

## ОСОБЕННОСТИ РЕАЛИЗАЦИИ РАСПОЗНАВАТЕЛЯ ДИКТОРА НА РЕКУРРЕНТНОМ ПРОЦЕССОРЕ

Рождественский Ю.В., Дьяченко Ю.Г., Морозов Н.В.

### 1. Введение

Распознаватель диктора является немаловажным компонентом интеллектуальных диалоговых систем, обеспечивающих комфорт и безопасность среды обитания человека. Кроме того, задача распознавания диктора привлекает все больше внимания как средство защиты конфиденциальной информации и ограничения права доступа во многих практических приложениях, начиная от автомобильной и бытовой электроники и кончая сферой банковской деятельности и охраны режимных объектов [1–4].

Использование голоса как индивидуальной характеристики человека является привлекательным по целому ряду причин:

- голос – наиболее удобное и естественное средство идентификации,
- голос уникален для каждого человека и является его неотъемлемым идентификатором,
- голос человека невозможно украсть и довольно трудно подделать (имитировать),
- оборудование для приема “отпечатка голоса” простое и миниатюрное.

Алгоритмическая база распознавателя диктора имеет много общего с задачей распознавания речи. И это не удивительно, поскольку в том и другом случае используются характерные особенности произнесения человеком заранее заданных слов или фраз. Однако в оценке результата распознавания используются различные подходы. В отличие от распознавателя речи, где точность распознавания определяется как относительное количество правильно распознанных слов (фраз), качество работы распознавателя диктора характеризуется двумя типами ошибок:

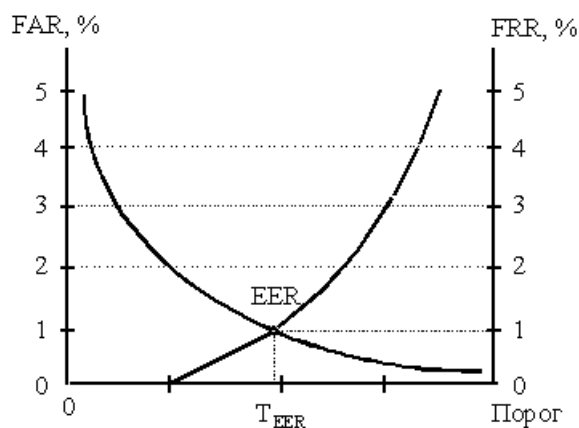
- ошибочный допуск “чужих” (False Acceptance Rate – FAR), не имеющих права доступа к запрашиваемому ресурсу,
- ошибочный отказ в доступе “своим” (False Rejection Rate – FRR), зарегистрированным в системе контроля доступа к ресурсу.

Усредненное качество распознавания определяется уровнем равной ошибки (Equal Error Rate – EER), при котором  $FAR = FRR$ . Распознаватели высокого класса надежности имеют  $EER \leq 0.1\%$  и используются в основном в сфере финансовых приложений и в системах высокой ответственности. Для своей реализации они требуют использования высокопроизводительных вычислительных систем. Распознаватели низкого класса надежности имеют  $EER \leq 5\%$ . Они используются, главным образом, в системах личного и домашнего применения и предъявляют гораздо меньшие требования к производительности вычислителя.

Уровень ошибок FAR, FRR и EER зависит от выбранных алгоритмов выделения речевых параметров и формирования модели говорящего. На рис.1 представлена зависимость уровня FAR и FRR от установленного порога

распознавания для некоторого распознавателя с  $EER=1\%$ . Порог распознавания – это значение вероятности соответствия тестируемого высказывания хранящейся в базе данных модели фразы или голоса, при котором тестируемое высказывание признается принадлежащим зарегистрированному пользователю. Из рис. 1 видно, что ошибки типа FAR и FRR – взаимозависимы для каждого распознавателя диктора: уменьшение одной из них за счет каких-то алгоритмических ухищрений неминуемо ведет к увеличению другой при фиксированном уровне EER.

В целом ряде применений превалирующим критерием является низкая стоимость распознавателя диктора при достаточно высоком качестве распознавания на уровне  $EER \leq 2\%$ . Это делает актуальной задачу дешевой реализации распознавателя диктора с приемлемыми качественными характеристиками. Анализ показывает, что появление процессоров с параллельной и рекуррентной архитектурой позволяет перевести эту задачу в практическую плоскость.



**Рис. 1. Зависимость уровня FAR и FRR от установленного порога распознавания  $T_{ERR}$**

По типу используемого речевого материала распознаватели диктора разделяются на *тексто-зависимые* и *тексто-независимые*. Тексто-зависимые распознаватели оперируют заданным набором фраз, надиктованным "своими" дикторами на этапе тренировки распознавателя, и используют модели этих фраз как шаблоны для оценки тестового произнесения диктора.. Тексто-независимые распознаватели допускают использование в качестве тестовой любой фразы и работают с шаблонами более "мелких" фрагментов речи: фонемы, дифоны и т.д. Последние распознаватели обладают большей степенью защиты, но и более требовательны к вычислительным ресурсам.

По характеру решаемой задачи распознаватели диктора делятся на *идентификаторы* и *верификаторы* говорящего. И если первым достаточно определить, принадлежит ли тестируемый голос диктору из группы "своих", то вторым необходимо принять решение, принадлежит ли голос именно тому человеку, за которого выдает себя диктор. Идентификаторы диктора требуют существенно меньше как вычислительных ресурсов, так и объемов памяти для хранения базы данных – основы распознавания.

Данная работа посвящена исследованию особенностей реализации распознавателя диктора, относящегося к классу тексто-зависимых идентификаторов и обеспечивающего уровень равной ошибки  $EER \leq 2\%$ , на

типовом рекуррентном процессоре с параллельной архитектурой, а также исследованию возможности его реализации на вычислителях малой производительности, типичными представителями которых являются 8-разрядные микроконтроллеры.

## 2. Алгоритмическая основа распознавателя

Для рассматриваемой задачи идентификации диктора традиционный алгоритм работы текстозависимого распознавателя может быть представлен в



*Рис. 2. Алгоритм распознавания диктора*

виде, изображенном на рис. 2. Он аналогичен алгоритму распознавателя речи, описанному в работе [5] на рис. 3.

Прошедший входную предобработку речевой сигнал сегментируется на фреймы (20-мс интервалы) и поступает на блок анализа, выделяющий индивидуальные для данного голоса (диктора) параметры для каждого фрейма: кепстральные коэффициенты, уровень энергии и т.д. Вычисленный вектор параметров (характеристический вектор, наблюдение) квантуется с помощью кодовой книги векторов, полученной на этапе обучения, и замещается одним индексом. Последовательность индексов (наблюдений), сформированная по результатам обработки всех фреймов входного речевого сигнала, тестируется на всех моделях дикторов (фраз) из базы данных с учетом предустановленного порога. Вычисляются вероятности описания полученной последовательности наблюдений каждой моделью и принимается решение о том, принадлежит ли тестируемая фраза "своему" диктору или "чужому".

Выделение речевых параметров происходит на основе анализа речевого сигнала в частотной области и использования особенностей слухового восприятия человека [6, 7] для упрощения последующих вычислений и повышения качества распознавания. Традиционный подход в этом случае основывается на использовании быстрого преобразования Фурье (БПФ) для вычисления энергетического спектра сигнала на текущем фрейме, преобразовании энергетического спектра в барковский спектр и применении RASTA фильтрации [6], существенно повышающей качество распознавания в условиях присутствия аддитивных и мультипликативных фоновых шумов. Кепстральные параметры, вычисляемые из фильтрованного барковского спектра, служат основой для формирования характеристического вектора, отражающего особенности анализируемого произнесения на текущем фрейме.

Этапы работы распознавателей речи и диктора совпадают. Различие состоит в параметрах характеристического вектора, формируемого в процессе анализа речи.

Распознаватель речи делает акцент на выделении особенностей произнесения одной речевой единицы (фонемы, дифона, трифона и т.д.) по сравнению с другой различными дикторами. При этом далеко не всегда важно, кто и как произносит конкретный звук. Главная задача распознавателя речи – идентифицировать сам звук. Поэтому характеристический вектор формируется на основе 6–8 кепстральных параметров и так называемых дельта-коэффициентов, отражающих скорость изменения кепстральных параметров от фрейма к фрейму. Дельта-коэффициенты дополнительно специфицируют различия между произнесениями звуков, повышая качество распознавания речи.

Распознаватель же диктора выявляет прежде всего особенности произнесения одного и того же звука разными дикторами. Для достижения этой цели характеристический вектор формируется из максимально возможного числа кепстральных параметров, исходя из вычислительных возможностей процессора, на котором предполагается реализация распознавателя. Как правило, 12 – 15 кепстральных параметров достаточно для получения хорошего качества распознавания. Здесь дельта-коэффициенты уже не играют существенной роли.

Оценка вычислительной сложности реализации описанного алгоритма идентификации говорящего равна примерно 20 MOPS для 16-разрядного ПЦОС с типовой архитектурой [7]. Львиная доля затрат (75 – 80%) приходится на преобразование в спектральной (частотной) области и квантование вектора параметров. Такие вычислительные затраты чрезмерны для простейших ПЦОС. Кроме того, анализ показывает, что они не являются необходимыми и аналогичного результата можно достичь при существенно меньших затратах. В связи с этим возникает задача модификации и максимального упрощения алгоритмов распознавателя.

Модификация алгоритма распознавателя речи, описанные в работе [5], позволяют аналогичным образом упростить и распознаватель диктора. В результате алгоритм распознавания диктора сводится к последовательности этапов, на каждом из которых основные затраты приходятся на фильтрацию сигнала, однополосную или многополосную, и на независимую обработку промежуточных данных в цикле по полосам спектра сигнала. Эти операции допускают параллельную обработку данных и эффективно реализуются на архитектуре рекуррентного процессора или параллельного ПЦОС. Рассмотрим более подробно этапы работы алгоритма распознавания диктора.

### **2.1. Предобработка речевого сигнала**

На этапе предобработки анализируемого речевого сигнала используется фильтр Баттерворта четвертого порядка. Его импульсная характеристика описывается формулой:

$$H(z) = \prod_{k=1}^2 \frac{1 + a_{1k}z^{-1} + a_{2k}z^{-2}}{1 + b_{1k}z^{-1} + b_{2k}z^{-2}},$$

где  $a_{1k}$ ,  $a_{2k}$ ,  $b_{1k}$  и  $b_{2k}$  — коэффициенты фильтра, имеющие следующие значения:  $a_{11} = -2.000125721$ ,  $a_{21} = 1.000125737$ ,  $a_{12} = -1.999873569$ ,  $a_{22} = 0.999873585$ ,  $b_{11} = -1.943779252$ ,  $b_{21} = 0.952444269$ ,  $b_{12} = -1.866892280$ ,  $b_{22} = 0.875214548$ .

Вычислительную сложность реализации фильтра Баттерворта удалось уменьшить за счет следующих упрощений. Каждый коэффициент фильтра может быть представлен разложением:  $A = -2 \pm \Delta A$ , или  $A = 1 \pm \Delta A$ . В результате в операциях умножения участвуют не сами коэффициенты фильтра,

а их дополнения до ближайшего целого значения, существенно меньшие по абсолютной величине. Это позволило использовать однобайтное представление множителей и за счет этого в 2.3 раза сократить сложность вычислений на этапе входной фильтрации.

Сравнительный анализ результатов фильтрации речевых сигналов на полномасштабной и упрощенной модели фильтра Баттерворта подтвердил правомерность использования такого упрощения.

Степень параллельности данной реализации фильтра Баттерворта – 4.

## 2.2. Вычисление барковского спектра

Исследования алгоритмов анализа речи и опыт реализации распознавателя речи показывают, что для получения барковского спектра можно успешно использовать полосовую фильтрацию вместо БПФ. При этом качественные характеристики программы идентификации диктора практически не изменяются.

Действительно, барковский спектр речевого сигнала строится на основе его энергетического частотного спектра. Он описывает характеристики речевого сигнала по полосам частот, значимым с точки зрения особенностей слухового восприятия человека [6]. Фактически он показывает уровень энергии анализируемого сигнала в соответствующей полосе частот.

В процессе фильтрации для каждой полосы подсчитывается суммарная энергия фильтрованного сигнала, значения которой и формируют спектр, аналогичный тому, который получается в результате применения преобразования Фурье и перевода энергетического спектра речевого сигнала в область барковских частот.

Количество фильтров равно числу барковских полос. Уравнение одного полосового фильтра описывается формулой:

$$H_B(z) = \frac{c_k}{1 + d_{1k}z^{-1} + d_2z^{-2}}, \quad k = 1, \dots, N_B,$$

где  $c_k$ ,  $d_{1k}$  и  $d_2$  — коэффициенты фильтра, первые два из которых зависят от номера полосы барковских частот, а коэффициент  $d_2$  имеет общее значение для всех фильтров.

Поскольку для построения характеристического вектора используется большее количество кепстральных коэффициентов, чем в распознавателе речи, применение односекционного фильтра вместо полномасштабного двухсекционного не столь существенно сказывается на качестве распознавания.

В свою очередь, существенное сокращение вычислительной сложности реализации банка фильтров на 8-разрядной архитектуре типового микроконтроллера достигается за счет использования следующих упрощений:

- однобайтного представления коэффициентов банка фильтров,
- однобайтного представления входного сигнала и результата вычислений в фильтрах верхних 9 полос,
- расчет энергии по полосам на основе только старших байт фильтрованного сигнала.

Оценки показывают, что такое упрощение позволяет достаточно точно реконструировать барковский спектр речевого сигнала. Среднее отклонение рассчитанной таким образом энергии сигнала по барковским полосам от энергии, вычисленной с помощью преобразования Фурье, не превышает 7%. Поскольку такая методика используется как на предварительном этапе

тренировки моделей, так и на этапе реального распознавания, итоговый уровень равной ошибки распознавания EER увеличивается примерно на 0.2%.

Степень параллельности такой реализации полосовой фильтрации – не менее 15 для частоты дискретизации входного речевого сигнала 8 кГц.

### ***2.3. Шумоподавление***

Шумоподавление осуществляется с помощью преобразования барковского спектра (логарифмирование, взвешивание, RASTA фильтрация, экспонирование) независимо по каждой из 15 барковских полос [5]. Каждая из этих операций допускает успешное распараллеливание обработки данных. При этом степень параллельности равна числу барковских полос – 15.

### ***2.4. Квантование вектора параметров***

Квантование вектора параметров осуществляется на основе векторной кодовой книги. Оно заключается в том, что полученные на этапе анализа характеристические вектора сопоставляются с векторами кодовой книги и замещаются номером (индексом) вектора, расстояние до которого минимально.

Для каждого вектора из кодовой книги выполняются одни и те же операции вычисления расстояния от него до квантуемого вектора. Следовательно, данный этап также эффективно реализуется на процессоре с параллельной архитектурой. Максимальная степень параллельности равна количеству векторов в кодовой книге (обычно 256). Практически же степень параллельности ограничивается возможностями используемого процессора.

Для реализации программы идентификации диктора на маломощных ПЦОС предлагается использовать манхеттенское расстояние [5] в качестве меры близости квантуемого вектора к табличному вместо Евклидова по аналогии с распознавателем речи.

Эксперименты показывают, что замена Евклидова расстояния на манхеттенское при использовании двухбайтного представления квантуемого вектора и векторов кодовой книги на 0.1% увеличивает уровень равной ошибки распознавания EER, в несколько раз сокращая вычислительные затраты при реализации распознавателя, например, на 8-разрядном микроконтроллере.

### ***2.5. Поиск ближайшей модели***

Этап анализа входного речевого сигнала и выделения характеристических векторов параметров, описывающих особенности речевого сигнала, заканчивается формированием последовательности квантованных наблюдений. Эта последовательность отображает особенности произнесения тестируемой фразы конкретным диктором и служит некоторым "отпечатком", на основе которого ищется наилучшим образом соответствующая ему модель фразы.

Среди различных методов и моделей, лежащих в основе такого поиска, наименьшей вычислительной сложностью при высоком уровне эффективности обладает метод Витерби на дискретных скрытых марковских моделях (Hidden Markov Models – НММ [4]). Работая в области логарифмических значений вероятностей, он позволяет обойтись без операций умножения в процессе вычислений.

Вариант НММ фразы, используемой в распознавателе диктора, аналогичен НММ слова, описанной в работе [5]. Отличие заключается лишь в количестве состояний модели: фраза, как более длительная по звучанию

единица речи, требует и большего числа состояний для описания последовательности межзвуковых переходов. В общем случае, чем длиннее фраза, тем большее число состояний модели требуется для ее представления.

Параллельность алгоритма Витерби основывается на том, что текстозависимый идентификатор диктора оперирует моделями фраз, индивидуальными для каждого диктора из группы "своих" дикторов, проверяя "отпечаток" произнесенной фразы на соответствие модели данной фразы для каждого диктора из списка "своих". Поскольку этот процесс происходит независимо для каждой модели, рекуррентный процессор имеет возможность обрабатывать одновременно столько моделей, сколько параллельных вычислителей он имеет в своем составе.

Степень параллельности алгоритма Витерби равна количеству "своих" дикторов.

Упрощение реализации алгоритма Витерби достигается за счет использования однобайтного представления логарифмов вероятностей наблюдений и переходов в процессе вычислений и двухбайтного накопления суммарной вероятности соответствия произнесенной фразы НММ из библиотеки. Практика показала, что такой точности вычислений достаточно для сохранения требуемого уровня качества распознавания.

### **3. Реализация распознавателя на рекуррентном процессоре**

Анализ рассмотренных в предыдущем разделе алгоритмов, лежащих в основе текстозависимого идентификатора диктора, показывает, общая задача распознавания диктора по произнесенной им фразе распадается на последовательность этапов, каждый из которых в значительной степени состоит из параллельных вычислений.

Для типового идентификатора диктора степень параллельности на каждом этапе варьируется от 4 до 256. И если число "своих" дикторов, на которое настроен распознаватель, не превышает 16, то наиболее целесообразным для реализации такого распознавателя представляется использование архитектуры рекуррентного процессора с 16 параллельными вычислителями. Опыт реализации аналогичных задач обработки речи (например, вокодеров) доказывает, что общие затраты времени на обработку анализируемого произнесения фразы на таком процессоре сократятся на 60 – 70% по сравнению с процессором с фон неймановской конвейерной архитектурой.

Параллельные операции, выполняемые на каждом из этапов, не имеют критериев досрочного окончания этапа: каждая параллельная ветвь вычислений должна быть полностью выполнена независимо от результата выполнения других ветвей. Это делает использование рекуррентного процессора еще более эффективным при реализации идентификатора диктора.

Особенности реализации этапов выделения барковского спектра на основе полосовой фильтрации и поиска ближайшей модели фразы с помощью алгоритма Витерби позволяют использовать одни и те же исходные данные для всех параллельных вычислителей. Это свойство также повышает эффективность применения рекуррентного процессора для решения задачи распознавания диктора.

### **4. Оценки реализуемости распознавателя на маломощном ПЦОС**

Описанные в разделе 2 упрощения алгоритмов распознавателя диктора приносят наиболее ощутимый эффект при реализации идентификатора диктора на маломощном ПЦОС, например, 8-миразрядном контроллере, вычислительные способности которого существенно ограничены. Дополнительное сокращение вычислительных затрат достигается использованием по возможности однобайтных вычислений вместо двухбайтных на всех этапах распознавания диктора. В совокупности все эти меры позволяют снизить требования к производительности микроконтроллера до допустимой величины, сохранив качество распознавания на приемлемом уровне.

В таблице показаны оценки вычислительной сложности реализации основных этапов программы идентификации диктора при использовании традиционных и предлагаемых алгоритмов на 8-разрядном типовом микроконтроллере. Они получены для случая, когда входной речевой сигнал представлен в двухбайтном РСМ формате, частота дискретизации равна 8 КГц, кодовая книга содержит 256 векторов, а размерность вектора параметров равна 13. Платой за почти десятикратное сокращение вычислительной сложности реализации алгоритмов распознавателя диктора на 8-разрядном микроконтроллере является увеличение примерно на 0.3% уровня равной ошибки распознавания EER.

**Таблица. Оценка реализуемости распознавателя диктора на 8-разрядном типовом микроконтроллере**

Этап работы	Вычислительные затраты, MOPS	
	традиционная реализация	упрощенная реализация
Предобработка входного сигнала	5.8	0.8
Вычисление барковского спектра	46.0	4.2
Шумоподавление и выделение характеристического вектора	2.4	1.5
Квантование вектора параметров	38.4	2.9
<b>Итого</b>	<b>92.6</b>	<b>9.4</b>

В режиме реального времени программа идентификации диктора выделяет характеристические вектора, квантует их и накапливает в памяти индексы квантованных наблюдений. При этом работают все алгоритмы, за исключением алгоритма Витерби.

Алгоритм Витерби работает на полной совокупности накопленных квантованных векторов и включается только после окончания произнесения диктором тестовой фразы. Поэтому его вычислительные затраты не оказывают влияния на суммарную сложность реализации распознавателя диктора и лишь определяют длительность интервала времени, в течение которого принимается решение о принадлежности диктора, произнесшего анализируемую фразу, к



числу зарегистрированных дикторов. Предварительные оценки показывают, что алгоритм Витерби на 8-разрядном микроконтроллере производительностью 10 MOPS способен обработать фразу длительностью 6 сек за 1 секунду реального времени, что вполне приемлемо с точки зрения эксплуатации распознавателя в режиме реального времени.

Для реализации программы идентификации диктора необходимы следующие объемы памяти:

- 1500 байт памяти данных (ОЗУ),
- 16 Кбайт программной памяти (ПЗУ),
- внешняя память для хранения НММ моделей, размер которой зависит от количества ключевых фраз и числа зарегистрированных пользователей системы идентификации (объем памяти для хранения НММ моделей для 64 клиентов и 10 ключевых фраз из 4 – 7 слов равен примерно 1 Мбайт).

Таковыми характеристиками обладает, например, семейство микроконтроллеров PIC18Cxxx фирмы Microchip, США [9].

Таким образом, программа идентификации диктора может быть успешно реализована на 8-разрядном микроконтроллере производительностью не более 10 MOPS, имеющем порт для подключения внешней памяти. Эксперименты показывают, что в условиях 15 дБ фонового белого шума и при фиксированных порогах распознавания для 64 зарегистрированных клиентов предлагаемый алгоритм обеспечивает точность распознавания на уровне равной ошибки распознавания  $EER < 2.0\%$ .

## 5. Заключение

Алгоритмическая основа текстозависимого распознавателя диктора в значительной мере базируется на выполнении параллельных вычислений. Степень параллельности при этом варьируется от 4 до 256, а исходные данные для разных ветвей параллельных вычислений часто совпадают. Поэтому использование рекуррентного процессора с параллельной архитектурой весьма эффективно для реализации такого класса распознавателей, позволяя на 60 – 70% ускорить обработку тестового речевого произнесения и принятие решения.

Реализация традиционных алгоритмов распознавателя диктора на ПЦОС предъявляет относительно высокие требования к производительности процессора (около 20 MOPS в терминах операций типового 16-разрядного ПЦОС) за счет, прежде всего, использования преобразования Фурье для анализа и обработки речевого сигнала в частотной области. Замена преобразования Фурье банком полосовых фильтров, наряду с менее значимыми упрощениями других алгоритмов в составе распознавателя, позволяет существенно снизить требования к производительности вычислителя – до 9 MOPS в терминах операций типового 16-разрядного ПЦОС без потери качества распознавания.

Дальнейшие упрощения алгоритмической базы и уменьшение точности наиболее трудоемких вычислений позволяет дополнительно снизить вычислительную сложность задачи (примерно до 10 MOPS в терминах операций типового 8-разрядного микроконтроллера) при сохранении приемлемого среднего качества распознавания на уровне 2% ошибочного отвержения "своих" и допуска "чужих". Таким образом, задача идентификации диктора может быть успешно решена с использованием относительно дешевых микросхем.

Упрощенная аппаратная реализация распознавателя диктора открывает

широкие возможности для его использования в качестве универсального блока речевого анализа в составе специализированных ПЦОС.

## 6. Литература

1. Magrin-Chagnolleau I., Wilke J. and Bimbot F., "A Further Investigation on AR-Vector Models for Text-Independent Speaker Identification", Telecom Paris (E.N.S.T.), 1996, <http://www.DSP.rice.edu/~imc/Publications/Icassp96/icassp96.html>.
2. Campbell J.P., "Testing with the YOHO CD-ROM Voice Verification Corpus", US Department of Defense, <http://www.biometrics.org/REPORTS/ICASSP95.html>.
3. Gauvain J.L., Lamel L.F. and Prouts B., "Speaker Identification and Verification", from LIMSI 1995 Scientific Report, March 1995, <http://www.limsi.fr/Recherche/TLP/reco /2pg95-sv.html>.
4. L.R.Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Proc. of the IEEE, 1989, Vol. 77, No 2, p. 257-286.
5. Рождественский Ю.В., Дьяченко Ю.Г., Морозов Н.В. Проблемы реализации распознавателя речи на рекуррентном процессоре. // Системы и средства информатики, вып. 13. – М.: Наука. 2003 - с.
6. Hermansky H., and Morgan N., "Rasta processing of speech". IEEE Transactions on Speech and Audio Processing, Special issue on Robust Speech Recognition, Vol. 2, No 4, 1994, pp. 578-589.
7. Hermansky H., "Perceptual Linear Predictive (PLP) Analysis of Speech", Journal of Acoustic Society of America, Vol. 87, No 4, 1990, pp.1738 - 1752.
8. Cohen A., "Speaker Recognition and Verification", DSP Group, <http://www.dspg.com>.
9. Microchip Technology Inc., <http://www.microchip.com>.

Рождественский Ю.В.

Дьяченко Ю.Г.

Морозов Н.В.