть определено. По этой причине выбор «скалярных» кодов состояний множества A_{δ_1} должен предшествовать выбору их структурных кодов. В свою очередь, задание структурных кодов состояний множества A_{δ_2} выполняется после задания структурных кодов состояний множества A_{δ_1} , а синтез схем KC_{d_2} и KC_{Inc} возможен лишь после задания структурных кодов всех состояний автомата.

5 РЕЗУЛЬТАТЫ

Применение принципа представления функции переходов автомата в виде множества частичных функций позволяет, в случае автомата на счетчике, сформировать следующий порядок задания алгебр:

1. Задание абстрактных подалгебр переходов (46) и (47) в соответствии со сформированными линейными последовательностями состояний.
2. Задание алгебры (51), изоморфной подалгебре (46).
3. Задание структурной подалгебры (44), изоморфной алгебре (51).
4. Задание структурной подалгебры (45), изоморфной подалгебре (44).

Учитывая то, что алгебра (51) играет в С-автомате вспомогательную роль при построении структурной алгебры переходов по заданной абстрактной алгебре переходов, назовем алгебру G_I *промежуточной алгеброй переходов*, понимая под нижним индексом «I» термин «intermediate» – «промежуточный». «Скалярные» коды состояний, образующие носитель промежуточной алгебры, назовем *промежуточными кодами состояний*. Операцию, образующую сигнатуру промежуточной алгебры, назовем *операцией переходов*.

В алгебраической форме эквивалентность абстрактного автомата и реализующего его структурного С-автомата выражается существованием следующих изоморфизмов:

1. Изоморфизм (53) промежуточной алгебры G_I с соответствующими ей абстрактной и структурной подалгебрами G_{δ_1} и G_{d_1} . Данный изоморфизм выражает реализацию части переходов автомата с помощью операции инкремента.
2. Изоморфизм (43) абстрактной и структурной алгебр переходов. Данный изоморфизм обеспечивается совместным заданием изоморфизмов (48) и (49), при котором сохраняются уникальность и однозначность структурных кодов состояний и входных сигналов независимо от способа разбиения функции переходов на частичные функции.

Если хотя бы один из данных изоморфизмов не может быть задан, то не может быть построен С-автомат, эквивалентный заданному абстрактному автомату. Таким образом, каждый из изоморфизмов является необходимым условием существования С-автомата, а достаточным условием является одновременное существование обоих изоморфизмов. Сведем изоморфизмы (53) и (43) в систему (54), которую будем рассматривать в качестве *математической модели автомата на счетчике*:

$$\begin{cases} G_{\delta_1} \leftrightarrow G_I \leftrightarrow G_{d_1}; \\ G_{\delta} \leftrightarrow G_d. \end{cases} \quad (54)$$

Процесс построения системы (54) включает выполнение следующих этапов структурного синтеза С-автомата: формирование линейных последовательностей состояний, кодирование состояний, формирование таблицы переходов [14].

После задания системы (54) следует завершающий этап структурного синтеза – реализация в заданном элементном базисе логической схемы автомата (комбинационных схем KC_{d_2} , KC_{Inc} и схемы формирования микроопераций) в соответствии со сформированными множеством структурных кодов состояний и разбиением функции переходов на частичные функции. Таким образом, метод структурного синтеза С-автомата может быть сокращенно представлен двумя этапами:

1. Задание системы изоморфизмов (54).

2. Реализация логической схемы автомата в заданном элементном базисе.

6 ОБСУЖДЕНИЕ

Представление функции переходов С-автомата в виде двух частичных функций наряду с использованием промежуточной алгебры переходов отражает совмещение в одной структуре двух различных способов реализации переходов автомата – канонического и инкрементного. Разбиение функции переходов на частичные функции в случае канонической структуры МПА не имеет смысла, поскольку в этом случае каждая частичная функция будет лишь реализовывать некоторый фрагмент единой системы канонических уравнений, а требуемая при этом реализация выбора частичной функции путем мультиплексирования приведет к увеличению аппаратных затрат в схеме устройства.

Приведенный пример использования промежуточной алгебры на базе операции инкремента позволяет предположить возможность формирования сигнатуры промежуточной алгебры из других операций, например, операции декремента. Подобный подход может способствовать снижению аппаратных затрат в схеме проектируемого микропрограммного автомата за счет использования в схеме формирования переходов таких комбинационных схем, аппаратные затраты в которых не зависят или зависят незначительно от количества реализуемых автоматных переходов.

ВЫВОДЫ

В настоящей работе предложены новые математические модели абстрактного и структурного автоматов, в основе которых лежит представление функции переходов в виде множества частичных функций. Результатом исследований является представление процессов преобразования кодов состояний в структурном автомате в виде промежуточной алгебры переходов, описывающей закон преобразования кодов состояний для некоторого подмножества автоматных переходов.

Использование различных операций переходов позволяет выделять на функции переходов структурного автомата различные законы преобразования кодов состояний, что может способствовать снижению аппаратных затрат в логической схеме автомата в сравнении с другими известными структурами МПА. Вопрос целесообразности применения предложенного подхода для оптимизации аппаратных затрат в логической схеме микропрограммного автомата не является очевидным из предложенных моделей и требует самостоятельного исследования.

СПИСОК ЛИТЕРАТУРЫ

1. Баранов С. И. Синтез микропрограммных автоматов / С. И. Баранов. – Л. : Энергия, 1979. – 232 с.
2. Алгебраическая теория автоматов, языков и полугрупп / под ред. М. Арбиба ; пер. с англ. – М. : Статистика, 1975. – 335 с.
3. Плоткин Б. И. Элементы алгебраической теории автоматов: учеб. пособие для вузов / Б. И. Плоткин, Л. Я. Гринглаз, А. А. Гварамия. – М. : Высшая школа, 1994. – 191 с.
4. Мальцев А. И. Алгебраические системы / А. И. Мальцев. – М. : Наука, 1970. – 392 с.
5. Плоткин Б. И. Универсальная алгебра, алгебраическая логика и базы данных / Б. И. Плоткин. – М. : Наука, гл. ред. физ.-мат. лит., 1991. – 448 с.
6. Судоплатов С. В. Элементы дискретной математики : учебник / С. В. Судоплатов, Е. В. Овчинникова. – М. : ИНФРА-М, Новосибирск : Изд-во НГТУ, 2002. – 280 с.
7. Богомолов А. М. Алгебраические основы теории дискретных систем / А. М. Богомолов, В. Н. Салий. – М. : Наука-Физматлит, 1997. – 368 с.
8. Новиков Ф. А. Дискретная математика для программистов / Ф. А. Новиков. – СПб. : Питер, 2000. – 304 с.
9. Кудрявцев В. Б. Введение в теорию автоматов / В. Б. Кудрявцев, С. В. Алешин, А. С. Подколзин. – М. : Наука, 1985. – 320 с.
10. Глушков В. М. Синтез цифровых автоматов / В. М. Глушков. – М. : Физматгиз, 1962. – 476 с.
11. Глушков В. М. Абстрактная теория автоматов / В. М. Глушков // Успехи математических наук. – 1961. – Т. XVI, Вып. 5. – С. 3–62.
12. Трахтенброт Б. А. Конечные автоматы (поведение и синтез) / Б. А. Трахтенброт, Я. М. Бардзинь. – М. : Наука, 1970. – 400 с.
13. Шиханович Ю. А. Введение в современную математику (начальные понятия) / Ю. А. Шиханович. – М. : Наука, 1965. – 376 с.
14. Баркалов А. А. Синтез устройств управления на программируемых логических устройствах / А. А. Баркалов. – Донецк : ДонНТУ, 2002. – 262 с.

Статья поступила в редакцию 24.09.2015.

После доработки 05.10.2015.

Бабаков Р. М.

Канд. техн. наук, доцент, доцент кафедри комп'ютерних наук Донецького національного технічного університету, м. Красноармійськ, Україна

ПРОМІЖНА АЛГЕБРА ПЕРЕХОДІВ В МІКРОПРОГРАМНОМУ АВТОМАТІ

Вирішено задачу формалізації завдання мікропрограмного автомата, в структурі якого частина автоматних переходів реалізується неканонічним шляхом. Запропоновано новий підхід до організації функції переходів мікропрограмного автомата, відповідно до якого функція переходів представляється у вигляді сімейства часткових функцій, кожна з яких визначена лише на частині області визначення функції переходів автомата і відповідає певній підмножині автоматних переходів.

З урахуванням запропонованого підходу традиційне представлення автомата у вигляді багатоосновної алгебри матиме певні зміни. По-перше, взаємна незалежність функцій переходів і виходів, що утворюють сигнатуру алгебри, дозволяє розглядати їх окремо одна від іншої, що призводить до представлення автомата у вигляді двох алгебр: алгебри переходів, сигнатура якої містить лише функцію переходів, і алгебри виходів, сигнатура якої містить лише функцію виходів. По-друге, представлення функції переходів у вигляді множини часткових функцій виводить до заміни алгебри переходів множиною підалгебр переходів, в кожній з яких сигнатура утворена частковою функцією переходів.

На прикладі мікропрограмного автомата з лічильником показано, що закон перетворення кодів станів в рамках певної підмножини переходів може бути заданий алгебраїчною функцією (операцією переходів), що використовує скалярну інтерпретацію кодів станів структурного автомата. Скалярну інтерпретацію кодів станів разом з операцією переходів пропонується представляти у вигляді так званої проміжної алгебри переходів, яка ізоморфна відповідним підалгебрам переходів абстрактного і еквівалентного йому структурного автоматів.

Ключові слова: мікропрограмний автомат, часткова функція переходів, проміжна алгебра переходів, автомат із лічильником.

Babakov R. M.

PhD, Associate Professor of the Department of Computer Sciences of the Donetsk National Technical University, Krasnoarmiysk, Ukraine

INTERMEDIATE ALGEBRA OF TRANSITIONS IN MICROPROGRAM FINAL-STATE MACHINE

The problem of formalization of representation of final-state machine, where the part of automaton transition is realized in non-canonical way, is solved. A new approach for organization of the function of transitions of the final-state machine is proposed. According to it the function of transitions is represented as a family of partial functions, each of which is defined only on the part of the domain of the function of transitions, and corresponds to a subset of the automaton transitions.

According to the proposed approach the traditional representation of the final-state machine as a polybasic algebra is changed. First, the mutual independence of the function of transitions and function of outputs that form the signature of algebra, allows us to consider them separately from each other. This way leads to presentation of the final-state machine as two algebras: the algebra of transitions whose signature contains only the function of transitions and algebra of outputs whose signature contains only the function of outputs. Second, the representation of the function of transitions in the form of a set of partial functions leads to the replacement of the algebra of transitions by the set of subalgebras of transitions, a signature of each of which is formed by partial function of transitions.

The example of the final-state machine with a counter shows that the law of transformation of codes of states within a certain subset of transitions can be set by an algebraic function (operation of transitions) using scalar interpretation of codes of states of the structural final-state machine. The representation of scalar interpretation of codes of states and the operation of transitions as so-called intermediate algebra of transitions isomorphic to both according subalgebras of transitions of the abstract and equivalent structural final-state machines is proposed.

Keywords: microprogram final-state machine, partial function of transitions, intermediate algebra of transitions, final-state machine with counter.

REFERENCES

1. Baranov S. I. Sintez mikroprogramnih avtomatov. Leningrad, Energiya, 1979, 232 p.
2. Algebraicheskaya teoriya avtomatov, yazikov i polugrupp. Pod red. M. Arbiba. Per. s angl. Moscow, Statistika, 1975, 335 p.
3. Plotkin B. I., Gringlaz L. Y., Gvaramiya A. A. Elementi algebraicheskoy teorii avtomatov: ucheb. posobie dlya vuzov. Moscow, Vish. shk., 1994, 191 p.
4. Maltsev A. I. Algebraicheskie sistemi. Moscow, Nauka, 1970, 392 p.
5. Plotkin B. I. Universalnaya algebra, algebraicheskaya logika i bazi danih. Moscow, Nauka. Gl. red. fiz.-mat. lit., 1991, 448 p.
6. Sudoplatov S. V., Ovchinnikova E. V. Elementi diskretnoy matematiki: uchebnik. Moscow, INFRA-M, Novosibirsk, Izdatelstvo NGTU, 2002, 280 p.
7. Bogomolov A. M., Saliy V. N. Algebraicheskie osnovi teorii diskretnih sistem. Moscow, Nauka, Fizmatlit, 1997, 368 p.
8. Novikov F. A. Diskretnaya matematika dlya programmistov. SPb, Piter, 2000, 304 p.
9. Kudryavtsev V. B., Alyoshin S. V., Podkolzin A. S. Vvedenie v teoriyu avtomatov. Moscow, Nauka, 1985, 320 p.
10. Glushkov V. M. Sintez tsifrovih avtomatov. Moscow, Fizmatgiz, 1962, 476 p.
11. Glushkov V. M. Abstraktnaya teoriya avtomatov, *Uspehi matematicheskikh nauk*, 1961, Vol. XVI, Issue 5, pp. 3–62.
12. Trahtenbrot B. A., Bardzin Y. M. Konechnie avtomati (povedenie i sintez). Moscow, Nauka, 1970, 400 p.
13. Shihanovich U. A. Vvedenie v sovremennuyu matematiku (nachalnie ponyatiya). Moscow, Nauka, 1965, 376 p.
14. Barkalov A. A. Sintez ustroystv upravleniya na programmiruemykh logicheskikh ustroystvah. Donetsk, DonNTU, 2002, 262 p.

¹Д-р техн. наук, професор, декан факультету комп'ютерних систем і автоматики Вінницького національного технічного університету, Вінниця, Україна

²Канд. техн. наук, доцент кафедри «Інформаційні системи та мережі» Національного університету «Львівська політехніка», Львів, Україна

ВИЯВЛЕННЯ КЛЮЧОВИХ СЛІВ НА ОСНОВІ МЕТОДУ КОНТЕНТ-МОНІТОРИНГУ УКРАЇНОМОВНИХ ТЕКСТІВ

Вирішено завдання розробки алгоритмічного забезпечення процесів контент-моніторингу для розв'язання задачі визначення ключових слів україномовного тексту. Розглянуто формальне обґрунтування методу контент-моніторингу тексту за допомогою стеммера Портера, в основу модифікації стемінгу покладено відомі результати класифікації морфемної і словотвірної структури дериватів української мови, виявлення закономірностей комбінаторики афіксів, моделювання структурної організації дієслів і суфіксальних іменників, а також морфологічних модифікацій у процесі словозміни дієслова та словозміни і словотворенні прикметників української мови. Проведено декомпозицію методу та розроблено алгоритмічне забезпечення його основних структурних складових за результатами контент-аналізу тексту. Теоретично виявлено способи покращення показників ефективності пошуку ключових слів, зокрема щільності ключовиків у тексті. На основі розробленого програмного забезпечення отримано результати експериментальної апробації запропонованого методу контент-моніторингу для визначення ключових слів в наукових текстах технічного профілю. Виявлено, що для обраної експериментальної бази зі 100 робіт найкращих результатів за критерієм щільності досягає метод аналізу статті без початкової обов'язкової інформації і без списку літератури, але із перевіркою уточнених заблокованих слів та уточненого тематичного словника.

Ключові слова: текст, україномовний, алгоритм, контент-моніторинг, ключові слова, контент-аналіз, стеммер Портера, лінгвістичний аналіз, синтаксичний аналіз.

НОМЕНКЛАТУРА

ІТ – інформаційні технології;

СЕКК – система електронної контент-комерції;

е-бізнес – електронний бізнес;

Е-комерція – електронна комерція;

ПЗ – програмне забезпечення;

$X = \{x_1, x_2, \dots, x_{n_X}\}$ – множина вхідних даних $x_i \in X$ з різних інформаційних ресурсів або від модераторів при $i = \overline{1, n_X}$;

$C = \{c_1, c_2, \dots, c_{n_C}\}$ – множина комерційного контенту $c_r \in C$ при $r = \overline{1, n_C}$;

C_0 – сформований комерційний контент;

C_1 – відфільтрований комерційний контент;

C_2 – відформатований комерційний контент;

C_3 – комерційний контент з визначеною множиною ключових слів;

$\langle U_C, U_G, U_K \rangle$ – набір критеріїв для текстового контенту X ;

$U_C = \{U_{C1}, U_{C2}, \dots, U_{Cn_C}\}$ – множина критеріїв створення комерційного контенту;

$U_G = \{U_{G1}, U_{G2}, \dots, U_{Gn_G}\}$ – множина критеріїв збирання комерційного контенту (фільтри);

$U_K = \{U_{K1}, U_{K2}, U_{K3}, U_{K4}\}$ – множина критеріїв визначення ключових слів в контенті;

U_{K1} – унікальність термів – іменників, словосполучень іменників або прикметника з іменником серед множини слів контенту;

U_{K2} – частота появи ключових слів комерційного контенту;

U_{K3} – кількість знаків без пробілів для $Noun \in U_{K1}$ при $Unicity \geq 80$;

U_{K4} – критерій формування множини ключових слів;

$T = \{t_1, t_2, \dots, t_{n_T}\}$ – час $t_p \in T$ транзакції формування контенту при $p = \overline{1, n_T}$;

$\alpha_0 : (X, U_C, T) \rightarrow C_0$ – оператор створення контенту – відображення даних з різних джерел у контент, який відрізняється актуальністю;

$\alpha_1 : (X, U_G, T) \rightarrow C_0$ – оператор збирання контенту – відображення даних від авторів у контент, який відрізняється достовірністю та актуальністю;

$\alpha_2 : (C_0, T, U_B) \rightarrow C_1$ – оператор виявлення дублювання контенту – відображення контенту в новий стан, який відрізняється унікальністю;

$\alpha_3 : (C_1, U_{FR}, T) \rightarrow C_2$ – оператор форматування контенту – відображення контенту в новий стан, який відмінний від попереднього форматом подання;

$\alpha_4 : (C_2, U_K, T) \rightarrow C_3$ – оператор виявлення ключових слів контенту – відображення контенту в новий стан, який відрізняється наявністю множини ключових слів, що загально описують його зміст.

ВСТУП

Активний розвиток мережі Інтернет сприяє зростанню потреб в отриманні оперативних даних виробничого/стратегічного характеру і реалізації нових форм інформаційного обслуговування через сучасні ІТ е-бізнесу [1–3]. Документована інформація, підготовлена відповідно до потреб користувачів, є інформаційним продуктом або комерційним контентом, наприклад, електронний матеріал Інтернет-видавництва, маркетингові дослідження,

консалтингові послуги тощо. Дії для забезпечення користувачів комерційним контентом є інформаційною послугою. Інтернет-ринок є сукупністю економічних, правових, організаційних і програмних відносин з продажу інформаційних продуктів/послуг між виробниками, постачальниками та користувачами [1–3].

Комерційний контент визначають як:

- вміст інформаційних ресурсів в СЕКК;
- об’єкт бізнес-процесів в СЕКК, наприклад, стаття, ПЗ, книга тощо;
- структурована та логічно завершена множина даних, що є об’єктом взаємовідносин між користувачем та СЕКК;
- набір електронних даних без наперед визначеної структури;
- дані комерційного призначення, що неподільні в часі;
- основний чинник формування області діяльності, функціонування та призначення СЕКК.

Сьогодні е-комерція є об’єктивною реальністю та перспективним бізнес-процесом. Інтернет є бізнес-середовищем, а комерційний контент є товаром з найбільшим попитом у ньому та основним об’єктом процесів електронної контент-комерції. Комерційний контент можна зразу замовити, оформити, оплатити і отримати on-line як товар. Через Інтернет продають весь спектр комерційного контенту – наукові та публіцистичні статті, музика, книги, фільми, фото, ПЗ тощо. Відомими корпораціями, які реалізують електронну контент-комерцію, є Google через Play Market, Apple – Apple Store, Amazon – Amazon.com [1].

Більшість рішень та досліджень зроблено на рівні реальних прикладних проектів, а сучасні СЕКК побудовані за закритим принципом як разові проекти та орієнтовані на реалізацію комерційного контенту, створеного за їх межами. Тому для проектування, створення, впровадження та супроводу СЕКК потребують розробки загальні методи та інформаційні технології формування, управління та супроводу комерційного контенту. З огляду на важливість для функціонування СЕКК ключових слів об’єктом дослідження обрано процес виявлення ключових слів в україномовних текстах у режимі реального часу, предмет дослідження – методи та моделі контент-моніторингу таких текстів.

1 ПОСТАНОВКА ЗАДАЧ

Нехай україномовний текстовий контент X з різних джерел інформації у вигляді $X = \{x_1, x_2, \dots, x_{n_x}\}$ має ста-

ти основою відфільтрованого контенту C_1 , відформатованого контенту C_2 та його модифікації C_3 з визначеною множиною ключових слів $KeyWords \in U_{K4}$. За відомими критеріями $\langle U_C, U_G, U_K \rangle$ потрібно визначити оператор виявлення ключових слів комерційного контенту $\alpha_4 : (C_2, U_K, T) \rightarrow C_3$ та експериментально перевірити параметр частоти появи ключових слів комерційного контенту U_{K2} за різними режимами роботи алгоритмічного забезпечення запропонованого методу контент-моніторингу.

2 ОГЛЯД ЛІТЕРАТУРИ

Сталою сучасною тенденцією можна вважати постійний ріст темпів виробництва текстового контенту в Інтернет-просторі. Цей процес є об’єктивним і позитивним, але виникла проблема – прогрес у галузі виробництва текстового контенту призводить до пониження загального рівня інформованості потенційного користувача Інтернет-простору [1–3]. Крім збільшення обсягів текстового контенту до масштабів, яке унеможливило його безпосереднє опрацювання та помітно гальмує його поширення виникає низка специфічних проблем (табл. 1).

Негативні чинники у формуванні текстового контенту ускладнюють процес пошуку необхідних даних при скануванні різних джерел інформації. Збільшення фізичного обсягу та змінність співвідношення актуальності/динаміки контентних потоків (наслідок систематичного або нерегулярного оновлення) призводить до виникнення дублювання, інформаційного шуму та надмірності результатів пошуку контенту. Охоплення та узагальнення великих динамічних потоків контенту, які неперервно генерують в Інтернет-джерелах, вимагає якісно нових методів/підходів пошуку – таких як контент-моніторинг (рис. 1) на основі аналізу ключових слів [1–32]. Вхідною інформацією для контент-моніторингу є текст на природній мові як послідовність символів, вихідна інформація – це таблиці розділів, речень і лексем аналізованого тексту. Контент-моніторинг є програмним засобом автоматизації знаходження найбільш важливих складових в потоках контенту за допомогою алгоритмів стемінгу [1–32]. Це змістовний аналіз потоків контенту з метою постійного отримання необхідних якісних/кількісних зрізів на протязі наперед не визначеного проміжку часу.

Мета роботи полягає у створенні алгоритмічного забезпечення методу контент-моніторингу україномовних текстів на основі стеммера Портера та його застосуванні для виявлення значущих ключових слів. Для досягнення

Таблиця 1 – Основні негативні чинники у формуванні текстового контенту

Назва	Основна причина	Рішення
Інформаційний шум	Структурованість масивів контенту.	Фільтри, контент-моніторинг, аналіз сайту, контент-аналіз.
Паразитичний контент	Поява в якості додатків.	Фільтри, контент-моніторинг, контент-аналіз.
Нерелевантність контенту	Невідповідність потребам користувачів.	Створення анотованої бази даних, пошукових образів первинного контенту та їх кластеризація, контент-аналіз.
Дублювання контенту	Дублювання в джерелах.	Контент-аналіз, сканери і фільтри на базі статистики та критеріїв.
Навігація в потоці контенту	Швидкий ріст обсягу і поширення контенту.	Аналіз сайту, фільтри, контент-моніторинг, контент-аналіз.
Надмірність пошуку	Дублювання і нерелевантність.	Анотований пошук, контент-аналіз та реферування.

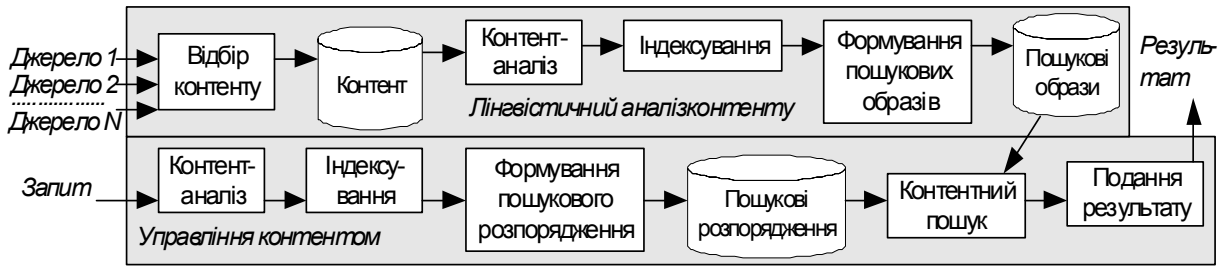


Рисунок 1 – Структурна схема процесу контент-моніторингу текстових масивів даних

мети пропонується розв’язати такі задачі дослідження: провести контент-аналіз текстової інформації; забезпечити визначення множини ключових слів; провести лінгвістичний аналіз текстового контенту; розробити синтаксичний аналізатор текстового контенту.

3 МАТЕРІАЛИ І МЕТОДИ

Головною складовою контент-моніторингу є контентний пошук та контент-аналіз тексту. Контент-аналіз призначений для пошуку контенту в масиві даних за змістовими лінгвістичними одиницями (алг. 1). Одиниця рахунку є кількісною мірою одиниці аналізу, що дозволяє реєструвати частоту (регулярність) появи ознаки категорії аналізу в тексті (кількість певних слів або їх поєднань, рядків, друкованих знаків, сторінок, абзаців, авторських аркушів, площа тексту тощо).

Алгоритм 1. Контент-аналіз текстового контенту.

Етап 1. Визначення набору критеріїв $\langle U_C, U_G \rangle$ для текстового контенту X .

Крок 1. Формування набору критеріїв як тип джерела (форум, електронна пошта, Інтернет-газета, чат, Інтернет-журнал); тип контенту (стаття, е-лист, банер, коментарій); учасники комунікації (відправник, одержувач, реципієнт).

Крок 2. Визначення розміру (мінімальний обсяг або довжина), частоти появи, способу/місця розповсюдження та час появи контенту.

Крок 3. Фільтрування згідно сформованого набору критеріїв контентного потоку та зберігання ідентифікованого релевантного контенту X .

Етап 2. Контент-аналітичний відбір. Формування вибіркової сукупності контенту X' за критеріями обмеженої вибірки $\langle U_C, U_G \rangle$ з більшого масиву X .

Етап 3. Виявлення змістовних одиниць аналізу $\langle U'_C, U'_G \rangle$ текстового комерційного контенту X' (словосполучення, речення, тема, ідея, автор, персонаж, соціальна ситуація, частина тексту, кластеризована за змістом категорії аналізу) за модифікованим алгоритмом Потера. Вимоги до вибору лінгвістичної одиниці аналізу: велика для інтерпретації значення; мала, щоб не інтерпретувати багато значень; легко ідентифікується; кількість одиниць велика для проведення вибірки.

Етап 4. Виділення одиниць рахунку аналізу текстового контенту X' .

Крок 1. Якщо одиниці рахунку $\langle U_C, U_G \rangle$ збігаються з одиницями аналізу $\langle U'_C, U'_G \rangle$, то знаходять частоти появи виділеної змістовної одиниці, інакше перейти до кроку 2.

Крок 2. Модератор на основі аналізованого контенту висуває та доповнює одиниці рахунку $\langle U_C, U_G \rangle$, наприклад, протяжність текстів; площа тексту, заповнена змістовними одиницями; кількість рядків (абзаців, знаків, колонок тексту); розмір/вид файлу; кількість рисунків з певним змістом/сюжетом тощо.

Етап 5. Порівняння змістовних одиниць аналізу $\langle U'_C, U'_G \rangle$ з одиницями $\langle U_C, U_G \rangle$.

Крок 1. Класифікація за угрупованнями із оціненням ваги змістовних категорій в загальному обсязі тексту. Класифікатором є загальна таблиця, в яку зведені всі категорії аналізу і одиниці аналізу. Фіксують одиниці виразу категорій.

Крок 2. Статистичні розрахунки зрозумілості та атрактивності контенту.

Етап 6. Розроблення інструменту контент-аналізу.

Крок 1. Створення закодованого протоколу контенту X' для компактності подання даних та швидкого порівняння результатів аналізу різного контенту.

Крок 2. Заповнення протоколу контенту X' властивостями (автор, час видання, обсяг тощо).

Крок 3. Заповнення протоколу контенту X' підсумками його аналізу (кількість вживання в ньому певних одиниць аналізу і висновки щодо категорій аналізу). Протокол кожного контенту X' заповнюється на основі підрахунку даних всіх його реєстраційних карток.

Етап 7. Розроблення таблиці контент-аналізу. Тип таблиці визначають у вигляді системи скоординованих і субординованих категорій аналізу: кожна категорія (питання) передбачає ряд ознак (відповідей), за якими квантифікується зміст тексту X' .

Етап 8. Розроблення кодувальної матриці контент-аналізу.

Крок 1. Якщо обсяг вибірки ≥ 100 одиниць, то аналізується набір матричних листів, інакше виконати крок 2.

Крок 2. Якщо вибірка < 100 одиниць, то проводиться двовимірний аналіз. В цьому випадку для кожного контенту X' формується кодувальна матриця.

Етап 9. Проведення аналізу тексту X' згідно створених кодувальних матриць.

Етап 10. Інтерпретація результатів $\alpha_0 : (X, U_C, T) \rightarrow C_0$ та $\alpha_1 : (X, U_G, T) \rightarrow C_0$. Виявляють і оцінюють характеристики контенту X' на основі статистичного набору підрахованих коефіцієнтів за певний період часу на визначену категорію. Охоплює всі здобуті фрагменти тексту C_0 , висновки спираються не на частину результатів, а враховуються всі без винятку. Фільтрування $\alpha_3 : (C_1, U_{FR}, T) \rightarrow C_2$ та формування $\alpha_3 : (C_1, U_{FR}, T) \rightarrow C_2$ комерційного контенту.

Застосування контент-аналізу при моніторингу Інтернет-джерел даних дозволяє автоматизувати процес знаходження найбільш важливих складових в потоках контенту при відборі даних з цих джерел. Це усуває дублювання контенту, інформаційний шум, паразитичний контент, надмірність результатів пошуку тощо. Даний метод застосовують в подальших етапах формування контенту для отримання більш точного релеван-

ного результату – створення унікального комерційного контенту, який користується попитом серед користувачів СЕKK. З метою реалізації контент-аналізу текстових масивів даних для формування множини ключових слів було розроблено програму на основі стеммера Потера, адаптованого до української мови (алг. 2), а також таблиці основ основних тематичних слів для подальшої рубрикації, текстів, що досліджуються (табл. 2). Блок-схему алгоритму наведено на рис. 2.

Таблиця 2 – Основні складові програми формування ключових слів

№	Назва	Пояснення
1	Filter	список потенційних ключовиків з аналізованого тексту з розрахунком відносної частоти їх появи в тексті
2	Input	вхідний текст для аналізу та визначення ключовиків
3	Format	тестувальний відформатований вхідний текст
4	Parser	парсер, адаптований до української
5	Stemer	правила стеммера Потера, адаптований до української
6	Object	класи об'єктів для стеммера Потера
7	Resvoc	список ключовиків (частоти вживання яких в тексті попали у визначений діапазон згідно алг. 2 на рис. 2 та відповідають тематичному словнику Thematic.txt)
8	Thematic	тематичний словник (формується модератором)
9	Vocab	список слів з аналізованого тексту з розрахунком абсолютної частоти їх появи в тексті

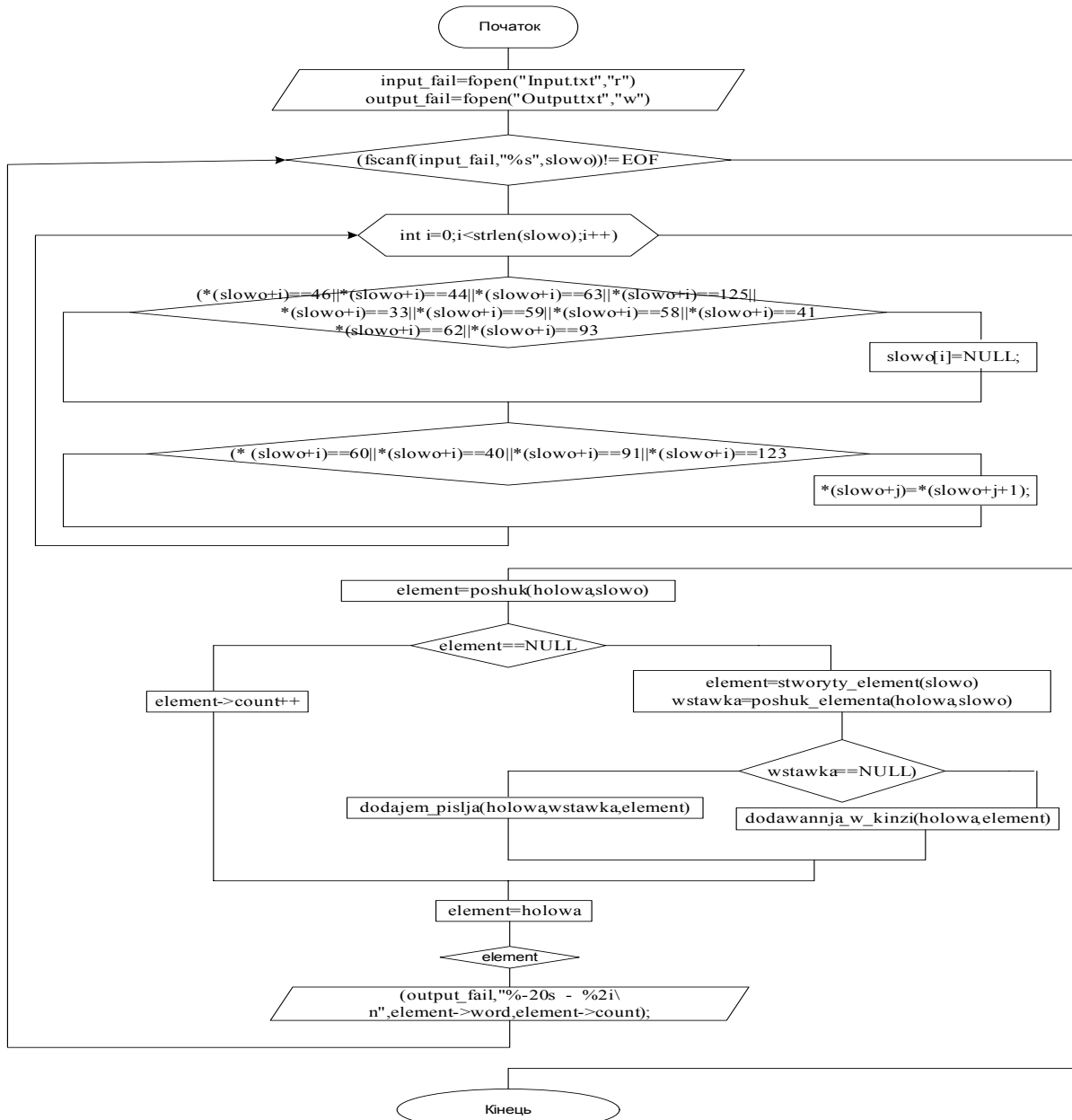


Рисунок 2 – Структурна схема алгоритму статистичного аналізу вживання слів в тексті

Алгоритм 2. Визначення множини ключових слів

Етап 1. В *Input* зберегти текст, який необхідно дослідити.

Етап 2. Відформатувати вхідний текст (однакові апострофи, забрати зайві символи, які не входять в абетку, окрім службових як пробіл, апостроф). В *Format* зберегти текст, який відформатовано.

Етап 3. При необхідності редагувати тематичний словник *Thematic*.

Етап 4. Запустити процес визначення множини ключових слів.

Крок 1. Будуємо спочатку алфавітно-частотний словник (абсолютні частоти) – *Vocab*.

Крок 2. Будуємо потім з *Vocab* алфавітно-частотний словник (відносні частоти) слів, тобто список слів за алфавітом та їх відносні частоти відносно загального обсягу тексту.

Крок 3. Будуємо скорочений список слів, частоти яких відповідають умовам формування ключовиків $U_K = \{U_{K1}, U_{K2}, U_{K3}, U_{K4}\}$, тобто список потенційних ключовиків – *Filter*.

Крок 4. Зв'язуємо сформований скорочений список з *Filter* зі списком *Thematic* та відповідно формуємо новий список входжень потенційних ключовиків з *Filter* в *Thematic* – список ключовиків в *Resvoc*.

Цей алгоритм не враховує пошук ключовиків по основах, у зв'язку з цим – результати дослідження текстів на формування ключових слів були негативні, зокрема:

– відсутні взагалі ключові слова у вихідному файлі (знайдені слова не відповідали вимогам до ключових слів – не попадали в діапазон частоти вживання в тексті);

– в списку ключових слів попали службові слова, дієприкметники, дієслова, як ніяк не можуть бути ключовими словами (некоректно прописана база правил заблокованих слів);

– були присутні декілька слів з одною основою, але з різними флексіями (некоректно прописана база правил визначення основ, наприклад, пошук – пошукowymi, користувач – користувачам, рейтинг – високорейтингового, рейтингу, контент – контентного, інформація – інформаційний, або були присутні граматичні помилки).

Тому був розроблений інший алгоритм знаходження множини ключових слів з врахуванням основ тематичних слів (рис. 3), та розміщений у відкритому доступі за адресою <http://victana.lviv.ua/index.php/kliuchovi-slova>.

4 ЕКСПЕРИМЕНТИ

Лінгвістичною базою для експериментального дослідження обрано 100 наукових публікацій Вісника Національного університету «Львівська політехніка» серії «Інформаційні системи та мережі» (<http://science.lp.edu.ua/sisn>), двох номерів 783 (<http://science.lp.edu.ua/SISN/SISN-2014>) та 805 (<http://science.lp.edu.ua/sisn/vol-cur-805-2014-2>). Аналіз статистики функціонування системи виявлення множини ключових слів із 100 наукових статей було проведено у два етапи, зокрема:

1. Проаналізувати всі статті із перевіркою загальних заблокованих слів та тематичного словника.

2. Проаналізувати всі статті із перевіркою уточнених заблокованих слів та уточненого тематичного словника (з більшою кількістю запуску системи формується множина невідомих слів (відсутніх і в тематичному словнику і в множині заблокованих)).

Окрім того на кожному етапі перевірка відбувалась в два кроки для кожної статті: аналіз всієї статті (рис. 4а) та аналіз статті без початку (назва, автори, удк, анотації двома мовами, авторські ключові слова двома мовами, місце роботи авторів) і без списку літератури (рис. 4б) для того, щоб визначити похибки точності формування множини ключових слів.

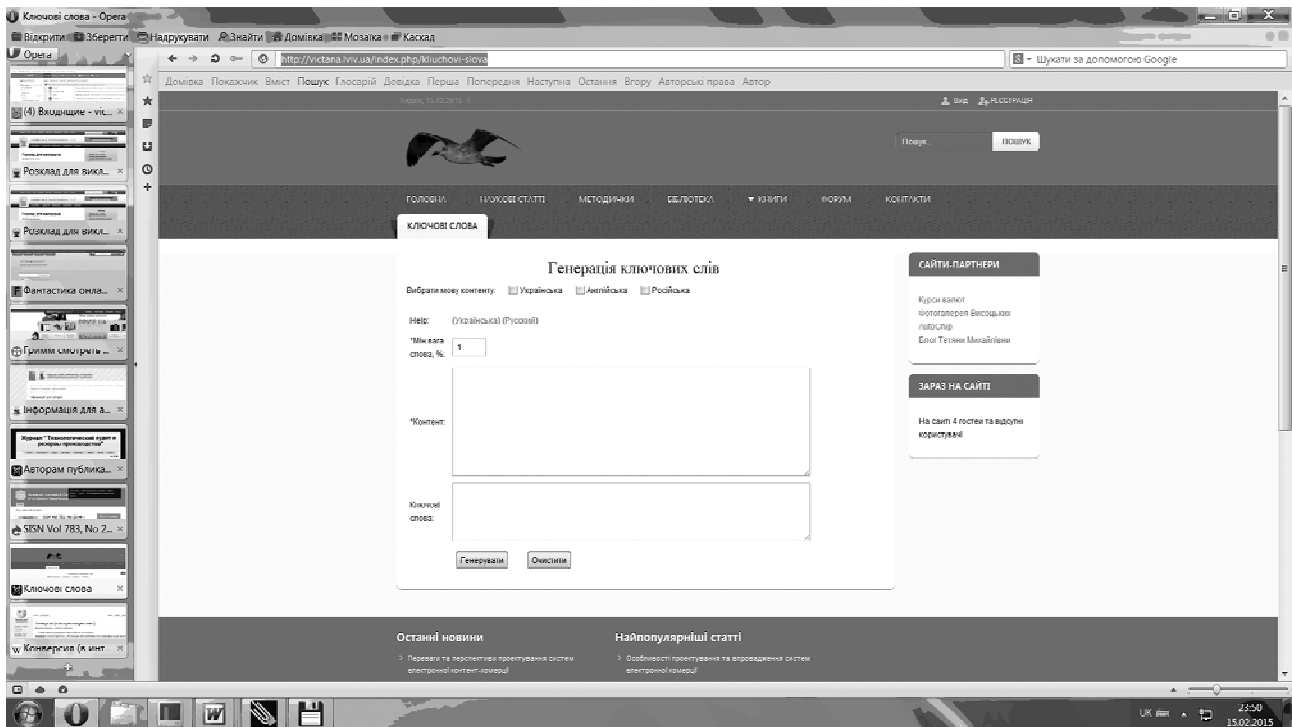


Рисунок 3 – Інформаційний ресурс визначення ключових слів з тексту

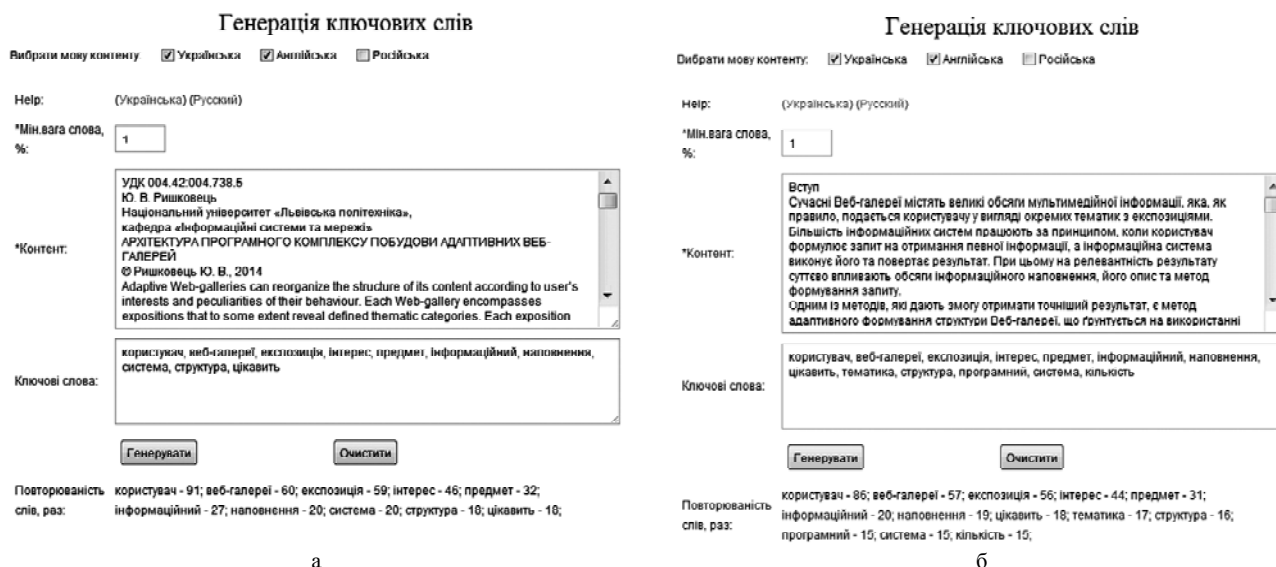


Рисунок 4 – Результати перевірки статті: а – приклад аналізу всієї статті, б – приклад аналізу статті без початку і без списку літератури

5 РЕЗУЛЬТАТИ

Аналіз статистики здійснювався за принципом порівняння множини авторських ключових слів (визначені та прописані в статті самими авторами цих робіт), множини ключових слів визначених за першим та другим етапами з різними вагами слів (але більше, за визначене в опції *Мінімум ваги слова, % в межах [1,5]) з повними та скороченими текстами робіт (табл. 3) при середньому арифметичному значенні авторських ключових словосполучень / слів біля 5 (4,77), які в середньому утворені з 10 (9,82) слів. Вага слова розраховується як відносна частота появи основи цього слова у всьому тексті. В табл. 4 присутні такі позначення, як *A* (всього ключових слів,

Таблиця 3 – Статистичні дані досліджених обсягів текстів статей

Назва обсягу статті	Крок 1		Крок 2	
	Всього	Середнє арифметичне	Всього	Середнє арифметичне
Сторінок	956	9,56	828	8,28
Абзаців	16497	164,97	15263	152,63
Рядків	42553	425,53	36965	369,65
Слів	345580	3455,8	291247	2912,47
Знаків	2327209	23272,09	1974773	19747,73
Знаків та пробілів	2674889	26748,89	2265917	22659,17

визначених системою при заданій вазі слова), *B* (змістовних слів зі списку утворених, тобто без невідомих аббревіатур, дієслів, службових слів тощо), *C* (збіг слів з визначеними автором статті), *D* (точність збігу знайдених ключовиків з авторським ключовими словами), *E* (додаткові ключові слова, визначені системою, але не визначені автором статті).

6 ОБГОВОРЕННЯ

На рис. 5 наведена порівняльна діаграма відсотків вживання знайдених системою ключових слів в відфільтрованому тексті (без початку (назва, автори, удк, анотації двома мовами, авторські ключові слова двома мовами, місце роботи авторів) і без списку літератури) Per_f та первинному авторському тексті Per_0 без уточнення модератором тематичного словника через поповнення заблокованих слів.

Отримані середні значення для 100 текстів $Per_f = 0,28$ та $Per_0 = 0,19$ показують, що така фільтрація наукових статей покращує щільність ключовиків у 1,48 раз або на 47,83 відсотка. На рис. 6 наведена порівняльна діаграма відсотків вживання знайдених системою ключових слів в відфільтрованому тексті (без початку (назва, автори, удк, анотації двома мовами, авторські ключові слова двома мовами, місце роботи авторів) і без списку

Таблиця 4 – Статистичні дані досліджених змісту текстів статей

Назва	Вага слова	Етап 1					Етап 2				
		A	B	C	D	E	A	B	C	D	E
Крок 1	≥ 1	5,46	3,92	2,51	2,08	1,74	7,43	7,03	3,27	3	4,18
	≥ 2	1,08	0,88	0,63	0,59	0,26	2,67	2,64	1,65	1,54	1,12
	≥ 3	0,41	0,38	0,22	0,21	0,16	1,21	1,2	0,85	0,79	0,41
	≥ 4	0,15	0,13	0,09	0,09	0,04	0,46	0,45	0,33	0,31	0,15
	≥ 5	0	0	0	0	0	0	0	0	0	0
Крок 2	≥ 1	6,51	5,02	2,68	2,23	2,37	8,35	7,78	3,25	2,91	4,99
	≥ 2	1,34	1,11	0,74	0,72	0,39	3,12	3,07	1,81	1,67	1,43
	≥ 3	0,51	0,45	0,29	0,27	0,17	1,42	1,4	0,93	0,85	0,54
	≥ 4	0,19	0,17	0,12	0,12	0,05	0,73	0,72	0,45	0,42	0,31
	≥ 5	0,11	0,1	0,06	0,06	0,04	0,33	0,32	0,25	0,23	0,1

літератури) Per_f^v та первинному авторському тексті Per_0^v з врахуванням уточнення модератором тематичного словника через поповнення заблокованих слів. Отримані середні значення для 100 текстів $\overline{Per_f^v} = 0,34$ та $\overline{Per_0^v} = 0,25$ показують, що фільтрація з одночасною модерацією тематичного словника покращує щільність ключовиків у 1,35 раз або на 35,44%.

На рис. 7 наведена порівняльна діаграма відсотків вживання знайдених системою ключових слів в початковому первинному авторському тексті без уточнення модератором тематичного словника через поповнення

заблокованих слів (Per_0) та з врахуванням уточнення модератором тематичного словника через поповнення заблокованих слів (Per_0^v).

Порівняння значень $\overline{Per_0} = 0,19$ та $\overline{Per_0^v} = 0,25$ демонструє ефективність модерації тематичного словника у початковому тексті – щільність ключовиків збільшується у 1,34 раз або на 34,33 відсотка. На рис. 8 наведена порівняльна діаграма відсотків вживання знайдених системою ключових слів в відфільтрованому авторському тексті без уточнення модератором тематичного словника через поповнення заблокованих слів (Per_f)

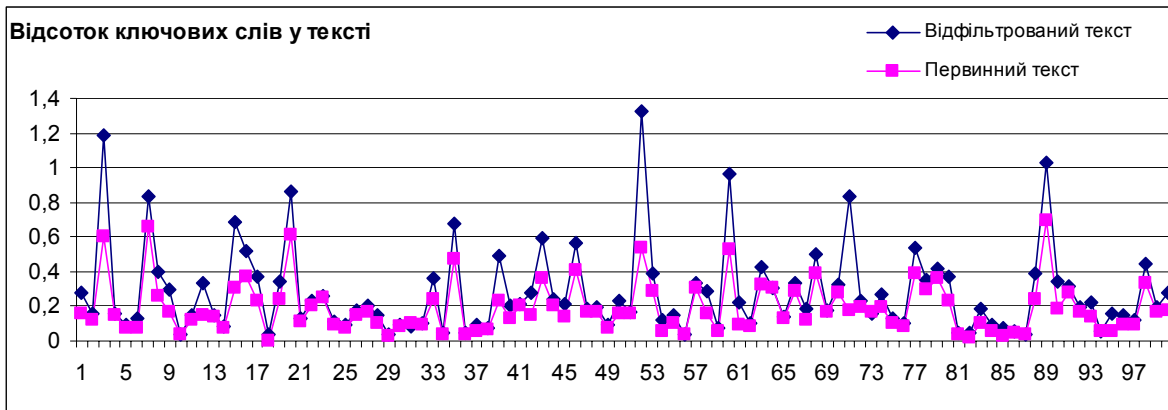


Рисунок 5 – Результати перевірки статей без уточнення модератором тематичного словника

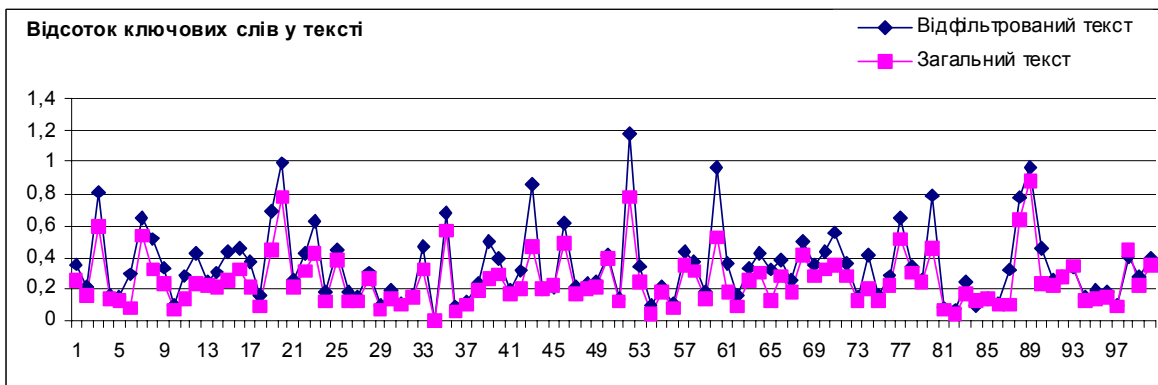


Рисунок 6 – Результати перевірки статей з врахуванням уточнення модератором тематичного словника

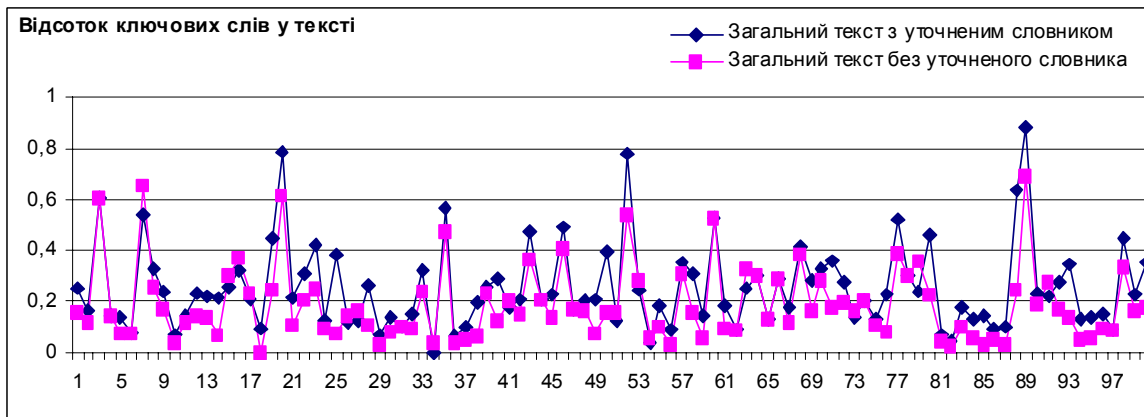


Рисунок 7 – Результати перевірки первинних авторських статей з різними словниками

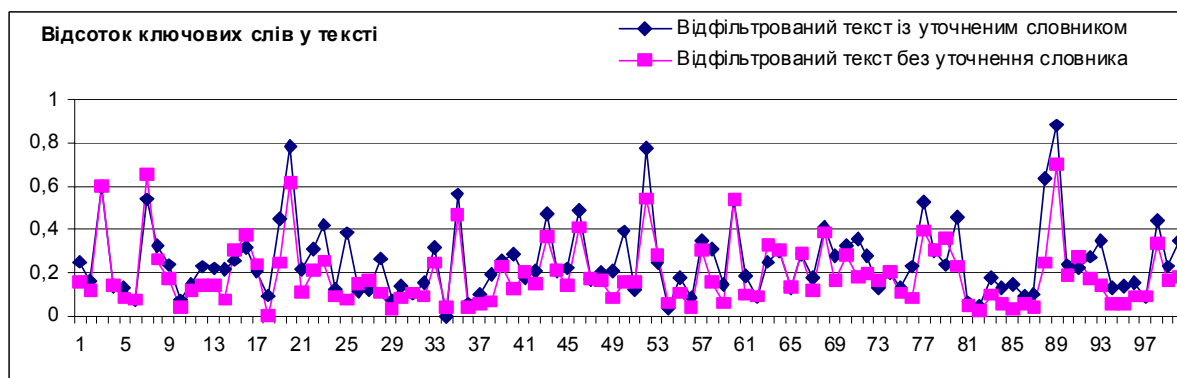


Рисунок 8 – Результати перевірки відфільтрованих статей з різними словниками

та з врахуванням модерації тематичного словника (Per_f^v). Порівняння значень $\overline{Per_f} = 0,28$ та $\overline{Per_f^v} = 0,34$ демонструє ефективність модерації тематичного словника у відфільтрованому тексті – щільність ключовиків збільшується у 1,23 раз або на 23,14 відсотка.

ВИСНОВКИ

У статті наведено теоретичне та експериментальне обґрунтування методу контент-моніторингу україномовного тексту на основі стемінгу Портера. Метод спрямовано на автоматичне виявлення значущих ключових слів україномовного тексту за рахунок запропонованого формального підходу до реалізації стемінгу україномовного контенту. Проведено декомпозицію методу контент-моніторингу на взаємопов'язані складові контент-аналізу текстової інформації та визначення множини ключових слів. Розроблено алгоритмічне забезпечення основних структурних складових запропонованого методу, а основу якого покладено адаптований до української мови алгоритм (стеммер) Портера. Теоретично виявлено способи покращення показників ефективності пошуку ключових слів, зокрема щільності ключовиків у тексті. Експериментальне дослідження 100 наукових публікацій з двох номерів (783 та 805) Вісника Національного університету «Львівська політехніка» серії «Інформаційні системи та мережі» (<http://science.lp.edu.ua/sisn>) продемонструвало позитивний вплив фільтрації тексту статті та модерації тематичного словника на визначення ключових слів. Виявлено, що для технічних наукових текстів експериментальної бази найкращих результатів досягає метод аналізу статті без початку (назва, автори, удк, анотації двома мовами, авторські ключові слова двома мовами, місце роботи авторів) і без списку літератури із перевіркою уточнених заблокованих слів та уточненого тематичного словника – для нього середнє значення щільності ключовиків у тексті досягає $Per_f^v = 0,34$, що на 81% більше за

аналогічне значення щільності первинного тексту $Per_0 = 0,19$. Потребує подальшого експериментального дослідження визначення ключових слів для інших категорій текстів – наукових гуманітарного про-філю, художніх, публіцистичних тощо.

ПОДЯКИ

У статті розв'язана науково-практична задача автоматичного виявлення значущих ключових слів та рубрикації україномовного контенту в Інтернет-системах на основі попереднього опрацювання відповідної текстової інформації. Роботу виконано в рамках спільних наукових досліджень кафедри інформаційних систем та мереж Національного університету

«Львівська політехніка» на тему «Дослідження, розроблення і впровадження інтелектуальних розподілених інформаційних технологій та систем на основі ресурсів баз даних, сховищ даних, просторів даних та знань з метою прискорення процесів формування сучасного інформаційного суспільства», а також кафедри автоматики та інформаційно-вимірювальної техніки Вінницького національного технічного університету у межах діяльності науково-дослідного центру прикладної та комп'ютерної лінгвістики. Результати досліджень здійснювались у рамках держбюджетних науково-дослідних робіт за темами «Розробка методів, алгоритмів і програмних засобів моделювання, проектування та оптимізації інтелектуальних інформаційних систем на основі Web-технологій «ВЕБ» та «Інтелектуальна інформаційна технологія образного аналізу тексту та синтезу інтегрованої бази знань природно-мовного контенту». Наукові дослідження провадилися також в рамках ініціативної тематики досліджень кафедри ІСМ Національного університету «Львівська політехніка» на тему «Розроблення інтелектуальних розподілених систем на основі онтологічного підходу з метою інтеграції інформаційних ресурсів».

СПИСОК ЛІТЕРАТУРИ

1. Берко А. Системи електронної контент-комерції / А. Берко, В. Висоцька, В. Пасічник. – Л. : НУЛП, 2009. – 612 с.
2. Математична лінгвістика / [В. Висоцька, В. Пасічник, Ю. Щербина, Т. Шестакевич]. – Л. : «Новий Світ-2000», 2012. – 359 с.
3. Найефективніші методи залучення потенційних клієнтів [Електронний ресурс] / Центр ресурсів якості трафіку оголошень, Google AdWords. – Режим доступу: http://www.google.com/intl/uk_ALL/ads/adtrafficquality/advertisers/best-practices-for-generating-leads.html. – Назва з титул. екрану.
4. Нечеткий пошук в тексті і словаре [Електронний ресурс]. – Режим доступу: <http://habrahabr.ru/post/114997/>. – Назва з титул. екрану.
5. Реализации алгоритмов. Расстояние Левенштейна [Електронний ресурс]. – Режим доступу: http://ru.wikibooks.org/wiki/Реализации_алгоритмов/Расстояние_Левенштейна. – Назва з титул. екрану.
6. Задача о расстоянии Дамерау-Левенштейна [Електронний ресурс]. – Режим доступу: http://neerc.ifmo.ru/wiki/index.php?title=%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%BE_%D1%80%D0%B0%D1%81%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D0%B8_%D0%94%D0%B0%D0%BC%D0%B5%D1%80%D0%B0%D1%83-%D0%9B%D0%B5%D0%B2%D0%B5%D0%BD%D1%88%D1%82%D0%B5%D0%B9%D0%BD%D0%B0. – Назва з титул. екрану.
7. Насонов Д. Функция Левенштейна [Електронний ресурс] / Д. Насонов. – Режим доступу: <http://rain.ifmo.ru/cat/data/theory/unsorted/levenshtein-2006/article.pdf>. – Назва з титул. екрану.

8. Левенштейн, который сравнивает строки [Электронный ресурс] / Веб-разработка. – Режим доступа: <http://dayte2.com/levenshtein>. – Назва з титул. екрану.
9. Вычисление расстояния Левенштейна между двумя строками [Электронный ресурс]. – Режим доступа: <http://wm-help.net/lib/b/book/827961078/78>. – Назва з титул. екрану.
10. Стеммер Потера [Электронный ресурс]. – Режим доступа: <http://labs.abcvg.com/stemmer/index.php>. – Назва з титул. екрану.
11. Moseichuk V. Porter stemming algorithm for Ukrainian languages [Electronic resource] / V. Moseichuk. – Access mode: http://www.marazm.org.ua/document/stemer_ua/. – Title from the screen.
12. Стемінг [Электронный ресурс]. – Режим доступа: <https://uk.wikipedia.org/wiki/Стемінг>. – Назва з титул. екрану.
13. Russian stemming algorithm [Electronic resource]. – Access mode: <http://snowball.tartarus.org/algorithms/russian/stemmer.html>. – Title from the screen.
14. Porter stemmer – реализация алгоритма стеммера Портера для русского языка на чистом функциональном языке Clojure [Электронный ресурс]. – Режим доступа: <https://github.com/allaud/porter-stemmer>. – Назва з титул. екрану.
15. The Porter Stemming Algorithm – Porter’s homepage. [Электронный ресурс]. – Режим доступа: <http://tartarus.org/~martin/PorterStemmer/>. – Назва з титул. екрану.
16. The Porter Stemming Algorithm – Project «Snowball» [Electronic resource]. – Access mode: <http://snowball.tartarus.org/algorithms/porter/stemmer.html>. – Title from the screen.
17. The English (Porter2) stemming algorithm – Project «Snowball» [Electronic resource]. – Access mode: <http://snowball.tartarus.org/algorithms/english/stemmer.html>. – Title from the screen.
18. Porter M. F. An algorithm for suffix stripping [Electronic resource] / M. F. Porter // Program. – 1980. – Т. 14, № 3. – С. 130–137. – Access mode: http://telemat.det.unifi.it/book/2001/wchange/download/stem_porter.html. – Title from the screen.
19. Willett P. The Porter stemming algorithm: then and now [Electronic resource] / P. Willett // Program: Electronic Library and Information Systems. – 2006. – В. 3, Т. 40. – С. 219–223. – ISSN 0033-0337. – Access mode: <http://eprints.whiterose.ac.uk/1434/>. – Title from the screen.
20. Сенік М. Вільний алгоритм стемінгу для української мови [Электронный ресурс] / М. Сенік. – Режим доступа: http://www.senyk.poltava.ua/projects/ukr_stemming/stemming_about.html. – Назва з титул. екрану.
21. Сенік М. Інструмент для пошуку слів з однаковими закінченнями [Электронный ресурс] / М. Сенік. – Режим доступа: http://www.senyk.poltava.ua/projects/ukr_stemming/word_by_ending.html. – Назва з титул. екрану.
22. Сенік М. Статичне дерево закінчень [Электронный ресурс] / М. Сенік. – Режим доступа: http://www.senyk.poltava.ua/projects/ukr_stemming/ukr_endings.html#dyn. – Назва з титул. екрану.
23. Сенік М. Демо стемінгу для української мови [Электронный ресурс] / М. Сенік. – Режим доступа: http://www.senyk.poltava.ua/projects/ukr_stemming/demo.html. – Назва з титул. екрану.
24. Вероятностный морфологический анализатор русского и украинского языков [Электронный ресурс]. – Режим доступа: <http://www.keva.ru/stemka/stemka.html>. – Назва з титул. екрану.
25. Стеммінг [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Стеммінг>. – Назва з титул. екрану.
26. Lovins J. B. Development of a stemming algorithm / J. B. Lovins // Mechanical Translation and Computational Linguistics 11:22–31. – 1968.
27. Jongejan, B. Automatic training of lemmatization rules that handle morphological changes in pre-, in- and suffixes alike [Electronic resource] / B. Jongejan, H. Dalanis // In the Proceeding of the ACL-2009, Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Singapore, August 2–7, 2009, pp. 145–153. – Access mode: <http://www.aclweb.org/anthology/P/P09/P09-1017.pdf>. – Title from the screen.
28. Вірогідний морфологічний аналізатор російської та української [Электронный ресурс]. – Режим доступа: <http://www.keva.ru/stemka/stemka.html>. – Назва з титул. екрану.
29. Модуль Drupal для стемінгу українською. Новий модуль для алгоритму Стема для Українського пошуку з виділенням коренів [Электронный ресурс]. – Режим доступа: <http://drupal.ua/node/1170>. – Назва з титул. екрану.
30. Стемінг Портера для української мови [Электронный ресурс]. – Режим доступа: http://www.marazm.org.ua/document/stemer_ua/. – Назва з титул. екрану.
31. Hardcoded stemmer for Ukrainian [Electronic resource]. – Access mode: <https://github.com/vgrichina/ukrainian-stemmer>. – Title from the screen.
32. Perestoronin P. Стеммер Портера для русского языка [Электронный ресурс] / P. Perestoronin. – Режим доступа: <http://blog.eigene.in/post/49598738049/snowball>. – Назва з титул. екрану.

Стаття надійшла до редакції 23.12.2015.

Після доробки 04.01.2016.

Бисикало О. В.¹, Высоцкая В. А.²

¹Д-р техн. наук, професор, декан факультета комп’ютерних систем і автоматики Вінницького національного технічного університету, Вінниця, Україна

²Канд. техн. наук, доцент кафедри «Информационные системы и сети» Национального университета «Львовская политехника», Львов, Україна

ВЫЯВЛЕНИЕ КЛЮЧЕВЫХ СЛОВ НА ОСНОВЕ МЕТОДА КОНТЕНТ-МОНИТОРИНГА УКРАИНОЯЗЫЧНЫХ ТЕКСТОВ

Решена задача разработки алгоритмического обеспечения процессов контент-мониторинга для решения задачи определения ключевых слов русскоязычного текста. Рассмотрено формальное обоснование метода контент-мониторинга текста с помощью Стеммер Портера, в основу модификации стемминг положены известны результаты классификации морфемной и словообразовательной структуры дериватов украинского языка, выявление закономерностей комбинаторики аффиксов, моделирование структурной организации глаголов и суффиксальных существительных, а также морфонологичных модификаций в процессе словоизменения глагола и словоизменении и словообразовании прилагательных украинского языка. Проведения декомпозиции метода и разработано алгоритмическое обеспечение его основных структурных составляющих по результатам контент-анализа текста. Теоретически обнаружены способы улучшения показателей эффективности поиска ключевых слов, в том числе плотности ключевиков в тексте. На основе разработанного программного обеспечения получены результаты экспериментальной апробации предложенного метода контент-мониторинга для определения ключевых слов в научных текстах технического профиля. Выявлено, что для выбранной экспериментальной базы из 100 работ лучших результатов по критерию плотности достигает метод анализа статьи без начальной обязательной информации и без списка литературы, но с проверкой уточненных заблокированных слов и уточненного тематического словаря.

Ключевые слова: текст, украиноязычный, алгоритм, контент-мониторинг, ключевые слова, контент-анализ, Стеммер Портера, лингвистический анализ, синтаксический анализ.

Bisikalo O. V.¹, Vysotska V. A.²

¹F.D., professor, Dean of Faculty for Computer Systems and Automation, Vinnytsia National Technical University, Vinnytsia, Ukraine

²Phd, associate professor of Information Systems and Networks Department, Lviv Polytechnic National University, Lviv, Ukraine

IDENTIFYING KEYWORDS ON THE BASIS OF CONTENT MONITORING METHOD IN UKRAINIAN TEXTS

The task of developing algorithmic providing processes of content monitoring for the problem solution of determining a keyword in Ukrainian text is solved. The formal justification of content monitoring in text using Porter stemmer is considered. The basis of the stemming

modification is the known results of morpheme and word building structure derivatives classification in Ukrainian language, affix combinatorics patterns identification, modeling the structural organization of verbs and suffixal nouns and morphological modifications in the verb inflection and word formation and inflection of adjectives in Ukrainian language. The method decomposition is conducted and the algorithmic software of its basic structural components of the text content analysis results is developed. Theoretically means to improve the performance indicators of keywords search are identified, including keyword density in text. Based on the software obtained results of experimental testing of the proposed method of content monitoring to keywords identification in scientific texts of technical profile are developed. It is detected that the chosen experimental base of 100 works the article analysis method the without the initial required information and without the reference list reaches the best results for the density criterion, but with the specified blocked words and qualifying thematic dictionary verification.

Keywords: text, a Ukrainian, algorithm, content monitoring, keywords, content analysis, Porter stemmer, linguistic analysis, parsing.

REFERENCES

1. Berko A., Vysotska V., Pasichnyk V. *Systemy elektronnoyi kontent-komertsiyi*. Leningrad, NULP, 2009, 612 p.
2. Vysotska V., Pasichnyk V., Scherbyna J., Shestakevych T. *Matematychna lnhvistyka*. Leningrad, Novyy Svit-2000, 2012, 359 p.
3. Nayefektyvnishi metody zaluchennya potentsiynih kliyentiv [Electronic resource]. Tsentr resursiv yakosti trafiku oholoshen, Google AdWords. Access mode: http://www.google.com/intl/uk_ALL/ads/adtrafficquality/advertisers/best-practices-for-generating-leads.html. Title from the screen.
4. Nechetkyy poysk v tekste y slovare [Electronic resource]. Access mode: <http://habrahabr.ru/post/114997/>. Title from the screen.
5. Realyzatsyy alhorytmov. Rasstoyanye Levenshteyna [Electronic resource]. Access mode: http://ru.wikibooks.org/wiki/Реализация_алгоритмов/Расстояние_Левенштейна. Title from the screen.
6. Zadacha o rasstoyaniyu Damerau-Levenshteyna [Electronic resource]. Access mode: http://neerc.ifmo.ru/wiki/index.php?title=%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%BE_%D1%80%D0%B0%D1%81%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D0%B0%D0%BC%D0%B5%D1%80%D0%B0%D1%83%D0%9B%D0%B5%D0%B2%D0%B5%D0%BD%D1%88%D1%82%D0%B5%D0%B9%D0%BD%D0%B0. Title from the screen.
7. Nasonov D. *Funktsyya Levenshteyna* [Electronic resource]. Access mode: <http://rain.ifmo.ru/cat/data/theory/unsorted/levenshtein-2006/article.pdf>. Title from the screen.
8. Levenshteyn, kotoryy sravnivaet stroki [Electronic resource]. Web development. Access mode: <http://dayte2.com/levenshtein>. – Title from the screen.
9. Vychislenie rasstoyaniya Levenshteyna mezhdu dvumya strokami [Electronic resource]. Access mode: <http://wm-help.net/lib/book/827961078/78>. Title from the screen.
10. Porter stemmer [Electronic resource]. Access mode: <http://labs.abcvg.com/stemmer/index.php>. Title from the screen.
11. Moseichuk V. Porter stemming algorithm for Ukrainian languages [Electronic resource]. Access mode: http://www.marazm.org.ua/document/stemer_ua/. Title from the screen.
12. Steming [Electronic resource]. Access mode: <https://uk.wikipedia.org/wiki/Стемінг>. Title from the screen.
13. Russian stemming algorithm [Electronic resource]. Access mode: <http://snowball.tartarus.org/algorithms/russian/stemmer.html>. Title from the screen.
14. Porter stemmer-realizatsiya algoritma stemmera Portera dlya russkogo yazyka na chistom funktsionalnom yazyke Clojure [Electronic resource]. Access mode: <https://github.com/allaud/porter-stemmer>. Title from the screen.
15. The Porter Stemming Algorithm-Porter's homepage [Electronic resource]. Access mode: <http://tartarus.org/~martin/PorterStemmer/>. Title from the screen.
16. The Porter Stemming Algorithm – Project «Snowball» [Electronic resource]. Access mode: <http://snowball.tartarus.org/algorithms/porter/stemmer.html>. – Title from the screen.
17. The English (Porter2) stemming algorithm – Project «Snowball» [Electronic resource]. Access mode: <http://snowball.tartarus.org/algorithms/english/stemmer.html>. Title from the screen.
18. Porter M. F. An algorithm for suffix stripping [Electronic resource], Program, 1980, Vol. 14, No. 3, pp. 130–137. Access mode: http://telemat.det.unifi.it/book/2001/wchange/download/stem_porter.html. Title from the screen.
19. Willett P. The Porter stemming algorithm: then and now [Electronic resource], Program: Electronic Library and Information Systems, 2006, B. 3, Vol. 40, pp. 219–223. ISSN 0033-0337. – Access mode: <http://eprints.whiterose.ac.uk/1434>. Title from the screen.
20. Senyk M. Vilnyy alhorytm steminhu dlya ukrayinskoyi movy [Electronic resource]. Access mode: http://www.senyk.poltava.ua/projects/ukr_stemming/stemming_about.html. Title from the screen.
21. Senyk M. Instrument dlya poshuku sliv z odnakovymy zakinchenyamy [Electronic resource], Access mode: http://www.senyk.poltava.ua/projects/ukr_stemming/word_by_ending.html. Title from the screen.
22. Senyk M. Statychno derevo zakinchen [Electronic resource]. Access mode: http://www.senyk.poltava.ua/projects/ukr_stemming/ukr_endings.html#dyn. Title from the screen.
23. Senyk M. Demo steminhu dlya ukrayinskoyi movy [Electronic resource]. Access mode: http://www.senyk.poltava.ua/projects/ukr_stemming/demo.html. Title from the screen.
24. Veroyatnostny morfologicheskyy analizator russkogo i ukrainskogo yazykov [Electronic resource]. Access mode: <http://www.keva.ru/stemka/stemka.html>. Title from the screen.
25. Steming [Electronic resource]. Access mode: <https://ru.wikipedia.org/wiki/Стеминг>. Title from the screen.
26. Lovins J. B. Development of a stemming algorithm, *Mechanical Translation and Computational Linguistics*, 11:22–31. – 1968.
27. Jongejan B., Dalianis H. Automatic training of lemmatization rules that handle morphological changes in pre-, in- and suffixes alike [Electronic resource], In the Proceeding of the ACL-2009, Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Singapore, August 2–7, 2009, pp. 145–153. Access mode: <http://www.aclweb.org/anthology/P/P09/P09-1017.pdf>. Title from the screen.
28. Virohidnyy morfologichnyy analizator rosiyskoyi ta ukrayinskoyi [Electronic resource]. – Access mode: <http://www.keva.ru/stemka/stemka.html>. – Title from the screen.
29. Modul Drupal dlya steminha ukrayinskoyi. Novyy modul dlya alhorytmu Stema dlya Ukrayinskoho poshuku z vydilennyam koreniv [Electronic resource]. – Access mode: <http://drupal.ua/node/1170>. – Title from the screen.
30. Steminh Portera dlya ukrayinskoyi movy [Electronic resource]. – Access mode: http://www.marazm.org.ua/document/stemer_ua/. – Title from the screen.
31. Hardcoded stemmer for Ukrainian [Electronic resource]. – Access mode: <https://github.com/vgrichina/ukrainian-stemmer>. – Title from the screen.
32. Perestoronin, P. Stemmer Portera dlya russkogo yazyka [Electronic resource]. – P. Perestoronin // Access mode: <http://blog.eigene.in/post/49598738049/snowball>. – Title from the screen.

¹Канд. техн. наук, доцент, доцент кафедры автоматизации и компьютерно-интегрированных технологий Харьковского национального автомобильно-дорожного университета, Харьков, Украина

²Канд. техн. наук, доцент, доцент кафедры автоматизации и компьютерно-интегрированных технологий Харьковского национального автомобильно-дорожного университета, Харьков, Украина

³Инженер, младший научный сотрудник кафедры автоматизации и компьютерно-интегрированных технологий Харьковского национального автомобильно-дорожного университета, Харьков, Украина

ПОВЫШЕНИЕ ТОЧНОСТИ ОЦЕНКИ СОСТОЯНИЯ ДИНАМИЧНЫХ ОБЪЕКТОВ КОМПЛЕКСОМ MATLAB-ARDUINO ПРИ ПРОЕКТИРОВАНИИ КИБЕР-ФИЗИЧЕСКИХ СИСТЕМ

Решена задача повышения эффективности взаимодействия MATLAB и Arduino при проектировании кибер-физических систем путем внесения изменений в реализацию стандартного протокола обмена со стороны Arduino. Предложено при запросе MATLAB на чтение данных с первого аналогового порта Arduino выполнять аналогово-цифровое преобразование данных со всех требуемых аналоговых портов с дальнейшей последовательной передачей полученных данных в MATLAB, что позволяет повысить качество управления динамичными процессами за счет уменьшения области неопределенности состояния многомерной быстродействующей системы управления. Кроме того, предложено функции предварительной обработки показаний датчиков выполнять средствами Arduino, что повышает гибкость кибер-физической системы за счет возможности изменения аппаратного обеспечения без изменения программного кода и протокола обмена MATLAB. Формальное описание взаимодействия MATLAB и Arduino позволяет реализовать протокол обмена с использованием беспроводных интерфейсов, микропроцессорных устройств и платформ, не совместимых с Arduino.

Ключевые слова: кибер-физическая система, MATLAB, Arduino, протокол обмена, период опроса.

НОМЕНКЛАТУРА

CPS – кибер-физическая система;

e – ошибка управления;

k – момент дискретизации по времени;

K_p – коэффициент усиления пропорционального канала регулятора;

T_i, T_d – постоянные времени интегрирующего и дифференцирующего каналов регулятора;

T_R – период чтения данных;

T_s – период дискретизации по времени;

T_w – период записи данных;

u – управляющее воздействие;

val – символьная переменная.

ВВЕДЕНИЕ

В результате взаимодействия встроенных систем управления и сетевых технологий возникла новая технология управления объектами и процессами – кибер-физическая система (cyber-physical system, CPS), которая характеризуется тесной интеграцией и координацией между вычислительными и физическими процессами при помощи сетевых технологий [1]. CPS широко внедряются в самые разнообразные отрасли производства: транспорт, энергетику, машиностроение, робототехнику, металлургию, а также в биомедицинские системы.

Проектирование CPS требует наличия более надежных моделей физических процессов, протекающих в системах управления [2, 3, 4]. От того, как модель соотносится с реальностью, зависит работоспособность CPS.

В промышленной практике аппаратные и программные составляющие многих систем управления до сих пор разрабатываются отдельно, без учета их взаимодействия между собой и с физическим миром. И уже после разработки системы управления, проверки ее на моделях,

устраняется влияние различного рода неопределенностей путем использования специальных методов настройки. Этот процесс является трудоемким и дорогостоящим, а с усложнением систем – практически неосуществимым.

Альтернативным подходом к проектированию CPS является использование технологии модельно-ориентированного проектирования [4]. Ее суть заключается в том, что вместо физических прототипов в специализированном программном пакете создается имитационная модель объекта управления, на которой отрабатывается и совершенствуется разрабатываемый алгоритм управления. В случае достижения алгоритмом поставленной задачи выполняется автоматическая генерация программного кода для целевой платформы. Чаще всего такое проектирование проводят в средах LabVIEW компании National Instruments и MATLAB/Simulink компании Mathworks [5, 6]. Возможности этих продуктов существенно расширяются при использовании связи с физическим миром, для чего в них предусмотрены коммуникационные функции. Различные датчики и исполнительные устройства могут подключаться через COM порт или USB. Это позволяет перейти от имитационных моделей к гибридным, в которых сочетаются как модели сложных объектов, так и реальные физические устройства.

Однако использование указанных функций при необходимости реализации протоколов обмена вызывает определенные трудности, и до недавнего времени было доступно лишь специалистам узкого профиля, поэтому встречалось довольно редко.

Ситуация изменилась с появлением платформы Arduino, которая существенно упрощает разработку CPS. В частности, унифицированные функции взаимодействия с сервером на Arduino получили поддержку в MATLAB путем разработки приложения «Arduino

Support from MATLAB» [7]. Данная работа направлена на повышение эффективности взаимодействия MATLAB и платформы Arduino при гибридном (модель/устройство) подходе к проектированию систем управления.

1 ПОСТАНОВКА ЗАДАЧИ

Привлечение MATLAB позволяет не только упростить процедуру проектирования, но и решать достаточно сложные задачи управления, используя ПЭВМ с установленным MATLAB в качестве ПЭВМ верхнего уровня, в то время как платформа Arduino решает задачи по получению и преобразованию информации о значениях контролируемых параметров системы управления и реализует протокол обмена доступа к серверу (рис. 1).

Реализация текстового обмена вида запрос/ответ между MATLAB и Arduino осуществляется по последовательному USB интерфейсу с использованием механизма виртуальных COM-портов (рис. 2) при помощи команд, сходных с командами среды Arduino. Известны реализации серверных протоколов [7], где для повышения гибкости управления каждая команда оперирует одним из 14 цифровых или одним из 6 аналоговых выводов. В текстовых запросах/ответах каждому порту (pin 0...13) Arduino ставится в соответствие символьное значение a...n переменной val (табл. 1).

Основные команды, которые содержит стандартный текстовый протокол обмена MATLAB и Arduino, реализованный в MATLAB и в скетче adiosrv_UNO.ino для Arduino (настройка линий цифрового ввода-вывода, собственно цифровой ввод-вывод, аналоговый ввод и псевдоаналоговый вывод ШИМ), содержат код команды (функции) и строку параметров (рис. 3):

```
<function><pin>
<function><pin><val>
```

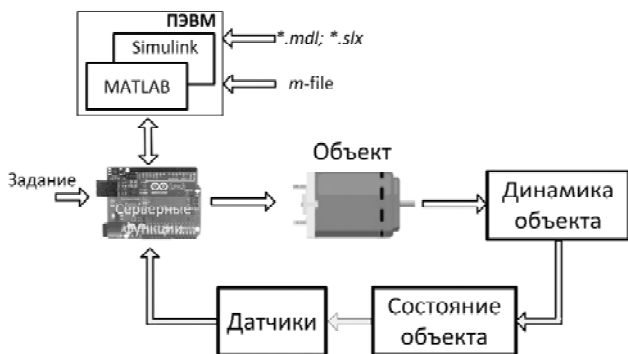


Рисунок 1 – Структура системы управления с Arduino

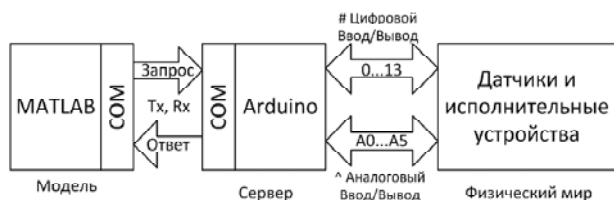


Рисунок 2 – Структурная схема взаимодействия MATLAB и Arduino

$function \in \{ '0' \dots '4' \} = \{ 0x30 \dots 0x34 \} = \{ 48 \dots 52 \}$ – код запроса/ команды MATLAB;

$pin \in \{ 'a' \dots 'n' \} = \{ 0x61 \dots 0x6E \} = \{ 97 \dots 110 \}$ – символьные обозначения цифровых линий pin0 ... pin13;

$pin \in \{ 'a' \dots 'f' \} = \{ 0x61 \dots 0x66 \} = \{ 97 \dots 102 \}$ – символьные обозначения аналоговых линий A0 ... A5;

$val \in \{ 0 \dots 255 \}$ – байтовые значения псевдоаналогового вывода (ШИМ).

$val \in \{ '0', '1' \}$ – бинарное значение для задания режима или состояния линии цифрового ввода-вывода.

Например, команда '0e1' устанавливает цифровой порт 4 ('e') в состояние вывода; команда '3b' дает указание на выполнение аналого-цифрового преобразования сигнала с A1 ('b'), а команда '4fz' выдает через pin 5 ('f') параметр ШИМ 122 ('z' в ASCII равен 122).

Однако использование рассмотренного стандартного протокола обмена MATLAB – Arduino выявило следующие недостатки, которые затрудняют его применение при исследовании и управлении динамичными объектами в режиме реального времени.

1) Непостоянство периода опроса портов в зависимости от загрузки ПЭВМ. В то же время при реализации обратной связи информацию от датчиков о состоянии объекта необходимо получать со строгой привязкой ко времени.

Кроме того, последовательный опрос нескольких сигналов, определяющих состояние многомерной системы, приводит к значительному временному сдвигу получаемых отсчетов (τ_1, τ_2 на рис. 4), который также существенно зависит от скорости работы и загрузки компьютера и изменяется во времени (среднее время записи/ считывания с одного порта составляет 15 мс). Это затрудняет оценку состояния быстро меняющихся, динамичных процессов. Подобная проблема была также описана в [8–11]. Практически целесообразно, чтобы информация от всех портов поступала одновременно за одно обращение, либо, при последовательном чтении, соответствовала состояниям объекта в предельно близкие моменты времени.

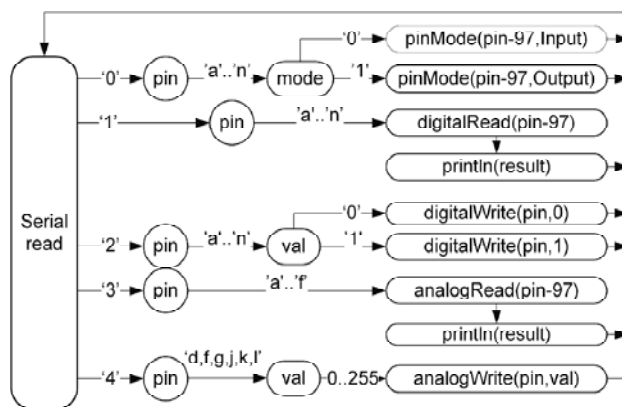


Рисунок 3 – Графическое представление протокола обмена MATLAB – Arduino

Таблица 1 – Соответствие номера порта символьной переменной val

val	a	b	c	d	e	f	g	h	i	j	k	l	m	n
pin	0	1	2	3	4	5	6	7	8	9	10	11	12	13

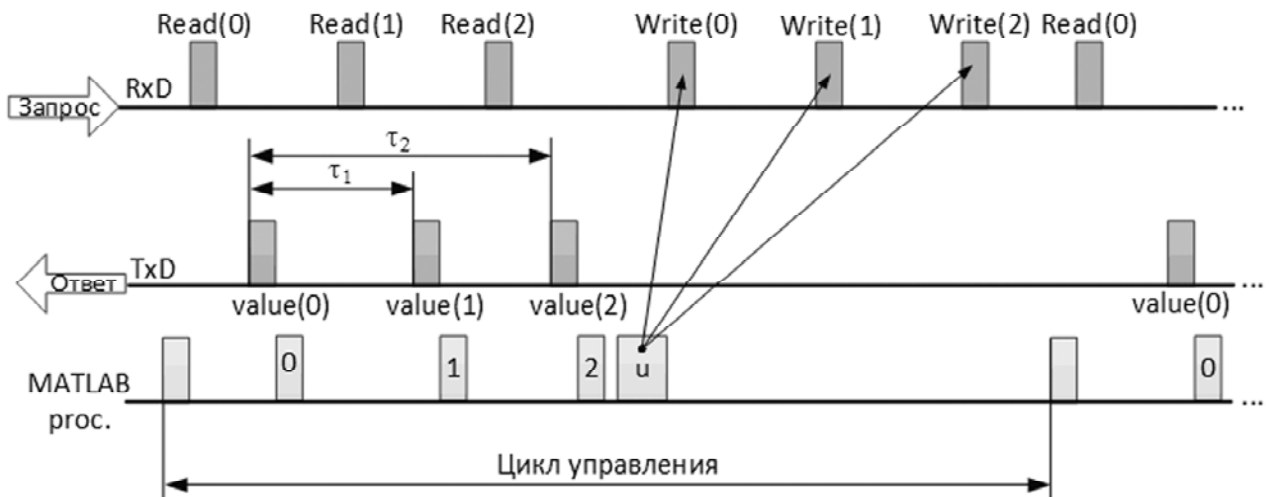


Рисунок 4 – Задержка определения состояния объекта при последовательном опросе портов Arduino

2) При использовании датчиков возникает необходимость предварительной обработки их показаний. Выполнять ее средствами MATLAB нецелесообразно, т.к. в этом случае имеет место жесткая привязка программы MATLAB к аппаратной части системы управления.

2 ОБЗОР ЛИТЕРАТУРЫ

Совместное использование MATLAB/Simulink и Arduino при проектировании кибер-физических систем в литературе освещено достаточно широко. Так в [9, 10, 12, 13] приведены примеры систем управления скоростью вращения вала двигателя постоянного тока как сервопривода различных объектов. В работе [14] представлено микропроцессорное устройство контроля параметров трехфазной сети по текущей информации от датчиков тока и напряжения. Устройство реализовано на плате Arduino Galileo. Для анализа точности выполняемых расчетов в среде MATLAB/Simulink разработана компьютерная модель микропроцессорного устройства, учитывающая дискретные преобразования аналоговых входных сигналов токов и напряжений по времени и уровню, измеренных непрерывными датчиками в трехфазной сети.

В [15] описана система управления уровнем жидкости в резервуаре на основе аппарата нечеткой логики. При этом для синтеза нечеткого регулятора использовались возможности MATLAB, а плата Arduino служила для получения данных от датчиков и генерации ШИМ-сигнала для управления насосом.

В [16] предложена система управления движением мобильного робота на основе Arduino, для программирования которой используются возможности MATLAB/Simulink.

Однако приведенные выше, и подавляющее большинство других публикаций посвящены описанию процедуры проектирования и принципов работы соответствующих систем управления и практически не рассматривают процессы, возникающие при взаимодействии Arduino и MATLAB. Результаты экспериментов с платой Arduino для оценки реализуемой ею точности измерения временных интервалов представлены в [11]. Однако в этой работе не предлагается методов повышения эффективности использования Arduino для управления объектами в реальном масштабе времени.

3 МАТЕРИАЛЫ И МЕТОДЫ

Рассмотренные выше недостатки стандартного протокола обмена между MATLAB и Arduino снижают эффективность использования гибридного подхода к проектированию систем управления. В связи с этим ниже предлагается модифицированный протокол обмена между MATLAB и Arduino, позволяющий устранить указанные недостатки.

Синхронизацию считываемых с портов данных предлагается осуществлять следующим образом. При получении запроса от MATLAB на чтение данных с первого аналогового порта (рис. 5), Arduino производит опрос всех требуемых аналоговых входов, тем самым формируется выборка данных, которые затем, при поступлении соответствующего запроса (функция '3') передаются в MATLAB. И хотя в Arduino опрос осуществляется также последовательно и занимает определенное время (считывание значения с аналогового входа занимает 14 тактов частоты синхронизации АЦП с периодом 8мкс, тогда $T_{\text{ацп}} = 148 = 112 \text{ мкс}$), это значительно меньше, чем при программном опросе с помощью MATLAB (около 17 мс).

На рис. 5 жирными (красными) стрелками показано прохождение потоков данных между Arduino, MATLAB и в приложениях *.m, *.ino.

Функцию предварительной обработки показаний с датчиков предлагается возложить на платформу Arduino, тем более что она имеет значительные вычислительные возможности. Так, после выполнения аналогово-цифрового преобразования данных со всех требуемых аналоговых портов, Arduino производит обработку полученного кода и определяет численные значения контролируемых параметров в соответствующих единицах измерения, а затем передает полученные данные из Arduino на ПЭВМ через последовательный порт. Приведенная процедура позволяет более гибко подходить к выбору технических средств получения информации о состоянии физических параметров системы управления. При этом нет необходимости вносить изменения в программу MATLAB.

С учетом всего сказанного выше в данном параграфе, соответствующий рис. 5 фрагмент модернизированного протокола обмена MATLAB-Arduino отображается рис. 6.

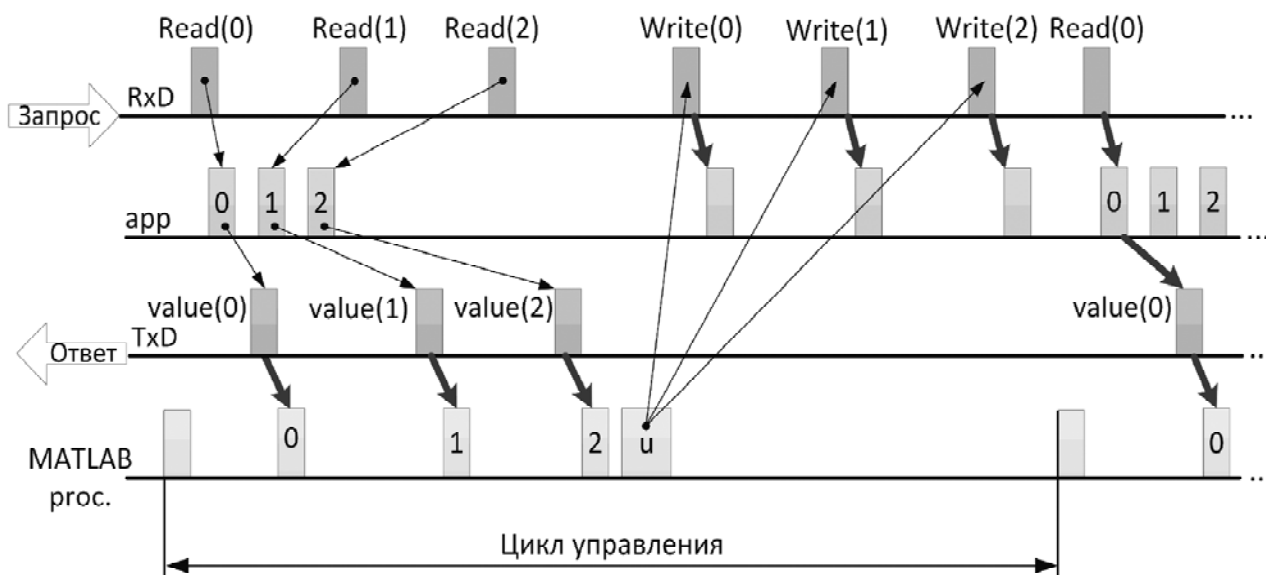


Рисунок 5 – Диаграмма опроса/ответа обмена при синхронизации считывания аналоговых сигналов

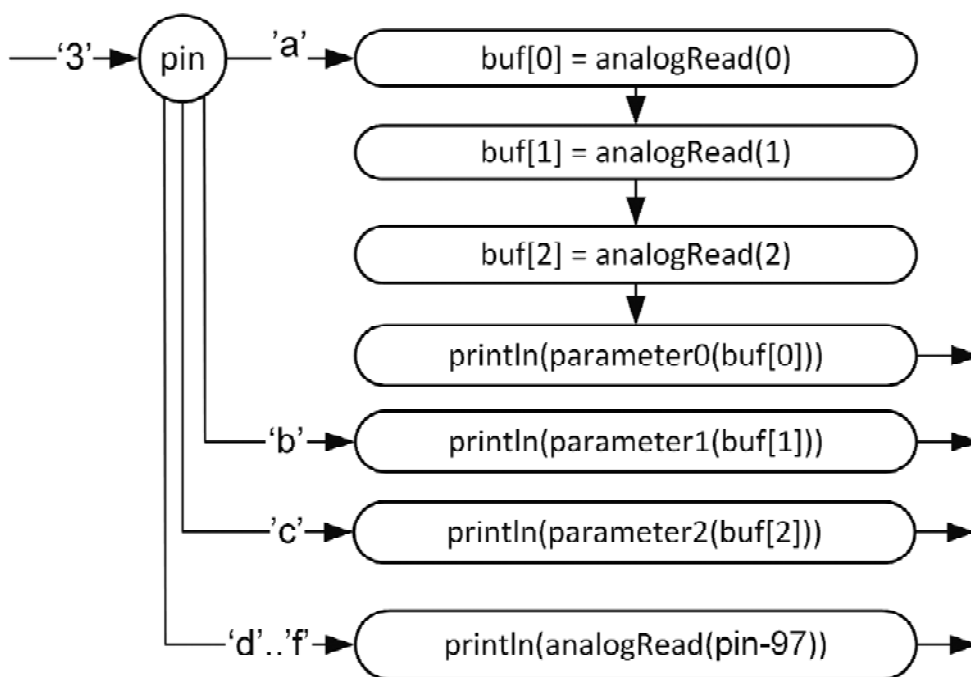


Рисунок 6 – Фрагмент модифицированного протокола обмена MATLAB-Arduino

На рис. 6 функция '3a' запрашивает результат аналого-цифрового преобразования сигнала с входа A0 ('a'), но последовательно выполняется преобразование для A0, A1, A2 и результаты сохраняются в буфере памяти (buf[0..2]) и впоследствии подвергаются обработке с помощью различных, в общем случае, функций parameter0(buf[0]), parameter1(buf[1]), parameter2(buf[2]). Преобразование обработанного кода buf[0] в текстовый формат и отправка по последовательному интерфейсу в ПЭВМ реализует функция println(parameter0(buf[0])). По запросам MATLAB '3b' и '3c' возвращаются результаты преобразования buf[1] и buf[2]. В то же время по запросам '3d'..'3f' выполняется аналого-цифровое преобразование сигналов с входов A3..A5 и возвращаются результаты в текстовом виде – println(analogRead(pin-97)).

Например, если 'a' ... 'c' (A0..A2) – входы для датчиков углового положения с разными характеристиками, то при указанной организации протокола в MATLAB возвращаются результаты преобразования напряжение-угол, скрывающие отличия датчиков. Аналогично можно организовать обработку для 1..6 произвольных аналоговых входов.

При необходимости жесткой цикличности опроса портов, например, при считывании информации с датчиков, функции отсчета временных интервалов целесообразно, как и буферизации данных, целесообразно возложить на Arduino. В этом случае MATLAB будет получать поток данных с переменной цикличностью, но массив полученных данных привязан к временной шкале.

4 ЭКСПЕРИМЕНТЫ

Для подтверждения наличия описанных в п.1 недостатков стандартного протокола обмена MATLAB-Arduino, а также проверки эффективности предлагаемой модификации протокола проведен ряд экспериментов на стенде, предназначенном для управления двигателем постоянного тока (ДПТ) посредством MATLAB и Arduino. Стенд (рис. 7) состоит из двигателя постоянного тока DC напряжением питания 5..12 В, ПЭВМ с MATLAB 2011b, Arduino Uno (MCU) с микроконтроллером Atmega328 с 32 кБ флеш-памяти (Flash), 1кБ ЭСППЗУ (EEP ROM) и 2 кБ SRAM, электронного ключа (ES), датчика скорости вращения двигателя (S). В качестве ПЭВМ использовался ноутбук с процессором Intel Core i3-330M 2,13 ГГц, ОЗУ 3ГБ и операционной системой Windows 7 Professional x64.

Задание желаемой скорости V_3 вращения осуществляется напряжением $U(V_3) = 0..5В$ с выхода потенциометра, а в качестве датчика скорости используется датчик Холла, частота импульсов $f(V)$ на выходе которого определяется скоростью V вращения электродвигателя с 12-полюсным магнитом на валу. Преобразователь частота-напряжение (f/U) формирует напряжение $U(V)$, пропорциональное скорости вращения двигателя. Управление двигателем производится посредством ШИМ-сигнала (PWM), параметры которого определяются на основе ПИД закона (1):

$$u_k = (u_{k-1} + K_p(e_k - e_{k-1}) + \frac{K_p T_s}{T_i} e_k + \frac{K_p T_d}{T_s} (e_k - 2e_{k-1} + e_{k-2})) \cdot \frac{255}{1023}. \quad (1)$$

При расчете параметров ПИД-регулятора использовались встроенные возможности MATLAB.

Взаимодействие MATLAB с Arduino осуществлялось на основе приведенного в п. 3 модифицированного протокола обмена.

5 РЕЗУЛЬТАТЫ

Рис. 8 отображает осциллограммы обмена между MATLAB и Arduino UNO посредством стандартного протокола. Период обмена включает: посылку MATLAB команды чтения аналогового входа A0 на вход последовательного обмена Arduino (RxD); прием значения (value), передаваемого по линии TxD Arduino; посылку команды аналогового вывода на pin 9, на которую ответ не передается. На рис. 8 видно изменение периода опроса D, зависящее от загрузки компьютера. Горизонтальная развертка составляет 2,5 мс/деление, вертикальная – 5В/деление, т.е. период T_1 составляет 12,5 мс, а период T_2 – 10 мс.

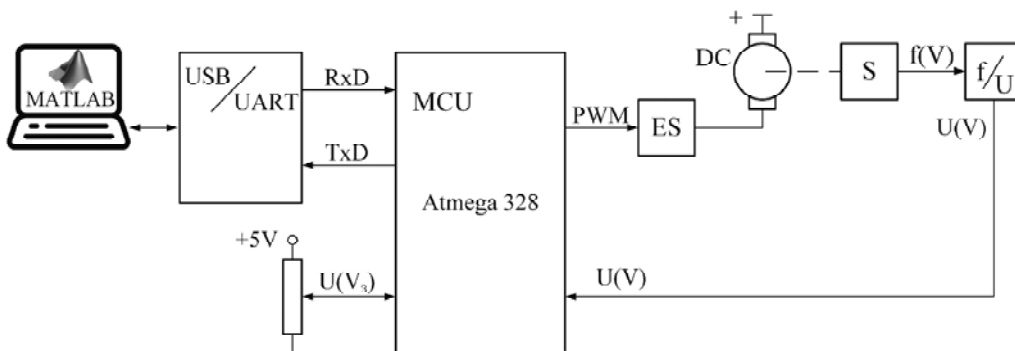


Рисунок 7 – Структурная схема стенда

Результаты серии экспериментов с обращением к аналоговым портам Arduino через MATLAB свидетельствуют о том, что основной вклад в вариацию периодов обмена вносит период чтения T_R (рис. 8), который, при обращении к одному порту, изменяется случайным образом в диапазоне $T_R = 8 \div 14$ мс. В то же время, период записи T_W изменяется не так заметно и в среднем составляет около 3 мс. С увеличением количества задействованных портов период их опроса увеличивается практически пропорционально.

Динамику системы управления двигателем при действии внешней нагрузки отображает рис. 9, где $U_x = U(V_3)/5 \cdot 1023$ – результат преобразования напряжения $U(V_3)$, снимаемого с переменного резистора, в код 0..1023; $U_v = U(V)/5 \cdot 1023$ – результат преобразования напряжения $U(V)$, снимаемого с датчика скорости в код 0..1023; pwm – параметр ШИМ в диапазоне 0..255; int – выход интегрирующего канала ПИД-регулятора.

6 ОБСУЖДЕНИЕ

Результаты экспериментальных исследований подтверждают преимущества предложенного в данной работе протокола. Реализация процедуры синхронизации считываемых с портов данных позволяет на два порядка (с 15 ÷ 17 мс до 112 мкс) уменьшить время работы с аналоговыми портами Arduino. Это, а также уменьшение зависимости системы управления от непостоянства периода опроса портов Arduino, приводит к уменьшению неопределенности относительно области возможных состояний объекта. Очевидно, что при управлении инерционными объектами полученный результат не окажет заметного влияния на качество управления, однако при управлении быстро меняющимися, динамичными процессами, использование предложенного протокола дает ощутимый эффект.

ВЫВОДЫ

Распространению кибер-физических систем управления способствует применение современных методов модельно-ориентированного проектирования. При этом важную роль играет использование недорогих, относительно простых в использовании и в то же время эффективных периферийных устройств, обеспечивающих коммуникацию между моделью объекта или процесса управления и физическим миром. Одной из таких устройств является платформа Arduino.

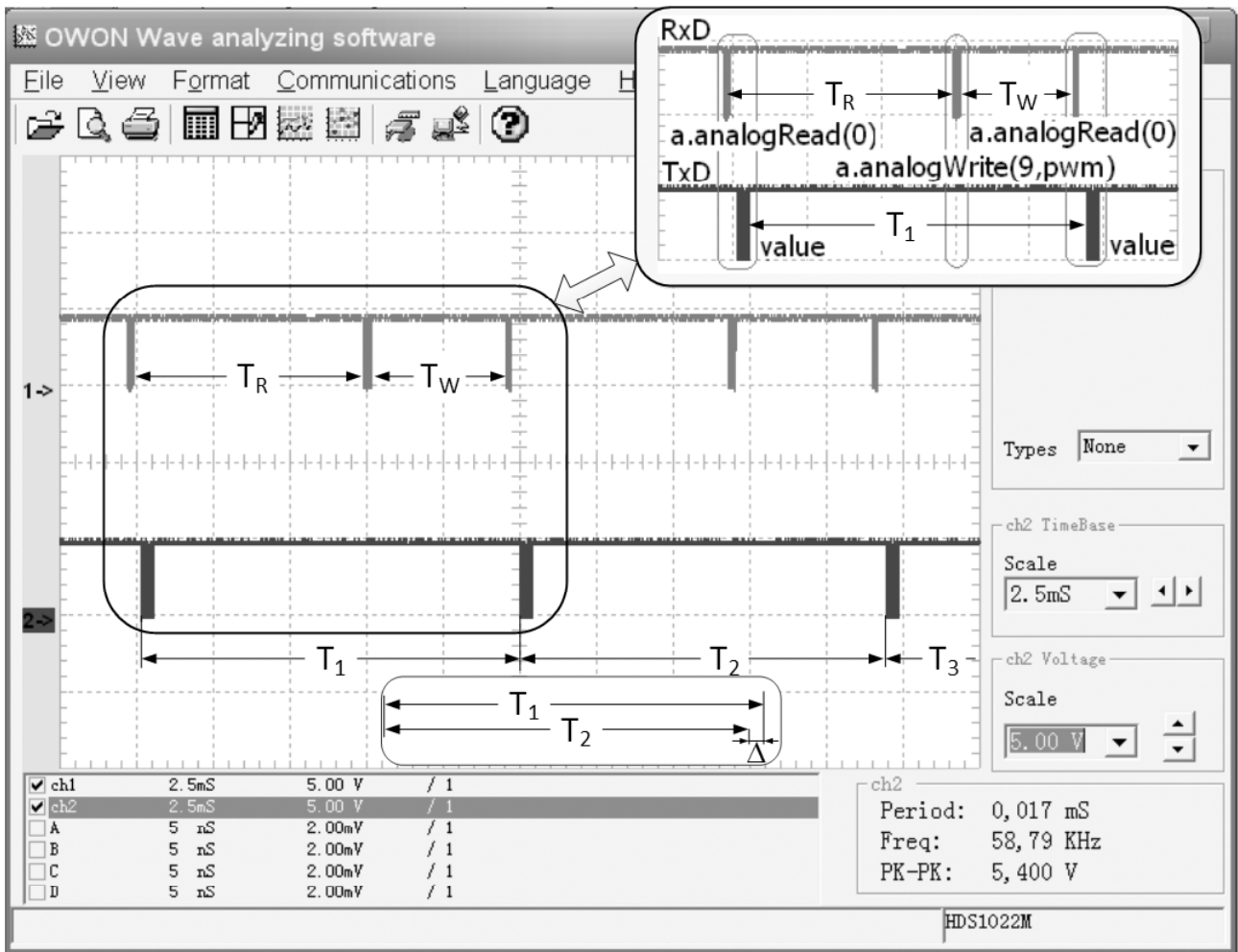


Рисунок 8 – Осциллограммы обмена между MATLAB и Arduino

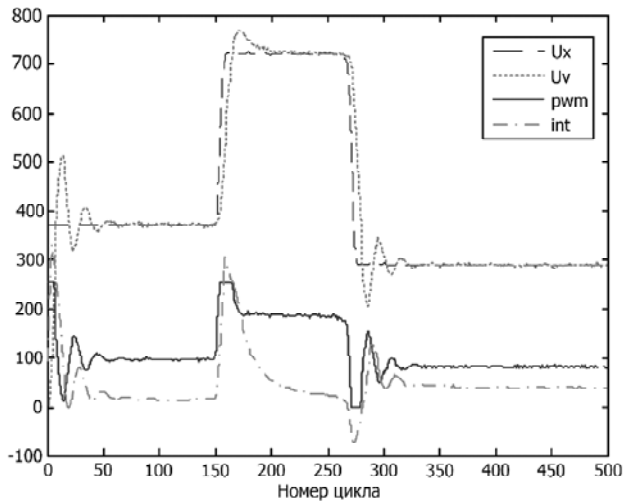


Рисунок 9 – Поведение системы управления двигателем

Однако использование стандартного протокола Arduino для обмена данными между ПЭВМ и физическим миром имеет определенные недостатки, связанные с непостоянством периода времени между получением выборки данных, невозможностью параллельного получения данных о состоянии объекта, а также нерациональным использованием ресурсов ПЭВМ.

Предложенный в данной работе протокол обмена данными между MATLAB и Arduino позволяет оптимизировать процесс взаимодействия системы управления с объектом и повысить качество управления динамичными многомерными процессами за счет уменьшения запаздывания в определении состояния многомерных объектов при последовательном обращении к аналоговым портам Arduino.

Выполнение предварительной обработки показаний датчиков средствами Arduino не влияет непосредственно на качество управления, однако повышает гибкость CPS за счет возможности изменения аппаратного обеспечения без изменения программного кода MATLAB.

Рассмотренный протокол может быть модифицирован для применения с другими средами моделирования, например, LabVIEW, а также реализован на основе иных платформ (Raspberry и проч.) и микроконтроллерах известных семейств – PIC, MSP430, MCS51, ARM, MIPS и т.д. Последовательный обмен может быть построен с использованием беспроводных Bluetooth (HC05 и др.), WiFi (ESP8266 и др), XBee и прочих модемов. Это расширяет область применения предлагаемых подходов.

СПИСОК ЛИТЕРАТУРЫ

1. Sangiovanni-Vincentelli A. Taming Dr. Frankenstein: Contract-Based Design for Cyber-Physical Systems / A. Sangiovanni-

- Vincentelli, W. Damm, R. Passerone // *European Journal of Control.* – 2012. – Vol. 18, № 3. – P. 217–238. DOI:10.3166/ejc.18.217-238.
2. Черняк Л. Киберфизические системы на старте / Л. Черняк // *Открытые системы. СУБД.* – 2014. – № 02 (198) – С. 10–13.
 3. Park Kyung-Joon Cyber-physical systems: Milestones and research challenges / Kyung-Joon Park, Rong Zheng, Xue Liu // *Computer Communications* – 2012. – № 36. – P. 1–7. DOI 10.1016/j.comcom.2012.09.006.
 4. Introduction to Embedded Systems – A Cyber-Physical Systems Approach / E. A. Lee and S. A. Seshia. Second Edition. [Electronic resource]. – Access mode: <http://LeeSeshia.org>, 2015.
 5. Jensen J. C. A model-based design methodology for cyber-physical systems / J. C. Jensen, D. H. Chang, E. A. Lee // *Wireless Communications and Mobile Computing Conference (IWCMC): 7th International conference, Istanbul, 4–8 July 2011: proceedings.* – IEEE, 2011. – P. 1666–1671. DOI: 10.1109/IWCMC.2011.5982785.
 6. Модельно-ориентированное проектирование программного обеспечения для встраиваемых систем в среде Matlab/Simulink / [Г. К. Топораш, А. В. Мазур, Д. А. Ковальчук, А. А. Пушкин] // *Автоматизація технологічних і бізнес-процесів.* – 2014. – Т. 17, № 17. – С. 26–29.
 7. Практические стратегии для перехода на модельно-ориентированное проектирование встроенных приложений / [Д. Венси, Д. Эрик, К. Ларри, Р. Винод] // *Компоненты и технологии.* – 2011. – Т. 10, №. 123. – С. 172–180.
 8. Arduino Support from MATLAB [Electronic resource]. – Access mode: <http://www.mathworks.com/hardware-support/arduino-matlab.html>.
 9. Al-Busaidi A. M. Development of an educational environment for online control of a biped robot using MATLAB and Arduino / A. M. Al-Busaidi // *Mechatronics: 9th France-Japan & 7th Europe-Asia Congress on and Research and Education in Mechatronics (REM), 21–23 Nov. 2012, 13th Int'l Workshop on.* – IEEE, 2012. – P. 337–344. DOI: 10.1109/MECATRONICS.2012.6451030.
 10. Barber R. Control Practices using Simulink with Arduino as Low Cost Hardware / R. Barber, M. Horra, J. Crespo. // *The 10th IFAC Symposium Advances in Control Education, August 28-30, 2013: proceedings.* – University of Sheffield, Sheffield, UK, 2013. – P. 250–255. DOI: 10.3182/20130828-3-UK-2039.00061.
 11. D'Ausilio A. Arduino: A low-cost multipurpose lab equipment / Alessandro D'Ausilio // *Behavior Research Methods.* – 2012. – Vol. 44, Issue 2. – P. 305–313. DOI 10.3758/s13428-011-0163-z
 12. Online Performance Optimization of a DC Motor Driving a Variable Pitch Propeller / [R. Cohen, D. Miculescu, K. Reilley and other] // *arXiv preprint arXiv:1310.0133.* – 2013.
 13. Bawa D. Fuzzy control based solar tracker using Arduino Uno / D. Bawa, C.Y. Patil // *International Journal of Engineering and Innovative Technology (IJEIT).* – 2013. – Vol. 2, Issue 12. – P. 179–187.
 14. Омельченко Е. Я. Устройство контроля трехфазной сети на основе Arduino-совместимых микроконтроллеров / Е. Я. Омельченко, А. В. Белый и др. // *Электротехника.* – 2014. – Т. 1, № 1. – С. 17–22.
 15. Krivić S. Design and implementation of fuzzy controller on embedded computer for water level control / S. Krivić, M. Hujdur, A. Mrzić, S. Konjicija // *MIPRO 2012: 35th International Convention, Opatija, Croatia, 21–25 May 2012: proceedings.* – IEEE, 2012. – P. 1747–1751.
 16. Gartseev I. B. A low-cost real-time mobile robot platform (ArEduBot) to support project-based learning in robotics & mechatronics / I. B. Gartseev, L. F. Lee, V. N. Krovi // *2nd International Conference on Robotics in Education (RiE 2011), Vienna, Austria, September 15–16, 2011: proceedings* [Electronic resource]. – Access mode: http://mechatronics.eng.buffalo.edu/publications/conference/ICRE2011_IlyaLeeKrovi_ArEduBot.pdf

Статья поступила в редакцию 28.10.2015.
После доработки 09.11.2015.

Гурко О. Г.¹, Плахтеев А. П.², Плахтеев П. А.³

¹Канд. техн. наук, доцент, доцент кафедры автоматизації та комп'ютерно-інтегрованих технологій Харківського національного автомобільно-дорожнього університету, Харків, Україна

²Канд. техн. наук, доцент, доцент кафедри автоматизації та комп'ютерно-інтегрованих технологій Харківського національного автомобільно-дорожнього університету, Харків, Україна

³Інженер, молодший науковий співробітник кафедри автоматизації та комп'ютерно-інтегрованих технологій Харківського національного автомобільно-дорожнього університету, Харків, Україна

ПІДВИЩЕННЯ ТОЧНОСТІ ОЦІНКИ СТАНУ ДИНАМІЧНИХ ОБ'ЄКТІВ КОМПЛЕКСОМ MATLAB-ARDUINO ПРИ ПРОЕКТУВАННІ КІБЕР-ФІЗИЧНИХ СИСТЕМ

Розв'язано задачу підвищення ефективності взаємодії MATLAB і Arduino при проектуванні кібер-фізичних систем шляхом внесення змін у реалізацію стандартного протоколу обміну з боку Arduino. Запропоновано при запиті MATLAB на читання даних з першого аналогового порту Arduino виконувати аналогово-цифрове перетворення даних з усіх необхідних аналогових портів з подальшою послідовною передачею отриманих даних в MATLAB, що дозволяє підвищити якість управління динамічними процесами за рахунок зменшення області невизначеності стану багатовимірної швидкодіючої системи управління. Крім того, запропоновано функції попередньої обробки показань датчиків виконувати засобами Arduino, що підвищує гнучкість кібер-фізичної системи за рахунок можливості зміни апаратного забезпечення без зміни програмного коду і протоколу обміну MATLAB. Формальний опис взаємодії MATLAB і Arduino дозволяє реалізувати протокол обміну з використанням бездротових інтерфейсів, мікропроцесорних пристроїв та платформ, не сумісних з Arduino.

Ключові слова: кібер-фізична система, MATLAB, Arduino, протокол обміну, період опитування.

Gurko A. G.¹, Plakhteev A. P.², Plakhteev P. A.³

¹PhD, Associate professor, Associate professor of department of automation and computer-integrated technologies, Kharkiv National Automobile and Highway University, Kharkiv, Ukraine

²PhD, Associate professor, Associate professor of department of automation and computer-integrated technologies, Kharkiv National Automobile and Highway University, Kharkiv, Ukraine

³Engineer, Junior researcher of department of automation and computer-integrated technologies, Kharkiv National Automobile and Highway University, Kharkiv, Ukraine

ACCURACY INCREASE OF DYNAMIC OBJECTS STATE ESTIMATION BY A COMPLEX MATLAB-ARDUINO WHEN CYBER-PHYSICAL SYSTEMS DESIGNING

The problem of accuracy increase of MATLAB and Arduino interaction when cyber-physical systems designing by making changes into the implementation of the standard protocol of Arduino has been solved. It has been proposed to carry out the analog-digital conversion of all required analog pins with further sequential data transmission to MATLAB when requesting to read data from the first analog pin of Arduino.

It allows improving control quality of dynamic processes by reducing the uncertainty range of a high-speed multi-dimensional control system state. In addition, it has been offered to perform the preprocessing of sensors data by means of Arduino that increases the flexibility of cyber-physical systems due to the possibility of the hardware changing without MATLAB software and protocol changing. A formal description of MATLAB and Arduino interaction allows realizing the communications protocol using wireless interfaces as well as microprocessor units and platforms, which are not compatible with Arduino.

Keywords: cyber-physical system, MATLAB, Arduino, communications protocol, polling period.

REFERENCES

1. Sangiovanni-Vincentelli A., Damm W., Passerone R. Taming Dr. Frankenstein: Contract-Based Design for Cyber-Physical Systems, *European Journal of Control*, 2012, Vol. 18, No. 3, pp. 217–238. DOI:10.3166/ejc.18.217-238.
2. Chernjak L. Kiberfizicheskie sistemy na starte, *Otkrytye sistemy. SUBD*, 2014, No. 02 (198), pp. 10–13.
3. Park, Kyung-Joon, Rong Zheng, Xue Liu Cyber-physical systems: Milestones and research challenges, *Computer Communications*, 2012, No. 36, pp. 1–7. DOI 10.1016/j.comcom.2012.09.006.
4. Lee E. A. and Seshia S. A. Introduction to Embedded Systems – A Cyber-Physical Systems Approach. Second Edition. [Electronic resource]. Access mode: <http://LeeSeshia.org>, 2015.
5. Jensen J. C., Chang D. H., Lee E. A. A model-based design methodology for cyber-physical systems, *Wireless Communications and Mobile Computing Conference (IWCMC): 7th International conference, Istanbul, 4–8 July 2011: proceedings. IEEE*, 2011, P. 1666–1671. DOI: 10.1109/IWCMC.2011.5982785.
6. Toporash G. K., Mazur A. V., Koval'chuk D. A., Pushkin A. A. Model'no-orientirovanoe proektirovanie programnogo obespechenija dlja vstraivaemyh sistem v srede Matlab/Simulink, *Avtomatizacija tehnologichnih i biznes-procesiv*, 2014, Vol. 17, No. 17, pp. 26–29.
7. Vensi D., Jerik D., Larri K., Vinod R. Prakticheskie strategii dlja perehoda na model'no-orientirovanoe proektirovanie vstroennyh prilozhenij, *Komponenty i tehnologii*, 2011, Vol. 10, No. 123, pp. 172–180.
8. Arduino Support from MATLAB [Electronic resource]. Access mode: <http://www.mathworks.com/hardware-support/arduino-matlab.html>.
9. Al-Busaidi A. M. Development of an educational environment for online control of a biped robot using MATLAB and Arduino, *Mechatronics: 9th France-Japan & 7th Europe-Asia Congress on and Research and Education in Mechatronics (REM), 21–23 Nov. 2012, 13th Int'l Workshop on. – IEEE, 2012*, pp. 337–344. DOI: 10.1109/MECATRONICS.2012.6451030.
10. Barber R., Horra M., Crespo J. Control Practices using Simulink with Arduino as Low Cost Hardware, *The 10th IEAC Symposium Advances in Control Education, August 28–30, 2013: proceedings. – University of Sheffield, Sheffield, UK, 2013*, pp. 250–255. DOI: 10.3182/20130828-3-UK-2039.00061.
11. D'Ausilio A. Arduino: A low-cost multipurpose lab equipment, *Behavior Research Methods*, 2012, Vol. 44, Issue 2, pp. 305–313. DOI 10.3758/s13428-011-0163-z
12. Cohen R., Miculescu D., Reilley K., Pakmehr M., Feron E. Online Performance Optimization of a DC Motor Driving a Variable Pitch Propeller, *arXiv preprint arXiv:1310.0133*, 2013.
13. Bawa D., Patil C. Y. Fuzzy control based solar tracker using Arduino Uno, *International Journal of Engineering and Innovative Technology (IJEIT)*, 2013, Vol. 2, Issue 12, pp. 179–187.
14. Omel'chenko E. Ja., Belyj A. V. i dr. Ustrojstvo kontrolja trehfaznoj seti na osnove Arduino-sovmestimyh mikrokontrollerov, *Jelektrotehnika*, 2014, Vol. 1, No. 1, pp. 17–22.
15. Krivić S., Hujdur M., Mrzić A., Konjicija S. Design and implementation of fuzzy controller on embedded computer for water level control, *MIPRO 2012: 35th International Convention, Opatija, Croatia, 21–25 May 2012: proceedings. IEEE, 2012*, pp. 1747–1751.
16. Gartsev I. B., Lee L. F., Krovi V. N. A low-cost real-time mobile robot platform (ArEduBot) to support project-based learning in robotics & mechatronics, *2nd International Conference on Robotics in Education (RiE 2011), Vienna, Austria, September 15–16, 2011: proceedings [Electronic resource]*. Access mode: http://mechatronics.eng.buffalo.edu/publications/conference/ICRE2011_IlyaLeeKrovi_ArEduBot.pdf

EVALUATION OF COMPONENT ALGORITHMS IN AN ALGORITHM SELECTION APPROACH FOR SEMANTIC SEGMENTATION BASED ON HIGH-LEVEL INFORMATION FEEDBACK

In this paper we discuss certain theoretical properties of the algorithm selection approach to the problem of semantic segmentation in computer vision. High quality algorithm selection is possible only if each algorithm's suitability is well known because only then the algorithm selection result can improve the best possible result given by a single algorithm. We show that an algorithm's evaluation score depends on final task; i.e. to properly evaluate an algorithm and to determine its suitability, only well formulated tasks must be used. When algorithm suitability is well known, the algorithm can be efficiently used for a task by applying it in the most favorable environmental conditions determined during the evaluation. The task dependent evaluation is demonstrated on segmentation and object recognition. Additionally, we also discuss the importance of high level symbolic knowledge in the selection process. The importance of this symbolic hypothesis is demonstrated on a set of learning experiments with a Bayesian Network, a SVM and with statistics obtained during algorithm selector training. We show that task dependent evaluation is required to allow efficient algorithm selection. We show that using symbolic preferences of algorithms, the accuracy of algorithm selection can be improved by 10 to 15% and the symbolic segmentation quality can be improved by up to 5% when compared with the best available algorithm.

Keywords: algorithm selection, algorithm suitability, computer vision.

NOMENCLATURE

ALE is Automated Labeling Environment;

BN is Bayesian Network;

CPMC is Constrained Parametric Min-Cuts for Automatic Object Segmentation;

FFT is Fast Fourier Transform;

HOG is Histogram of Oriented Gradients;

IA is Iterative Analysis;

MSER is Maximally Stable Extremal Regions;

SIFT is Scale Invariant Feature Transform;

SDS is Simultaneous Detection and Segmentation;

SVM is Support Vector Machine.

INTRODUCTION

The algorithm selection problem has been introduced by Rice [1] and since has been used in various applications. Recently it has been applied to machine vision and image processing [2, 3]. While in general the algorithm selection process works well [4–7], for more complex problem spaces, problems that are related to feature selection, evaluation and algorithm suitability have been recorded and reported [3, 8].

Algorithm selection is in general seen as a secondary solution to a problem because to select best algorithm from a set of available algorithms several preconditions must be satisfied: knowledge about the problem, knowledge about the algorithms, algorithm suitability and distinctive features for each algorithm must be known. Consequently, algorithm selection is neither easy to apply nor the least expensive solution. However, for complex problems that have large feature spaces including problems that deal with real-world situations and environment, algorithm selection is a viable alternative. The concept behind algorithm selection is in the algorithm separation: an algorithm that would deal with the problem successfully for all combinations of environmental conditions will be too complex but a set of more specific algorithms for subsets of conditions will

provide better and cheaper solutions when applied on a case by case basis.

To obtain result improvement from a case by case selected set of algorithms, high quality selection mechanism with a minimal precision of selection is required: for a set of inputs, the selected algorithms must be such that the cumulative result is better than the best of the available algorithms. This implies that the algorithm selection mechanism must be able to select the best algorithm as often as possible.

A reliable algorithm selection implies that the set of available algorithms have been evaluated in a very strict setting and in a task dependent manner. As will be shown, task specific evaluation provides data that can be used for algorithm selection because only such evaluation results can be used to predict reliably algorithm results on new untested input data. This means that evaluation of a single algorithm cannot be seen as a holistic process but rather as a precise and specific process that is not generalizable.

Finally, the algorithm selection presented in this paper is situated within the framework for high level image understanding. We show that unlike standard feature-only based algorithm selection approaches, the high level symbolic description greatly improves the accuracy of algorithm selection as well as the final result of high level understanding.

1 PROBLEM STATEMENT

In this paper we analyze several problems of the algorithm selection:

- the impact of the high level symbolic understanding of image content on the accuracy of algorithm selection;
- the impact of algorithm evaluation on the algorithm selection process;
- the impact of feature for object recognition evaluation on the algorithm selection process.

2 REVIEW OF LITERATURE

Algorithm selection was introduced by Rice [1] in the context of selection of scheduling algorithm in computer operating system. Since then it has been applied to various problems and fields of research in different ways and granularity. In image processing and computer vision the algorithm selection has been used to determine the best algorithm for segmentation of artificially generated images of noisy geometrical shapes [4]. In [7] algorithm selection was used for determining the best algorithm for the segmentation of biological cell images and [5] used algorithm selection in a performance predicting framework.

For segmentation of more complex natural images [2] proposed an algorithm selection approach using machine learning and composition: final segmentation was created from partial segmentation from best algorithms for different regions of the image. The method showed that despite results with high accuracy of selection the final result was only as good as the best available algorithm. Finally, a more specific approach was used to select parameters in single algorithm for segmentation in [9].

In [10] uses depth information to estimate whole image properties such as occlusions, background and foreground isolation and point of view estimation to determine type of objects in the image. All the modules of this approach are processed in parallel and integrated in a final single step. An airport apron analysis is performed in [11] where the authors use motion tracking and understanding inspired by cognitive vision techniques. Finally, the image understanding can also be approached from a more holistic approach such as for instance in [12] where the intent is only to estimate the nature of the image and distinguish between mostly natural or artificial content.

Currently there is a large amount of work combining segmentation and recognition and some of them are [13, 14]. In [15] uses an interleaved object recognition and segmentation in such manner that the recognition is used to seed the segmentation and obtain more precise detected objects contours. In [16] objects are detected by combining part detection and segmentation in order to obtain better shapes of objects. More general approaches such as [17] build a list of available objects and categories by learning them from data samples and reducing them to relevant

information using some dictionary tool. However this approach does not scale to arbitrary size because the labels are not structured and ultimately require complete knowledge of the whole world.

3 MATERIALS AND METHODS

In [8] an alternative approach to image understanding was proposed: an algorithm selection platform with verification of the high-level symbolic interpretation of the image content was proposed. This platform is used as basis of research in this paper and is shown in Fig. 1.

The platform works in two distinct modes and integrates both the algorithm selection from features and algorithm selection from high-level feedback. Initially, the input image is processed (box 1) by algorithm selected using the algorithm selection mechanism (box 3) that uses only the image features (Loop 1). The resulting high-level description of the image (obtained from the object recognition), is verified for logical contradictions (box 4) both on the context, on the part-level, on the location and on the relative size level. If the verification does not detect any high-level symbolic contradiction the processing stops and outputs the current high level description. If however a logical contradiction is detected, a hypothesis that solves the contradiction is generated (box 5). The image region that corresponds to the contradiction and to the hypothesis is used to extract local features, to determine local context information and to estimate attributes of the possible objects located in the selected image region. These three sources of information are used in the meta level to estimate what other algorithm should be used to correct the contradiction (Loop 2). This second loop is iterated over all contradictions until all contradictions are resolved. For the rest of this paper the presented system will be referred to as Iterative Analysis (IA).

Notice that the proposed system uses a twofold processing convergence. First convergence of the approach is to obtain a non-contradictory high-level description (contradiction resolution). The second convergence is the match between a description without contradiction and a set of algorithms (algorithm matching). The proposed approach thus combines processing quality with the meta-processing algorithm matching. This approach thus enables to exploit each algorithm's strongest points on an application, image features and image content basis.

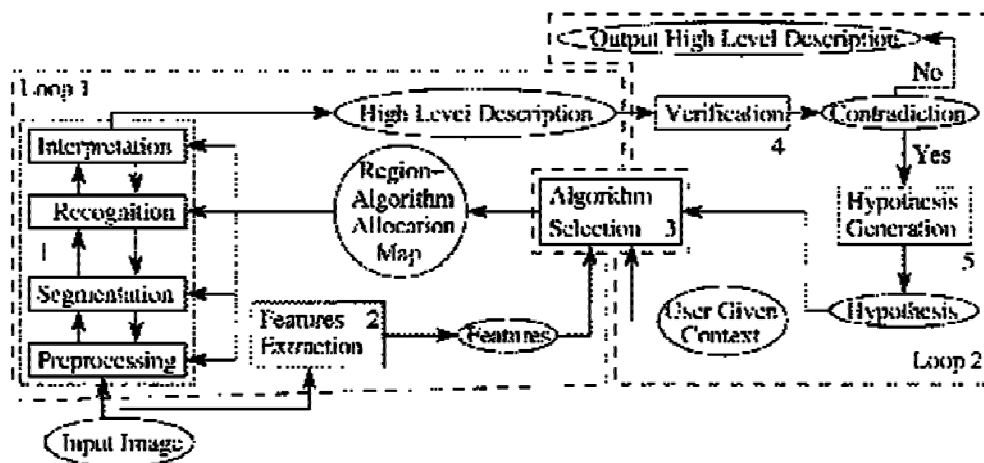


Figure 1 – Algorithm selection platform with verification of the high-level symbolic interpretation of the image content

The concept behind the processing in box 1, Figure 1 is that each algorithm used is a network of various component algorithms. Box 1 shows the general classical robotic sequential processing that uses four components processing levels: the preprocessing, segmentation, recognition and interpretation. However as in this paper the algorithms used are performing the semantic segmentation the interpretation is obtained by a single common algorithm. Also the selection is not limited to these four processing blocks but rather is intended to accommodate various algorithm networks.

As a final note some specific information about the selection process is required. In the initial loop of the IA processing, features extracted from the input images are FFT coefficients, Gabor features, wavelets, gist, color average, intensity average, edges, covariant features, SIFT, HOG, MSER and textures. All these features are transformed into a histogram and are concatenated into a single vector per image (or per region) of 5000 values.

For all loops after the initial one, the hypothesis is represented as a set of attributes. These attributes are obtained using the regprops function in Matlab. These attributes have been discretized in order to simplify the representation but to allow discrete representation of each of the available hypotheses.

In this paper the platform uses algorithms performing semantic segmentation: first segment an image and recognize regions as objects. The result of such processing is fed to the interpretation and verification according to the above platform description.

4 EXPERIMENTS

In order to assess an algorithm processing quality, it is necessary to evaluate its performance with respect to some training data set and ground truth. Each evaluation

experiment was designed using real algorithm selection data. The algorithms used in our classification task are the ALE [11, 18], CPMC with recognition [14] and the SDS [19]. The three algorithms have similar performance results shown in Table 1. Here the numbers given in the original papers may vary due to different set up, initialization and training conditions of the original and this experiments.

Consequently most of the algorithms that perform the semantic segmentation task first segments an image using some well-known segmentation algorithm and then apply the object recognition (there are other algorithms that are not using this order such as [15]).

Let us assume that an algorithm is evaluated for the quality of segmentation i.e. it evaluates whole image segmentation by comparing the result of processing to a human provided ground truth. Figure 2a shows an example of input image, Fig. 2b – human generated ground truth and Fig. 2c–Fig. 2d – the result of a segmentation algorithm. Fig. 2b – Fig. 2c have also their f-value shown in the parentheses. F-value is one of the standard measures used to determine the accuracy of computer generated segmentation [20]. According to the f-measure in this case of evaluation the algorithm generating the result shown in Fig. 2c is superior (closer when pixel-to-pixel comparison is done with human segmentation in Fig. 2b) to the algorithm which result is shown in Fig. 2d.

Now let’s look at the same algorithms in the task of semantic segmentation. In semantic segmentation and input

Table 1 – Results of semantic segmentation algorithms on the VOC2012 validation dataset

Name	Result
ALE	47.8%
CPMC	48.3%
SDS	49.9%

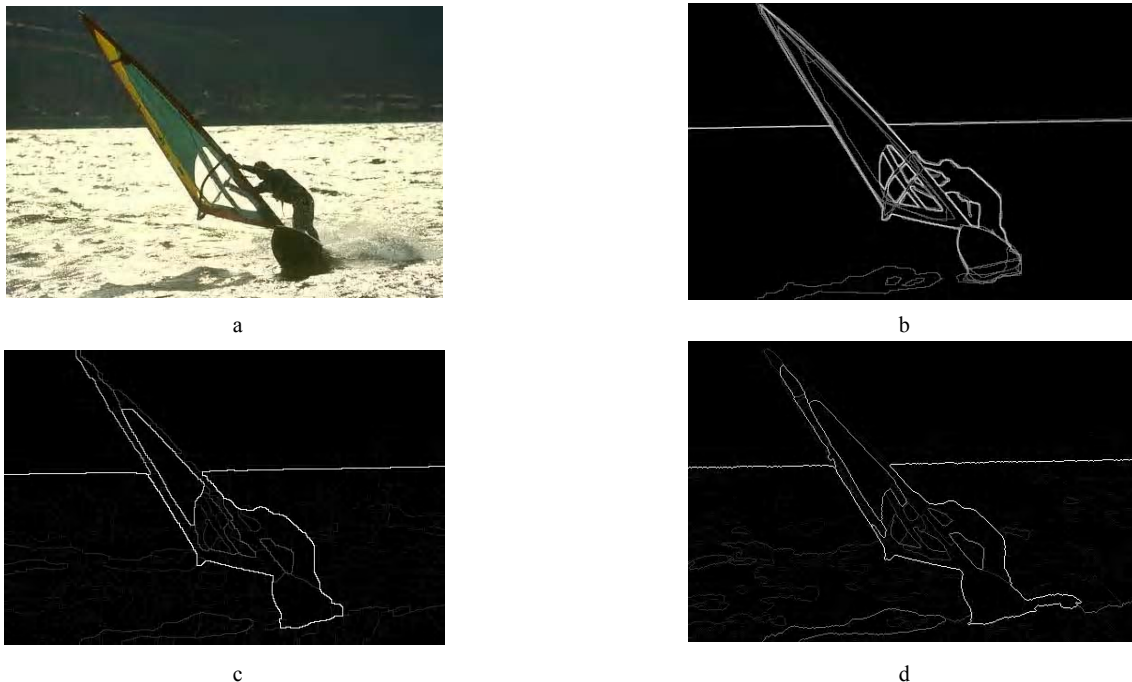


Figure 2 – Example of algorithmic Segmentation: a – input image, b – human ground truth (0.9), c – result of algorithm from [21] (0.92), d – result of algorithm from [22] (0.87)

image is segmented and then each region is labeled from a set of available object label set. In the task of semantic segmentation, the two best algorithms for image segmentation shown in Fig. 2c, Fig. 2d will not have the same f-values. In fact, algorithms with much lower f-value $f = 0.77$ (in the task of segmentation and with result shown in Fig. 3b) will have much higher resulting score because the regions obtained from the detected regions are more precise for object detection and labeling.

The reason for such change of the score is possible because in image segmentation the algorithm's result is evaluated by comparing the obtained boundaries with the ground truth generated by human. However, in the case of semantic segmentation the evaluation is made first by determining the boundaries of the target object and then the detection of the correct object is tested. This means that segmenting the whole image and comparing it to a set of human generated ground truth will result in more variation because even humans will not generally agree on how to segment a whole scene. This is because the evaluation is done with respect to a human segmentation that depends on feeling and intuition. However, when segmenting an image to determine object boundary the disparity between humans is much smaller. The semantic segmentation can be automatically judged on whether or not the correct object was correctly detected. Consequently despite the fact that some segmentation might be close enough to a human like segmentation it might not be well suited for the segmentation of a particular object.

Thus for two different tasks, the score of the final evaluation of a same algorithm might not be the same and the algorithm that had a good result in one tasks will have much lower result score for another task. But a statistical evaluation of algorithms might not be sufficient to determine advantages and disadvantages precisely enough. Figure 4 shows the standard model of robotics where multiple processes are formed into a set of consecutive algorithmic steps. The combination of algorithms can result in non-linear result that would not be observed otherwise. Thus it is necessary to evaluate the component algorithms as well

so that individual suitabilities can be determined and impact on the result of the entire computation.

Similarly to the segmentation study a change of result can be obtained in recognition. Various features have different accuracy and ability to detect and recognize an object.

Using various features for detection (using the bag of words recognition model) it can be shown that depending on the region used to extract the feature descriptors and on the features extracted, the recognition accuracy will change. For instance assume that a segmentation algorithm such as [21, 25] is used for segmentation. The results of the segmentation are boundaries that indicate main regions of the image where the features for recognition should be extracted and the recognition model should be applied. Depending on what features are extracted the accuracy will change depending on the image. In some cases there will be no detection and in some other cases the detection will be a success.

Figure 5 shows the results of calculating the bounding box after two different features (SIFT and HOG) have been used for object recognition. In this case we extract features from the whole images. The features and the descriptors extracted are used to recognize a motor-bike and then the same feature descriptors are used to generate a bounding box. The idea behind this experiment is to assess the importance of a region in recognition of a motor-bike given that segmentation occurred prior to recognition.

The bounding box method determination is shown in Fig. 6 and Fig. 7.

Following the standard bag of words object recognition method a model of the object being detected is available as a set of histograms of feature clustered centers. Input image is first used as input for feature extraction, the feature descriptors are then clustered into k centers and a new histogram is constructed with bins corresponding to the k centers. Once the histogram is obtained, it is compared to all histograms in the model database and four closest matches are saved. Finally features corresponding to four best matching bins (each from one of the model histograms) are used to determine which descriptors and consequently which key points are used to determine the bounding box (Figure 7).

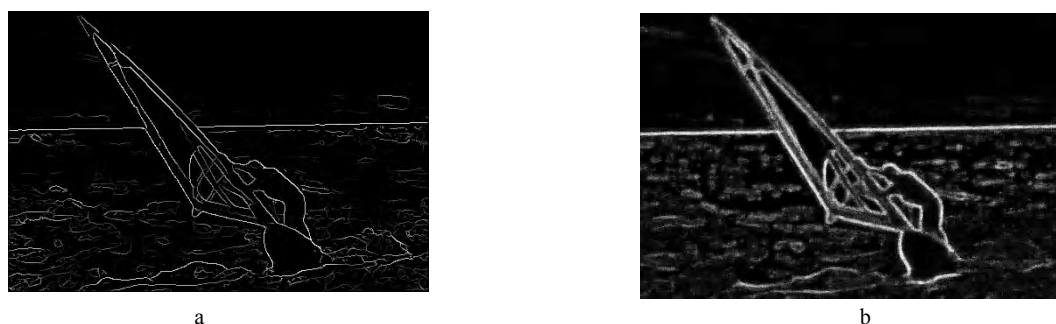


Figure 3 – Example of two algorithms for segmentation with lower f-value:

a – result of segmentation of input image from Figure 2a using the algorithm from [23], b – result of segmentation of input image from Figure 2a using the algorithm from [24]



Figure 4 – Typical example of processing required for semantic segmentation

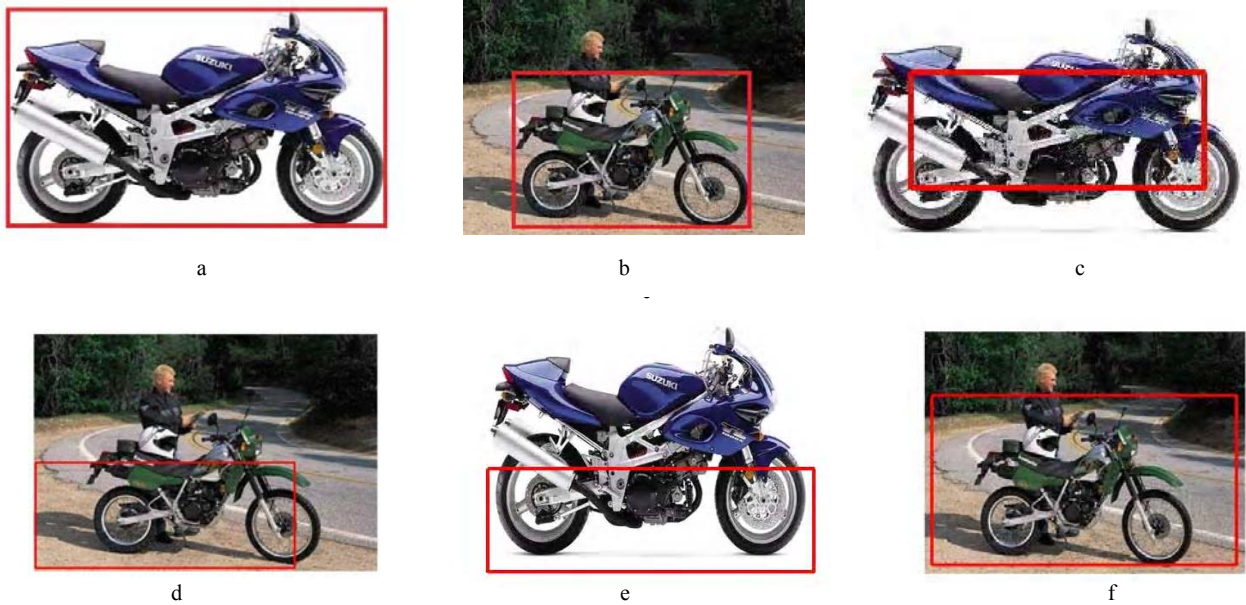


Figure 5 – Comparison of Bounding boxes obtained from SIFT and HOG features: a – Input Image 1 and Bounding Box by Human, b – Input Image 2 and Bounding Box by Human, c – Bounding box from SIFT, d – Bounding box from SIFT, e – Bounding Box from HOG, f – Bounding Box from HOG

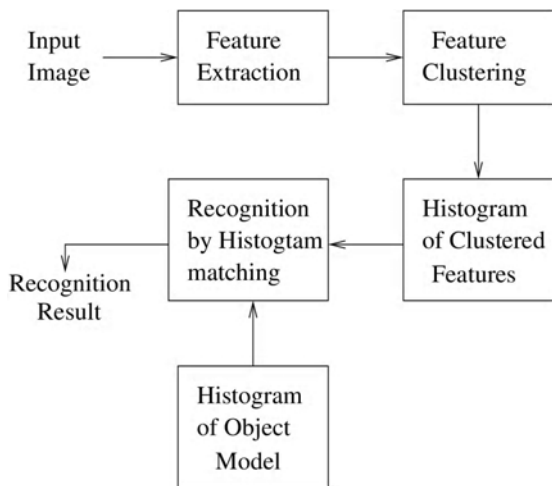


Figure 6 – Schema of Bag-of-words recognition algorithm

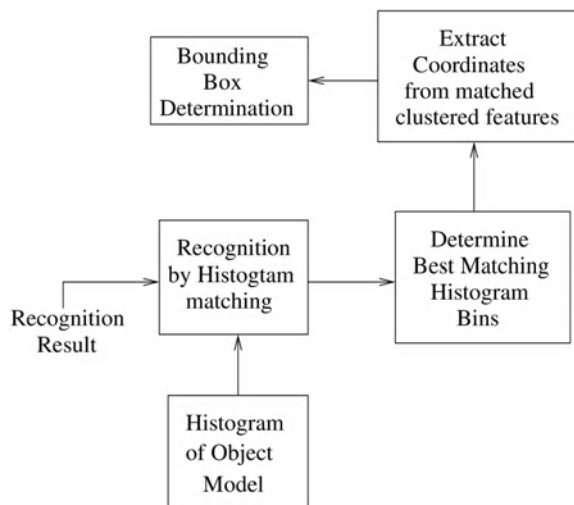


Figure 7 – Schema of Bounding Box extraction from a successful object recognition using Bag-of-words recognition algorithm

Using the method for determining the bounding box as an evaluation of the feature descriptor relevance to the motor-bicycle model, it can be seen that having different regions used to extract the feature descriptors would have a significant impact on object detection. For instance if only the upper left region (Region 1 in Fig. 8) of the bicycle would be contained in a single region no detection would happen if the HOG features would be used. Also if the SIFT features would be used it is possible that positive detection might not occur as not enough of the significant SIFT descriptors for successful detection are contained in the Region 1 only. On the other hand if the bottom of the motorcycle would be contained in a single region (Region 3 in Fig. 8) the HOG features would not be able to detect the motorcycle.

Consequently, using various features for only recognition or for semantic segmentation can have considerably different results as both the segmentation and the recognition are sensitive and difficult operations. Their evaluation is thus highly task dependent.

In the software platform previously introduced the algorithm selection is iterated through several iterations. The stopping condition for the processing of the image is either no more improvement is possible due to having tried all available algorithms or no more improvement is possible as the new hypothesis generated is the same as the previous one.

Initially, the algorithms are selected using only the image features but after the first processing loop the hypothesis generated is used for algorithm selection. The features have been successfully used for algorithm selection in various approaches, however in general such algorithm selector is limited due to the fact that many algorithms are designed for particular symbolic and semantic context.

The semantic segmentation results in a set of symbolically labeled regions and thus analyzing the obtained regions by various algorithms it is possible to conclude that various algorithms have affinities for different objects.

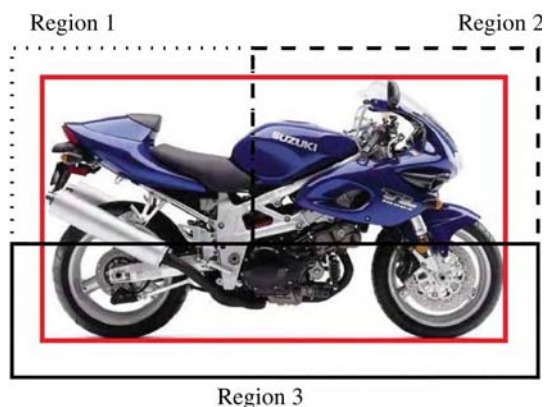


Figure 8 – Example of three different regions obtained as a result of a possible segmentation

Such affinity for particular objects can be due to the following reasons:

- The environment in which the particular object is captured has particular interaction with the object that is favorable to be detected by a particular algorithm.
- The object itself has a set of features that a particular algorithm is better suited for detection and segmentation.

Consequently we asked: what is the impact of symbolic information (content related) on the accuracy of algorithm selection?

To answer the above question we conducted a set of experiments. The experiments were carried using the VOC2012 [26] database. The dataset used is not the standard VOC2012 validation set but a reduced one in order to allow applying our platform. This means that only images where multiple objects to be segmented are present. The dataset is thus reduced and contains only ~300 images out of ~1500 images contained in the VOC2012 dataset.

5 RESULTS

The platform was initially designed to use Bayesian Network (BN) because the probabilistic inference is well suited to deal with missing variables. Consequently and because the two different modes of algorithm selection (features only and features with high level description), a single trained BN can be used. However selection of algorithms using Bayesian Network is still problematic and thus two alternative algorithm selectors were used for comparison. These two selection mechanisms are SVM and Statistics from training.

In a first experiment we compared the BN and the SVM because both of these algorithm selectors work on similar principles. Both BN and SVM are used in the initial and all further iterations of the IA platform. In the first iterations only features are used to select algorithm while in all further loops the features from the contradiction region as well as the hypothesis is used. The main difference between using the BN and SVM is that SVM requires incomplete input information imputation [28] while the BN is well suited to handle missing input information by design. This means that for the first iteration, the SVM is provided with average values of the hypothesis in order for the input vector has the desired and fixed length.

The results of comparison of the precision of the BN and of the SVM are shown in Table 2.

The problem of using the Bayesian Network is that it requires discrete input information. However most of the features extracted from input image are continuous and unbounded. Consequently it is required to cluster the input information and only then use it as input to the BN. This however has in most of the cases a dramatic influence on the performance of the probabilistic algorithm selection.

As can be seen the impact of hypothesis attributes is significant in the case of SVM, however in the case of BN it is difficult to evaluate as the overall precision is too low. The general increase of algorithm selection using the features and attributes compared to the selection using only features is up to 10% of accuracy.

The final evaluation of the high level information (feedback) in our system is the usage of statistical accuracy of each algorithm. The accuracy represents the percentile average of the f-measure of each semantic segmentation algorithm. Table 3 shows the accuracy of semantic segmentation for each of the three used algorithms: ALE [18], CPMC [14] and SDS [19]. Each columns shows average accuracy for each of the categories of objects that are to be recognized and segmented and overall average accuracy in the bottom row. The statistical information provided was obtained by evaluating the VOC2012 validation dataset that contains approximately 1500 images.

The last column in Table 3 shows for each class of objects the best algorithm based on the statistical accuracy of each algorithm. This means that in the platform and during iterations all but the first one, for each hypothesis algorithm will be selected only using the best algorithm listed in the rightmost column. Using this approach we evaluated the proposed Iterative Analysis method described in this paper. The result comparison is shown in Table 4. It shows average precision for each algorithm for the test dataset.

Table 2 – Comparison of BN and SVM in selection accuracy using only features vs. features and attributes

Task	Algorithm	Features	Features+Attributes
2-class	SVM	58%	65%
2-class	BN	46%	55%
3-class	SVM	43%	46%
3-class	BN	39%	43%

Table 3 – Average accuracy of different algorithms on the training set

Accuracy for each class (intersection/union measure)	Algorithm Names			Best Algorithm
	ALE	SDS	CPMC	
Background	71.711	84.937	83.098	SDS
Aeroplane	52.096	60.927	64.404	CPMC
Bicycle	27.558	26.823	17.965	ALE
Bird	36.684	56.239	50.783	SDS
Boat	38.656	47.003	45.036	SDS
Bottle	43.643	48.465	41.605	SDS
Bus	65.787	70.559	69.104	SDS
Car	58.338	60.723	60.733	CPMC
Cat	63.789	59.847	56.524	ALE
Chair	24.001	20.815	11.663	ALE
Cow	64.853	42.112	52.842	ALE
Dining Table	41.339	38.694	19.406	ALE
Dog	55.190	51.535	48.995	ALE
Horse	58.998	43.653	43.899	ALE
Motorbike	56.909	52.300	52.858	ALE
Person	49.107	61.649	46.707	SDS
Potted Plant	31.408	37.360	40.563	CPMC
Sheep	53.563	51.829	49.285	ALE
Sofa	38.598	22.375	49.208	ALE
Train	53.910	56.288	58.319	CPMC
Average Accuracy	48.473	50.089	47.048	

Table 4 – Results of semantic segmentation accuracy on the test data set

Accuracy for each class (intersection/union measure)	Algorithm's Name				Best Algorithm
	ALE	SDS	CPMC	IA	
Background	54.878	80.061	77.478	62.157	SDS
Aeroplane	0.000	0.000	0.000	0.000	--
Bicycle	26.799	31.913	13.515	27.624	SDS
Bird	22.070	37.042	59.947	21.932	CPMC
Boat	0.000	0.000	0.000	0.000	--
Bottle	37.445	50.990	39.280	50.226	SDS
Bus	44.212	12.034	71.156	45.412	CPMC
Car	52.788	34.924	31.873	56.241	IA
Cat	63.939	65.552	62.707	63.802	SDS
Chair	19.113	22.355	7.800	19.014	SDS
Cow	33.093	0.000	0.000	30.991	ALE
Dining Table	39.155	50.907	23.997	40.169	SDS
Dog	60.085	49.253	49.827	59.148	ALE
Horse	46.406	27.761	27.155	47.128	IA
Motorbike	61.154	28.477	33.949	61.697	IA
Person	46.362	63.940	46.068	57.947	SDS
Potted Plant	25.762	36.391	25.045	23.245	SDS
Sheep	69.008	66.129	27.191	69.008	IA
Sofa	29.672	17.062	11.806	29.702	IA
Train	43.602	0.000	28.651	51.174	IA
TVmonitor	31.320	62.904	53.201	37.091	SDS
Average Accuracy	38.422	35.128	32.935	40.653	IA

6 DISCUSSION

Notice that not all algorithms tested have an average score in all categories: this is due to the fact for the images that contained the object cow was not detected not even once by neither SDS nor CPMC. Moreover observe that our approach IA is best only in few categories but in most of the categories is relatively close to the best one. As a result of using the statistical information for algorithm selection the IA approach results in the best semantic segmentation.

As a final comment on the importance of high level image description and content understanding, Figure 9 shows the results of three different semantic segmentation algorithms (Fig. 9c–Fig. 9e) and the result obtained by IA platform (Fig. 9f) that uses features and features and hypothesis attributes for algorithm selection. In the experiment illustrated in Fig. 9 the input image is shown in Fig. 9a. The first algorithm selected generated the result shown in Fig. 9c. The obtained semantic segmentation was analyzed for shape, proximity, position and relative size contradiction [27] and a hypothesis solving the contradiction is generated.

Using this hypothesis a new algorithm (Fig. 9e) was selected and the two results of semantic segmentation are merged. The result is shown in Fig. 9f. Notice the replacement of the chair (red region) from the initial result without removing any part of the sofa (green region).

CONCLUSION

In this paper we described some theoretical properties of algorithm selection. In particular we discussed the importance of the proper evaluation and the importance of hypothesis in the algorithm selection. The results show that for algorithms that are context sensitive – and most of algorithms used in real world application are context sensitive – the iterative approach proposed in this paper improves the overall computer vision and image understanding. The high level information was demonstrated to be very important – using only the statistics on the class level segmentation accuracy the algorithm selection approach provides best results and outperforms all the used algorithms.

Several extensions to this work are planned. The statistical information obtained during testing is not precise enough and thus will be explored in combination of features from the contradiction regions for increased accuracy of algorithm selection. The features used so far in the algorithm selection also require accuracy improvement by finding richer features. Such features have been recently obtained by the use of convolutional neural networks and we plan to integrate them into the IA platform. Finally, the hypothesis used is a simple object label obtained from measured properties such as objects

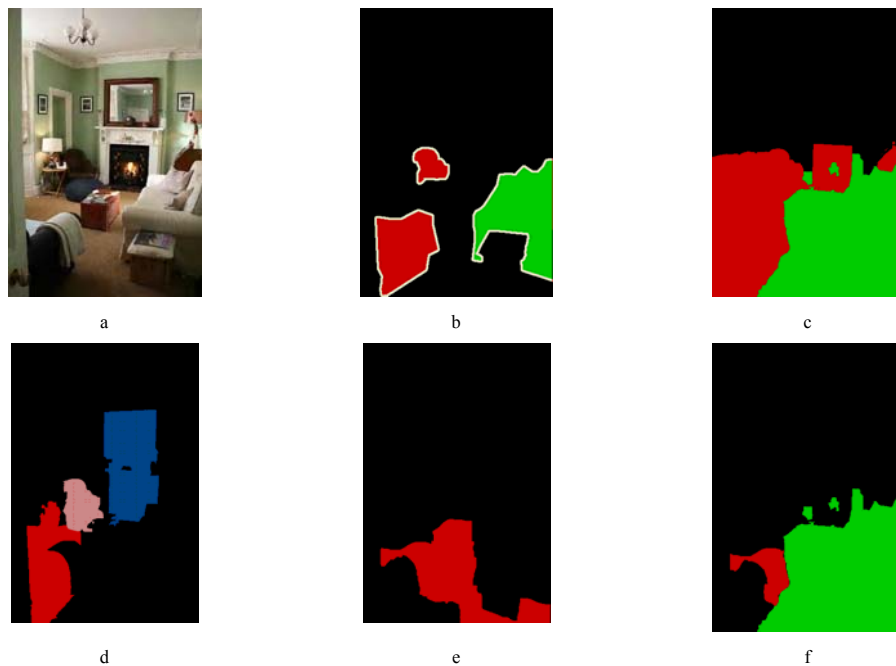


Figure 9 – An example of different stages of processing an input image using the algorithm selection platform: a – Input Image, b – Ground Truth, c – ALE Result, d – SDS Result, e – CPMC Result, f – IA Result

proximity, relative size and so on: to increase the accuracy of hypothesis generation a deeper semantic model connecting more object attributes and the objects with the environment of the world is to be build and used in close future.

REFERENCES

1. Rice J. The algorithm selection problem / J. Rice // *Advances in Computers*. – 1976. – Vol. 15. – P. 65–118.
2. Lukac M. Machine learning based adaptive contour detection using algorithm selection and image splitting / M. Lukac, R. Tanizawa, M. Kameyama // *Interdisciplinary Information Sciences*. – 2012. – Vol. 18, № 2. – P. 123–134.
3. Lukac M. Natural image understanding using algorithm selection and high level feedback / M. Lukac, M. Kameyama, K. Hiura // *SPIE Intelligent Robots and Computer Vision XXX: algorithms and Techniques*. – 2013. DOI: 10.1117/12.2008593
4. Zhang Y. Optimal selection of segmentation algorithms based on performance evaluation / Y. Zhang and H. Luo // *Optical Engineering*. – 2000. – Vol. 39, № 6. – P. 1450–1456.
5. Yong X. Optimal selection of image segmentation algorithms based on performance prediction / X. Yong, D. Feng, Z. Rongchun // *Proceedings of the Pan-Sydney Area Workshop on Visual Information Processing (VIP2003)*. – 2003. – P. 105–108.
6. Yu L. Feature selection for high-dimensional data: A fast correlation-based filter solution / L. Yu, H. Liu // *Proceedings of the 20th International Conference on Machine Learning*. – 2004. – P. 856–863.
7. Takemoto S. Algorithm selection for intracellular image segmentation based on region similarity / S. Takemoto, H. Yokota // *Ninth International Conference on Intelligent Systems Design and Applications*. – 2009. – P. 1413–1418. DOI: 10.1109/ISDA.2009.205
8. Lukac M. Bayesian-network-based algorithm selection with high level representation feedback for real-world intelligent systems / M. Lukac, and M. Kameyama // *Information Technology in Industry*. – 2015. – Vol. 3, № 1. – P. 10–15.
9. Peng B. Parameter selection for graph cut based image segmentation / B. Peng, V. Veksler // *In Proceedings of the British Conference on Computer Vision*. – 2008. – P. 16.1–16.10. DOI: 10.5244/C.22.16
10. Hoiem D. Closing the loop on scene interpretation / D. Hoiem, A. A. Efros, M. Hebert // *Proc. Computer Vision and Pattern Recognition (CVPR)*. – 2008. – P. 1–8. DOI: 10.1109/CVPR.2008.4587587
11. Ferryman J. Automated scene understanding for airport aprons / [J. Ferryman, M. Borg, D. Thirde and other] // *Proceedings of 18th Australian Joint Conference on Artificial Intelligence*. – 2005. – P. 593–603. DOI: 10.1007/11589990_62
12. Oliva A. Modeling the shape of the scene: a holistic representation of the spatial envelope / A. Oliva, A. Torralba // *International Journal of Computer Vision*. – 2001. – Vol. 42, № 3. – P. 145–175.
13. Ladicky L. Graph cut based inference with co-occurrence statistics / L. Ladicky, C. Russell, P. Kohli, and P. Torr // *In Proceedings of the 11th European conference on Computer vision*. – 2010. – P. 239–253. DOI: 10.1007/978-3-642-15555-0_18
14. Carreira J. Object recognition by sequential figure-ground ranking / J. Carreira, F. Li, C. Sminchisescu // *International Journal of Computer Vision*. – 2012. – Vol. 98, № 3. – P. 243–262.
15. Leibe B. Robust object detection with interleaved categorization and segmentation / B. Leibe, A. Leonardis, B. Schiele // *International Journal of Computer Vision*. – 2008. – Vol. 77. – P. 259–289.
16. Finding animals: Semantic segmentation using regions and parts / [Arbelaez P., Hariharan B., Gu C. and other] // *International Conference on Computer Vision and Pattern Recognition*. – 2012. – P. 3378–3385. DOI: 10.1109/CVPR.2012.6248077
17. Li L.-J. Towards total scene understanding: classification, annotation and segmentation in an automatic framework / L.-J. Li, R. Socher, L. Fei-Fei // *Computer Vision and Pattern Recognition (CVPR)*. – 2009. – P. 2036–2043. DOI:10.1109/CVPR.2009.5206718
18. Ladicky L. Inference methods for crfs with co-occurrence statistics / L. Ladicky, C. Russell, P. Kohli, P. Torr // *International Journal of Computer Vision*. – 2013. – Vol. 103, № 2. – P. 213–225.
19. Hariharan B. Simultaneous detection and segmentation / B. Hariharan, P. Arbelaez, R. Girshick, J. Malik // *European Conference on Computer Vision (ECCV)*. – 2014. – P. 297–312. DOI: 10.1007/978-3-319-10584-0_20
20. Martin M. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics / M. Martin, C. Fowlkes, D. Tal, J. Malik // *Proceedings of 8th International Conference on Computer Vision*. – 2001. – P. 416–423. DOI: 10.1109/ICCV.2001.937655
21. Arbelaez P. Contour detection and hierarchical image segmentation / P. Arbelaez, M. Maire, C. Fowlkes, J. Malik // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. – 2011. – Vol. 33, № 5. – P. 898–916.
22. Arbelaez P. Boundary extraction in natural images using ultrametric contour maps / P. Arbelaez // *Computer Vision and Pattern Recognition Workshop*. – 2006. – P. 182–190. DOI: 10.1109/CVPRW.2006.5.
23. Ren X. Multi-scale improves boundary detection in natural images / X. Ren // *Proceedings of the 10th European Conference on Computer Vision*. – 2008. – P. 533–545. DOI: 10.1007/978-3-540-88690-7_40
24. Dollar P. Supervised learning of edges and object boundaries / P. Dollar, Z. Tu, S. Belongie // *IEEE Computer Vision and Pattern Recognition (CVPR)*. – 2006. – P. 1964–1971. DOI: 10.1109/CVPR.2006.298
25. Using contours to detect and localize junctions in natural images / [M. Maire, P. Arbelaez, C. Fowlkes, J. Malik] // *Conference on Vision and Pattern Recognition*. – 2008. – P. 1–8. DOI: 10.1109/CVPR.2008.4587339
26. The pascal visual object classes (VOC) challenge / [M. Everingham, L. Van Gool, C. K. I. Williams and other] // *International Journal of Computer Vision*. – 2010. – Vol. 88, № 2. – P. 303–338.
27. Lukac M. Bayesian-network-based algorithm selection with high level representation feedback for real-world information processing / M. Lukac, M. Kameyama // *IT in Industry*. – 2015. – Vol. 3, № 1. – P. 10–15.
28. Handling missing values in support vector machine classifier / [K. Pelckmans, J. De Brabanter, J.A.K. Suykens and other] // *Neural Networks*. – 2005. – Vol. 18, № 5–6. – P. 684–692.

Article was submitted 22.09.2015.

After revision 06.10.2015.

Лукач М.¹, Абдиева К.², Каменя М.³

¹Д-р философии, ассистент кафедры компьютерных наук, Университет им. Назарбаева, Астана, Казахстан

²Аспирант, лаборатория ROSE, Наньянский технологический Университет, Сингапур

³Д-р наук, профессор, профессор школы информатики, Университете Тохоку, Сендай, Япония

ОЦЕНИВАНИЕ КОМПОНЕНТНЫХ АЛГОРИТМОВ ДЛЯ ВЫБОРА АЛГОРИТМА СЕМАНТИЧЕСКОЙ СЕГМЕНТАЦИИ НА ОСНОВЕ ОБРАТНОЙ СВЯЗИ С ВЫСОКИМ УРОВНЕМ ИНФОРМАЦИИ

Обсуждаются некоторые теоретические свойства подхода по выбору алгоритма для решения проблемы семантической сегментации в компьютерном зрении. Высококачественный выбор алгоритма возможен, только если пригодность каждого алгоритма хорошо известна, потому что только тогда результат выбора алгоритма может улучшить наилучший возможный результат, полученный одним алгоритмом. Показано, что оценка алгоритма зависит от конечной задачи; т.е. для того чтобы правильно оценивать алгоритм и определить его пригодность, необходимо использовать только хорошо сформулированные задачи. Когда пригодность алгоритма известна алгоритм может быть эффективно использован для задачи, применяясь в наиболее благоприятных условиях, определяемых в ходе оценивания. Оценивание, зависящее от задачи, продемонстрировано на сегментации и распознавании объектов. Кроме того, обсуждается важность символического знания высокого уровня в процессе отбора. Важность этой символической гипотезы продемонстрировано на наборе экспериментов по обучению байесовской сети и SVM, а также с помощью статистических данных, полученные во время обучения селектора алгоритма. Показано, что для выбора эффективного алгоритма требуется оценивание, зависящее от

задачі. Показано, що используя символические предпочтения алгоритмов, точность выбора алгоритма может быть улучшена на 10–15%, а качество символической сегментации может быть улучшено до 5% по сравнению с наилучшим доступным алгоритмом.

Ключевые слова: выбор алгоритма, пригодность алгоритма, компьютерное зрение.

Лукач М.¹, Абдієв К.², Камеяма М.³

¹Д-р філософії, асистент кафедри комп'ютерних наук, Університет ім. Назарбаєва, Астана, Казахстан

²Аспірант, лабораторія ROSE, Наньянський технологічний Університет, Сінгапур

³Д-р наук, професор, професор школи інформатики Універсам Тохоку, Сендай, Японія

ОЦІНЮВАННЯ КОМПОНЕНТНИЙ АЛГОРИТМІВ ДЛЯ ВИБОРУ АЛГОРИТМА СЕМАНТИЧНОЇ СЕГМЕНТАЦІЇ НА ОСНОВІ ЗВОРОТНОГО ЗВ'ЯЗКУ З ВИСОКИМ РІВНЕМ ІНФОРМАЦІЇ

Показано, що оцінка алгоритму залежить від кінцевого завдання; тобто для того щоб правильно оцінювати алгоритм і визначити його придатність, необхідно використовувати тільки добре сформульовані завдання. Коли придатність алгоритму відома, алгоритм може бути ефективно використаний для завдання, застосовуючись у найбільш сприятливих умовах, обумовлених у ході оцінювання. Оцінювання, залежне від завдання, продемонстровано на сегментації і розпізнаванні об'єктів. Крім того, обговорюється важливість символічного знання високого рівня у процесі відбору. Важливість цієї символічної гіпотези продемонстровано на наборі експериментів з навчання байєсівської мережі та SVM, а також за допомогою статистичних даних, отриманих під час навчання селектора алгоритму. Показано, що для вибору ефективного алгоритму потрібно оцінювання, залежне від завдання. Показано, що використовуючи символічні переваги алгоритмів, точність вибору алгоритму може бути поліпшена на 10–15%, а якість символічної сегментації може бути покращена до 5% у порівнянні з найкращим доступним алгоритмом.

Ключові слова: вибір алгоритму, придатність алгоритму, комп'ютерний зір.

REFERENCES

- Rice J. The algorithm selection problem, *Advances in Computers*, 1976, Vol. 15, pp. 65–118.
- Lukac M., Tanizawa R., Kameyama M. Machine learning based adaptive contour detection using algorithm selection and image splitting, *Interdisciplinary Information Sciences*, 2012, Vol. 18, No. 2, pp. 123–134.
- Lukac M., Kameyama M., Hiura K. Natural image understanding using algorithm selection and high level feedback, *SPIE Intelligent Robots and Computer Vision XXX: algorithms and Techniques*, 2013. DOI: 10.1117/12.2008593
- Zhang Y., Luo H. Optimal selection of segmentation algorithms based on performance evaluation, *Optical Engineering*, 2000, Vol. 39, No. 6, pp. 1450–1456.
- Yong X., Feng D., Rongchun Z. Optimal selection of image segmentation algorithms based on performance prediction, *Proceedings of the Pan-Sydney Area Workshop on Visual Information Processing (VIP2003)*, 2003, pp. 105–108.
- Yu L., Liu H. Feature selection for high-dimensional data: A fast correlation-based filter solution, *Proceedings of the 20th International Conference on Machine Learning*, 2004, pp. 856–863.
- Takemoto S., Yokota H. Algorithm selection for intracellular image segmentation based on region similarity, *Ninth International Conference on Intelligent Systems Design and Applications*. 2009, pp. 1413–1418. DOI: 10.1109/ISDA.2009.205
- Lukac M., Kameyama M. Bayesian-network-based algorithm selection with high level representation feedback for real-world intelligent systems, *Information Technology in Industry*, 2015, Vol. 3, No. 1, pp. 10–15.
- Peng B., Veksler V. Parameter selection for graph cut based image segmentation, *In Proceedings of the British Conference on Computer Vision*, 2008, pp. 16.1–16.10. DOI: 10.5244/C.22.16
- Hoiem D., Efros A. A., Hebert M. Closing the loop on scene interpretation / D. Hoiem, // Proc. Computer Vision and Pattern Recognition (CVPR), 2008, P. 1–8. DOI: 10.1109/CVPR.2008.4587587
- Ferryman J. Borg M., Thirde D., Fusier F., Valentin V., Bremond F., Thonnat M., Aguilera J., Kampel M. Automated scene understanding for airport aprons, *Proceedings of 18th Australian Joint Conference on Artificial Intelligence*, 2005, pp. 593–603. DOI: 10.1007/11589990_62
- Oliva A., Torralba A. Modeling the shape of the scene: a holistic representation of the spatial envelope, *International Journal of Computer Vision*, 2001, Vol. 42, No. 3, pp. 145–175.
- Ladicky L., Russell C., Kohli P., and Torr P. Graph cut based inference with co-occurrence statistics, *In Proceedings of the 11th European conference on Computer vision*, 2010, pp. 239–253. DOI: 10.1007/978-3-642-15555-0_18
- Carreira J. Li F., Sminchisescu C. Object recognition by sequential figure-ground ranking, *International Journal of Computer Vision*, 2012, Vol. 98, No. 3, pp. 243–262.
- Leibe B. Leonardis A., Schiele B. Robust object detection with interleaved categorization and segmentation, *International Journal of Computer Vision*, 2008, Vol. 77, pp. 259–289.
- Arbelaez P. Hariharan B., Gu C., Gupta S., Bourdev L., Malik J. Finding animals: Semantic segmentation using regions and parts, *International Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3378–3385. DOI: 10.1109/CVPR.2012.6248077
- Li L.-J., Socher R., Fei-Fei L. Towards total scene understanding: classification, annotation and segmentation in an automatic framework, *Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 2036–2043. DOI:10.1109/CVPR.2009.5206718
- Ladicky L., Russell C., Kohli P., Torr P. Inference methods for crfs with co-occurrence statistics, *International Journal of Computer Vision*, 2013, Vol. 103, No. 2, pp. 213–225.
- Hariharan B., Arbelaez P., Girshick R., Malik J. Simultaneous detection and segmentation, *European Conference on Computer Vision (ECCV)*, 2014, pp. 297–312. DOI: 10.1007/978-3-319-10584-0_20
- Martin M., Fowlkes C., Tal D., Malik J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, *Proceedings of 8th International Conference on Computer Vision*, 2001, P. 416–423. DOI: 10.1109/ICCV.2001.937655
- Arbelaez P., Maire M., Fowlkes C., Malik J. Contour detection and hierarchical image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011, Vol. 33, No. 5, pp. 898–916.
- Arbelaez P. Boundary extraction in natural images using ultrametric contour maps, *Computer Vision and Pattern Recognition Workshop*, 2006, pp. 182–190. DOI: 10.1109/CVPRW.2006.5.
- Ren X. Multi-scale improves boundary detection in natural images, *Proceedings of the 10th European Conference on Computer Vision*, 2008, pp. 533–545. DOI: 10.1007/978-3-540-88690-7_40
- Dollar P., Z. Tu, S. Belongie Supervised learning of edges and object boundaries, *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2006, pp. 1964–1971. DOI: 10.1109/CVPR.2006.298
- Maire M. Arbelaez P., Fowlkes C., Malik J. Using contours to detect and localize junctions in natural images, *Conference on Vision and Pattern Recognition*, 2008, P. 1–8. DOI: 10.1109/CVPR.2008.4587339
- Everingham M., Van Gool L., Williams C. K. I., Winn J., Zisserman A. The pascal visual object classes (VOC) challenge, *International Journal of Computer Vision*, 2010, Vol. 88, No. 2, pp. 303–338.
- Lukac M., Kameyama M. Bayesian-network-based algorithm selection with high level representation feedback for real-world information processing, *IT in Industry*, 2015, Vol. 3, No. 1, pp. 10–15.
- Pelckmans K., De Brabanter J., Suykens J.A.K., De Moor B. Handling missing values in support vector machine classifier, *Neural Networks*, 2005, Vol. 18, No. 5–6, pp. 684–692.

УПРАВЛІННЯ У ТЕХНІЧНИХ СИСТЕМАХ

УПРАВЛЕНИЕ В ТЕХНИЧЕСКИХ СИСТЕМАХ

CONTROL IN TECHNICAL SYSTEMS

УДК 629.7

Хацько Н. Е.

Канд. техн. наук, старший преподаватель кафедры систем и процессов управления Национального технического университета «Харьковский политехнический институт», Харьков, Украина

РЕШЕНИЕ ТЕРМИНАЛЬНОЙ ЗАДАЧИ УПРАВЛЕНИЯ С ИСПОЛЬЗОВАНИЕМ ИНФОРМАЦИИ ИНЕРЦИАЛЬНОГО БЛОКА

Рассматривается задача терминального управления летательным аппаратом по информации бесплатформенной инерциальной навигационной системы, в которой за заданное время требуется перевести аппарат в заданное пространственное положение с требуемым значением вектора конечной скорости. Построено программное управление для невозмущенного движения летательного аппарата. Синтезировано управление летательного аппарата по текущему значению его вектора состояния, определенного инерциальным навигационным блоком. Получены выражения для оценки точности попадания в конечную точку с учетом ошибки акселерометра. Проанализирована точность выведения летательного аппарата в конечную точку замкнутой системой управления при использовании информации от инерциальных датчиков с известным уровнем ошибок.

На примере упрощенной модели проведены компьютерное моделирование движения летательного аппарата и анализ точности приведения вектора состояния в заданную конечную точку в зависимости от уровня внешних возмущений и погрешности входной информации. Проведены эксперименты, позволяющие выявить влияние численного значения глубины прогноза, использованного в синтезе управления по методу преследования ведущей точки, на точность решения терминальной задачи. Сформулированы рекомендации для проектирования алгоритмов автоматического управления движением в комплексе с проектированием информационно-измерительной системы.

Ключевые слова: система управления, бесплатформенная инерциальная навигационная система, ошибка измерения вектора состояния, динамическое возмущение, точность терминального управления.

НОМЕНКЛАТУРА

БИНС – бесплатформенная инерциальная навигационная система;

ИКАО – (от англ. International Civil Aviation Organization) – международная организация гражданской авиации;

ИИСНС – интегрированная инерциально-спутниковая навигационная система;

ЛА – летательный аппарат;

МЭМС – микро-электро-механическая система;

САУП – система автоматического управления;

τ – глубина прогноза, интервал времени;

$a(t)$ – ускорение ЛА;

δa – постоянная погрешность ускорения;

b_i – коэффициенты полинома;

$f(t)$ – возмущающее воздействие;

f^* – максимальная величина воздействия;

$r(t)$ – координата объекта;

$r^*(t)$ – программная (эталонная) траектория;

$\delta r(t)$ – координатная ошибка;

r_0 – координата объекта в начальный момент времени;

ни;

r_T – координата объекта в конечный момент времени;

ни;

\dot{r}_0 – скорость объекта в начальный момент времени;

\dot{r}_T – скорость объекта в конечный момент времени;

T – общее время движения по траектории;

t – текущее время;

t' – переменная локального времени;

$u(t, r, \dot{r})$ – закон управления ЛА;

$\delta u(t)$ – ошибка управления.

ВВЕДЕНИЕ

Неотъемлемым элементом систем автоматического управления полетом (САУП) современных летательных аппаратов (ЛА) является бортовая навигационная система, обеспечивающая получение информации о местоположении, скорости и ориентации аппарата в про-

странстве. Учитывая зональные требования к точности самолетовождения, регламентированные нормативными документами ИКАО [1], бортовое оборудование должно обеспечивать высокоточную навигацию, особенно на критических участках движения, таких, как взлет, посадка, маневры в горизонтальной и вертикальной плоскости.

Эффективным способом достижения требуемой точности самолетовождения является использование так называемых интегрированных инерциально-спутниковых навигационных систем (ИИСНС), получивших широкое распространение благодаря успехам спутниковой радионавигации [2]. Принцип работы таких систем основан на комплексировании информации, получаемой от инерциальных датчиков (гироскопов и акселерометров), и спутниковой информации о текущих координатах и скорости ЛА. При этом регулярное поступление и использование достоверных спутниковых измерений позволяет сколь угодно долго удерживать ошибки определения навигационных параметров практически в окрестности нуля. Однако, как показано в [3], даже при полностью развернутых спутниковых орбитальных группировках в течение суток в некоторых областях Земного шара могут встречаться интервалы времени, более 15 минут, когда спутниковая информация вследствие особого взаимного расположения спутников в пространстве (геометрического фактора) становится не пригодной для высокоточной навигации. В этих условиях пилотирование осуществляется по информации исключительно инерциальной подсистемы, роль которой исполняет бортовая бесплатформенная инерциальная навигационная система (БИС).

Вышесказанное определяет актуальность и место рассматриваемой в данной статье проблемы в контексте современной технологии управления полетом ЛА.

1 ПОСТАНОВКА ЗАДАЧИ

Рассматривается задача управления ЛА по информации БИС, а именно – терминальная задача, в которой за заданное время требуется перевести аппарат в заданное пространственное положение с требуемым значением вектора конечной скорости.

Полагаем, что в начальный момент времени положение и скорость ЛА известны. Необходимо с помощью управляющих воздействий за заданное время перевести ЛА в требуемую точку пространства с требуемой конечной скоростью. При этом информация о текущем значении вектора состояния ЛА доступна с некоторой ошибкой, соответствующей принятой модели ошибок инерциального датчика.

Для удобства изложения принципиального хода решения задачи и некоторых аналитических результатов рассмотрим одномерный случай, для которого невозмущенное управляемое движение ЛА описывается уравнением

$$\ddot{r}(t) = u(t), \quad (1)$$

где $r(t)$ – координата объекта, $u(t)$ – искомое управляющее ускорение. В начальный ($t = 0$) и конечный ($t = T$) моменты времени состояние управляемого объекта характеризуется параметрами

$$r(0) = r_0, \quad \dot{r}(0) = \dot{r}_0, \quad (2)$$

$$r(T) = r_T, \quad \dot{r}(T) = \dot{r}_T. \quad (3)$$

Время движения по траектории определено величиной T .

С учетом получаемой информации о векторе состояния необходимо синтезировать закон управления $u(t, r, \dot{r})$, обеспечивающий в силу (1), (2) выполнение условий (3).

2 ОБЗОР ЛИТЕРАТУРЫ

На практике часто возникают задачи выполнения полета ЛА в заданную точку пространства с заданными параметрами движения за отведенное время. Это терминальные задачи, решению которых посвящены работы [4–7]. Понятие терминального управления формулируется как управление, цель которого заключается в переводе объекта управления в заданное конечное состояние в заданный момент времени [8]. Наилучшим решением терминальной задачи в алгоритмическом смысле является получение явных аналитических зависимостей для управляющих функций [5]. Современная техника управления базируется на теории обратной связи и анализе линейных систем [9]. Контроль текущих параметров состояния ЛА и формирование управления, корректирующего полет, повышает эффективность работы САУП. Обычно процесс управления включает следующие основные этапы:

- выработку эталонной траектории движения ЛА к заданной цели, то есть получение требуемого состояния ЛА через заданный промежуток времени в идеальных условиях полета;
- получение информации о текущем состоянии ЛА на шаге работы САУП;
- анализ полученной информации и выработку управляющих воздействий для продолжения движения ЛА.

Система управления является соединением отдельных элементов в определенную конфигурацию, имеющую причинно-следственные связи между элементами и обеспечивающую заданные характеристики [10], поэтому точность системы управления зависит от точности работы всех элементов системы. Навигационный модуль следует рассматривать как элемент системы управления.

Основным инструментом для получения информации о текущем векторе состояния ЛА чаще всего является ИИСНС, объединяющая измерители различной физической природы – спутниковую и инерциальную навигационную систему – в единый интегрированный навигационный комплекс. Для повышения точности определения навигационных и угловых параметров ориентации в ИИСНС используют различные способы комплексирования информации: раздельные, слабо связанные, жестко связанные и глубоко интегрированные [11]. Но, независимо от степени интеграции данных, при задержке получения достоверной информации от бортовой спутниковой навигационной аппаратуры потребителя, БИС работает в автономном режиме. В этом случае только полученные от нее данные могут быть использованы для формирования оптимального управляющего воздействия на ЛА.

Следует отметить, что в последнее время существует устойчивая тенденция к удешевлению инерциальных систем. Уже 20 лет стремительно развивается технология

микро-электро-механических систем (МЭМС). Если ранее точность датчиков, построенных с использованием этой технологии была низкой, и их можно было использовать только в автомобилях и бытовых приборах, то в последнее время точность МЭМС-акселерометров повысилась до навигационной и точность МЭМС-гироскопов также находится на высоком уровне [12]. Ведущие мировые компании уже наладили выпуск навигационных модулей с МЭМС-датчиками для малой и средней авиации.

Известно [13], что в ходе функционирования БИНС накапливает ошибки оценивания навигационных параметров, вызванные ошибками измерений инерциальных датчиков. Связь ошибок БИНС с ошибками датчиков чрезвычайно сложная, т.к. характер накопления ошибок БИНС существенно зависит также и от маневров, совершаемых ЛА, в процессе функционирования БИНС. Поэтому исследовать в общем виде влияние ошибок БИНС в целом и инерциальных датчиков, в частности, на точность управления пространственным движением ЛА удается только путем численного моделирования. В то же время, для выяснения некоторых особенностей взаимодействия задач управления и идентификации вектора состояния ЛА представляется необходимым провести аналитические исследования на упрощенной модели управляемого процесса.

В данной статье проводится исследование точности достижения конечных условий в задаче терминального управления движением объекта, решаемой с помощью замкнутой САУП по информации инерциальных датчиков.

3 МАТЕРИАЛЫ И МЕТОДЫ

В основу решения задачи терминального управления полагается некоторая программно заданная траектория, соединяющая краевые точки, для которой согласно концепции обратных задач динамики [5] определяется программное управление $u^*(t)$.

Зададим программную траекторию, как экстремаль функционала $\int_0^T \dot{j}^2(t) \cdot dt = \int_0^T u^2(t) \cdot dt$ [14]. В этом случае

$$r^*(t) = b_3 \cdot t^3 + b_2 \cdot t^2 + b_1 \cdot t + b_0, \quad (4)$$

а коэффициенты b_i , $i = \overline{0,3}$ определяются из условия согласования решения (4) с краевыми условиями (2), (3):

$$\begin{aligned} b_0 &= r_0, \\ b_1 &= \dot{r}_0, \\ b_2 &= \frac{3(r_T - r_0)}{T^2} - \frac{\dot{r}_T + 2\dot{r}_0}{T}, \\ b_3 &= \frac{2(r_0 - r_T)}{T^3} + \frac{\dot{r}_T + \dot{r}_0}{T^2}. \end{aligned} \quad (5)$$

Таким образом, располагая уравнением программной траектории (4), из (1) находим программное управление в виде:

$$u^*(t) = 6b_3t + 2b_2$$

или, после подстановки (5):

$$u^*(t) = \left(12 \frac{r_0 - r_T}{T^3} + 6 \frac{\dot{r}_T + \dot{r}_0}{T^2} \right) \cdot t + 6 \frac{r_T - r_0}{T^2} - 2 \frac{\dot{r}_T + 2\dot{r}_0}{T}. \quad (6)$$

В дальнейшем, учитывая предполагаемый идеальный характер движения, программное управление (6) и порождаемую им программную траекторию (4) будем называть эталонными.

Найденное программное управление соответствует разомкнутой схеме управления и, по понятным причинам, не может использоваться на практике. Для осуществления движения вдоль программной траектории в условиях возможных параметрических и динамических возмущений предлагается воспользоваться алгоритмом синтеза по методу преследования ведущей точки [5].

Будем считать, что в каждый момент времени доступны измерения координаты и скорости $\hat{r}(t)$, $\dot{\hat{r}}(t)$. В этих условиях решается задача перевода за некоторое время τ , в дальнейшем называемое глубиной прогноза, вектора состояния (r, \dot{r}) из точки $(\hat{r}(t), \dot{\hat{r}}(t))$ в точку $(r^*(t + \tau), \dot{r}^*(t + \tau))$. Как видно из постановки задачи, целевая точка выбирается на эталонной траектории (4), опережает текущую точку по времени на τ и к моменту времени T , согласно процедуре построения траектории (4), приходит в требуемое конечное положение, обеспечивая решение исходной задачи терминального управления.

Итак, дополнительно учитывая ограниченное по модулю возмущение $|f| \leq f^*$, математическая модель задачи синтеза имеет вид:

$$\ddot{r}(t') = u(t') + f(t'),$$

где $t' \in [0, \tau]$ – переменная локального времени, с краевыми условиями:

$$\begin{aligned} r(0) &= \hat{r}(t), & \dot{r}(0) &= \dot{\hat{r}}(t), \\ r(\tau) &= r^*(t + \tau), & \dot{r}(\tau) &= \dot{r}^*(t + \tau). \end{aligned} \quad (7)$$

Соединяя начальную и конечную точки (7) кубическим полиномом от переменной локального времени t' , получим

$$r(t') = a_3 t'^3 + a_2 t'^2 + a_1 t' + a_0,$$

где

$$\begin{aligned} a_0 &= \hat{r}(t), & a_1 &= \dot{\hat{r}}(t), \\ a_2 &= 3 \frac{r^*(t + \tau) - \hat{r}(t)}{\tau^2} - \frac{\dot{r}^*(t + \tau) + 2\dot{\hat{r}}(t)}{\tau}, \\ a_3 &= 2 \frac{\dot{\hat{r}}(t) - \dot{r}^*(t + \tau)}{\tau^3} + \frac{\dot{r}^*(t + \tau) + \dot{\hat{r}}(t)}{\tau^2}, \end{aligned}$$

для управления получим:

$$u(t') = 6a_3 t' + 2a_2. \quad (8)$$

Ограничивая временной интервал использования управления (8) моментом локального времени $t' = 0$, в качестве закона синтеза окончательно получим:

$$u(t, \hat{r}, \dot{\hat{r}}) = 6 \cdot \frac{r^*(t + \tau) - \hat{r}(t)}{\tau^2} - 2 \cdot \frac{\dot{r}^*(t + \tau) + 2 \cdot \dot{\hat{r}}(t)}{\tau}, \quad (9)$$

в котором использованы как фактические текущие значения оценки вектора состояния $r(t), \dot{r}(t)$, так и его прогнозируемые эталонные значение $r^*(t + \tau), \dot{r}^*(t + \tau)$, представляющие собой известную функцию времени.

Для анализа эффективности выбранной схемы синтеза исследуем замкнутую систему управления с точки зрения возможности реализации эталонной траектории.

Будем считать, что фактическое значение переменной $r(t)$, соответствующее решению возмущенного уравнения движения, отличается от эталонного значения $r^*(t)$, задаваемого (4), на величину $\delta r(t) = r(t) - r^*(t)$. Представим закон синтеза управления (9) в виде:

$$u(t, r, \dot{r}) = u^*(t) + \delta u(\delta r, \delta \dot{r}),$$

где второе слагаемое выражается как

$$\delta u(t) = -\frac{6}{\tau^2} \delta r(t) - \frac{4}{\tau} \delta \dot{r}(t). \quad (10)$$

В этих условиях уравнение для ошибки

$$\delta \ddot{r}(t) = \delta u + f(t)$$

с учетом (10) приводится к виду

$$\delta \ddot{r}(t) + \frac{4}{\tau} \delta \dot{r}(t) + \frac{6}{\tau^2} \delta r(t) = f(t) \quad (11)$$

и имеет решение

$$\delta r(t) = e^{-\frac{2}{\tau}t} \left(C_1 \cos \frac{\sqrt{2}}{\tau}t + C_2 \sin \frac{\sqrt{2}}{\tau}t \right) + \delta r^f(t), \quad (12)$$

в котором коэффициенты C_1, C_2 определяются из начальных условий $\delta r(0), \delta \dot{r}(0)$, а вынужденная составляющая

$$\delta r^f(t) = \frac{\tau}{\sqrt{2}} \int_0^t e^{-\frac{2}{\tau}\theta} \sin \left(\frac{\sqrt{2}}{\tau} \cdot \theta \right) \cdot f(t - \theta) \cdot d\theta \quad (13)$$

соответствует свертке импульсной переходной функции инерциального звена второго порядка, каковым является (11), и функции $f(t)$.

Для получения грубой оценки $\delta r^f(t)$ при $t \rightarrow \infty$ без конкретизации функциональной зависимости $f(t)$, воспользуемся соотношениями $|f(t)| \leq f^*$ и $\left| \sin \frac{\sqrt{2}}{\tau} \cdot \theta \right| \leq 1$,

где f^* – максимальная величина воздействия. Откуда

следует, что для любого $t \geq 0$ имеет место неравенство

$$|\delta r^f(t)| < \frac{\tau}{\sqrt{2}} \int_0^t e^{-\frac{2}{\tau}\theta} \cdot f^* \cdot d\theta = \frac{\tau^2}{2\sqrt{2}} f^* \left(1 - e^{-\frac{2}{\tau}t} \right) \xrightarrow{t \rightarrow \infty} \frac{\tau^2}{2\sqrt{2}} f^*. \quad (14)$$

Получить для (13) более точную оценку, соответствующую нестрогому неравенству, затруднительно, поэтому закономерен вопрос о достижимости, или о степени «грубости» полученной верхней оценки для $|\delta r^f(t)|$. Легко видеть, что в частном случае, когда $f(t) = f^* = Const$, вынужденная составляющая принимает вид $\delta r^f(t) = \frac{\tau^2}{6} f^*$ и отличается от оценки из (14) менее чем в два раза. Этот факт говорит о том, что использованные упрощения при получении оценки $|\delta r^f(t)|$ не приводят к существенному искажению результата.

Таким образом, из (12) следует, что при идеальных измерениях вектора состояния управление (9) при $f \equiv 0$ и не нулевых начальных условиях обеспечивает асимптотическую устойчивость движения относительно эталонной траектории, а при не нулевом, но ограниченном возмущении $|f(t)| \leq f^*$, точка гарантированно не отклоняется от эталонной траектории более чем на величину $\frac{\tau^2}{2\sqrt{2}} f^*$.

Перейдем к анализу влияния ошибки измерения вектора состояния на точность работы замкнутой системы управления и, в частности, на терминальную точность решения задачи управления.

Пусть оценка вектора состояния (r, \dot{r}) осуществляется с ошибкой. Будем считать, что исходным измерением является величина $a(t) = \ddot{r}(t) + \delta a$, соответствующая сумме истинного ускорения и постоянной погрешности δa , а оценка вектора состояния $(\hat{r}, \dot{\hat{r}})$ производится, согласно принципу инерциального счисления, путем двойного интегрирования измерений. Рассмотренный случай характерен для работы БИНС, в которой погрешность δa складывается из погрешности собственно акселерометра и ошибки компенсации вектора ускорения свободного падения, обусловленной погрешностью гироскопов. В некоторых режимах движения ЛА такую совокупную ошибку можно считать постоянной.

Итак, с учетом $\ddot{\hat{r}}(t) = a(t)$ для оценки $\hat{r}, \dot{\hat{r}}$, совпадающей в начальный момент времени с фактическими значениями одноименных переменных, имеем:

$$\dot{\hat{r}}(t) = \dot{r}(t) + \delta a \cdot t,$$

$$\hat{r}(t) = r(t) + \delta a \cdot \frac{t^2}{2}.$$

Таким образом, после подстановки (10), (11) в закон управления (9), для замкнутой системы уравнение ошибки реализации эталонного движения принимает вид:

$$\delta \ddot{r}(t) + \frac{4}{\tau} \delta \dot{r}(t) + \frac{6}{\tau^2} \delta r(t) = -\frac{4\delta a}{\tau} \cdot t - \frac{3\delta a}{\tau^2} \cdot t^2 + f(t). \quad (15)$$

Полагая, что в начальный момент времени $\delta r(0) = 0, \delta \dot{r}(0) = 0$, уравнение (15) имеет решение

$$\delta r(t) = -\frac{\delta a}{2} t^2 + \frac{\delta a}{6} \tau^2 - e^{-\frac{2}{\tau} t} \frac{\delta a \tau^2}{3} \left(\frac{1}{2} \cos \frac{\sqrt{2}}{\tau} t + \frac{1}{\sqrt{2}} \sin \frac{\sqrt{2}}{\tau} t \right) + \delta r^f(t). \quad (16)$$

В отличие от (12), где $\delta r(t)$ является ограниченной по модулю величиной, в (16) абсолютное значение $\delta r(t)$ неограниченно растет с увеличением времени t , что объясняется накапливаемым характером ошибки идентификации вектора состояния в БИНС.

Аналогичный вывод справедлив и для ошибки терминальной скорости:

$$\delta \dot{r}_T^{\delta a}(\tau) = \frac{\delta a}{\sqrt{2}} \tau \cdot e^{-\frac{2}{\tau} T} \cdot \sin \frac{\sqrt{2}}{\tau} \cdot T - \delta a \cdot T. \quad (17)$$

4 ЭКСПЕРИМЕНТЫ

Был проведен вычислительный эксперимент с использованием разработанного программного комплекса, имитирующего работу САУП [15].

Для проверки правильности аналитических зависимостей (16), (17) выбрано устойчивое прямолинейное движение, при котором изменяется только одна координата. В ходе эксперимента интегрировались как параметры реального движения, так и оценки алгоритма БИНС.

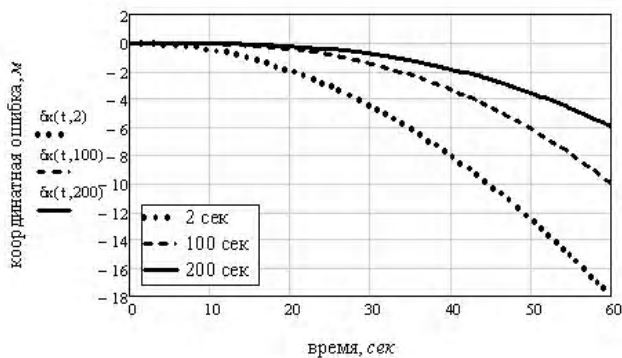


Рисунок 1 – Координатная ошибка при использовании закона управления с различными значениями $\tau \in \{2 \text{ сек}, 100 \text{ сек}, 200 \text{ сек}\}$

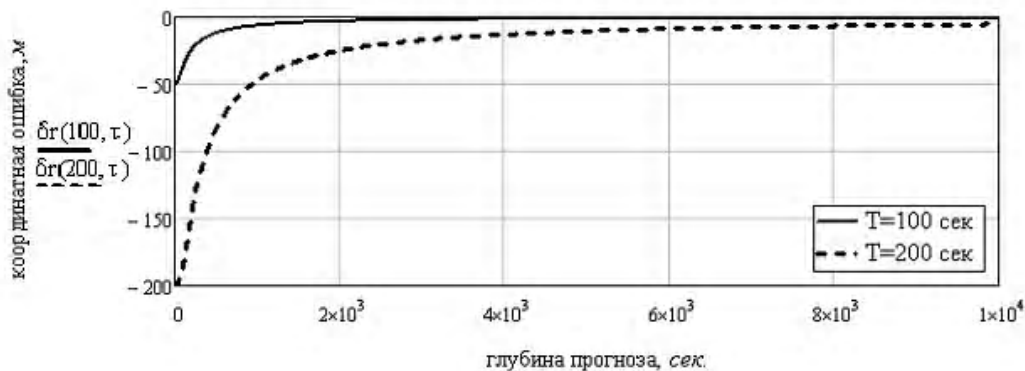


Рисунок 2 – Зависимость координатной ошибки в терминальной точке от глубины прогноза

5 РЕЗУЛЬТАТЫ

Эксперимент на малом интервале времени равном 100 сек., в котором зависимость $\delta r(t)$ при $f(t) \equiv 0$ и заданном уровне ошибок измерения акселерометра δa получена для различных значений глубины прогноза τ . Результаты эксперимента демонстрирует рис. 1.

Второй эксперимент проводился с целью демонстрации возможности нивелирования влияния ошибок измерений путем выбора параметра глубины прогноза τ на фиксированном интервале движения T . Поведение зависимости (16) как функции параметра τ показано на рис. 2.

6 ОБСУЖДЕНИЯ

Поведение координатной ошибки во время движения в терминальную точку (рис. 1) подтверждает вывод о том, что ошибка управления (16) неограниченно возрастает с увеличением времени t из-за накопления ошибки определения вектора состояния БИНС, причем, зависимость ошибки управления от времени t близка к квадратичной.

Приведенные на рис. 1 результаты примечательны также тем, что отражают зависимость скорости нарастания ошибки управления от глубины прогноза τ , являющейся свободным параметром закона управления (9). Из уравнения (15) видно, что чем больше τ , тем меньше при прочих равных условиях влияние δa на динамику δr . Так, при $\delta a = 0,08 \text{ м/с}^2$ и терминальной продолжительности $T = 60 \text{ с}$, модуль терминальной ошибки $|\delta r(T)|$ для параметра τ , равного 2 с, 100 с и 200 с, достигает значений 18 м, 10 м и 6 м соответственно (рис. 1). Указанная закономерность сохраняется и при других значениях T .

Во втором эксперименте рассматривалась координатная ошибка $\delta r(t)$ при фиксированном времени движения ($T = \text{const}$), что позволило представить (16) в виде:

$$\delta r_T^{\delta a}(\tau) = -\frac{\delta a}{2} T^2 + \frac{\delta a}{6} \tau^2 - e^{-\frac{2}{\tau} T} \frac{\delta a \tau^2}{3} \left(\frac{1}{2} \cos \frac{\sqrt{2}}{\tau} T + \frac{1}{\sqrt{2}} \sin \frac{\sqrt{2}}{\tau} T \right). \quad (18)$$

Для двух вариантов терминальной продолжительности $T=100 \text{ сек}$ и $T=200 \text{ сек}$ параметр τ менялся от 0,01 сек до 10^4 сек при отсутствии динамического возмущения

($f(t) \equiv 0$) и ненулевых ошибках измерения δa . Обе кривые на рис. 2 стремятся к оси абсцисс, что показывает монотонное уменьшение терминальной ошибки при увеличении τ . При этом, анализируя (18), несложно показать, что для любого конечного T имеет место $\lim_{\tau \rightarrow \infty} (\delta r_T(\tau)) = 0$. Аналогичный вывод справедлив и для ошибки терминальной скорости:

$$\delta \dot{r}_T^{\delta a}(\tau) = \frac{\delta a}{\sqrt{2}} \tau \cdot e^{-\frac{2}{\tau} T} \cdot \sin \frac{\sqrt{2}}{\tau} \cdot T - \delta a \cdot T.$$

Таким образом, при пренебрежимо малых динамических возмущениях (f^* – мало) параметр τ , входящий в закон управления (9), целесообразно выбирать большим. Такой выбор позволяет уменьшить влияние ошибки измерений ускорения на терминальную точность управления. Однако данный вывод не может быть окончательным, поскольку в условиях ненулевых динамических возмущений увеличение параметра τ , как следует из (13), приводит к увеличению ошибки реализации эталонной траектории и, как следствие, к уменьшению точности выполнения терминальных условий.

ВЫВОДЫ

Проведено исследование точности функционирования замкнутой системы управления с учетом динамического возмущения и погрешности измерений параметров движения инерциальной навигационной системой. Решена задача терминального управления. Закон управления включает в себя программную составляющую управления, соответствующую невозмущенному движению вдоль эталонной траектории, и поправку, осуществляющую синтез по методу преследования ведущей точки. Синтез управления осуществляется по информации, полученной путем интегрирования измерений фактического ускорения и содержащей ошибку, характерную для акселерометров. В результате анализа терминальной точности функционирования замкнутой системы получены следующие выводы-рекомендации:

- при увеличении постоянной времени передаточной функции замкнутой системы для ошибки влияние ошибок измерений уменьшается;
- для уменьшения влияния динамического возмущения на терминальную точность следует уменьшать коэффициент усиления передаточной функции замкнутой системы для ошибки;
- при выборе варьируемого параметра в законе синтеза по методу преследования ведущей точки следует руководствоваться следующими соображениями: если информационно-измерительная система обеспечивает получение координат и скорости объекта с высокой точностью, то выбираемый параметр прогноза должен быть минимальным. При использовании датчиков низкой точности параметр прогноза следует увеличить. Окончательное решение проблемы настройки алгоритма синтеза проводить с использованием моделирования типовых режимов работы.

В данной работе сделана попытка проанализировать требования к замкнутой системе управления с позиций влияния на точность как динамического возмущения, так и ошибок измерения вектора состояния. Показано, что влияние упомянутых факторов прямо противоположно. Сформулированные рекомендации могут быть полезны для проектирования алгоритмов автоматического управления движением в комплексе с проектированием информационно-измерительной системы.

СПИСОК ЛИТЕРАТУРЫ

1. Performance Based Navigation. Doc 9613. – Montreal : ICAO, 2008. – 290 p.
2. Ревнивых С. Г. Тенденции развития глобальной спутниковой навигации / С. Г. Ревнивых // Международная конференция по интегрированным навигационным системам: XIX международная конференция, Санкт-Петербург, 28–30 мая 2012 г.: сборник докладов. – СПб. : ГНЦ РФ ЦНИИ «Электроприбор», 2012. – С. 266–276.
3. Разработка и испытание интегрированной инерциально-спутниковой навигационной системы НСИ-2000MTG с расширенной доступностью спутниковых измерений / [Т. Н. Вахитов, А. Б. Колчев, К. Ю. Счастливцев, В. Б. Успенский, П. В. Ларионов, А. А. Фомичев] // Международная конференция по интегрированным навигационным системам: XIX международная конференция, Санкт-Петербург, 28–30 мая 2012 г.: сборник докладов. – СПб. : ГНЦ РФ ЦНИИ «Электроприбор», 2012. – С. 226–235.
4. Крутько П. Д. Алгоритмы терминального управления линейных динамических систем / П. Д. Крутько // Известия РАН. Теория и системы управления. – 1998. – № 6. – С. 33–45.
5. Крутько П. Д. Обратные задачи динамики управляемых систем. Линейные модели / П. Д. Крутько. – М. : Наука, 1987. – 304 с.
6. Батенко А. П. Системы терминального управления / А. П. Батенко. – М. : Радио и связь, 1984. – 160 с.
7. Salychev O. S. Applied Inertial Navigation: Problems and Solutions / O. S. Salychev. – М. : BMSTU Press, 2004. – 304 p.
8. Галиулин А. С. Методы решения обратных задач динамики / А. С. Галиулин. – М. : Наука, 1986. – 224 с.
9. Дорф Р. Современные системы управления / Р. Дорф, Р. Бишоп. – М. : Лаборатория Базовых Знаний, 2002. – 832 с.
10. Теория управления. Терминология. – М. : Наука, 1988. – Вып. 107. – 56 с.
11. Неусыпин К. А. Современные системы и методы наведения, навигации и управления летательными аппаратами / К. А. Неусыпин. – М. : МГОУ, 2009. – 500 с.
12. Barbour N. M. Inertial Navigation Sensors [Electronic resource] / N. M. Barbour // NATO RTO Lecture Series, RTO-EN-SET-116, Low-Cost Navigation Sensors and Integration Technology, 2010. – Access mode: [http://ftp.rta.nato.int/public/PubFullText/RTO/EN/RTO-EN-SET-116-2010/EN-SET-116\(2010\)-02.pdf](http://ftp.rta.nato.int/public/PubFullText/RTO/EN/RTO-EN-SET-116-2010/EN-SET-116(2010)-02.pdf)
13. Голован А. А. Математические основы навигационных систем. Часть I. Математические модели инерциальной навигации / А. А. Голован, Н. А. Парусников. – М. : Изд-во МГУ, 2011. – 132 с.
14. Успенский В. Б. Теоретические основы гиросилового управления ориентацией космического летательного аппарата / В. Б. Успенский. – Харьков : НТУ «ХПИ», 2006. – 328 с.
15. Хацько Н. Е. Исследование возможности управления полетом по данным инерциальных датчиков низкого класса точности / Н. Е. Хацько // Проблемы машиностроения. – Харьков : ПИМАШ. – 2013. – Т. 16, № 5. – С. 32–35.

Статья поступила в редакцию 07.12.2015.
После доработки 14.12.2015.

Хацько Н. Є.

Канд. техн. наук, старший викладач кафедри систем і процесів управління Національного технічного університету «Харківський політехнічний інститут», Харків, Україна

РОЗВ'ЯЗАННЯ ТЕРМІНАЛЬНОЇ ЗАДАЧІ УПРАВЛІННЯ З ВИРИСТАННЯМ ІНФОРМАЦІЇ ІНЕРЦІАЛЬНОГО БЛОКА

Розглянуто задачу термінального управління літальним апаратом за інформацією безплатформної інерціальної навігаційної системи, в якій за заданий час потрібно перевести апарат у задане просторове положення з необхідним значенням вектора кінцевої швидкості. Побудовано програмне управління для незбуреного руху літального апарата. Синтезовано функцію управління літального апарату за поточним значенням його вектора стану, визначеного інерціальним навігаційним блоком. Отримано вирази для оцінки точності вирішення термінальної задачі з урахуванням похибки акселерометра. Проаналізовано точність виведення літального апарату в кінцеву точку замкнутою системою управління при використанні інформації від інерціальних датчиків з відомим рівнем похибок.

На прикладі спрощеної моделі проведено комп'ютерне моделювання руху літального апарату. Доведено, що точність приведення вектора стану в задану кінцеву точку залежить від рівня збурень і похибки вхідної інформації. Проведено чисельні експерименти, що дозволяють виявити вплив чисельного значення глибини прогнозу, використаного в синтезі управління за методом переслідування провідною точки, на точність рішення термінальної задачі. Сформульовано рекомендації для проектування алгоритмів автоматичного управління рухом в комплексі з проектуванням інформаційно-виміральної системи.

Ключові слова: система управління, безплатформна інерціальна навігаційна система, похибка виміру вектору стану, динамічне збурення, точність термінального управління.

Khatsko N. E.

PhD, Senior Lecturer of Systems and processes of control department, National Technical University «Kharkiv Polytechnic Institute», Kharkiv, Ukraine

APPROACH TO SOLVING THE TERMINAL CONTROL PROBLEM ON CONDITION OF IMPRECISE MEASURING OF STATE VECTOR

The article considers the problem of terminal control of an aircraft using the data from the strapdown inertial navigation system, specifically the case when the aircraft is moved to a specific spatial position having the required vector of terminal velocity. Built program model for not altered motion of aircraft. Developed software for aircraft control, based on current state-vector taken from inertial navigation system. Method of analysis of aircraft positioning accuracy in the final location of a closed-loop control system that is based on the information taken from inertial sensors that have a deliberately known level of errors. Deduced formulas for estimation of aircraft final position taking into account accelerometer errors.

On the base of a simplified model the analysis of the accuracy of the state vector bringing to a given endpoint is performed, depending on the input information level of the perturbation and inaccuracy of input information.

Conducted experiments demonstrate dependency of software accuracy on accuracy of terminal control problem solution.

The following recommendations were obtained after performing analysis of accuracy of a close-loop system functioning:

– With time, the impact of errors from inertial navigation system decreases;

– To reduce impact of dynamic altered motion onto the finale state, we should decrease coefficient of gain transfer function;

– When choosing a variable parameter in the law of Synthesis method of leading point persecution, you should be guided with the following considerations: if the information-measuring system provides the coordinates and velocity of the object with high accuracy, the selected parameter of the forecast should be minimal. When using low-accuracy sensors, the value of forecast parameter should be increased.

Keywords: control system, strapdown inertial navigation system, the measurement the error of the state vector measurement, dynamical disturbance, the accuracy of the terminal management.

REFERENCES

1. Performance Based Navigation. Doc 9613. Montreal: ICAO, 2008, 290 p.
2. Revniviykh S. G. Tendentsii razvitiya globalnoy sputnikovoy navigatsii, *Mezhdunarodnaya konferentsiya po integririvannym navigatsionnym sistemam: XIX mezhdunarodnaya konferentsiya, Sankt-Peterburg, 28–30 maya 2012 g.: sbornik dokladov*. SPb, GNTs RF TsNII «Elektropribor», 2012, pp. 266–276.
3. Vakhitov T. N., Kolchev A. B., Schastlivets K. Yu., Uspenskiy V. B., Larionov P. V., Fomichev A. A. Razrabotka i ispytanie integririvannoy inertsialno-sputnikovoy navigatsionnoy sistemy NSI-2000MTG s rasshirennoy dostupnostyu sputnikovyykh izmereniy, *Mezhdunarodnaya konferentsiya po integririvannym navigatsionnym sistemam: XIX mezhdunarodnaya konferentsiya, Sankt-Peterburg, 2–30 maya 2012 g.: sbornik dokladov*. SPb, GNTs RF TsNII «Elektropribor», 2012, pp. 226–235.
4. Krutko P. D. Algoritmy terminalnogo upravleniya lineynykh dinamicheskikh sistem, *Izvestiya RAN. Teoriya i sistemy upravleniya*, 1998, No. 6, pp. 33–45.
5. Krutko P. D. Obratnye zadachi dinamiki upravlyaemykh sistem. Lineynye modeli. Moscow, Nauka, 1987, 304 p.
6. Batenko A. P. Sistemy terminalnogo upravleniya. Moscow, Radio i svyaz, 1984, 160 p.
7. Salychev O. S. Applied Inertial Navigation: Problems and Solutions. Moscow, BMSTU Press, 2004, 304 p.
8. Galliulin A. S. Metody resheniya obratnykh zadach dinamiki. Moscow, Nauka, 1986, 224 p.
9. Dorf R., Bishop R. *Sovremennye sistemy upravleniya*. Moscow, Laboratoriya Bazovykh Znaniy, 2002, 832 p.
10. Teoriya upravleniya. Terminologiya. Moscow, Nauka, 1988, Vyp. 107, 56 p.
11. Neusypin K. A. *Sovremennye sistemy i metody navedeniya, navigatsii i upravleniya letatelnyimi apparatami*. Moscow, MGOU, 2009, 500 p.
12. Barbour N. M. Inertial Navigation Sensors [Electronic resource], *NATO RTO Lecture Series, RTO-EN-SET-116, Low-Cost Navigation Sensors and Integration Technology, 2010*. Access mode: [http://ftp.rta.nato.int/public/PubFullText/RTO/EN/RTO-EN-SET-116-2010/EN-SET-116\(2010\)-02.pdf](http://ftp.rta.nato.int/public/PubFullText/RTO/EN/RTO-EN-SET-116-2010/EN-SET-116(2010)-02.pdf)
13. Golovan A. A., Parusnikov N. A. *Matematicheskie osnovy navigatsionnykh sistem*. Chast I. Matematicheskie modeli inertsialnoy navigatsii. Moscow, Izd-vo MGU, 2011, 132 p.
14. Uspenskiy V. B. *Teoreticheskie osnovy girosilovogo upravleniya orientatsiy kosmicheskogo letatel'nogo apparata*. Kharkov, NTU «KhPI», 2006, 328 p.
15. Khatsko N. Ye. *Issledovanie vozmozhnosti upravleniya poletom po dannym inertsialnykh datchikov nizkogo klassa tochnosti, Problemi mashinobudovannya*. Kharkiv, IPMASH, 2013, vol.16, No. 5, pp. 32–35.

Наукове видання

**Радіоелектроніка,
інформатика,
управління**

№ 1/2016

Науковий журнал

Головний редактор – д-р фіз.-мат. наук В. В. Погосов

Заст. головного редактора – д-р техн. наук С. О. Субботін

Комп'ютерне моделювання та верстання
Редактор англійських текстів

С. В. Зуб
С. О. Субботін

Оригінал-макет підготовлено у редакційно-видавничому відділі ЗНТУ

Свідоцтво про державну реєстрацію
КВ № 6904 від 29.01.2003.

*Підписано до друку 05.04.2016. Формат 60×84/8.
Папір офс. Різогр. друк. Ум. друк. арк. 12,56.
Тираж 300 прим. Зам. № 253.*

69063, м. Запоріжжя, ЗНТУ, друкарня, вул. Жуковського, 64

Свідоцтво суб'єкта видавничої справи
ДК № 2394 від 27.12.2005.