

ПЕРЕДАЧА И ОБРАБОТКА ИНФОРМАЦИИ

УДК 621.396.96

С.Н. Кириллов, В.М. Бердников

УСТОЙЧИВЫЙ АЛГОРИТМ ОПРЕДЕЛЕНИЯ МЕСТОПОЛОЖЕНИЯ ДИКТОРА ДЛЯ РАСПРЕДЕЛЕННОЙ СЕТИ МИКРОФОНОВ

Предложен устойчивый алгоритм определения местоположения диктора для распределенной сети микрофонов на основе двухэтапной обработки «взрывных» компонентов речевого сигнала во временной области, позволяющий оценить позицию диктора с точностью до 0,16 м и вероятностью 0,73 при частоте оценки в среднем один раз в секунду. Кроме того, представлен конечный алгоритм решения разностно-дальномерной задачи на плоскости, который по сравнению с итеративным методом Ньютона обеспечивает в 1,5 раза меньшую ошибку местоположения диктора.

Ключевые слова: определение местоположения, распределенная сеть микрофонов, речевой сигнал, диктор, разность времени прихода сигнала, двухэтапная обработка.

Введение. Возможность определения местоположения диктора на основе обработки речевого сигнала (РС) представляет интерес для целого ряда приложений:

- в автоматизированных системах слежения видеокамеры за местоположением докладчиков на конференциях, собраниях, выставках и т.п. [1, 2];
- обнаружение РС и определение направления на него для задач управления роботами [3, 4];
- локализации и идентификации лиц в системах безопасности;
- определение местоположения диктора вне зоны видимости (например, за преградой, стеной и т.п.).

Как правило [5, 6], системы определения местоположения диктора строятся на основе микрофонных решеток (МР) при расстоянии d между соседними микрофонами

$$d_{MP} < \lambda_{\min} / 2 = c / 2f_{\max}, \quad (1)$$

где λ_{\min} - минимальная длина волны звука, c - скорость звука (около 330 м/с [4]), f_{\max} - максимальная частота звука.

Как следует из формулы (1), расстояние d_{MP} для РС при $f_{\max} \leq 4000$ Гц будет составлять несколько сантиметров. Использование большего расстояния d приводит к неоднозначностям взаимнокорреляционной функции (ВКФ) РС

между выбранной парой микрофонов, вследствие высокого уровня боковых лепестков (БЛ) [сравнимых с главным лепестком (ГЛ)], что проявляется в связи с природой РС. В то же время МР для определения местоположения диктора работают только в ближней зоне (расстояние до источника речи сравнимо с d_{MP}), так как в дальней зоне возникают грубые ошибки разрешения источников РС в пространстве [7].

Применение нескольких разнесенных МР, определяющих направление на источник РС и, следовательно, оценивающих его местоположение по пересечению направлений, является достаточно дорогой и сложной технической задачей, обеспечивающей не очень высокую точность [7, 8].

Целью работы является разработка устойчивого двухэтапного алгоритма определения местоположения источника РС в дальней зоне на основе распределенной сети микрофонов ($d \gg d_{MP}$).

Описание алгоритма. Основными методами определения местоположения источников РС являются:

- 1) DOA (Direction Of Arrive) – определение угла прихода звуковой волны с помощью направленных микрофонов или нескольких МР [5, 7];
- 2) TDOA (Time Difference Of Arrive) – определение разности времени прихода звуковой

волны в синхронизированных микрофонах [9];

3) комбинированное решение, основанное на сочетании DOA и TDOA [5].

В данной работе предлагается модификация метода TDOA на основе двухэтапной обработки РС, обеспечивающая снижение требований к характеристикам микрофонов и их калибровке, в то время как использование метода DOA требует применения нескольких калиброванных МР с достаточно узкой диаграммой направленности (соответственно большим количеством микрофонов).

Хотя метод TDOA для МР хорошо зарекомендовал себя в случае вычисления ВКФ соответствующих пар микрофонов, но при $d \gg d_{MP}$ он оказывается неустойчивым и даже непригодным, вследствие описанной выше неоднозначности ВКФ. В современной технической литературе не представлены пояснения или описание работающих алгоритмов на основе метода TDOA для данного случая.

Далее рассмотрим предлагаемый алгоритм.

Пусть N_{mic} - число используемых и синхронизированных микрофонов, t_{rec} - время записи, f_s - частота дискретизации РС. Основой метода TDOA является определение разности времени прихода $\Delta\tau$ одинаковых фрагментов РС между выбранными парами микрофонов (каналов).

В связи с неидеальностью канала распространения РС, сигналы одного и того же фрагмента речи в разных микрофонах отличаются друг от друга и сдвинуты по времени. Кроме того, отличия РС в разных каналах обусловлены влиянием шумов, реверберацией, отличием переходных характеристик микрофонов и различного рода помех. В то же время общая тенденция изменения формы сигналов со временем сохраняется.

Из анализа записей РС можно отметить, что форма огибающей РС имеет периодически повторяющиеся спады или паузы, а также подъемы [«взрывные» компоненты (ВК)] разной интенсивности.

Для разработки устойчивого алгоритма определения разности времени прихода фрагментов РС между соответствующими парами микрофонов целесообразно использовать двухэтапную процедуру обработки РС, когда на первом этапе производится грубая оценка местоположения диктора, а на втором более точная оценка на основе корреляционной обработки. Использование данного алгоритма позволит обойти проблему неоднозначности ВКФ и обеспечит более высокую точность местоположения диктора.

Блок-схема определения местоположения

диктора представлена на рисунке 1, а подробное описание последовательности действий в соответствующих блоках приведено ниже.



Рисунок 1 – Блок-схема определения местоположения диктора

В блоке 1 осуществляется вычисление дисперсии шума D_{ni} в паузах РС и дисперсии РС D_{iq} в пределах окон шириной Δ_{var} каждого i -го канала (здесь q - номер окна). Далее, ВК идентифицируются при следующих условиях:

- превышении значения D_{iq} над D_{ni} ;

- возрастание значения величины D_{iq} последовательно как минимум в n_{q1} окнах, где середина первого окна является грубой границей ВК слева \tilde{t}_{ij}^e ;

- последующее уменьшение величины D_{iq}

последовательно как минимум в n_{q2} окнах, середина n_{q2} -го окна является грубой границей ВК справа \tilde{t}_{ij}^l .

Соответственно, чем больше значений n_{q1} и n_{q2} , тем более выраженные ВК будут определены в РС, но при этом возрастает вероятность пропуска «слабых» ВК.

В блоке 2 производится уточнение границ ВК (t_{ij}^e и t_{ij}^l) путем оценок минимума дисперсии $D_{\min ij}^e$ и $D_{\min ij}^l$ в окрестностях величин \tilde{t}_{ij}^e и \tilde{t}_{ij}^l соответственно шириной Δ_{var} посредством уменьшения окна поиска до $\Delta_{\text{var}}/5$ (выбрано экспериментально) и расчета в нем среднего значения РС. Обозначим среднее значение РС слева как M_{ij}^{\min} .

Сопоставление соответствующих ВК друг другу осуществляется в каждом из каналов на основе сравнения левых границ t_{ij}^e и реального расстояния d между соответствующими микрофонами как

$$\left| t_{ij}^e - t_{i-1j}^e \right| \leq d_{i-1} + \delta, \quad (2)$$

где δ - величина, учитывающая погрешность определения границ ВК.

В ходе эксперимента над различными записями РС было установлено, что целесообразно использовать следующие значения параметров:

$$\Delta_{\text{var}} = 25 \text{ мс}; n_{q1} = 3; n_{q2} = 3; \delta = 30 \text{ см.}$$

В блоке 3 происходит вычисление среднеквадратических отклонений (СКО) σ_{ij} для каждого j -го ВК шириной w_{ij} во всех N_{mic} каналах на основе границ t_{ij}^e и t_{ij}^l , определенных на 3-м шаге ($w_{ij} = t_{ij}^l - t_{ij}^e$).

Далее определяются соответствующие пороги для каждого ВК

$$z_{ij} = n \cdot \sigma_{ij}, \quad (3)$$

где коэффициент n может варьироваться в пределах от единицы до значения, при котором пороги z_{ij} не превышают соответствующих максимумов данного ВК в любом из каналов.

В блоке 4 осуществляется грубая оценка разности времени прихода соответствующих ВК между выбранными каналами $\Delta\tau_{ii-1}^j$ посредством аппроксимации фронта ВК (первый этап оценки).

Вследствие того, что РС имеет сильно изрезанный и нестационарный характер, а нарастание амплитуды ВК достаточно медленное (в сравнении с резкими звуками), определение разности времени прихода сигнала между выбранными каналами по переднему фронту ВК прямо на основе порогов z_{ij} , как это было сделано для резких звуков в [10] является крайне неустойчивым решением, приводящим к грубым ошибкам определения местоположения источника РС.

В результате этого предлагается параболическая аппроксимация по критерию минимума СКО фронта ВК, позволяющая «смягчить» последствия изрезанности фронта ВК РС и тем самым повысить устойчивость определения разности времени прихода сигнала $\Delta\tau$ между выбранными каналами.

Параболическая аппроксимация фронта ВК осуществляется следующим образом:

а) на основе последовательного вычисления окон дисперсии от t_{ij}^e до t_{ij}^l шириной $\Delta_{\text{var}}/10$ (выбрано экспериментально) определяется окно с максимальным значением $D_{\max ij}$ и правой границей фронта, соответствующего ВК считается середина данного окна с индексом t_{ij}^f и величиной M_{ij}^{\max} ;

б) оцениваются дисперсии p окон $D_{\text{inter } ij}^p$ между t_{ij}^e и t_{ij}^f и расчет средних значений РС в этих окнах M_{ijp}^{inter} , а также соответствующего индекса t_{ijp}^{inter} ;

в) построение фронта ВК с помощью параболической аппроксимации по точкам $\{(t_{ij}^e, M_{ij}^{\min}), \dots, (t_{ijp}^{\text{inter}}, M_{ijp}^{\text{inter}}), \dots, (t_{ij}^f, M_{ij}^{\max})\}$.

Пример параболической аппроксимации по критерию минимума СКО фронта ВК РС показан на рисунке 2.

Затем определяются пересечения фронтов ВК с соответствующими усредненными по n порогами z_{ij} и, следовательно, оценка моментов времени T_{ij} . Далее определяются разности времени прихода соответствующих ВК между выбранными каналами как

$$\Delta\tau_{ii-1}^j = T_{ij} - T_{i-1j}. \quad (4)$$

В блоке 5 производится уточнение величин $\Delta\tau_{ii-1}^j$ посредством оценки ближайшего пика ВКФ вблизи задержки $\Delta\tau_{ii-1}^j$ грубо оцененной на первом этапе оценки (второй этап). ВКФ вычислялась между ВК соответствующих друг

другу в выбранной паре микрофонов.

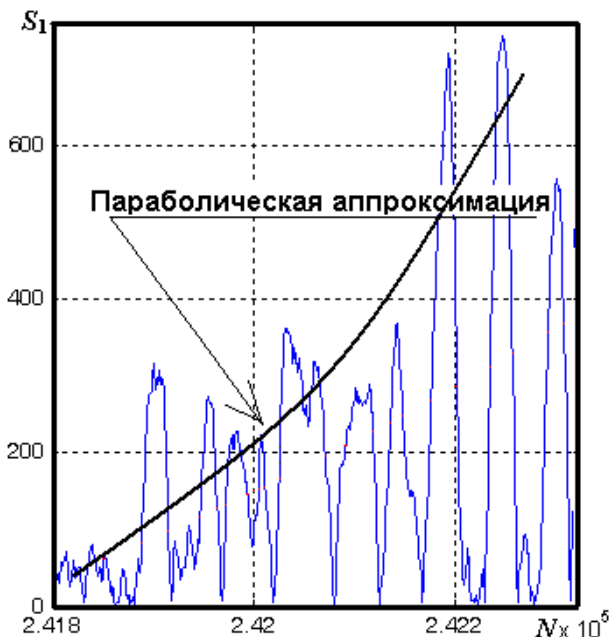


Рисунок 2 – Пример аппроксимации фронта ВК (здесь N – отсчеты РС)

Таким образом, на первом этапе на основе грубой оценки задержек устранялась неоднозначность ВКФ, а на втором этапе определялся пик соответствующего лепестка ВКФ.

В блоке 6 находится решение разностно-дальномерной задачи итеративным или конечным методом.

Хотя итеративный метод Ньютона более простой и быстросходящийся по сравнению с конечным алгоритмом, ему присущи следующие недостатки [11]:

- требуется задание начальных координат объекта, достаточно близких к реальному местоположению диктора, иначе возможно попадание в локальный минимум;

- непригодность в случае, если объект расположен вне периметра ограниченного микрофонами (например, для линейного распределения сети микрофонов).

В то же время конечные алгоритмы обеспечивают решение разностно-дальномерной задачи при различном взаимном расположении микрофонов и источника РС, но более чувствительны к ошибкам определения взаимных задержек $\Delta\tau$ между выбранными каналами при близком расположении соответствующих пар микрофонов [11, 12].

Конечные алгоритмы, предложенные в [11, 12], требуют большого числа микрофонов для определения источника РС в пространстве.

На основании этого является целесообразной разработка простого конечного алгоритма для определения местоположения источника РС

на плоскости при его расположении как в стороне от сети микрофонов, так и внутри ее периметра. Данный алгоритм требует минимум 3 микрофона.

Пусть для каждой пары микрофонов на основе вычисленной разности времени прихода сигнала $\Delta\tau_{ii-1}^j$ можно определить угол θ_{ii-1}^j между прямой линией, соединяющей два микрофона относительно ее центра, и направлением на источник РС (направляющая линия):

$$\theta_{ii-1}^j = \arccos(c \cdot \Delta\tau_{ii-1}^j / r_{ii-1}), \quad (5)$$

где r_{ii-1} - расстояние между соответствующими микрофонами.

Соответственно имея минимум две пары микрофонов, можно определить точку пересечения двух направляющих.

Пусть имеем два уравнения прямых

$$y_1 = k_1(x_1 - a_1) + b_1, \quad (6)$$

$$y_2 = k_2(x_2 - a_2) + b_2. \quad (7)$$

Так как в точке пересечения прямых (вероятном местоположении источника РС) соответствующие координаты двух прямых в теории должны быть равны, то выше представленные уравнения можно переписать следующим образом

$$y_0 = k_1(x_0 - a_1) + b_1, \quad (8)$$

$$y_0 = k_2(x_0 - a_2) + b_2. \quad (9)$$

После преобразования уравнений (8) и (9) получим решение в виде

$$y_0 = [k_2(a_1 - a_2 - b_1/k_1) + b_2] / (1 - k_2/k_1), \quad (10)$$

$$x_0 = (y_0 + k_1 a_1 - b_1) / k_1. \quad (11)$$

Для определения параметров k , a и b всех прямых, участвующих в решении разностно-дальномерной задачи, необходимо изначально задать точку отсчета прямоугольной системы координат и привести данные параметры к этой системе.

Таким образом, используя доступное количество независимых пар микрофонов и усреднив полученные координаты источника РС от пересечения различных направляющих можно получить приближенное решение на плоскости.

Структурная схема предложенного алгоритма представлена на рисунке 3.

Экспериментальная часть. Проведены экспериментальные исследования предложенного алгоритма при следующих условиях:

- количество микрофонов $N_{mic} = 4$, модель - Olympus LS-10 (стерео диктофон), чувствитель-

ность -40 дБ, диаграмма направленности кардиоидная для каждого моно канала;

- частота дискретизации $f_s = 44100$ Гц;
- длительность каждой записи РС $t_{rec} = 10$ с;

- ширина окна дисперсии $\Delta_{var} = 1000$ отсчетов;

- коэффициент n менялся в пределах всего фронта ВК с шагом 0,1 с последующим усреднением порогов до величины z_{ij} ;

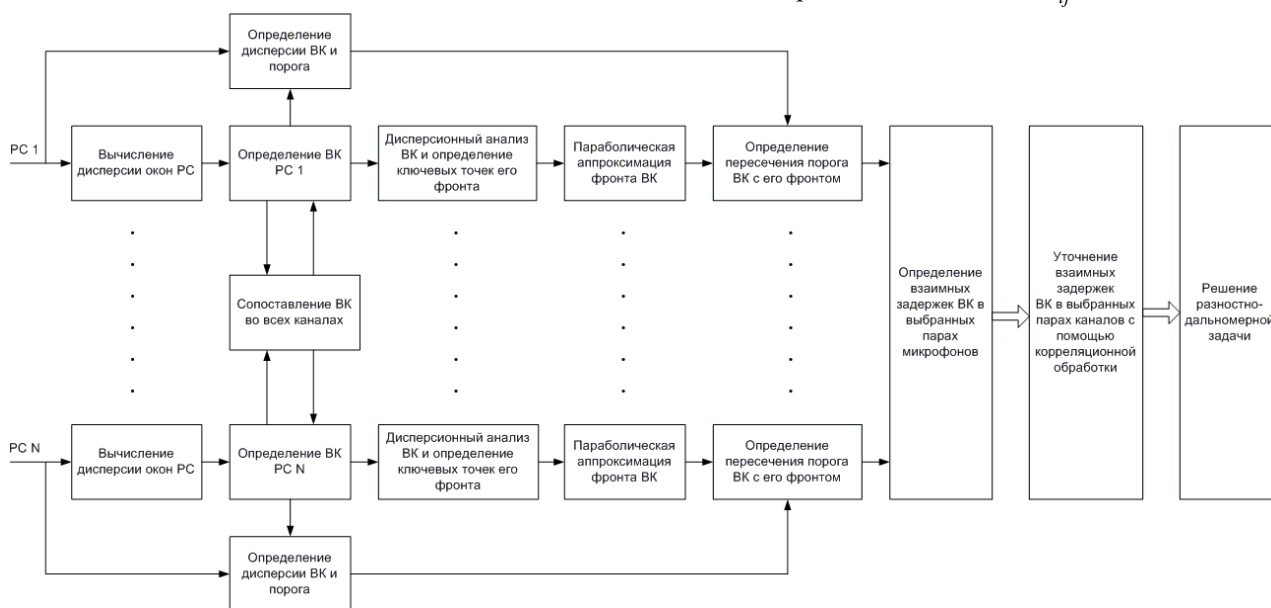


Рисунок 3 – Структурная схема алгоритма определения местоположения диктора

- размер прямоугольного помещения составлял: $x = 6$ м, $y = 3$ м, $h = 3$ м (здесь x - длина, y - ширина, h - высота);

- внутри помещения отсутствовали какие-либо преграды и объекты.

- микрофоны располагались в углах помещения на высоте $h = 0,85$ м от пола (см. таблицу 1);

Таблица 1 – Координаты микрофонов

Номер микрофона	Координаты, м		
	x	y	z
1	0,13	0,08	0,85
2	0,13	3,00	0,85
3	5,9	2,95	0,85
4	5,9	0,08	0,85

- в целях проверки работоспособности и эффективности предложенного алгоритма для каждой точки начитывался различный текст 10-ю дикторами (6 мужчин и 4 женщины);

- в ходе эксперимента определялись ошибки местоположения диктора как на основе только первого этапа оценки задержек, так и двух вместе;

- для решения разностно-дальномерной задачи использовались итеративный метод и конечный алгоритм;

- диктор последовательно размещался по всей площади помещения с шагом 0,5 м (итого 91 позиция).

План исследований.

1. Определение вероятности попадания оценки взаимных задержек ВК между каналами в область ГЛ ВКФ на первом этапе алгоритма.

2. Сравнение одноэтапного и двухэтапного алгоритмов оценки взаимных задержек ВК посредством вычисления средних ошибок позиционирования диктора на плоскости для итеративного и конечного методов решения разностно-дальномерной задачи в случае попадания оценки задержек в область ГЛ ВКФ.

3. Аналогично п. 2, но для случаев попадания оценки задержек в область БЛ ВКФ.

В результате анализа множества записей РС экспериментально было установлено, что минимальная ширина ГЛ ВКФ между ВК различных пар каналов равна 1 мс или 34 см по дальности. Вычисленная вероятность попадания оценки задержек в данную область составила. $P_{ML} = 0,73$.

Средняя ошибка определения местоположения диктора при использовании только первого этапа алгоритма оценки задержек и их попадания в область ГЛ ВКФ для итеративного метода равна 0,61 м, а для конечного – 0,34 м. В случае попадания оценки задержек в область БЛ средняя ошибка составила 1,23 м и 0,81 м соответственно.

При двухэтапном алгоритме в случае попадания оценки задержек в область ГЛ ВКФ средняя ошибка составила для итеративного метода 0,45 м, а для конечного алгоритма 0,16 м.

При попадании оценки задержек в область БЛ средняя ошибка составила 1,17 м и 0,97 м соответственно.

Оценим средние ошибки определения местоположения диктора для различных случаев с учетом вероятности попадания оценки взаимных задержек между ВК в область ГЛ ВКФ.

Для случая одноэтапного алгоритма оценки задержек и итеративного метода:

$$\delta_{OI} = P_{ML} \cdot 0,61 + (1 - P_{ML}) \cdot 1,23 = 0,78 \text{ м.}$$

Для случая одноэтапного алгоритма оценки задержек и конечного метода:

$$\delta_{OC} = P_{ML} \cdot 0,34 + (1 - P_{ML}) \cdot 0,81 = 0,47 \text{ м.}$$

Для случая двухэтапного алгоритма оценки задержек и итеративного метода:

$$\delta_{TI} = P_{ML} \cdot 0,45 + (1 - P_{ML}) \cdot 1,17 = 0,64 \text{ м.}$$

Для случая одноэтапного алгоритма оценки задержек и конечного метода:

$$\delta_{TC} = P_{ML} \cdot 0,16 + (1 - P_{ML}) \cdot 0,97 = 0,38 \text{ м.}$$

Для примера результаты эксперимента в точке с координатами (4,5; 1,7; 1,2) м представлены в таблице 2 (величины ошибок при одноэтапной оценке задержек между ВК представлены в круглых скобках, а ВК, оценка задержек которых на первом этапе не попала в ГЛ ВКФ, выделены полужирным шрифтом).

Таблица 2 – Результаты эксперимента в точке (4,5, 1,7, 1,2)

Номер ВК	Время ВК, с	Ошибка позиции, м	
		Итеративный	Конечный
1	0,13	0,42 (0,76)	0,04 (0,37)
2	0,45	1,17 (1,22)	1,02 (0,81)
3	1,56	0,26 (0,54)	0,16 (0,26)
4	2,54	0,45 (0,72)	0,20 (0,34)
5	3,61	0,51 (0,59)	0,05 (0,39)
6	4,17	1,10 (1,30)	0,91 (0,91)
7	4,76	0,49 (0,57)	0,26 (0,49)
8	5,67	1,43 (1,24)	1,03 (0,90)
9	6,81	0,46 (0,68)	0,17 (0,44)
10	8,16	0,61 (0,73)	0,50 (0,57)
11	9,18	0,38 (0,54)	0,21 (0,06)
12	9,78	0,48 (0,42)	0,02 (0,44)
Среднее значение		0,64 (0,78)	0,38 (0,47)

По результатам проведенного эксперимента можно сделать следующие выводы:

- предложенный алгоритм определения местоположения диктора на основе ВК РС обеспечивает среднюю точность позиционирования при двухэтапной оценке задержек на 20 % выше, чем при одноэтапной с частотой примерно один раз в секунду;

- вероятность попадания оценки задержек в область ГЛ ВКФ составила 0,73 при этом преимущество в точности оценок двухэтапной процедуры над одноэтапной равна 53 %. В случае попадания оценки задержек в область БЛ ВКФ на первом этапе, уточнение на втором этапе может как ухудшить, так и улучшить точность определения задержек ВК;

- точность определения местоположения диктора существенно зависит от предложенного метода решения разностно-дальномерной задачи, так предложенный конечный алгоритм, основанный на пересечении направляющих линий, позволяет достичь средней ошибки на уровне 0,38 м, в то время как итеративный - 0,64 м, т.е. примерно в 1,5 раза меньше.

Заключение. Предложен устойчивый алгоритм определения местоположения диктора для распределенной сети микрофонов на основе двухэтапной обработки ВК РС во временной области, позволяющий оценить позицию диктора с точностью до 0,16 м и вероятностью 0,73 при частоте оценки в среднем один раз в секунду. Кроме того, представлен простой конечный алгоритм решения разностно-дальномерной задачи на плоскости, который по сравнению с итеративным методом Ньютона обеспечивает в 1,5 раза меньшую ошибку местоположения диктора.

Библиографический список

1. H. Wang and P. Chu. Voice source localization for automatic camera pointing system in video-conferencing, Proc. of ICASSP, 1997.
2. Y. Huang, J. Benesty, and G. Elko. Passive acoustic source location for video camera steering, Proc. of ICASSP, 2000.
3. K. Nakadai, T. Matsui, H.G. Okuno, and H. Kitano. Active audition system and humanoid exterior design. In Proceedings International Conference on Intelligent Robots and Systems, 2000.
4. K. Nakadai, H.G. Okuno, and H. Kitano. Realtime sound source localization and separation for robot audition. In Proceedings IEEE International Conference on Spoken Language Processing, pages 193-196, 2002.
5. M. Brandstein and D. Ward (Eds.). Microphone Arrays signal processing techniques and applications, Springer, 2001.
6. Q. Lin, E. Jan, and J. Flanagan, "Microphone arrays and speaker identification," IEEE Trans on Speech and Audio Processing, vol. 2, no. 4, pp. 622-629, 1994.
7. Hui Liu, Evangelos Miliotis. Acoustic Positioning Using Multiple Microphone Arrays. Technical Report CS-2004-01. University Ave. Canada, 2004.
8. Wang Peng, Wee Ser. Sound Source Tracking Using Microphone Arrays. Nanayang Technological University. Singapore, 2000.
9. Y. Rui and D. Florencio, "Time delay estimation in the presence of correlated noise and reverberation," in

ICASSP'04, 2004, vol. 2, pp.: 133–136.

10. Бердников В.М. Алгоритм определения местоположения источников резких звуков для распределенной микрофонной решетки // 16-я всероссийская научно-техническая конференция студентов, молодых ученых и специалистов. НИТ-2011. РГРТУ. Рязань, 2011. – С. 34-37.

11. M.S. Brandstein, J.E. Adcock, J.H. DiBiase, and H.F. Silverman, “A closed-form method for finding

source locations from microphone array time-delay estimates,” in Proc. ICASSP-1995, Detroit, MI, May 1995, pp. 3019–3022.

12. H.C. Schau and A.Z. Robinson, “Passive source localization employing intersecting spherical surfaces from time-of-arrival differences” IEEE Trans. Acoust., Speech, Signal Process., vol. ASSP-35, no. 8, pp. 1223–1225, Aug. 1987.

УДК 681.325.5

С.В. Челебаев, В.Н. Локтюхин, А.В. Антоненко

СИНТЕЗ СТРУКТУРЫ АЦП СЛЕДЯЩЕГО ПРИНЦИПА ДЕЙСТВИЯ НА ОСНОВЕ МОДИФИЦИРОВАННОЙ СЕТИ ХОПФИЛДА

Обоснована и реализована возможность применения нейросетей с обратными связями для построения преобразователей частотно-временных и амплитудных параметров сигналов в цифровой код. Разработана процедура синтеза их структур с использованием сети Хопфилда с настраиваемыми в ходе ее обучения синаптическими связями между нейронами.

Ключевые слова: *нейронные сети, сеть Хопфилда, преобразователи формы представления информации, частота, код.*

Введение Основное содержание методологических положений и подходов к структурному синтезу аналого-цифровых преобразователей (АЦП) на основе нейросетевых технологий базируется на определяющей роли выбора архитектуры исходной нейросети [1]. Ее эффективный выбор обеспечивает и достижение необходимых технико-экономических характеристик преобразователя, и успешное обучение сети на решение поставленной задачи преобразования.

При этом в формулировке задачи также учитывается принцип действия АЦП, например циклический или следящий. Применение второго обеспечивает непрерывный характер аналого-цифрового (АЦ) преобразования, когда результат преобразования непрерывно «отслеживает» любое изменение входного аналогового сигнала в отличие от устройств циклического типа, в которых результат формируется только через интервал времени, определяемый циклом преобразования.

На настоящий момент известен целый ряд нейросетевых преобразователей частотно-временных и амплитудных параметров сигналов в цифровой код [1, 2], представляющих собой устройства разомкнутого циклического типа, не обеспечивающих поддержку оперативной адаптации выходного кода к изменению входного аналогового сигнала. При этом синтез указанных

устройств базируется в основном на применении исходно выбранных перцептронных однонаправленных сетей, что предопределяет циклический вид АЦ-преобразования. В связи с этим является актуальным выбор архитектуры нейросети как важного этапа процедуры синтеза структур нейросетевого АЦ-преобразователя замкнутого типа.

Укрупненный порядок проектирования (синтеза) АЦП следящего типа с функцией обучения Представленная в данном подразделе детализированная процедура синтеза нейропреобразователей с функцией обучения базируется на сформированных в [1] методологических положениях построения преобразователей.

Этап 1. Выбор и обоснование архитектуры исходной нейросети. Проведение этапа выбора и обоснования архитектуры исходной нейросети требует определения: вида сети (перцептронная, рекуррентная и т.д.), возможного количества ее слоев, структуры связей между ними, разновидностей применяемых активационных функций (эти характеристики могут уточняться). Наряду с этим, необходимо проведение анализа исходной сети по ее принадлежности к классу сетей, отличающихся способом решения задачи обучения: формируемые сети, сети с формируемой матрицей связи, обучаемые и комбинированные (смешанные) [3].

Выбор вида сети в значительной степени обосновывается ее характерными свойствами, а также известными предпочтениями в ее применении, зависящими от конкретных характеристик задачи преобразования информации. Так, для построения преобразователей замкнутого типа ориентированы сети с обратными связями, относящиеся к классу следящих систем. При необходимости возможно комплексирование различных видов сетей. Таким образом, можно утверждать, что уже на уровне выбора вида сети предопределяется направление успешного выполнения поставленной задачи преобразования.

Так как сети с обратными связями имеют пути, передающие сигналы от выходов к входам, то отклик таких сетей является динамическим, т.е. после приложения нового входа вычисляется выход и, передаваясь по сети обратной связи, модифицируется вход. Затем выход повторно вычисляется и процесс повторяется снова и снова. Для устойчивой сети последовательные итерации приводят к все меньшим изменениям выхода, пока выход не становится постоянным. Для многих сетей процесс никогда не заканчивается. Поэтому такие сети называют неустойчивыми.

Проблема устойчивости ставила в тупик первых исследователей. Из работы [4] известна теорема, описывающая подмножество сетей с обратными связями, выходы которых в конце концов достигают устойчивого состояния.

Дж. Хопфилд внёс важный вклад как в теорию, так и в применение нейросистем с обратными связями [5]. Поэтому некоторые их конфигурации известны как сети Хопфилда. Характерной особенностью данных сетей является их принадлежность к классу сетей с формируемой матрицей связей, так как, как правило, они имеют одинаковую структуру и различаются лишь матрицей связей.

На рисунке 1 показана сеть Хопфилда с обратными связями, состоящая из двух слоев [6, 7]. Нулевой слой не выполняет вычислительной функции, а лишь распределяет выходы сети обратно на входы. Каждый нейрон первого слоя вычисляет взвешенную сумму своих входов в виде сигнала S_j , который затем с помощью нелинейной функции активации F преобразуется в сигнал y_j . Эти операции сходны с нейронами других видов сетей, например персептронных или рекуррентных.

В работе Хопфилда [5] функция F являлась пороговой. Выход y_j нейрона 1-го слоя равен единице, если взвешенная сумма выходов с других нейронов больше порога T_j , в противном

случае она равна нулю. Значение y_j вычисляется следующим образом:

$$\begin{aligned} y_j &= 1, \text{ если } S_j > T_j, \\ y_j &= 0, \text{ если } S_j < T_j, \\ y_j &\text{ не изменяется, если } S_j = T_j, \end{aligned} \quad (1)$$

где $S_j = \sum_{i \neq j} w_{ij} y_i + x_j$.

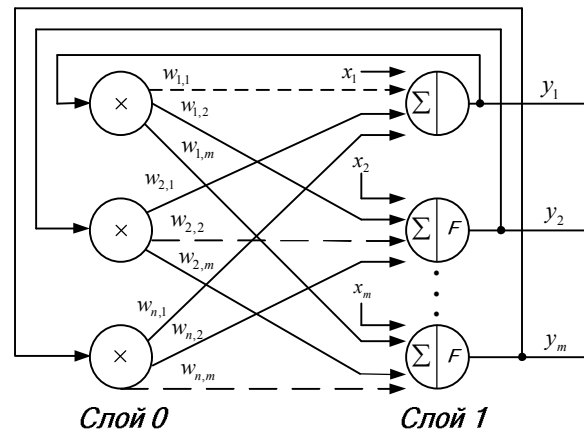


Рисунок 1 – Однослойная сеть с обратными связями

Состояние сети – это множество текущих значений сигналов y_j от всех нейронов. В первоначальной сети Хопфилда состояние каждого нейрона менялось в дискретные случайные моменты времени, а в последующей модификации состояния нейронов могли меняться одновременно. Поскольку выходом бинарного нейрона является ноль или единица, то текущее состояние сети является двоичным числом, каждый бит которого является сигналом y_j некоторого нейрона.

Функционирование сети описывается следующим образом. Когда подается новый входной вектор, сеть переходит из вершины в вершину, пока не стабилизируется. Устойчивая вершина определяется сетевыми весами, текущими входами и величиной порога функций активации. Если входной вектор частично неправилен или неполон, то сеть стабилизируется в вершине, ближайшей к желаемой.

Веса между слоями в этой сети могут рассматриваться в виде матрицы W . В работе [5] показано, что сеть с обратными связями является устойчивой, если ее матрица симметрична и имеет нули на главной диагонали, т.е. если $w_{ij} = w_{ji}$ и $w_{ii} = 0$ для всех i .

Этап 2. Определение базовой конфигурации структуры преобразователя как аппроксимирующей сети. Данный этап включает в

себя следующие действия: определение числа, вида и специфики представления входных и выходных сигналов нейросетевой модели преобразователя и ее нейроэлементов, их количества во всех слоях сети, а также видов возможной технологии задания весов $w_{j,i}^{(1)}$ и организации синаптических связей $w_{j,i}^{(1)} \cdot x_j$ между элементами. Итогом этого этапа является получение исходного аналитического и матричного описания функционирования всей ИНС, характеризующейся возможностью ее обучения конкретной задаче линейного или нелинейного преобразования [1].

Определение числа входов M и выходов K сети и особенностей задания на них сигналов производится с учетом формы представления преобразуемой величины x в виде частоты f_x или напряжения U_x и способа кодирования результата y_N^* в виде двоичного позиционного кода $y_N^* = b_m b_{m-1} \dots b_2 b_1 = \sum_{i=1}^m b_i \cdot 2^{i-1}$, поэтому количество выходов сети K равно числу разрядов m выходного кода.

Нейросеть, ориентированная на выполнение операции аналого-цифрового преобразования, будет иметь два входных сигнала: x_1 и x_2 и соответственно такое же число μ_0 нейронов в ее входном слое. В качестве одного из сигналов, например x_1 , может выступать преобразуемая величина x , а в качестве x_2 – эталонная (опорная) величина x_{on} .

Значение эталона x_{on} определяет меру преобразования (измерения). Эталон x_{on} обычно представляется в виде дискрета преобразования $\Delta x_0 = x_{max} / 2^m$, максимального значения x_{max} или величин, связанных с ними по обратно пропорциональной зависимости.

Так как относительная погрешность δ_{np} преобразования $x \rightarrow y_N^*$ находится в соответствии с выражением $\delta_{np} = 1/2^m$, то зависимость, связывающая δ_{np} с числом $\mu_1 = m$ нейронов $HV_i^{(1)}$ выходного слоя, имеет вид:

$$\delta_{np} = 1/2^{\mu_1}.$$

Предлагаемая модифицированная сеть Хопфилда (рисунок 2) отличается от сети (см. рисунок 1) наличием входа эталонной величины преобразования, а также связями, обеспечивающими нормирование значений разрядов

выходного кода в цепи обратной связи величиной эталона преобразования. С учетом количества входов и выходов однослойная структура преобразователя на основе модифицированной сети имеет вид, представленный на рисунке 2.

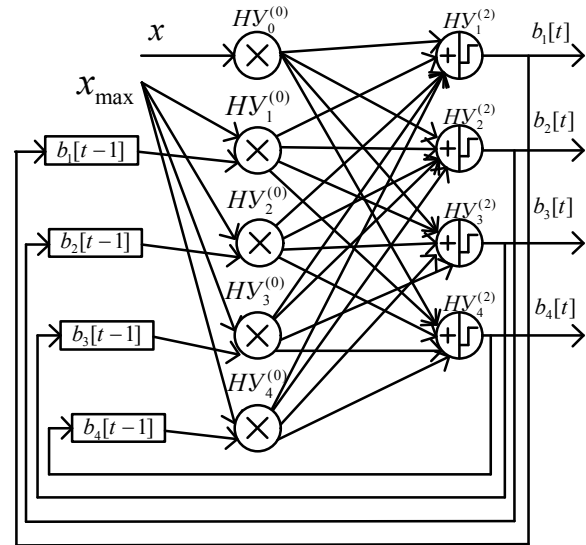


Рисунок 2 – Аналого-цифровой преобразователь на основе модифицированной сети Хопфилда

Сеть (см. рисунок 2) описывается совокупностью зависимостей (например, для 4-разрядного преобразователя):

$$b_1[t] = F \left(\begin{matrix} w_{1,1} b_1[t-1] x_{max} + w_{2,1} b_2[t-1] x_{max} + \\ w_{3,1} b_3[t-1] x_{max} + w_{4,1} b_4[t-1] x_{max} + w_{0,1} x \end{matrix} \right),$$

$$b_2[t] = F \left(\begin{matrix} w_{1,2} b_1[t-1] x_{max} + w_{2,2} b_2[t-1] x_{max} + \\ w_{3,2} b_3[t-1] x_{max} + w_{4,2} b_4[t-1] x_{max} + w_{0,2} x \end{matrix} \right),$$

$$b_3[t] = F \left(\begin{matrix} w_{1,3} b_1[t-1] x_{max} + w_{2,3} b_2[t-1] x_{max} + \\ w_{3,3} b_3[t-1] x_{max} + w_{4,3} b_4[t-1] x_{max} + w_{0,3} x \end{matrix} \right),$$

$$b_4[t] = F \left(\begin{matrix} w_{1,4} b_1[t-1] x_{max} + w_{2,4} b_2[t-1] x_{max} + \\ w_{3,4} b_3[t-1] x_{max} + w_{4,4} b_4[t-1] x_{max} + w_{0,4} x \end{matrix} \right),$$

где $F(S) = \begin{cases} 1, & \text{если } S_i \geq T_i, \\ 0, & \text{если } S_i < T_i; \end{cases}$ – функция активации;

$y_N^* = b_4 b_3 b_2 b_1$ – позиционный двоичный код;
 x – преобразуемая аналоговая величина, представленная в виде частоты f_x , временного интервала τ_x или напряжения U_x ;

x_{max} – эталон преобразования, представленный в виде максимального значения преобразуемой величины;

T_i – пороги нейронов.

Структура (см. рисунок 2) может быть

описана также совокупностью зависимостей (2 ... 5), что позволяет упростить ее аппаратную реализацию за счет уменьшения количества произведений $w_{i,j}b_j$:

$$b_1[t] = F \left(\begin{pmatrix} w_{1,1}b_1[t-1] + \\ w_{2,1}b_2[t-1] + \\ w_{3,1}b_3[t-1] + \\ w_{4,1}b_4[t-1] \end{pmatrix} \cdot x_{\max} + w_{0,1}x \right), \quad (2)$$

$$b_2[t] = F \left(\begin{pmatrix} w_{1,2}b_1[t-1] + \\ w_{2,2}b_2[t-1] + \\ w_{3,2}b_3[t-1] + \\ w_{4,2}b_4[t-1] \end{pmatrix} \cdot x_{\max} + w_{0,2}x \right), \quad (3)$$

$$b_3[t] = F \left(\begin{pmatrix} w_{1,3}b_1[t-1] + \\ w_{2,3}b_2[t-1] + \\ w_{3,3}b_3[t-1] + \\ w_{4,3}b_4[t-1] \end{pmatrix} \cdot x_{\max} + w_{0,3}x \right), \quad (4)$$

$$b_4[t] = F \left(\begin{pmatrix} w_{1,4}b_1[t-1] + \\ w_{2,4}b_2[t-1] + \\ w_{3,4}b_3[t-1] + \\ w_{4,4}b_4[t-1] \end{pmatrix} \cdot x_{\max} + w_{0,4}x \right). \quad (5)$$

Этап 3. Обучение АЦП на основе сети с обратными связями. Далее показывается процедура настройки сети на примере 4-разрядного АЦП для выполнения операции линейного преобразования. Процедура состоит из четырех стадий (их число определяется количеством разрядов преобразователя).

Стадия 0. Задание пороговых значений нейронов.

Для упрощения нахождения весовых коэффициентов сети пороговые значения заданы пропорционально весам разрядов:

$$t_1 = \frac{1}{16}, \quad t_2 = \frac{1}{8}, \quad t_3 = \frac{1}{4} \quad \text{и} \quad t_4 = \frac{1}{2}.$$

Или в общем виде:

$$t_i = \frac{1}{2^{n-i+1}},$$

где n – количество разрядов преобразователя.

Стадия 1. Нахождение весовых коэффициентов, влияющих на вычисление значения

старшего b_4 разряда результата, осуществляется в результате выполнения 5 шагов.

Шаг 1.1. Нахождение коэффициента $w_{0,4}$.

Для нахождения $w_{0,4}$ следует рассмотреть переход из состояния $b_4b_3b_2b_1[t-1] = 0000$ в состояние $b_4b_3b_2b_1[t] = 1zzz$ ("0000" → "1zzz"), где $z \in \{0; 1\}$. Очевидно, граничным условием перехода будет значение $x = x_{\max}/2$. Исходя из этого на основе выражения (5) можно составить условие:

$$w_{0,4} \cdot \frac{x_{\max}}{2} \geq \frac{x_{\max}}{2},$$

откуда следует, что

$$w_{0,4} \geq 1. \quad (6)$$

Для ограничения значения коэффициента $w_{0,4}$ необходимо рассмотреть переход "0000" → "0zzz", где граничным условием перехода будет значение $x = (x_{\max}/2)_-$, которое можно заменить значением $x = x_{\max}/2 - x_{\max}/2^m$, где $m \rightarrow \infty$. Получится условие:

$$w_{0,4} \cdot \left(\frac{x_{\max}}{2} - \frac{x_{\max}}{2^m} \right) < \frac{x_{\max}}{2},$$

из которого следует:

$$w_{0,4} < 2^m / (2^m - 2). \quad (7)$$

Найдя область пересечения неравенств (6) и (7), получим, что

$$w_{0,4} = 1. \quad (8)$$

Шаг 1.2. Нахождение коэффициента $w_{4,4}$.

Для нахождения $w_{4,4}$ необходимо рассмотреть переход из состояния $b_4b_3b_2b_1[t-1] = 1000$ в состояние $b_4b_3b_2b_1[t] = 1zzz$ ("1000" → "1zzz"), где $z \in \{0; 1\}$. Очевидно, граничным условием перехода будет значение $x = x_{\max}/2$. Исходя из этого на основе выражения (5) с учетом значения (8) получим условие:

$$1 \cdot \frac{x_{\max}}{2} + w_{4,4} \cdot x_{\max} \geq \frac{x_{\max}}{2},$$

откуда следует, что

$$w_{4,4} \geq 0. \quad (9)$$

По аналогии с шагом 1.1 в результате рассмотрения перехода "1000" → "0zzz", где гранич-

ным условием перехода будет значение $x = (x_{\max}/2)_-$, которое можно заменить значением $x = x_{\max}/2 - x_{\max}/2^m$, где $m \rightarrow \infty$, получится условие:

$$w_{4,4} < 1/2^m. \tag{10}$$

Найдя область пересечения неравенств (9) и (10), получим:

$$w_{4,4} = 0. \tag{11}$$

Шаг 1.3. Нахождение коэффициента $w_{3,4}$.

Для нахождения $w_{3,4}$ следует рассмотреть переход из состояния $b_4b_3b_2b_1[t-1] = 0100$ в состояние $b_4b_3b_2b_1[t] = 1zzz$ ("0100" → "1zzz"), где $z \in \{0; 1\}$. Граничным условием перехода будет значение $x = x_{\max}/2$. Исходя из этого на основе выражения (5) с учетом значения (8) можно составить условие:

$$1 \cdot \frac{x_{\max}}{2} + w_{3,4} \cdot x_{\max} \geq \frac{x_{\max}}{2},$$

откуда следует, что

$$w_{3,4} \geq 0. \tag{12}$$

По аналогии с шагами 1.1, 1.2 в результате рассмотрения обратного перехода "0100" → "0zzz" с учетом условия (12) найдено:

$$w_{3,4} = 0. \tag{13}$$

Шаг 1.4. Нахождение коэффициента $w_{2,4}$.

Для нахождения $w_{2,4}$ необходимо рассмотреть переход из состояния $b_4b_3b_2b_1[t-1] = 0010$ в состояние $b_4b_3b_2b_1[t] = 1zzz$ ("0010" → "1zzz"), где $z \in \{0; 1\}$. Граничным условием перехода будет значение $x = x_{\max}/2$. Исходя из этого на основе выражения (5) с учетом значения (8) получим условие:

$$1 \cdot \frac{x_{\max}}{2} + w_{2,4} \cdot x_{\max} \geq \frac{x_{\max}}{2},$$

откуда следует, что

Таблица 1 – Нахождение весовых коэффициентов, влияющих на вычисление значения разряда b_3 результата

Шаг	Переход $b_4b_3b_2b_1[t-1] \rightarrow b_4b_3b_2b_1[t]$	Значение граничного условия перехода	Условие перехода	Решение условия	Значение веса с учетом противоположного перехода
2.1	"0000" → "z1zz"	$x = \frac{x_{\max}}{4}$	$w_{0,3} \cdot \frac{x_{\max}}{4} \geq \frac{x_{\max}}{4}$	$w_{0,3} \geq 1$	$w_{0,3} = 1$

$$w_{2,4} \geq 0. \tag{14}$$

По аналогии с шагами 1.1 – 1.3 в результате рассмотрения обратного перехода "0010" → "0zzz" с учетом условия (14) найдено:

$$w_{2,4} = 0. \tag{15}$$

Шаг 1.5. Нахождение коэффициента $w_{1,4}$.

Для нахождения $w_{1,4}$ рассматривается переход из состояния $b_4b_3b_2b_1[t-1] = 0001$ в состояние $b_4b_3b_2b_1[t] = 1zzz$ ("0001" → "1zzz"), где $z \in \{0; 1\}$. В качестве граничного условия перехода выступает значение $x = x_{\max}/2$. Исходя из этого на основе выражения (5) с учетом значения (8) можно составить условие:

$$1 \cdot \frac{x_{\max}}{2} + w_{1,4} \cdot x_{\max} \geq \frac{x_{\max}}{2},$$

откуда следует, что

$$w_{1,4} \geq 0. \tag{16}$$

По аналогии с шагами 1.1 – 1.3 (1.4) в результате рассмотрения обратного перехода "0001" → "0zzz" с учетом условия (16) найдено:

$$w_{1,4} = 0. \tag{17}$$

В результате 1-го этапа для операции линейного преобразования получено выражение:

$$b_4[t] = F(x), \tag{18}$$

где $F(x) = \begin{cases} 1, & \text{если } x \geq x_{\max}/2, \\ 0, & \text{если } x < x_{\max}/2. \end{cases}$

Стадия 2. Нахождение весовых коэффициентов, влияющих на вычисление значения следующего b_3 за старшим разрядом результата.

Последовательность действий для нахождения весовых коэффициентов, влияющих на вычисление значения следующего b_3 за старшим разрядом результата, аналогична последовательности, рассмотренной на стадии 1. В таблице 1 показаны результаты нахождения весовых коэффициентов, полученных на 2-й стадии.

Продолжение таблицы 1.

2.2	"0100" → "z1zz"	$x = \frac{x_{\max}}{4}$	$1 \cdot \frac{x_{\max}}{4} + w_{3,3} \cdot x_{\max} \geq \frac{x_{\max}}{4}$	$w_{3,3} \geq 0$	$w_{3,3} = 0$
2.3	"1000" → "z1zz"	$x = \frac{12x_{\max}}{16}$	$1 \cdot \frac{12x_{\max}}{16} + w_{4,3} \cdot x_{\max} \geq \frac{x_{\max}}{4}$	$w_{4,3} \geq -\frac{1}{2}$	$w_{4,3} = -\frac{1}{2}$
2.4	"0010" → "z1zz"	$x = \frac{x_{\max}}{4}$	$1 \cdot \frac{x_{\max}}{4} + w_{2,3} \cdot x_{\max} \geq \frac{x_{\max}}{4}$	$w_{2,3} \geq 0$	$w_{2,3} = 0$
2.5	"0001" → "z1zz"	$x = \frac{x_{\max}}{4}$	$1 \cdot \frac{x_{\max}}{4} + w_{1,3} \cdot x_{\max} \geq \frac{x_{\max}}{4}$	$w_{1,3} \geq 0$	$w_{1,3} = 0$

В результате 2-го этапа для операции линейного преобразования получено выражение:

$$b_3[t] = F\left(x - \frac{1}{2} \cdot b_4[t-1] \cdot x_{\max}\right), \quad (19)$$

где $F(x) = \begin{cases} 1, & \text{если } x \geq x_{\max}/4, \\ 0, & \text{если } x < x_{\max}/4. \end{cases}$

Стадия 3. Нахождение весовых коэффициентов, влияющих на вычисление значения разряда b_2 результата.

В таблице 2 показаны результаты нахождения весовых коэффициентов, полученных на 3-м этапе.

В результате 3-го этапа для операции линейного преобразования получено выражение:

$$b_2[t] = F\left(x - \frac{1}{2} \cdot b_4[t-1] \cdot x_{\max} - \frac{1}{4} \cdot b_3[t-1]\right), \quad (20)$$

где $F(x) = \begin{cases} 1, & \text{если } x \geq x_{\max}/8, \\ 0, & \text{если } x < x_{\max}/8. \end{cases}$

Таблица 2 – Нахождение весовых коэффициентов, влияющих на вычисление значения разряда b_2 результата

Шаг	Переход $b_4b_3b_2b_1[t-1] \rightarrow b_4b_3b_2b_1[t]$	Значение граничного условия перехода	Условие перехода	Решение условия	Значение веса с учетом противоположного перехода
3.1	"0000" → "zz1z"	$x = \frac{x_{\max}}{8}$	$w_{0,2} \cdot \frac{x_{\max}}{8} \geq \frac{x_{\max}}{8}$	$w_{0,2} \geq 1$	$w_{0,2} = 1$
3.2	"1000" → "zz1z"	$x = \frac{10x_{\max}}{16}$	$1 \cdot \frac{10x_{\max}}{16} + w_{4,2} \cdot x_{\max} \geq \frac{x_{\max}}{8}$	$w_{4,2} \geq -\frac{1}{2}$	$w_{4,2} = -\frac{1}{2}$
3.3	"0100" → "zz1z"	$x = \frac{6x_{\max}}{16}$	$1 \cdot \frac{6x_{\max}}{16} + w_{3,2} \cdot x_{\max} \geq \frac{x_{\max}}{8}$	$w_{3,2} \geq -\frac{1}{4}$	$w_{3,2} = -\frac{1}{4}$
3.4	"0010" → "zz1z"	$x = \frac{2x_{\max}}{16}$	$1 \cdot \frac{2x_{\max}}{16} + w_{2,2} \cdot x_{\max} \geq \frac{x_{\max}}{8}$	$w_{2,2} \geq 0$	$w_{2,2} = 0$
3.5	"0001" → "zz1z"	$x = \frac{2x_{\max}}{16}$	$1 \cdot \frac{2x_{\max}}{16} + w_{1,2} \cdot x_{\max} \geq \frac{x_{\max}}{8}$	$w_{1,2} \geq 0$	$w_{1,2} = 0$

Стадия 4. Нахождение весовых коэффициентов, влияющих на вычисление значения младшего b_1 разряда результата.

В таблице 3 показаны результаты нахождения весовых коэффициентов, полученных на 4-м этапе.

Таблица 3 – Нахождение весовых коэффициентов, влияющих на вычисление значения разряда b_1 результата

Шаг	Переход $b_4b_3b_2b_1[t-1] \rightarrow b_4b_3b_2b_1[t]$	Значение граничного условия перехода	Условие перехода	Решение условия	Значение веса с учетом противоположного перехода
4.1	"0000" → "zzz1"	$x = \frac{x_{\max}}{16}$	$w_{0,1} \cdot \frac{x_{\max}}{16} \geq \frac{x_{\max}}{16}$	$w_{0,1} \geq 1$	$w_{0,1} = 1$
4.2	"1000" → "zzz1"	$x = \frac{9x_{\max}}{16}$	$1 \cdot \frac{9x_{\max}}{16} + w_{4,1} \cdot x_{\max} \geq \frac{x_{\max}}{16}$	$w_{4,1} \geq -\frac{1}{2}$	$w_{4,1} = -\frac{1}{2}$

Продолжение таблицы 3.

4.3	"1000" → "zzz1"	$x = \frac{5x_{\max}}{16}$	$1 \cdot \frac{5x_{\max}}{16} + w_{3,1} \cdot x_{\max} \geq \frac{x_{\max}}{16}$	$w_{3,1} \geq -\frac{1}{4}$	$w_{3,1} = -\frac{1}{4}$
4.4	"0010" → "zzz1"	$x = \frac{3x_{\max}}{16}$	$1 \cdot \frac{3x_{\max}}{16} + w_{2,1} \cdot x_{\max} \geq \frac{x_{\max}}{16}$	$w_{2,1} \geq -\frac{1}{8}$	$w_{2,1} = -\frac{1}{8}$
4.5	"0001" → "zzz1"	$x = \frac{x_{\max}}{16}$	$1 \cdot \frac{x_{\max}}{16} + w_{1,1} \cdot x_{\max} \geq \frac{x_{\max}}{16}$	$w_{1,1} \geq 0$	$w_{1,1} = 0$

В результате 4-го этапа для операции линейного преобразования получено выражение:

$$b_1[t] = F \left(\begin{array}{l} x - \frac{1}{2} \cdot b_4[t-1] \cdot x_{\max} - \frac{1}{4} \cdot b_3[t-1] - \\ - \frac{1}{8} \cdot b_2[t-1] \end{array} \right), \quad (21)$$

где $F(x) = \begin{cases} 1, & \text{если } x \geq x_{\max}/16, \\ 0, & \text{если } x < x_{\max}/16. \end{cases}$

На основе выражений (18) – (21) получено выражение для расчета весовых коэффициентов n -разрядного преобразователя частотно-временных параметров сигналов в цифровой код следящего типа:

$$w_{0,j} = 1, \quad j = \overline{[1, n]}, \quad (22)$$

$$w_{i,j} = 0, \quad i \leq j, \quad (23)$$

$$w_{i,j} = -\frac{1}{2^{n-i+1}}, \quad i > j. \quad (24)$$

На основе зависимостей (22 ... 24) найдены весовые коэффициенты и произведено моделирование работы 10-разрядного преобразователя частоты в код. На рисунке 3 показан процесс преобразования частоты $f_x = 0,521 \cdot f_{x_{\max}}$ в код $N_y = b_{10}b_9b_8b_7b_6b_5b_4b_3b_2b_1$, когда начальное значение $f_x[0] = 0$.

Заключение. Обосновано применение сетей с обратными связями для построения преобразователей формы представления информации «аналог – код», в частности преобразователей частоты и напряжения в цифровой код. Предложена процедура синтеза АЦП на основе модифицированной сети Хопфилда.

Недостатком рассматриваемого ИНС-преобразователя является наличие перерегулирования на начальных тактах преобразования, однако это компенсируется скоростью преобразования. Так, скорость следящего преобразования по сравнению с преобразователем астатического типа увеличилась в среднем в 5 раз. Во избежание снятия неверного результата преобразования

необходимо выдавать сигнал готовности. Следует отметить, что значение старшего разряда кода определяется на первом такте работы преобразователя (см. рисунок 3), значение второго разряда – на втором и т.д.

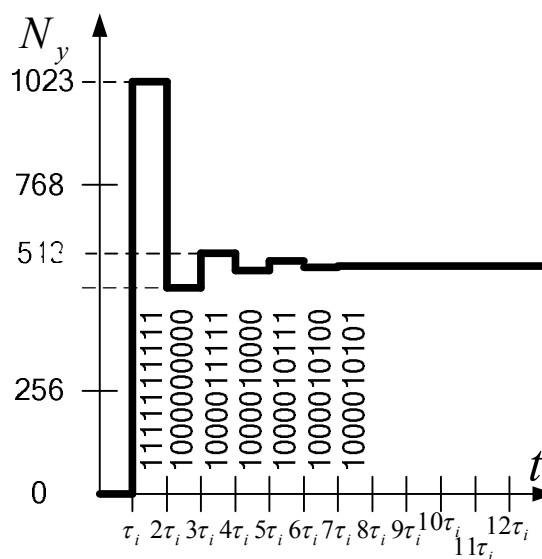


Рисунок 3 – Процесс преобразования частоты в код на основе модифицированной сети Хопфилда

Моделирование подтвердило, что n -разрядный преобразователь выдает значение кода с погрешностью, не превышающей единицу младшего разряда, за интервал времени, не превышающий n тактов. Длительность такта преобразования τ_i лежит в пределах $2 \cdot T_0 \leq \tau_i \leq 2^n \cdot T_0$ [2].

Дальнейшие исследования могут быть направлены на решение задач синтеза структур преобразователей следящего принципа действия с получением результата, начиная с младших разрядов, что позволяет исключить эффект перерегулирования.

Библиографический список

1. Локтюхин В.Н., Челебаев С.В. Нейросетевые преобразователи импульсно-аналоговой информации: организация, синтез, реализация / под общ. ред. А.И. Галушкина. М.: Горячая линия – Телеком, 2008. 144 с.
2. Локтюхин В.Н., Челебаев С.В. Принципы применения технологии искусственных нейронных

сетей для проектирования преобразователей частотно-временных параметров сигналов в код // Датчики и системы. № 2. 2006. С. 43-56.

3. Комарцова Л.Г., Максимов А.В. Нейрокомпьютеры. Изд-во МГТУ им. Н.Э. Баумана, 2002. 320 с.

4. Cohen M.A., Grossberg S.G. Absolute stability of global pattern formation and parallel memory storage by competitive neural networks // IEEE Transactions on Systems, Man and Cybernetics. 1983. № 13. Pp. 815-826.

5. Hopfield J.J. Neural networks and physical systems with emergent collective computational abilities

// Proceedings of the National Academy of Science. 1982. № 79. Pp. 2554-2558.

6. Hopfield J.J., Tank D.W. Neural computation of decisions in optimization problems // Biological Cybernetics. 1985. № 52. Pp. 141-152.

7. Tank D.W., Hopfield J.J. Simple «neural» optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit. Circuits and Systems IEEE Transactions on CAS. 1986. № 33 (5). Pp. 533-541.

УДК 621.391

С.Н. Бузыкканов, А.В. Петрунин

СЕГМЕНТАЦИЯ РЕЧЕВЫХ СИГНАЛОВ НА ОСНОВЕ ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЙ В ВЕСОВОМ ПРОСТРАНСТВЕ СОБОЛЕВА

Предложена модификация существующего в настоящее время алгоритма сегментации речевых сигналов, основанная на разложении по вейвлет-базисам, ортогональным в весовом пространстве Соболева W_2^1 . Показано, что точность сегментации речевых сигналов с помощью предложенных вейвлет-функций удастся повысить с 61% в пространстве L_2 до 76% в весовом пространстве Соболева и при этом понизить процент ложных границ с 61% до 46% соответственно.

Ключевые слова: *весовое пространство Соболева W_2^1 , сегментация речи, быстрое вейвлет-преобразование.*

Введение. Вейвлет-преобразования появились не так давно, но уже представляют серьезную альтернативу аппарату Фурье. Преимуществом вейвлет-преобразования является применение базисов разложения, локализованных не только в частотной, но и во временной области, что важно при обработке нестационарных сигналов.

Обычно вейвлет-преобразование подразумевает разложение сигнала по базисам, принадлежащим пространству L_2 , в котором норма функции определяется выражением:

$$\|f\| = \int_{-\infty}^{+\infty} f^2(t) dt < \infty. \quad (1)$$

Однако, как показали проведенные ранее исследования [1-5], более естественным при обработке речевых сигналов является проведение анализа в весовом пространстве Соболева W_2^1 .

Цель работы. Разработка алгоритмов сегментации речевых сигналов на основе вейвлет-функций, ортогональных в весовом пространстве Соболева W_2^1 .

Теоретические основы. В системах автома-

тического распознавания речи важной задачей является сегментация речи в соответствии с фонетикой языка. Процесс распознавания речи невозможен без точной сегментации, так как она в большей степени определяет надежность системы автоматического распознавания речи [6]. Сегментация речи – это процесс поиска границ между словами, слогами или фонемами – минимальными смысловыми единицами речи [7].

Рассмотрим процесс сегментации речевых сигналов. Если деление производится человеком, то это требует существенных затрат времени и сил, а если обработка проходит на ЭВМ, то это приводит к пропущенным границам и ложно обнаруженным переходам. Несмотря на то, что проблема сегментации стоит уже не один десяток лет, до настоящего времени так и не удалось получить алгоритм, который был бы одновременно устойчив к различным изменениям в речевом тракте, а также обладал стопроцентной точностью определения границ и отсутствием ложных срабатываний.

При разработке методов сегментации речевых сигналов сразу встает вопрос о том, что

принять за речевую единицу. В настоящее время в существующих алгоритмах сегментации за речевую единицу обычно принимается фонема [8]. Использование фонемной сегментации обеспечит минимальный, по сравнению с другими речевыми единицами, размер словаря, если назначение алгоритма – распознавание речи. В русской речи содержится 42 фонемы [7], что значительно меньше числа звуков, которые могут появиться в речевом тракте. Речевой сигнал считается квазистационарным на интервале 10...30 мс, поэтому практически все известные алгоритмы сегментации [8] используют интервал анализа не более 20 мс.

Для сегментации речевых сигналов применим алгоритм, основанный на сравнении вейвлет-коэффициентов на наиболее информативном уровне разложения [8]. Решение о наличии межфонемного перехода принимается по фиксированному порогу изменения энергии коэффициентов детализации:

$$|e_{i+1,j} - e_{i,j}| \geq 3,5, \quad (2)$$

где

$$e_{ij} = 10 \cdot \lg \sum_{k=0}^{n_j-1} d_{i-n_j+k}, \quad (3)$$

d_i - коэффициенты разложения (детализации). В формулах (2) и (3) под i подразумевается номер скользящего окна; j - количество уровней вейвлет-преобразования; $n = n/2^j$ - размер скользящего окна на j -ом уровне разложения; n - размер окна на нулевом уровне; k - счетчик.

Существующие алгоритмы сегментации речевых сигналов работают в пространстве L_2 , которое накладывает ограничение только на энергию сигнала. Однако речевые сигналы принадлежат более узкому классу сигналов – сигналам с ограниченной энергией производной, т.е. они принадлежат весовому пространству Соболева W_2^1 . Поэтому обработка сигналов в весовом пространстве Соболева является для них более естественной.

Под весовым пространством Соболева W_2^1 понимается пространство, в котором норма функции определяется следующим выражением [2...5]:

$$\|f\| = (1-\alpha) \int_{-\infty}^{+\infty} (f(t))^2 dt + \alpha \int_{-\infty}^{+\infty} (f'(t))^2 dt, \quad (4)$$

где α - весовой коэффициент пространства W_2^1 . Весовое пространство Соболева W_2^1 , в свою очередь, является подпространством простран-

ства L_2 . Как показано в [9], при обработке сигналов с помощью функций, ортогональных в пространстве W_2^1 , скорость сходимости будет выше и соответственно число значащих коэффициентов разложения уменьшится.

В ходе проведенных ранее исследований [5] были получены выражения для скейлинг- и вейвлет-функций, ортогональных в весовом пространстве Соболева, которые соответственно имеют следующий вид:

$$\begin{aligned} \widehat{\varphi}_N^{(j)}(w) &= \frac{1}{\sqrt{1-\alpha + i\sqrt{\alpha}2^j w}} \prod_{p=1}^{+\infty} m_{0,N}^{(j+p)}(2^{-p}w) = \\ &= \widetilde{\varphi}_0^{(j)}(w) e^{-iNw/2} \left(\frac{\sin(w/2)}{w/2} \right)^N \prod_{p=1}^{+\infty} L_N^{(j+p)}(2^{-p}w), \quad (5) \end{aligned}$$

$$\begin{aligned} \widehat{\psi}_N^{(j)}(2w) &= -e^{-iw} \overline{m_{0,N}^{(j+1)}(w+\pi)} \widehat{\varphi}_N^{(j+1)}(w) = \\ &= \frac{2^{1+j/2} \sqrt{4\alpha(1-\alpha)} e^{-iw} i^{-N}}{(1-\alpha + \alpha 2^{2(j+1)} w^2)} \left(\frac{\sin^2(w/2)}{w/2} \right)^N \times \\ &\quad \times \frac{ch \left(2^{-j-1} \sqrt{\frac{1-\alpha}{\alpha}} \right) - \cos(w)}{\sqrt{sh \left(2^{-j} \sqrt{\frac{1-\alpha}{\alpha}} \right)}} \times \\ &\quad \times \overline{L_N^{(j+1)}(w+\pi)} \prod_{p=1}^{+\infty} L_N^{(j+p+1)}(2^{-p}w). \quad (6) \end{aligned}$$

Тригонометрический полином степени N $L_N^{(j)}(w) = \sum_{k=0}^N b_{k,N}^{(j)} e^{-ikw}$ определяется таким образом, чтобы корни выражения $\sum_{k=0}^N b_{k,N}^{(j)} z^k$ удовлетворяли условию $|z| \geq 1$ и

$$L_N^{(j)}(0) = \sqrt{P_N^{(j)}(0)} = \sqrt{2-1/x_j}, \quad (7)$$

где $x_j = ch^2 \left(2^{-j-1} \sqrt{\frac{1-\alpha}{\alpha}} \right)$.

Также было показано, что применение полученных вейвлет-функций позволяет повысить качество изображений при сжатии по сравнению с аналогичными алгоритмами в пространстве L_2 . Рассмотрим применение данных вейвлет-функций в алгоритме сегментации речевых сигналов [8].

Экспериментальные исследования. При экспериментальных исследованиях был смоделирован алгоритм сегментации речевых сигна-

лов на основе быстрого вейвлет-преобразования [8] в пакете MATLAB 7.0.4. Данный алгоритм использует при анализе сигналов как вейвлет Хаара в пространстве L_2 , так и функцию-аналог вейвлета Хаара, полученную в [5] для весового пространства Соболева W_2^1 .

Речевая база содержала 10 фраз, записанных в реальной обстановке. В записи участвовали 3 диктора. Запись осуществлялась с частотой дискретизации 22050 Гц и разрядностью 16 бит на отсчет.

Для проведения тестирования работы алгоритмов полученная речевая база была «вручную» просегментирована на фонемы бригадой из 6 человек, и после разбиения границы были усреднены.

Для оценки качества работы алгоритма сегментации речевых сигналов использовался критерий Неймана – Пирсона, применяемый в радиолокации [10], где решаются аналогичные задачи – повышение вероятности обнаружения цели и минимизации количества ложных срабатываний.

По критерию оптимального обнаружения, сумма вероятности ложного срабатывания F и вероятности пропуска границы межфонемного перехода D_0 должна быть минимальна.

$$F + D_0 \rightarrow \min ,$$

в идеальном случае:

$$F + D_0 = 0 .$$

При применении алгоритма, предложенного в [8], и использовании базиса Хаара в пространстве L_2 были получены следующие результаты: 61% правильно найденных границ, 61% найденных ложных межфонемных переходов.

Применяя функции-аналоги вейвлета Хаара, полученные в [5] для весового пространства Соболева W_2^1 , удалось значительно повысить процент правильно найденных границ с 61 до 76 и при этом понизить долю ложных срабатываний с 61 до 46 процентов.

Для того чтобы оценить качество работы алгоритма сегментации, воспользуемся критерием оптимального обнаружения, который учитывает два вида нежелательных срабатываний (пропуск границы и нахождение ложного межфонемного перехода) при обработке вейвлетом Хаара в пространстве L_2 и функцией-аналогом вейвлета Хаара, полученной для весового пространства Соболева W_2^1 .

В пространстве L_2 :

вероятность ложного срабатывания равна 0,61, вероятность пропуска границы, т. е. необнаружение межфонемного перехода: $(1-0,61)=0,39$

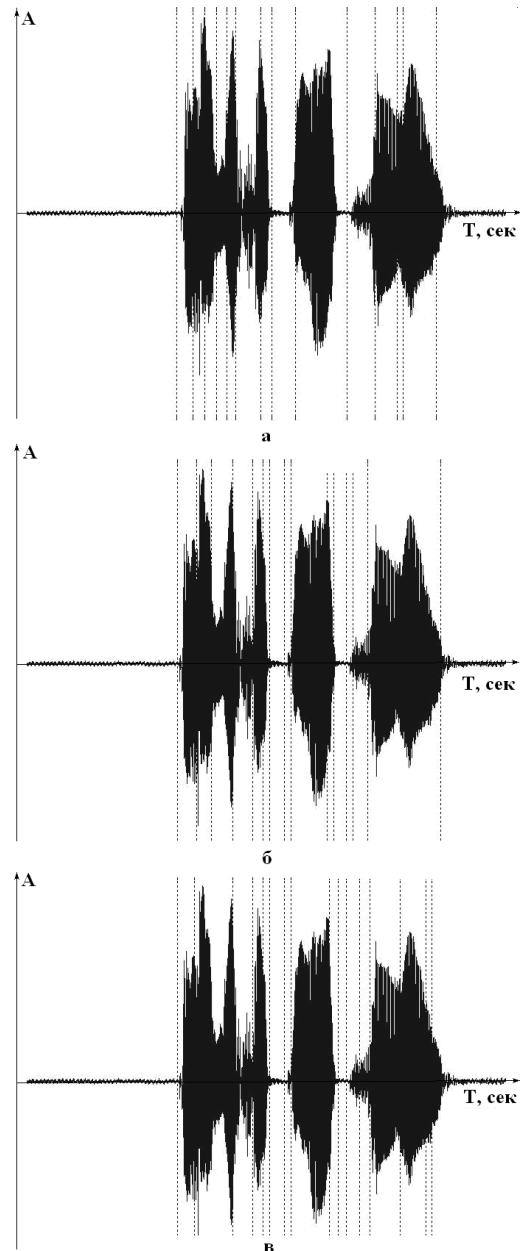
$$0,39 + 0,61 = 1 .$$

В весовом пространстве Соболева W_2^1 :

вероятность ложного срабатывания равна 0,46, вероятность пропуска границы: $(1-0,76)=0,24$

$$0,24 + 0,46 = 0,7 .$$

На рисунке представлены массивы границ межфонемных переходов для слова «квалификация», где a соответствует разбиению ручным способом; b - разбиению с помощью вейвлета Хаара в пространстве L_2 ; $в$ - границам, найденным с помощью функции-аналога вейвлета Хаара в весовом пространстве Соболева W_2^1 .



Массивы границ межфонемных переходов

Таким образом, в ходе проведенных экспериментальных исследований показано, что применение вейвлет-функций, ортогональных в весовом пространстве Соболева, позволяет существенно повысить точность автоматической сегментации речевых сигналов.

Заключение. Опираясь на экспериментальные данные, можно утверждать, что использование математического аппарата, позволяющего осуществлять поиск массива границ межфонемных переходов на основе вейвлет-функций, ортогональных в весовом пространстве Соболева W_2^1 , позволяет повысить точность сегментации речевых сигналов в 1,5 раза.

Библиографический список

1. Соболев С.Л. Некоторые применения функционального анализа в математической физике: монография. Изд. Ленингр. гос. универ., 1950 (1988). 336 с.
2. Кириллов С.Н., Бузыканов С.Н. Двухканальная система передачи и обработки сигналов в пространстве Соболева // Вестник РГРТА. Вып. N.10. 2002. С. 6-8.
3. Кириллов С.Н., Бузыканов С.Н. Оценка спек-

ральной плотности мощности сигналов в модифицированном пространстве Соболева// Радиоэлектроника, 2002. Т. 45. № 12. С. 46-51. (Известия высших учебных заведений).

4. Кириллов С.Н., Бузыканов С.Н. Алгоритм дискретного спектрального анализа сигналов в модифицированном пространстве Соболева// Автотметрия. 2003. № 1. С. 88-94.

5. Бузыканов С.Н. Вейвлет-обработка изображений в весовом пространстве Соболева// Вестник РГРТУ № 4. Вып. № 34. 2010. С. 12-19.

6. Сорокин В.Н., Цыплихин А.И. Сегментация и распознавание гласных// Информационные процессы. Т. 4. № 2. 2004. С. 202-220.

7. Лобанов Б.М., Цирульник Л.И. Компьютерный синтез и клонирование речи. Минск: Белорусская наука, 2008. 316 с.

8. Ермоленко Т., Шевчук В. Алгоритмы сегментации с применением быстрого вейвлет-преобразования// Статьи, принятые к публикации на сайте международной конференции Диалог'2003. www.dialog-21.ru.

9. Бехтин Ю.С. Теоретические основы вейвлет-кодирования зашумленных сигналов. Рязань:РГРТУ, 2009. 121 с.

10. Финкельштейн М.И. Основы радиолокации. М.: Радио и связь, 1983. 530 с.

УДК 629.705 : 621.391.258

П.В. Бабаян, А.Б. Фельдман

ОПРЕДЕЛЕНИЕ МЕСТОПОЛОЖЕНИЯ ОБЪЕКТА НА ИЗОБРАЖЕНИИ В СИСТЕМАХ ТЕХНИЧЕСКОГО ЗРЕНИЯ МОБИЛЬНЫХ РОБОТОТЕХНИЧЕСКИХ КОМПЛЕКСОВ

Рассмотрен подход к определению координат объекта на изображении для систем технического зрения мобильных робототехнических комплексов, особенностью которого является учет межкадровых геометрических деформаций, динамики поступающих изображений, присутствия пространственно неоднородного фона и посторонних движущихся объектов. Представлены результаты экспериментальных исследований, подтверждающие высокую эффективность предложенного подхода.

Ключевые слова: видеопоследовательность, измерение координат, геометрические деформации, выделение движения.

Введение. Развитие различных приложений интеллектуальных робототехнических комплексов, особенно мобильных, невозможно без совершенствования теории, методов и алгоритмов технического зрения. Интеллектуальные системы управления и робототехнические комплексы, снабженные системой технического зрения (СТЗ), широко и успешно применяются во многих отраслях промышленности на таких операциях, как контроль, отбраковка, учет и

складирование деталей, сварка, покраска, монтаж элементов на поверхность печатных плат. Незаменимы СТЗ в таких задачах, как охрана и безопасность, ориентирование и навигация мобильных аппаратов, аэрокосмическая разведка, управление высокоточным вооружением.

Функциональные задачи, решаемые СТЗ, в зависимости от конкретного приложения весьма разнообразны, их спектр колеблется от регистрации присутствия какого-либо объекта или

возникновения какой-либо ситуации до детального семантического описания структуры наблюдаемой сцены. Тем не менее, нетрудно установить, что базой для решения большинства задач технического зрения являются обнаружение и оценка параметров объектов, находящихся или вновь появляющихся в поле зрения датчика изображений.

Несмотря на важную роль указанных задач для приложений технического зрения, на данный момент трудно указать универсальный подход, позволяющий однозначно регистрировать присутствие произвольных объектов и осуществлять длительное помехоустойчивое слежение за ними в последовательности поступающих изображений. Это во многом объясняется тем, что задачи технического зрения плохо поддаются формализации и в большинстве приложений имеется высокая степень априорной неопределенности относительно условий наблюдения, характера фоновой составляющей, особенностей наблюдаемых объектов вкпе с параметрами движения носителя СТЗ и изменения ориентации датчика изображений. Последнее особенно характерно для систем технического зрения мобильных роботов.

В самом деле, ввиду перемещения носителя и изменения ориентации датчика изображений невозможно заранее предсказать особенности наблюдаемой сцены, которая к тому же имеет динамичный характер. При этом в наблюдаемой видеопоследовательности возникают геометрические деформации, учесть которые с использованием телеметрии, полученной от датчиков ориентации устройства формирования изображений, не всегда представляется возможным. Помимо этого, ситуация осложняется тем, что сами объекты интереса могут двигаться относительно неподвижного фона, причем характер их движения зачастую нельзя установить априори. Игрют роль также и другие мешающие обстоятельства, к которым можно отнести, например, неоднородность фоновой составляющей, пересечение траекторий и заслонение объектов, а также наличие посторонних объектов, визуально схожих с объектами интереса.

В настоящей работе предлагается подход к определению координат объекта на изображении текущего кадра видеопоследовательности для систем технического зрения мобильных робототехнических комплексов, существенными особенностями которого являются учет динамики поступающих изображений, использование предыстории движения объекта, а также обнаружение и удаление из рассмотрения посторонних движущихся объектов. На основе обработки

большого числа натуральных видеосюжетов показывается, что измерение координат объекта в текущем кадре на основе критерия, учитывающего характер фоновой составляющей, позволяет значительно увеличить длительность непрерывного слежения за объектом по сравнению с распространенным на практике разностным критерием сходства изображений.

Постановка задачи. Пусть некоторая сцена наблюдается датчиком изображений, положение и ориентация которого могут изменяться во времени. Будем считать, что сцена содержит объект интереса, изображение которого в n -м кадре описывается функцией $h(u, v, n)$, где целочисленные пространственные координаты (u, v) заданы на множестве H_n , связанном с центром объекта. Пространственно неоднородный фон будем описывать функцией $g(k, m, n)$, носителем которой в текущем кадре является множество пар целых чисел G_n , связанное с некоторой фиксированной точкой фона. Состояние яркости объекта и фона будем описывать моделями независимо протекающих в каждой точке процессов вида [1–3]:

$$\begin{aligned} h(u, v, n) &= h(u, v, n-1) + \omega_h(u, v, n), \\ g(k, m, n) &= g(k, m, n-1) + \omega_g(k, m, n), \end{aligned} \quad (1)$$

где $\omega_h(u, v, n) \sim N(0, \sigma_h^2)$ и $\omega_g(k, m, n) \sim N(0, \sigma_g^2)$ – некоррелированные по пространству и времени гауссовские случайные величины, представляющие собой формирующий шум.

Зафиксируем неподвижную систему координат, связав ее, например, с первым кадром видеопоследовательности. Пусть положение центра объекта в этой системе описывается вектором координат $\lambda(n) = [\lambda_x(n), \lambda_y(n)]^T$, а положение точки, принимаемой за «центр» фона, – соответственно вектором $\mu(n) = [\mu_x(n), \mu_y(n)]^T$.

Ввиду перемещения носителя СТЗ и изменения ориентации датчика изображений в видеопоследовательности возникают геометрические деформации. Если наблюдается достаточно удаленная сцена и поле зрения датчика изображений невелико, то возникающие деформации можно описывать аффинной моделью [4, 5]:

$$\begin{aligned} T(x, y, n) &= \\ &= \gamma(n) \begin{bmatrix} \cos \varphi(n) & -\sin \varphi(n) \\ \sin \varphi(n) & \cos \varphi(n) \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \psi_x(n) \\ \psi_y(n) \end{bmatrix}, \end{aligned} \quad (2)$$

где $\gamma(n)$ – масштабный множитель, $\varphi(n)$ – угол поворота, $\psi_x(n), \psi_y(n)$ – смещения по соответствующим координатам изображения теку-

шего n -го кадра относительно первого. Будем считать, что геометрические преобразования изображений объекта $T_h(x, y, n)$ и фона $T_g(x, y, n)$ отличаются только смещениями, которые для изображения объекта имеют значения $\psi_x(n) = \lambda_x(n)$, $\psi_y(n) = \lambda_y(n)$, а для изображения фона $\psi_x(n) = \mu_x(n)$, $\psi_y(n) = \mu_y(n)$.

При заданных моделях движения путем оптимальной фильтрации могут быть получены прогнозируемые плотности вероятности векторов координат центров изображений объекта и фона [1–3], которые будем считать гауссовскими с прогнозируемыми значениями $\bar{\lambda}(n)$, $\bar{\mu}(n)$ и прогнозируемыми ковариационными матрицами $\bar{\mathbf{K}}_\lambda(n)$, $\bar{\mathbf{K}}_\mu(n)$. Построить модели изменения параметров $\gamma(n)$ и $\varphi(n)$ несколько сложнее, поэтому прогнозируемые распределения этих параметров также будем считать гауссовскими с прогнозируемыми значениями $\bar{\gamma}(n) = \gamma(n-1)$, $\bar{\varphi}(n) = \varphi(n-1)$ и прогнозируемыми дисперсиями $\bar{\sigma}_\gamma^2(n)$ и $\bar{\sigma}_\varphi^2(n)$, которые могут быть найдены, например, усреднением по выборке значений соответствующих параметров на предыдущих кадрах.

Будем также считать, что, помимо объекта интереса, в видеопоследовательности присутствуют похожие на него посторонние движущиеся объекты, форма и яркость изображений которых априори неизвестны.

Цель работы. В настоящей работе ставится задача на основании прогнозируемых значений параметров геометрических деформаций определять местоположение объекта в текущем кадре, т.е. «измерять» вектор $\lambda(n) = [\lambda_x(n), \lambda_y(n)]^T$.

Решение задачи. Прежде всего, решим задачу измерения координат объекта интереса при условии, что посторонние объекты отсутствуют.

Модель формирования наблюдаемого изображения $l(i, j, n)$, $(i, j) \in L_n$, предполагая, что объект непрозрачен и закрывает некоторый участок фона, примем в виде:

$$l(i, j, n) = r(T_h(i, j, n))h(T_h(i, j, n), n) + (1 - r(T_h(i, j, n)))g(T_g(i, j, n), n) + \xi(i, j, n), \quad (3)$$

где $r(x, y)$, $(x, y) \in R^2$ – информационный параметр, равный 1 для точек объекта и 0 в противном случае, который будем полагать неизменным на интервале наблюдения; $\xi(i, j, n)$ – пространственно-временной гауссовский белый шум; значения яркостей $h(T_h(i, j, n), n)$ и

$g(T_g(i, j, n), n)$ определяются посредством интерполяции по ближайшему значению $h(u, v, n)$ и $g(k, m, n)$ соответственно, при этом пренебрегаем ошибками дискретизации.

Так как наблюдению доступны только зашумленные изображения, необходимо предпринимать специальные меры для получения оценок яркостей $h(u, v, n)$ и $g(k, m, n)$. Как показано в [1–3], на основании моделей состояния (1) могут быть синтезированы процедуры оптимальной временной фильтрации изображений в каждой точке пространственной системы координат. При этом эмпирически установлено, что уже после первых 20–25 итераций на выходе фильтров образуются достаточно «хорошие» оценки изображений объекта $\bar{h}(u, v, n)$ и фона $\bar{g}(k, m, n)$, которые назовем прогнозируемыми изображениями. При этом ошибками оценивания яркостей допустимо пренебречь, полагая изображения неслучайными в задаче измерения координат объекта в текущем кадре. Фактически это можно объяснить тем, что изображения меняются медленно по сравнению с частотой поступления кадров.

Введем в рассмотрение сигнальную функцию

$$\begin{aligned} \bar{s}(i, j, n, \gamma(n), \varphi(n), \lambda(n), \mu(n)) = \\ = r(T_h(i, j, n))\bar{h}(T_h(i, j, n), n) + \\ + (1 - r(T_h(i, j, n)))\bar{g}(T_g(i, j, n), n), \end{aligned} \quad (4)$$

составленную из прогнозируемых изображений $\bar{h}(u, v, n)$ и $\bar{g}(k, m, n)$ с учетом информационного параметра $r(i, j)$ и зависящую от параметров геометрических преобразований. В таком случае модель наблюдения можно более компактно представить в виде:

$$l(i, j, n) = \bar{s}(i, j, n, \gamma(n), \varphi(n), \lambda(n), \mu(n)) + \xi(i, j, n). \quad (5)$$

С учетом прогнозируемых распределений параметров геометрических деформаций для объекта и фона, в предположении, что параметры статистически независимы, апостериорная плотность вероятности координат объекта может быть записана в виде:

$$\begin{aligned} p(\lambda(n) | \mathbf{L}(n), \gamma(n), \varphi(n), \mu(n)) = \\ = \frac{1}{Z_n} P(\mathbf{L}(n) | \gamma(n), \varphi(n), \lambda(n), \mu(n)) \times \\ \times \bar{p}(\gamma(n))\bar{p}(\varphi(n))\bar{p}(\lambda(n))\bar{p}(\mu(n)), \end{aligned} \quad (6)$$

где

$$P(\mathbf{L}(n) | \gamma(n), \varphi(n), \lambda(n), \mu(n)) = \exp \left\{ -\frac{1}{2} \sum_{(i,j) \in L_n} \left(l(i,j,n) - \bar{s}(i,j,n, \gamma(n), \varphi(n), \lambda(n), \mu(n)) \right)^2 \right\} \quad (7)$$

$$= \frac{(2\pi\sigma_\xi^2)^{N/2}}{\dots}$$

представляет собой функцию правдоподобия наблюдаемого изображения; $\bar{p}(\gamma(n))$, $\bar{p}(\varphi(n))$, $\bar{p}(\lambda(n))$, $\bar{p}(\mu(n))$ – прогнозируемые плотности распределений соответствующих параметров; Z_n – константа, определяемая условиями нормировки.

Следует отметить, что сигнальная функция зависит от параметров нелинейно, что порождает негауссовость апостериорной плотности, поэтому применение байесовского критерия для нахождения координат объекта нецелесообразно. Так, например, использование байесовского критерия с квадратичной функцией потерь приводит, как известно, к вычислению апостериорного среднего, что предполагает усреднение вектора параметров по всему диапазону изменения последних. Ввиду большой размерности задачи это практически невозможно выполнить в реальном масштабе времени. В связи с этим будем использовать критерий максимума апостериорной плотности. Тогда после логарифмирования приходим к необходимости минимизации критерия

$$J_n^{(1)} = (\lambda(n) - \bar{\lambda}(n))^T \bar{\mathbf{K}}_\lambda(n) (\lambda(n) - \bar{\lambda}(n)) + (\mu(n) - \bar{\mu}(n))^T \bar{\mathbf{K}}_\mu(n) (\mu(n) - \bar{\mu}(n)) + \frac{1}{\sigma_\gamma^2(n)} (\gamma(n) - \bar{\gamma}(n))^2 + \frac{1}{\sigma_\varphi^2(n)} (\varphi(n) - \bar{\varphi}(n))^2 + \frac{1}{\sigma_\xi^2} \sum_{(i,j) \in L_n} \left(l(i,j,n) - \bar{s}(i,j,n, \gamma(n), \varphi(n), \lambda(n), \mu(n)) \right)^2, \quad (8)$$

в котором, как нетрудно заметить, последнее слагаемое представляет собой нормированную относительно дисперсии помехи меру сходства наблюдаемого изображения и сигнальной функции.

Вычисление критерия (8) также сопряжено со значительными трудностями. Поэтому обратимся к предположению, которое заключается в том, что объект занимает достаточно малую по сравнению со всем кадром площадь, следовательно, пренебрегая наличием объекта, можно оценить параметры геометрических преобразований фона $\gamma(n)$, $\varphi(n)$, $\mu_x(n)$, $\mu_y(n)$. Следует заметить, что предложено достаточно много подходов, позволяющих оценивать параметры аффинного преобразования, некоторые из которых ориентированы на работу в реальном масштабе времени. В частности, к разряду вычислительно эффективных можно отнести подход, основанный на известных свойствах Фурье-преобразования изображений [6, 7], а

также многоэталонный корреляционный алгоритм, описанный в [8].

После того как геометрические искажения скомпенсированы, остается определить положение объекта в кадре, для чего, пренебрегая ошибками компенсации, необходимо минимизировать критерий

$$J_n^{(2)} = (\lambda(n) - \bar{\lambda}(n))^T \bar{\mathbf{K}}_\lambda(n) (\lambda(n) - \bar{\lambda}(n)) + \frac{1}{\sigma_\xi^2} \sum_{(i,j) \in L_n} \left(l(i,j,n) - \bar{s}^*(i,j,n, \lambda(n)) \right)^2, \quad (9)$$

где $\bar{s}^*(i,j,n, \lambda(n))$ – сигнальная функция, преобразованная к системе координат текущего кадра, которая зависит теперь только от координат объекта и не зависит от параметров геометрических преобразований фона.

Квадратичная форма на месте первого слагаемого (9) соответствует прогнозируемому распределению координат объекта. Вообще говоря, весьма непросто получить оценку прогнозируемой ковариационной матрицы $\bar{\mathbf{K}}_\lambda(n)$, для вычисления которой необходимо знать априорные распределения и ковариационную матрицу ошибок оценивания координат объекта. Выражения для вычисления оценок диагональных элементов $\bar{\mathbf{K}}_\lambda(n)$ представлены, например, в работе [9], где использовалось предположение о нулевом фоне и отсутствии корреляции между ошибками измерения разноименных координат объекта, что далеко не всегда выполняется в реальных ситуациях.

В принципе можно поступить и другим образом, отказавшись от явного учета первого слагаемого в формуле (9), ограничить при этом зону поиска областью $W_n(\bar{\lambda}_x, \bar{\lambda}_y) \subset L_n$ с центром в точке $(\bar{\lambda}_x, \bar{\lambda}_y)$, соответствующей прогнозируемому положению объекта. В таком случае приходим к необходимости минимизации критерия

$$J_n^{(3)} = \sum_{(i,j) \in W_n} \left(l(i,j,n) - \bar{s}^*(i,j,n, \lambda(n)) \right)^2 \quad (10)$$

по векторному параметру $\lambda(n) = [\lambda_x(n), \lambda_y(n)]^T$.

Критерий (10) удобно преобразовать следующим образом:

$$J_n^{(4)} = \sum_{(i,j) \in W_n} \left(l(i,j,n) - \bar{g}^*(i,j,n) \right)^2 + \sum_{(u,v) \in H_n^*} \left(l(\lambda_x + u, \lambda_y + v, n) - \bar{h}^*(u,v,n) \right)^2 - \sum_{(u,v) \in H_n^*} \left(l(\lambda_x + u, \lambda_y + v, n) - \bar{g}^*(\lambda_x + u, \lambda_y + v, n) \right)^2, \quad (11)$$

где H_n^* – множество точек, получаемое из H_n путем поворота на угол $\varphi(n)$ и масштабирования в $\gamma(n)$ раз, с центром в начале координат. Первое слагаемое не зависит от положения объекта и, следовательно, может быть опущено. Поэтому для измерения координат объекта в текущем кадре достаточно минимизировать по параметрам λ_x, λ_y эквивалентный критерий

$$J_n^{(5)}(\lambda_x, \lambda_y) = \sum_{(u,v) \in H_n^*} \left(l(\lambda_x + u, \lambda_y + v, n) - \bar{h}^*(u, v, n) \right)^2 - \sum_{(u,v) \in H_n^*} \left(l(\lambda_x + u, \lambda_y + v, n) - \bar{g}^*(\lambda_x + u, \lambda_y + v, n) \right)^2. \quad (12)$$

При этом измерения координат могут быть найдены по правилу

$$(\tilde{\lambda}_x(n), \tilde{\lambda}_y(n)) = \underset{(\lambda_x, \lambda_y) \in W_n(\tilde{\lambda}_x, \tilde{\lambda}_y)}{\operatorname{argmin}} J_n^{(5)}(\lambda_x, \lambda_y). \quad (13)$$

Интересно сравнить (12) с распространенным в практических приложениях критерием сходства изображений [4, 9]:

$$F_n = \sum_{(u,v) \in H_n^*} \left| l(\lambda_x + u, \lambda_y + v, n) - \bar{h}^*(u, v, n) \right|. \quad (14)$$

Понятно, что в выражении (12) для участков фона, похожих на объект, второе слагаемое будет близко к нулю, поэтому результирующее значение критерия $J_n^{(5)}(\lambda_x, \lambda_y)$ окажется выше, чем для интересующего объекта. Видно, что при использовании (14) ничего подобного не происходит. Таким образом, применение критерия (12) помогает получить корректные измерения координат объекта интереса в присутствии похожих на него участков фонового изображения.

Тем не менее, в силу эффектов пространственной дискретизации, чересстрочной развертки, наличия неучтенного шума и ряда других причин измерение координат по правилу (13) может сопровождаться ошибкой. Если задана модель движения, то для уточнения местоположения объекта в текущем кадре, оценивания компонентов вектора скорости и прогнозирования дальнейших наблюдений могут быть использованы методы калмановской фильтрации [1–3].

Как отмечалось, для учета динамики яркостей изображений на основе моделей (1) может быть получен алгоритм оптимальной фильтрации калмановского типа. Заметим, что вычисление оценок дисперсий формирующих шумов ω_h и ω_g , необходимых для расчета оптимальных

коэффициентов усиления, не всегда может быть реализовано с достаточной точностью. Поэтому целесообразно вместо калмановской фильтрации ограничиться применением экспоненциальных фильтров [7]. Одновременно с оценкой яркостей изображений разумно также с помощью рекурсивного алгоритма находить оценки $\hat{\sigma}_\xi^2(i, j, n)$ дисперсии аддитивного шума ξ в каждой точке пространственной системы координат.

Модернизированный алгоритм измерения координат. Обратимся теперь к проблеме, которая заключается в том, что, помимо объекта интереса, в видеопоследовательности могут присутствовать посторонние похожие на него движущиеся относительно фона объекты. Как можно видеть, этот факт никак не учтен в критерии (12). Следовательно, наличие посторонних объектов может приводить к грубым ошибкам измерения координат объекта интереса.

Чтобы обнаружить присутствие посторонних объектов, можно воспользоваться методами, основанными на выделении изменений в видеопоследовательностях [7]. Заметим, что это не приведет к существенному увеличению вычислительных затрат, так как к этому моменту уже получены оценки фоновой составляющей $\bar{g}^*(i, j, n)$ и дисперсии аддитивного шума $\hat{\sigma}_\xi^2(i, j, n)$. При этом целесообразно воспользоваться алгоритмом, описанным в работе [10], в котором с помощью аппарата несимметричных распределений Джонсона учитываются случайные геометрические деформации, вызванные турбулентностью атмосферы, и ошибки совмещения изображений смежных кадров.

В результате применения алгоритма выделения изменений будет получено бинарное изображение $b(i, j, n)$, в котором с единичными пикселями будут сопоставлены участки текущего кадра, соответствующие движущимся объектам, среди которых может оказаться также объект интереса. Сегмент, соответствующий интересующему объекту, нетрудно исключить из изображения $b(i, j, n)$ с помощью простых эвристических процедур. В таком случае критериальную функцию в задаче измерения координат можно модернизировать, например, следующим образом:

$$J_n^{(6)}(\lambda_x, \lambda_y) = \sum_{(u,v) \in H_n^*} \frac{\left(l(\lambda_x + u, \lambda_y + v, n) - \bar{h}^*(u, v, n) \right)^2}{\hat{\sigma}_\xi^2(\lambda_x + u, \lambda_y + v, n)}$$

$$- \sum_{(u,v) \in H_n^*} \frac{\left(l(\lambda_x + u, \lambda_y + v, n) - \bar{f}^*(\lambda_x + u, \lambda_y + v, n) \right)^2}{\hat{\sigma}_\xi^2(\lambda_x + u, \lambda_y + v, n)}, \quad (15)$$

где

$$\bar{f}^*(i, j, n) = \tilde{b}(i, j, n)l(i, j, n) + (1 - \tilde{b}(i, j, n))\bar{g}^*(i, j, n), \quad (16)$$

$\tilde{b}(i, j, n)$ – результат выделения изменений, из которого исключены точки, принадлежащие

объекту интереса.

Экспериментальные исследования. Наибольший интерес для исследований представляет сравнение способов измерения координат объекта, основанных на использовании критериев (14) и (15). При этом, чтобы подчеркнуть различие в качестве работы алгоритмов, связанных с указанными критериями, было решено отказаться от возможностей траекторной фильтрации измерений. Вместо этого использовалась упрощенная процедура, согласно которой $\bar{\lambda}(n) = \tilde{\lambda}(n-1)$.

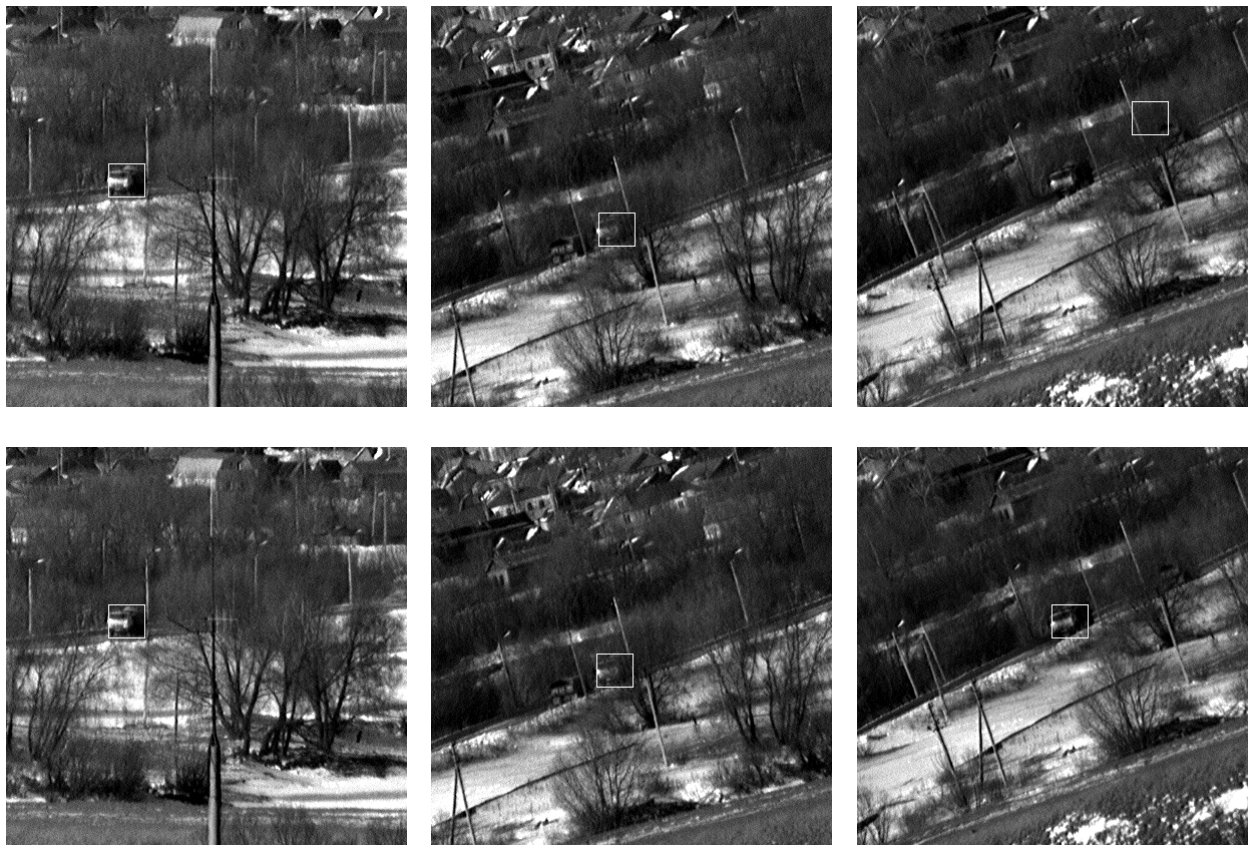


Иллюстрация работы алгоритмов слежения: верхний ряд соответствует алгоритму, использующему критерий (14), нижний ряд – алгоритму, использующему критерий (15)

В первой серии экспериментов ставилась задача сравнения способов измерения координат в условиях геометрических искажений видеопоследовательностей, полученных с помощью датчика, ориентация которого может меняться, что характерно для систем технического зрения мобильных робототехнических комплексов. На рисунке представлены результаты работы алгоритмов слежения за объектом. Верхний ряд изображений демонстрирует ситуацию, когда из-за наличия неоднородного фона и постороннего движущегося объекта происходит потеря объекта интереса алгоритмом слежения, основанным на критерии сходства (14). Алгоритм, использующий критерий (15), при тех же условиях

продолжает непрерывное слежение за объектом. Результаты экспериментов, проведенных на 10 видеосюжетах с аналогичными характеристиками, показывают, что применение критерия (15) более оправданно, так как с помощью последнего можно получить большую длительность непрерывного слежения по сравнению с разностным критерием (14).

Вторая серия экспериментов была посвящена количественному сравнению алгоритмов слежения за объектами, использующих критерии (14) и (15). При проведении исследований использовался тестовый набор, состоящий из 58 видеосюжетов видимого и ИК диапазонов, для которых человеком-экспертом были предвари-

тельно построены «эталонные» траектории объектов интереса. Фон представлял собой пересеченный городской ландшафт. Размер объектов интереса на разных сюжетах варьировался от 5x5 до 40x40 элементов разложения. На ряде сюжетов вблизи объекта интереса наблюдались похожие на него посторонние движущиеся объекты. В ходе исследований анализировалось число кадров до срыва слежения N_i , где

$i = 1$ для критерия (14) и $i = 2$ для критерия (15). Для непосредственного сравнения алгоритмов использовалась величина $\Delta N = N_2 - N_1$. Экспериментальные исследования, результаты которых представлены в таблице, подтверждают высокую эффективность подхода к измерению координат объекта, основанного на использовании оценки фона и выделении посторонних движущихся объектов.

Результаты сравнительных исследований

	$\Delta N < -25$	$-25 \leq \Delta N < -5$	$-5 \leq \Delta N \leq 5$	$5 < \Delta N \leq 25$	$\Delta N > 25$
Число сюжетов	2	4	15	1	36

Заключение. В рамках проведенных исследований был разработан подход к измерению координат объекта для систем технического зрения мобильных робототехнических комплексов, работоспособный в условиях геометрических преобразований видеопоследовательностей, при наличии неоднородного по пространству фона и посторонних движущихся объектов. Результаты экспериментов, выполненных с использованием натуральных видеосюжетов, подтверждают высокую эффективность предложенного подхода.

Исследования проводились при поддержке гранта Президента РФ для молодых ученых (МК-20.2011.10).

Библиографический список

1. Аллатов Б.А. Алгоритм оценивания местоположения изменяющего яркость объекта в последовательности изображений // Изв. вузов. Сер. Приборостроение. – 1991. – № 7. – С. 76 - 81.
2. Аллатов Б.А. Оценивание параметров движущегося объекта в последовательности изменяющихся двумерных изображений // Автометрия. – 1991. – № 3. – С. 21-24.
3. Аллатов Б.А. Оптимальное оценивание параметров движущегося объекта в последовательности изображений // Автометрия. – 1994. – № 2. – С. 32 - 37.

4. Андросов В.А., Бойко Ю.В., Бочкарев А.М., Однорог А. П. Совмещение изображений в условиях неопределенности // Зарубежная радиоэлектроника. – 1985. – № 4. – С. 54 - 70.

5. Астапов Ю.М., Васильев Д.В., Заложнев Ю.И. Теория оптико-электронных следящих систем. – М.: Наука. Гл. ред. физ.-мат. лит., 1988. – 328 с.

6. Reddy B.S., Chatterji B.N. An FFT-based Technique for Translation, Rotation and Scale-invariant Image Registration. – IEEE Transaction on Image Processing. – 1996. – vol.5., № 8. – PP. 1266 – 1271.

7. Аллатов Б.А., Бабаян П.В., Балашов О.Е., Степанов А.И. Методы автоматического обнаружения и сопровождения объектов. Обработка изображений и управление. – М.: Радиотехника, 2008. – 176 с.: ил.

8. Стротов В.В. Выбор опорных участков в многоэталонном алгоритме определения параметров геометрических преобразований изображений // Вестник Рязанского государственного радиотехнического университета. – Рязань, 2009. – № 2 (28). – С. 93 - 96.

9. Аллатов Б.А., Бабаян П.В., Корепанов С.Е. Корреляционный алгоритм слежения за движущимися объектами при наличии близкорасположенных похожих объектов // Вестник Рязанского государственного радиотехнического университета. – Рязань, 2010. – № 2 (32). – С. 3 - 8.

10. Бабаян П.В. Выделение объектов в условиях случайных геометрических преобразований изображений // Автометрия. – 2010. – № 3. – С. 41 - 47.

УДК 004.056.52/53:004.72

В.А. Гончаров

РАЗРАБОТКА МЕТОДОВ ОБНАРУЖЕНИЯ СЕТЕВЫХ АТАК В ВЫСОКОСКОРОСТНЫХ ВЫЧИСЛИТЕЛЬНЫХ СЕТЯХ

Рассмотрена проблема сигнатурного поиска сетевых атак в вычислительных сетях, работающих со скоростями передачи информации выше 1 Гбит/с. Разработан специализированный язык для высокопроизводительного поиска сигнатур. Создан конвертор для сигнатур COA Snort.

Выполнена оптимизация метода поиска преобразованных сигнатур. Достигнуты результаты, позволяющие применять разработанные методы сигнатурного поиска сетевых атак в вычислительных сетях, использующих каналы передачи информации со скоростями 10 Гбит/с.

Ключевые слова: сетевая атака, система обнаружения атак, сигнатурный поиск, конвертор сигнатур, оптимизация поиска.

Введение. Одним из факторов, воздействующих на защищаемую информацию в информационной системе (ИС) на базе вычислительной сети (ВС) и способных приводить к нарушению ее безопасности, является сетевая атака (СА), реализуемая как внешним, так и внутренним нарушителем-злоумышленником [1]. За последние десять лет СА являются составной частью наиболее часто реализуемых и наиболее опасных угроз безопасности информации, обрабатываемой в защищаемых ИС на базе ВС, имеющей подключение к сетям связи общего пользования и (или) сетям международного информационного обмена [2]. Поэтому защита от негативных последствий СА является одной из актуальных проблем обеспечения безопасности информации, обрабатываемой в таких ИС.

СА является целенаправленным несанкционированным воздействием на ресурсы защищаемой ИС или получением несанкционированного доступа к ним с применением программных или программно-аппаратных средств и протоколов межсетевое взаимодействие [1]. Для обнаружения СА используются программно-аппаратные средства защиты информации, называемые сетевыми системами обнаружения атак, или сетевыми СОА. В настоящее время разработано достаточно большое количество различных сетевых СОА, однако при использовании в ВС, работающих со скоростями передачи данных более 1 Гбит/с, для сетевых СОА требуются специализированные аппаратные решения [3]. Использование программных средств сетевых СОА на стандартных аппаратных платформах приводит к фактическому пропуску СА [4].

Для сетевых СОА чаще всего используется сигнатурный метод обнаружения атак, в котором сигнатура представляет собой простое или составное условие, применяемое к анализируемым событиям. Каждое условие соответствует одной обнаруживаемой СА. Выполнение условия означает обнаружение СА.

На практике, для снижения вычислительных затрат, в сетевых СОА используется упрощенный поиск СА, включающий в себя неполный анализ протокола и простой поиск набора подстрок в необработанных данных. Такой упрощенный поиск СА приводит к увеличению количества ошибок первого и второго рода [5].

Эффективность работы сетевой СОА, использующей сигнатурный метод обнаружения атак, зависит от используемого набора сигнатур, который необходимо создавать и постоянно актуализировать. Для свободно распространяемых сетевых СОА стало общепринятым использование набора сигнатур СОА Snort [6]. Таким образом, альтернативой созданию и актуализации собственного набора сигнатур СА является использование набора сигнатур СОА Snort. Для исследовательских СОА, использующих сигнатурный метод, это единственная возможность для обеспечения их практического использования.

Цель работы. Целью работы является разработка методов сигнатурного обнаружения СА для использования в сетевых СОА, работающих в ВС со скоростями передачи информации более 1 Гбит/с, с возможностью использования набора сигнатур СОА Snort.

Язык описания сигнатур. Объектом анализа сетевых СОА является сетевой трафик – совокупность передаваемых по ВС пакетов определенного протокола, каждый из которых содержит служебные заголовки, соответствующие используемому протоколу, и инкапсулированные (вложенные) в пакет данные протокола следующего уровня [7]. Для извлечения инкапсулированных данных сетевыми СОА используется процедура декапсуляции. В случае, если инкапсулированные данные представляют собой передаваемые данные пользователей ВС либо протокол не содержит инкапсулированных данных, считается, что выполнена полная декапсуляция протокола. После выполнения процедуры декапсуляции для служебных заголовков выполняется процедура декодирования, позволяющая представить декодированный заголовок в виде пар "аргумент-значение" (AV-пар). В результате выполнения процедур декапсуляции и декодирования формируется событие для анализа сетевой СОА.

Для решения задачи быстрого сигнатурного поиска СА был разработан специализированный язык описания сигнатур, в котором применен подход, аналогичный используемому в BPF-фильтрах [8]: вместо описания множества условий реализуется программа, названная QF-программой, вызываемая один раз для каждого обрабатываемого события. Программа

состоит из набора функций, названных QF-функциями, которые последовательно выполняются с помощью интерпретатора. Опишем структуру QF-программы. Множества:

$\mathbf{T} = \mathbf{D} \cup \mathbf{A} \cup \mathbf{S}$ – множество значений параметров, где \mathbf{D} – множество целых чисел, \mathbf{A} – множество адресов, \mathbf{S} – множество двоичных строк.

$\mathbf{R} = \{\text{истина, ложь, пусто, ошибка}\}$ – множество результатов выполнения QF-функции.

$\mathbf{Q} = (q_i | q_i \in \mathbf{QF})$ – QF-программа, представляющая собой последовательность QF-функций q_i из множества \mathbf{QF} .

Описание QF-функции представляет собой пару:

$$q_i = \langle c, \mathbf{Q} \rangle, \quad (1)$$

где $c \in \mathbf{T}$ – статический параметр QF-функции.

При вызове QF-функции с описанием q_i происходит выполнение функции f_{q_i} , где:

$$r = f_{q_i}(q, p, e), \quad (2)$$

где $r \in \mathbf{R}$ – элемент результат выполнения функции, $p \in \mathbf{T}$ – динамический параметр функции, а e – событие сетевой СОА, имеющее следующий вид:

$$e = \langle av, u \rangle, \quad (3)$$

где $av \in \mathbf{A} \times \mathbf{T}$ – элементы декодированных заголовков протоколов, \mathbf{A} – множество возможных элементов декодированных заголовков протоколов, $u \in \mathbf{S}$ – недекапсулированные данные. Так как событие $e \in \mathbf{E}$ доступно для всех QF-функций, то в некоторых случаях оно не будет явно указано в параметрах.

QF-функции можно сгруппировать по следующим основным типам.

1. Функции загрузки.

$$r_0 = f_{\text{загрузка}}(\langle \langle \emptyset, (q_1) \rangle, \emptyset \rangle), \quad (4)$$

где $r_0 = f_{q_1}(q_1, p_{\text{загрузка}})$, $p_{\text{загрузка}} = f_{\text{преобр.}}(e)$, $r_0 \in \mathbf{R}$, $q_1 \in \mathbf{Q}$, $e \in \mathbf{E}$, $p_{\text{загрузка}} \in \mathbf{T}$.

Таким образом, функция загрузки реализует следующее отображение:

$$f_{\text{загрузка}} : e \rightarrow p_{\text{загрузка}} \quad (5)$$

2. Функции обработки.

$$r_0 = f_{\text{обр.}}(\langle \langle c_{\text{обр.}}, (q_1) \rangle, p_{\text{вход}} \rangle), \quad (6)$$

где $r_0 = f_{q_1}(q_1, p_{\text{выход}})$, $r_0 \in \mathbf{R}$, $q_1 \in \mathbf{Q}$, $p_{\text{вход}} \in \mathbf{T}$, $p_{\text{выход}} \in \mathbf{T}$, $c_{\text{обр.}} \in \mathbf{T}$, а функция обработки определяет отображение:

$$f_{\text{обр.}} : \langle p_{\text{вход}}, c_{\text{обр.}} \rangle \rightarrow p_{\text{выход}} \quad (7)$$

3. Функции сравнения можно условно разделить на тип А (сравнение статического параметра с динамическим) и тип Б (обработка результатов вложенных функций). Функции сравнения типа А имеют следующий вид:

$$r_0 = f_{\text{сравн.А}}(\langle \langle c_{\text{сравн.}}, \emptyset \rangle, p_{\text{сравн.}} \rangle), \quad (8)$$

где $r_0 \in \mathbf{R}$, $p_{\text{сравн.}} \in \mathbf{T}$, $c_{\text{сравн.}} \in \mathbf{T}$, а функция сравнения определяет отображение:

$$f_{\text{сравн.А}} : \langle p_{\text{сравн.}}, c_{\text{сравн.}} \rangle \rightarrow r_0 \quad (9)$$

Функции сравнения типа Б имеют следующий вид:

$$r_0 = f_{\text{сравн.Б}}(\langle \langle \emptyset, \mathbf{Q}_1 \rangle, \emptyset \rangle), \quad (10)$$

где $r_0 \in \mathbf{R}$, $\mathbf{Q}_1 \subset \mathbf{Q}$, а функция сравнения определяет отображение:

$$f_{\text{сравн.Б}} : \mathbf{R}_{\text{сравн.}} \rightarrow r_0, \quad (11)$$

где $\mathbf{R}_{\text{сравн.}} = \{f_{q_i}(c_i, p_i) | q_i \in \mathbf{Q}_1\}$, $p_i \in \mathbf{T}$, $c_i \in \mathbf{T}$.

4. Функции условного перехода.

$$r_0 = f_{\text{условн.}}(\langle \langle \emptyset, \{q_1, q_2, q_3\} \rangle, \emptyset \rangle), \quad (12)$$

где $q_1 \in \mathbf{Q}$, $q_2 \in \mathbf{Q}$, $q_3 \in \mathbf{Q}$,

$$r_0 = f_{q_2}(q_2, \emptyset), \text{ если } f_{q_1}(q_1, \emptyset) = \text{истина},$$

$$r_0 = f_{q_3}(q_3, \emptyset), \text{ если } f_{q_1}(q_1, \emptyset) \neq \text{истина}.$$

5. Функции вывода.

$$r_0 = f_{\text{вывод}}(\langle \langle c_{\text{результат}}, \emptyset \rangle, p_{\text{результат}} \rangle), \quad (13)$$

где $r_0 = \text{пусто}$, $p_{\text{результат}} \in \mathbf{T}$, $c_{\text{результат}} \in \mathbf{T}$, а пара $\langle c_{\text{результат}}, p_{\text{результат}} \rangle$ выводится, как результат работы СОА.

В QF-функции обработки входят также функции работы с регулярными выражениями [9], что позволяет гибко обрабатывать поля текстовых протоколов. Язык может быть расширен новыми функциями, вызываемыми из модулей расширения СОА. Кроме того, язык содержит функции, позволяющие сохранять состояние QF-программы в виде битовой маски между ее вызовами внутри обработки одного сеанса взаимодействия клиента и сервера. Так как описание QF-программы имеет древовидную структуру, то для описания QF-программы был выбран язык XML.

Преобразование сигнатур СОА Snort. Сигнатуру СОА Snort можно представить в следующем виде:

$$ss_i = \langle \mathbf{SC}_i, \mathbf{SD}_i \rangle, \quad (14)$$

где ss_i – i -я сигнатура, \mathbf{SC}_i – множество элементов условия i -й сигнатуры, \mathbf{SD}_i – множество

элементов описания i -й сигнатуры, $ss_i \in SS$, SS – множество сигнатур.

Преобразование множества сигнатур COA Snort в QF-программу можно представить с помощью следующих правил:

$$SS \rightarrow \langle q_{\text{старт}}, Q_{\text{описание}}, q_{\text{конец}} \rangle, \quad (15)$$

где $q_{\text{старт}} \in Q$ – функция начала (пролог), $q_{\text{конец}} \in Q$ – функция конца (эпилог), $Q_{\text{описание}} \subset Q$ – функции описательной части.

$$ss_i \rightarrow \langle q_{\text{условие}}, q_{II}, Q_{\text{элементы}}, q_{\text{вывод}} \rangle, \quad (16)$$

где $\langle q_{\text{условие}}, q_{II}, Q_{\text{элементы}}, q_{\text{вывод}} \rangle \in Q_{\text{описание}}$,

$q_{\text{условие}} \in Q$ – функция условного перехода, $q_{II} \in Q$ – функция операции "И" (функция сравнения типа Б), $Q_{\text{элементы}} \subset Q$ – функции элементов условия сигнатуры.

$$sc_i \rightarrow \langle q_{\text{загрузка}}, q_{\text{сравнение}} \rangle, \quad (17)$$

где $\langle q_{\text{загрузка}}, q_{\text{сравнение}} \rangle \in Q_{\text{элементы}}$,

$$q_{\text{загрузка}} \in Q, q_{\text{сравнения}} \in Q.$$

Практическая проверка результатов работы такой QF-программы, образованной из сигнатур COA Snort, показала крайне низкую производительность обработки при росте числа сигнатур. На рисунке 1 приведена зависимость времени обработки 1 миллиона пакетов двух видов тестового трафика А и Б от количества сигнатур на одном ядре процессора. Трафик А представляет собой выборку ТСП-пакетов из типового трафика Интернет. Трафик Б представляет собой выборку коротких (менее 1 Кбайта) HTTP-сессий из типового трафика Интернет. Следует обратить внимание на постоянный рост времени обработки.

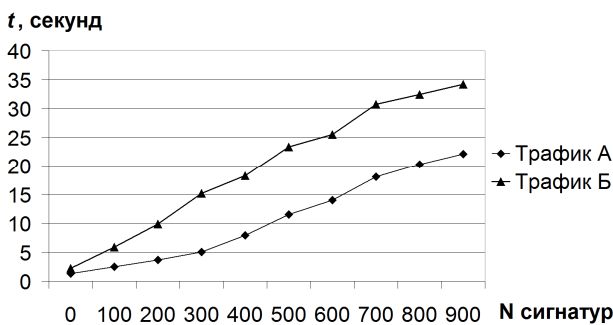


Рисунок 1 – Зависимость времени обработки 1 млн пакетов от количества преобразованных сигнатур без использования оптимизации

На рисунке 2 показана такая же зависимость для COA Snort. Следует обратить внимание на

практическое отсутствие роста времени обработки.

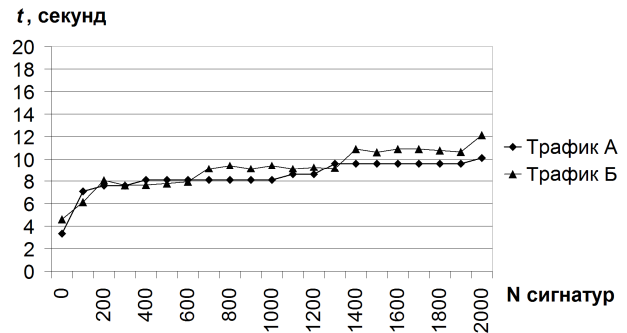


Рисунок 2 – Зависимость времени обработки 1 млн пакетов от количества сигнатур для COA Snort

Проведенное исследование показало наличие следующих проблем, вызывающих падение производительности.

1. Избыточное дублирование условий. Например, для всех сигнатур, связанных с протоколом HTTP, в каждом сгенерированном условии выполнялась одинаковая проверка на тип протокола.

2. Особенности организации современных процессоров. При обращении к фрагментам QF-программы, расположенным в различных частях оперативной памяти, неэффективно используется кэш-память процессора. При вызове внутри интерпретатора QF-программы динамических обработчиков QF-функций неэффективно используются конвейер процессора и возможность внеочередного выполнения инструкций.

Оптимизация выполнения сигнатур COA Snort. Для решения проблемы дублирования условий все условия сигнатур COA Snort были разбиты на две части:

- 1) условия, применимые к AV-парам декодированных служебных заголовков протоколов;
- 2) условия поиска подстрок в недекапсулированных данных.

Для условий первого типа формируется список уникальных условий:

$$Q_{U,Y} = \cup Q_{AV_i}, \quad (18)$$

где $Q_{\text{элементы}_i} = Q_{AV_i} \cup Q_{S_i}$, $Q_{AV_i} \subset Q$ – условия первого типа, $Q_{S_i} \subset Q$ – условия второго типа. В результате каждая Snort сигнатура будет преобразована следующим образом:

$$ss_i \rightarrow \langle q_{\text{условие}}, q_{II}, Q_{AV_i} \cup Q_{S_i}, q_{\text{вывод}} \rangle. \quad (19)$$

Интерпретатор QF-программы на первом шаге поочередно выполняет каждое выделенное уникальное условие и формирует битовую

маску, в которой единицей указывается истинность данного условия. Далее при выполнении QF-функции перед проверкой условий типа 2 для сигнатуры проверяется простая проверка битовых масок:

$$M_e \cap M_i = M_i, \quad (20)$$

где M_e – сформированная битовая маска для события СОА, а M_i – битовая маска i -й сигнатуры.

Использование битовых масок дало прирост производительности более чем в 3 раза.

Для решения проблемы особенности организации современных процессоров, была выполнена предварительная компиляция QF-условий в псевдоинструкции виртуального процессора. Структура инструкции имеет следующий вид:

$$\langle q_{\text{загрузка}}, q_{\text{сравнение}}, p \rangle, \quad (21)$$

где $q_{\text{загрузка}} \in \mathbf{Q}$, $q_{\text{сравнение}} \in \mathbf{Q}$, $p \in \mathbf{P}$. Для часто используемых QF-функций загрузки и сравнения не происходит вызовов их обработчиков, а их работа выполняется через встроенные (inline) операции.

Использование предварительной компиляции дало прирост производительности более 20 %.

Следующий шаг оптимизации использовал тот факт, что поиск подстрок в сигнатурах СОА Snort выполняется в 2 этапа: упрощенный (для быстрого отсеивания пакетов) и сложный (с полной проверкой, включая использование регулярных выражений). Для реализации упрощенного поиска подстрок было использовано единое префиксное дерево (бор), обеспечивающее поиск самой длинной строки из каждой сигнатуры. Таким образом, после выполнения обработки дерева подстрок становится возможным заранее отобрать сигнатуры, которые следует исключить из рассмотрения. Использование дерева подстрок дало прирост производительности более чем в 3 раза.

Для решения проблемы особенностей организации современных процессоров, связанных с организацией кэш-памяти, было предпринято переупорядочение хранимого в памяти дерева строк так, чтобы все элементы хранились последовательно. Использование переупорядочивания дерева в памяти дало прирост производительности более 10 %.

Последним шагом оптимизации стало преобразование префиксного дерева в конечный автомат (алгоритм Ахо-Корасик), что дало прирост производительности около 3 % и привело к результатам, показанным на рисунке 3.

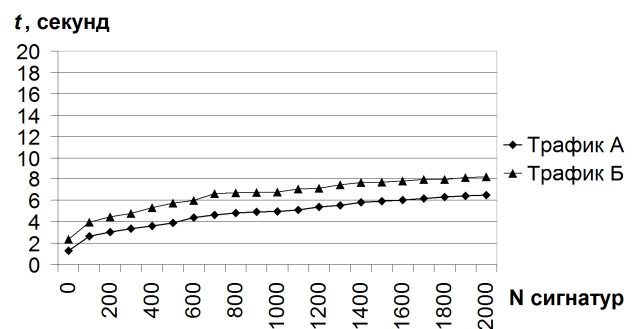


Рисунок 3 – Зависимость времени обработки 1 млн пакетов от количества преобразованных сигнатур с использованием разработанных методов оптимизации

Полученная скорость обработки сетевого трафика в тысячах пакетов в секунду для набора из 1000 и 2000 сигнатур СОА Snort для тестового трафика А и Б приведена в таблице.

N сигнатур, тип трафика	1000		2000	
	А	Б	А	Б
СОА Snort	123,5	106,3	98,8	82,4
Разработанный метод	202,8	148,3	154,9	123,0

При среднем размере пакета 600 байт разработанный метод позволяет обрабатывать трафик порядка 500 Мбит/с на одном ядре процессора.

Выводы. Предложенный метод реализации сигнатурного поиска в сетевой СОА позволяет использовать сигнатуры СОА Snort и обеспечивает в среднем на 30 % большую производительность данной СОА.

На базе данного метода была разработана сетевая СОА, реализующая сигнатурный метод выявления СА, а также программное обеспечение преобразования сигнатур СОА Snort в формат разработанной СОА. Разработанная сетевая СОА позволяет при использовании более чем 20-ядерного сервера и специализированных сетевых карт перехвата обрабатывать трафик в ВС, использующих скорости передачи информации 10 Гбит/с.

Библиографический список

- ГОСТ Р 51275-2006. Защита информации. Объект информатизации. Факторы, воздействующие на защищаемую информацию. Общие положения.
- Ивашко А.М. Некоторые методологические и организационные проблемы обнаружения и предупреждения компьютерных атак на критически важные объекты информационной инфраструктуры Российской Федерации // Материалы Второй международной научной конференции по проблемам безопасности и противодействия терроризму. М., 2006. С. 67-73.
- TILEmpower™ Platform. TILEmpower™

Platform. Product Brief. [Электронный ресурс] - 2009. - Режим доступа: http://www.tilera.com/products/platforms/TILEmpower_platform, свободный – Загл. с экрана.

4. *Ian Graham*. Achieving Zero-loss Multi-gigabit IDS – Results from Testing Snort® on Endace Accelerated Multi-CPU Platforms. [Электронный ресурс] - 2006. - Режим доступа: <http://www.touchbriefings.com/pdf/2259/graham.pdf>, свободный – Загл. с экрана.

5. *Ashley Thomas*. RAPID: Reputation based Approach for Improving Intrusion Detection Effectiveness. [Электронный ресурс] - 2009 – Режим доступа: <http://www.secureworks.com/research/whitepapers-shortcut/SWRX-RAPID.pdf>, свободный – Загл. с экрана.

6. SNORT Users Manual 2.9.1 [Электронный ресурс] / The Snort Project, 2011. – Режим доступа: http://www.snort.org/assets/166/snort_manual.pdf, свободный – Загл. с экрана.

7. OSI model [Электронный ресурс] / Wikipedia the Free Encyclopedia, 2011. – Режим доступа: http://en.wikipedia.org/wiki/OSI_model, свободный – Загл. с экрана.

8. *Steven McCanne, Van Jacobson*. The BSD Packet Filter: A New Architecture for User-level Packet Capture? - 1992. – Режим доступа: <http://www.tcpdump.org/papers/bpf-usenix93.pdf> свободный – Загл. с экрана.

9. *Jeffrey E.F. Friedl*. Mastering Regular Expressions, Third Edition // O'Reilly Media. 2006.