

УДК 621.391.266

# НОВЫЙ МЕТОД ВЫЧИСЛЕНИЯ ЛИНЕЙНЫХ СПЕКТРАЛЬНЫХ ЧАСТОТ РЕЧЕВЫХ СИГНАЛОВ, ОСНОВАННЫЙ НА УНИВЕРСАЛЬНОМ АЛГОРИТМЕ РЕШЕНИЯ ТРАНСЦЕНДЕНТНЫХ УРАВНЕНИЙ

В. Ю. СЕМЕНОВ

Институт гидромеханики НАН Украины, Киев

Получено 1.11.2002

Предложен новый эффективный метод вычисления линейных спектральных частот (ЛСЧ) речевых сигналов, который основан на разработанном алгоритме полного численного решения трансцендентных уравнений, не имеющих кратных корней. Принципиально алгоритм состоит из двух частей – выделения отрезков, содержащих единственный корень, и последующего нахождения корней с помощью одной из стандартных итерационных процедур. Эффективность различных модификаций предложенного метода поиска ЛСЧ проверена на реальных речевых сигналах. Исследованы два подхода к нахождению ЛСЧ – прямое решение трансцендентных уравнений относительно тригонометрических функций и решение полиномиальных уравнений, полученных в результате разложения по чебышевским полиномам. Свойство упорядоченности ЛСЧ использовано для снижения вычислительных затрат. В отличие от большинства существующих методов определения ЛСЧ, предложенный метод обладает произвольно высокой точностью вычислений и гарантирует устойчивость соответствующего авторегрессионного фильтра. Показано, что данный метод может быть применен в системах реального времени.

Запропоновано новий ефективний метод обчислення лінійних спектральних частот (ЛСЧ) мовних сигналів, який базується на розробленому алгоритмі повного чисельного розв'язку трансцендентних рівнянь, що не мають кратних коренів. Принципово алгоритм складається з двох частин – виділення відрізків, які містять єдиний корінь, та подальшого знаходження кореня за допомогою однієї зі стандартних ітераційних процедур. Ефективність різних модифікацій запропонованого методу пошуку ЛСЧ перевірено на реальних мовних сигналах. При цьому досліджені два підходи до знаходження ЛСЧ – пряме розв'язання трансцендентних рівнянь відносно тригонометричних функцій та розв'язання поліноміальних рівнянь, отриманих внаслідок розкладу за Чебишевськими поліномами. Властивість упорядкованості ЛСЧ використано для зниження обчислювальних витрат. На відміну від більшості існуючих методів обчислення ЛСЧ, запропонований метод забезпечує доволі високу точність обчислень та гарантує стійкість відповідного авторегресійного фільтра. Показано, що даний метод може бути застосований в системах реального часу.

The new efficient method of calculation of the line spectral frequencies (LSF) is proposed. The method is based on a developed algorithm of full numerical solution of transcendental equation having no multiple roots. This algorithm is composed of two parts – location of intervals containing a single root and following refinement of root's value by one of standard rootfinding procedures. Efficiency of different modifications of the proposed LSF calculation method is verified on real speech signals. Two approaches to computation of LSF are considered – the direct solution of transcendental equations containing trigonometric functions and the solution of polynomial equations obtained by the series expansion in Chebyshev polynomials. The LSF's ordering property is used for decreasing the computational expenses. In opposite to majority of existing LSF computation algorithms, the proposed method provides arbitrary high accuracy and guarantees the stability of a corresponding autoregressive filter. It is shown that the developed method can be applied in real-time processing systems.

## ВВЕДЕНИЕ

Развитие современных приложений цифровой связи, таких как сотовая и IP-телефония, привело к необходимости разработки высокоэффективных алгоритмов кодирования речи, работающих на все более низких скоростях (обеспечивающих большую степень сжатия речевой информации). История алгоритмов кодирования речи начинается с 1940-х годов, когда появились первые вокодеры [1]. Рассмотрим основные причины, ведущие к необходимости сжатия речевых сигналов.

1. Ограниченная пропускная способность типовых каналов связи. Стоимость цифровых каналов связи обычно пропорциональна количеству передаваемых по ним цифровых дан-

ных. Хотя стоимость этих средств с каждым годом уменьшается, потребность в них еще более увеличивается.

2. Необходимость криптографической защиты информации. Кодирование речи, особенно низкоскоростное (со скоростью менее 4 кбит/с), открывает широкие возможности для шифрования передаваемой информации. Только преобразовав речь в цифровую форму и применив шифрование, можно гарантированно защититься от перехвата. Аналоговые средства шифрования (скремблеры) дают лишь временную защиту.
3. Необходимость компактного хранения речевых сигналов. Стоимость цифровой памяти

примерно пропорциональна количеству накапливаемых данных. Стандартные алгоритмы сжатия (Хафмена, Лемпеля–Зива и др.), лежащие в основе популярных архиваторов zip и arj, применительно к речевым сигналам дают очень низкую степень сжатия (менее, чем  $1.5 \div 2$  раза).

Напомним вкратце основные понятия, связанные с кодированием речи и ее передачей по цифровым каналам связи.

Количество бит, необходимое для представления одной секунды речевого сигнала, называют скоростью кодирования. Как правило, под несжатым речевым сигналом подразумевается его представление с помощью импульсно-кодовой модуляции (ИКМ), при которой речь дискретизируется с частотой 8 кГц и оцифровывается по фиксированной 256-уровневой шкале, образуя скорость кодирования 64 кбит/с (отметим, что в мультимедийных приложениях исходная скорость кодирования, как правило, существенно более высокая).

Алгоритмы кодирования речи можно разбить на три основные группы, в зависимости от степени сжатия, которую они обеспечивают:

- 1) высокоскоростные (скорость кодирования  $16 \div 64$  кбит/с);
- 2) среднескоростные ( $4 \div 16$  кбит/с);
- 3) низкоскоростные (менее 4 кбит/с).

Высокоскоростные методы сжатия речи, такие как ИКМ, дифференциальная ИКМ (ДИКМ), адаптивно-дифференциальная ИКМ (АДИКМ), не используют специфические особенности речи и могут быть применены для кодирования широкого класса сигналов.

Методы среднескоростного и низкоскоростного сжатия, важность которых в настоящее время особенно высока, основаны на моделях речеобразования. Наибольшее распространение получила авторегрессионная (АР) модель речеобразования [1–3], в которой речевой сигнал моделируется, как результат прохождения управляющего (возбуждающего) процесса через авторегрессионный (полюсной) фильтр. Принципиально эта модель отображена на рис. 1.

Возбуждающий процесс моделирует поток воздуха на выходе голосовых связок человека. В данной модели он представляет собой белый шум в случае произнесения глухих звуков (невокализованная речь) или последовательность импульсов, идущих друг за другом с периодом, соответствующим частоте колебания голосовых связок (вокализованная речь).

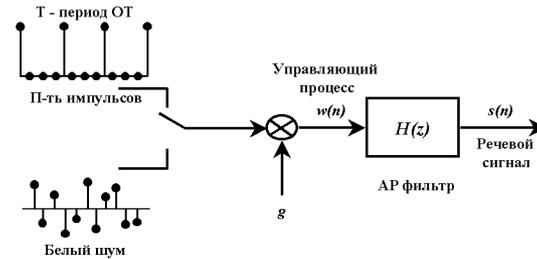


Рис. 1. Авторегрессионная модель формирования речевого сигнала

Работа полосного фильтра  $H(z)$ , моделирующего форму голосового тракта в момент произнесения звука, может быть описана в разностной форме:

$$s(n) = - \sum_{k=1}^p a_k s(n-k) + gw(n), \quad (1)$$

где  $s(n)$  – речевой сигнал;  $w(n)$  – возбуждающий процесс;  $g$  – коэффициент усиления;  $a_k$ ,  $k=1, 2, \dots, p$  – АР коэффициенты. Порядок АР модели  $p$  выбирается, как правило, от 8 до 20. Очевидно, что повышение порядка модели позволяет более точно моделировать спектральные характеристики речи.

Основным свойством параметров АР модели является их относительно медленное изменение с течением времени. Можно считать, что они остаются неизменными на отрезках длиной  $10 \div 30$  мс (свойство квазистационарности речевых сигналов).

Традиционно АР коэффициенты определяются с помощью методов линейного предсказания, в частности, автокорреляционного метода (АКМ) [1, 2]. На использовании данной модели речеобразования основано большинство средне- и низкоскоростных методов сжатия речи. Как правило, отличие между этими алгоритмами состоит лишь в способе кодирования возбуждающего процесса.

Таким образом, задача кодирования речи разбивается на две основные подзадачи: кодирование возбуждающего процесса (кодирование источника) и кодирование АР параметров  $a_k$ ,  $k=1, 2, \dots, p$ .

Однако непосредственное квантование АР коэффициентов на практике не используется. Это объясняется их чрезвычайно высокой спектральной чувствительностью. Небольшие ошибки квантования приводят к относительно большим спектральным искажениям в восстанавливаемом речевом сигнале и даже могут вызвать неустойчивость синтезирующего фильтра приемника. Кроме

того, АР коэффициенты не имеют четких динамических диапазонов изменения, что также серьезно затрудняет их квантование. Аналогичная проблема возникает в связи с задачей интерполяции АР коэффициентов: как уже отмечалось, в системах кодирования речи вычисление и квантование АР коэффициентов выполняется поблочно, на интервалах длиной от 10 до 30 мс. Столь медленное обновление параметров приводит к резким изменениям в АР коэффициентах на смежных фреймах, вследствие чего наблюдаются “щелчки” и другие заметные на слух нежелательные искажения. Для решения этой проблемы в принимающих устройствах необходимо вводить линейную интерполяцию АР коэффициентов.

Непосредственная интерполяция АР коэффициентов не может использоваться в силу тех же причин, что и в задаче квантования. Поэтому АР параметры перед кодированием обычно преобразуются нелинейным образом в эквивалентный набор параметров, например, в коэффициенты отражения или логарифмы отношений поперечных сечений модели речевого тракта [2, 3]. Существенную роль в последнее время играют также методы векторного квантования АР коэффициентов [3–5].

Наиболее популярным способом кодирования АР параметров в настоящее время становится их однозначное преобразование в набор линейных спектральных частот (ЛСЧ), квантование которых является наиболее эффективным по сравнению с другими эквивалентными представлениями АР модели.

Понятие ЛСЧ было впервые введено Итакура [6]. То, что ЛСЧ кодируют спектр речи более эффективно, чем другие наборы параметров, можно объяснить близостью их природы к природе формантных частот [7]. Поэтому ЛСЧ могут квантоваться с учетом особенностей восприятия речи человеком. Кроме того, понятие о ЛСЧ тесно связано с акустической моделью голосового тракта [8].

ЛСЧ применяются в американских федеральных стандартах CELP (code excited linear prediction, 4800 бит/с) [9], MELP (mixed excitation linear prediction, 2400 бит/с) [10], стандарте CS-ACELP (8000 бит/с) [11], используемого в IP-телефонии, и многих других алгоритмах сжатия речи. Например, в стандарте MELP речь разбивается на кадры длиной 22.5 мс, каждый из которых кодируется 54-ю битами (результатирующая скорость кодирования  $54 \times 1000 / 22.5 = 2400$  бит/с), из которых 34 бита отводится на кодирование десяти ЛСЧ, а остальные – на возбуждающий процесс и коэффициент усиления.

Кроме того, ЛСЧ продемонстрировали высокую эффективность в задачах распознавания речи [12] и идентификации диктора [13]. В работе [14] проиллюстрировано применение ЛСЧ к решению проблемы очистки речи от шума.

Таким образом, определение ЛСЧ по заданным АР коэффициентам  $a_k$ ,  $k=1, 2, \dots, p$  является актуальной научно-технической задачей, имеющей важное прикладное значение. В то же время, вычисление ЛСЧ, как будет показано, сопряжено со значительными вычислительными и аппаратными затратами. Существующие методы поиска ЛСЧ, как правило, являются результатом компромисса между количеством вычислений и их точностью. Обычно приемлемое быстродействие обеспечивается за счет увеличения погрешности определения ЛСЧ. Это может приводить к неустойчивости соответствующих авторегрессионных фильтров, что сказывается на качестве и разборчивости восстанавливаемого речевого сигнала. Вероятность таких сбоев особенно высока при использовании АР моделей относительно высоких порядков.

Цель данной работы – поиск более эффективных, по сравнению с известными, алгоритмов вычисления ЛСЧ. Основу предлагаемого подхода составляет развитый в работе универсальный метод решения трансцендентных уравнений  $f(x)=0$  для непрерывно дифференцируемых функций, не имеющих кратных корней.

Предложенный метод обладает высокой точностью вычислений и гарантирует устойчивость соответствующего авторегрессионного фильтра. В отличие от многих существующих методов, он не требует никакой априорной информации о расположении ЛСЧ. Кроме того, точности определения частот не зависят друг от друга и могут легко регулироваться. Показано, что данный метод может быть применен в системах реального времени.

## 1. СВОЙСТВА ЛИНЕЙНЫХ СПЕКТРАЛЬНЫХ ЧАСТОТ

Рассмотрим преобразование АР коэффициентов  $a_k$ ,  $k=1, 2, \dots, p$  в набор ЛСЧ. Пусть задан отбрасывающий фильтр с характеристикой

$$A(z) = 1 + \sum_{k=1}^p a_k z^{-k}. \quad (2)$$

Предположим, что этот фильтр является минимально-фазовым, то есть все его нули лежат внутри круга единичного радиуса (это всегда верно для АР коэффициентов, определенных

с помощью АКМ) [1].

Введем в рассмотрение следующие симметричный и асимметричный фильтры:

$$\begin{cases} F_s(z) = A(z) + z^{-p-1}A(z^{-1}), \\ F_a(z) = A(z) - z^{-p-1}A(z^{-1}). \end{cases} \quad (3)$$

Предполагая, что порядок модели  $p$  является четным числом ( $p=2M$ ), получим, что  $F_s(-1) = A(-1) - A(-1) = 0$ , т. е.  $F_s(z)$  имеет тривиальный корень  $z = -1$ . Аналогично получаем, что многочлен  $F_a(z)$  имеет тривиальный корень  $z = 1$ .

Деление полиномов (3) на сомножители, отвечающие за указанные корни, приводит к выражениям

$$\begin{cases} G_1(z) = \frac{F_s(z)}{1+z^{-1}}, \\ G_2(z) = \frac{F_a(z)}{1-z^{-1}}, \end{cases} \quad (4)$$

также являющимися многочленами, каждый из которых можно представить в виде

$$G_m(z) = \sum_{k=0}^p g_k^{(m)} z^{-k}, \quad g_0^{(m)} = g_p^{(m)} = 1, \quad (5)$$

$$m = 1, 2.$$

Коэффициенты этих симметричных полиномов могут быть вычислены по приводимым ниже рекуррентным зависимостям:

- полином  $G_1(z)$ :

$$g_k^{(1)} = \begin{cases} -1 + a_1 + a_p, & k=1, \\ -g_{k-1}^{(1)} + a_k + a_{p+1-k}, & k=2, \dots, M, \\ g_{p-k}^{(1)}, & k=M+1, \dots, p-1; \end{cases} \quad (6)$$

- полином  $G_2(z)$ :

$$g_k^{(2)} = \begin{cases} 1 + a_1 - a_p, & k=1, \\ g_{k-1}^{(2)} + a_k - a_{p+1-k}, & k=2, \dots, M, \\ g_{p-k}^{(2)}, & k=M+1, \dots, p-1. \end{cases} \quad (7)$$

Можно показать [16, 17], что полиномы  $G_1(z)$  и  $G_2(z)$  обладают рядом следующих важных свойств.

1. Все их корни расположены на единичной окружности, то есть имеют вид  $z_k = e^{i\omega_k}$ . Аргументы корней полиномов  $G_1(z)$  и  $G_2(z)$ , лежащие в диапазоне  $(0; \pi)$ , называются *линейными спектральными частотами* (ЛСЧ)  $\omega_k$ , соответствующими полиному (2). Количество ЛСЧ в точности равно  $p$ , поскольку каждый из многочленов (4) имеет  $p=2M$  корней на единичной окружности, ровно половина из которых лежит в верхней полуплоскости.

2. Если обозначить как  $\omega_1^{(1)}, \omega_2^{(1)}, \dots, \omega_M^{(1)}$  ЛСЧ, соответствующие  $G_1(z)$ , а  $\omega_1^{(2)}, \omega_2^{(2)}, \dots, \omega_M^{(2)}$  – ЛСЧ, соответствующие  $G_2(z)$ , то оказывается, что они перемежаются:

$$\omega_1^{(1)} < \omega_1^{(2)} < \omega_2^{(1)} < \omega_2^{(2)} < \dots < \omega_M^{(1)} < \omega_M^{(2)} \quad (8)$$

(отметим, что все неравенства здесь строгие, т. е. не существует ЛСЧ, совпадающих между собой). В дальнейшем мы будем именовать это свойство упорядоченностью ЛСЧ.

3. Одновременное выполнение первых двух свойств является необходимым и достаточным условием того, что корни многочлена  $A(z)$  находятся внутри круга единичного радиуса. Таким образом, если выбранная схема квантования ЛСЧ сохраняет первые два свойства, то этим гарантируется устойчивость синтезирующего фильтра  $1/A(z)$ .

Для получения компактных уравнений относительно ЛСЧ преобразуем многочлены (4), заменив в них  $z$  на  $e^{i\omega_k}$ :

$$G_m(z) = \sum_{k=0}^{2M} g_k^{(m)} z^{-k} = e^{-iM\omega} \times \left[ \sum_{k=0}^M g_k^{(m)} (e^{i(M-k)\omega} + e^{-i(M-k)\omega}) + g_M^{(m)} \right] = e^{-iM\omega} \left[ \sum_{k=0}^M 2g_k^{(m)} \cos(M-k)\omega + g_M^{(m)} \right] = e^{-iM\omega} \sum_{k=0}^M r_k^{(m)} \cos k\omega, \quad m=1, 2, \quad (9)$$

где

$$r_k^{(m)} = \begin{cases} g_M^{(m)}, & k=0, \\ 2g_{M-k}^{(m)}, & k=1, \dots, M. \end{cases} \quad (10)$$

Таким образом, задача поиска ЛСЧ сводится к решению относительно  $\omega$  двух уравнений вида

$$\sum_{k=0}^M r_k^{(m)} \cos(k\omega) = 0, \quad m = 1, 2, \quad (11)$$

где коэффициенты  $r_k^{(m)}$  связаны с параметрами исходной АР модели соотношениями (6), (7), (10).

Трансцендентное уравнение (11) может быть преобразовано к полиномиальному путем замены  $x = \cos \omega$ . Значения для кратного аргумента ( $\cos n\omega$ ) могут быть выражены с помощью чебышевских полиномов

$$T_n(x) = \cos(n \arccos(x)) = \sum_{k=0}^n t_k^{(n)} x^{n-k}.$$

Известно [18], что полиномы Чебышева могут быть рассчитаны по рекуррентным соотношениям

$$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x), \quad (12)$$

$$k = 2, 3, \dots :$$

$$T_0(x) = 1, \quad T_1(x) = x,$$

$$T_2(x) = 2x^2 - 1, \quad T_3(x) = 4x^3 - 3x,$$

$$T_4(x) = 8x^4 - 8x^2 + 1 \dots$$

Таким образом, уравнение (11) может быть записано в виде

$$r_0^{(m)} + \sum_{k=1}^M r_k^{(m)} T_k(x) = 0, \quad m = 1, 2. \quad (13)$$

С учетом рекуррентных соотношений (12) преобразуем уравнение (13) к виду

$$\sum_{k=0}^M r_k^{(m)} x^{M-k} = 0, \quad m = 1, 2. \quad (14)$$

После выполнения описанных преобразований получаем пару вещественных полиномов, корни которых:

- 1) различны;
- 2) перемежаются между собой;
- 3) действительны и лежат в диапазоне  $[-1; 1]$ .

Если определены корни уравнения (14) –  $x_k^{(1)}$ ,  $k=1, \dots, M$ , соответствующие  $F_s(z)$ , и  $x_k^{(2)}$ ,  $k=1, \dots, M$ , соответствующие  $F_a(z)$ , – то ЛСЧ могут быть вычислены с помощью обратного нелинейного преобразования  $\omega = \arccos x$ .

Таким образом, проиллюстрировано наличие двух следующих подходов к определению ЛСЧ.

1. **Прямое решение уравнения (11).** В рамках существующих методов решения данной задачи отрезки, содержащие единственный корень, определяются путем вычисления функции

$$R(e^{i\omega}) = r_0 + \sum_{k=1}^M r_k \cos k\omega$$

на большой сетке частот с помощью быстрого преобразования Фурье [15]. Такие методы имеют очевидный недостаток – заранее неизвестен размер сетки, которую необходимо взять, поскольку нули функции  $R(e^{i\omega})$  могут находиться сколь угодно близко друг к другу. Слишком большой же размер сетки приводит к существенному увеличению вычислительных затрат.

2. **Решение полиномиального уравнения (14) с последующим применением обратного нелинейного преобразования для возврата в частотную область.** Существующие методы решения полученных полиномиальных уравнений [7, 16, 17] можно разделить на две подгруппы.

Типичным представителем первой из них является метод Кабала – Рамачандрана [7]. Как и в случае уравнения (11), отрезки, содержащие единственный корень, выявляются путем вычисления значений полинома на большой (но фиксированной) сетке значений. Такая сетка вычисляется путем анализа ЛСЧ у достаточно большого количества дикторов. В статье [7] обсуждаются предельные значения погрешностей этого метода. К недостаткам такого подхода следует отнести необходимость исследования большого тестового объема речевых данных для формирования эффективных сеток для каждого отдельно взятого порядка АР модели. Кроме того, следует ожидать ухудшения производительности при работе с дикторами, которые не участвовали в исследованиях по необходимому размеру сетки.

Еще одна подгруппа методов [16, 17] с применением чебышевских полиномов использует методологию последовательного понижения степени. Определяется наибольший корень полинома, на который он затем делится (с помощью, например, модифицированной схемы Горнера [17]). У полученного многочлена, степень которого меньше исходной на единицу, опять определяется наибольший корень, на который затем делится многочлен. Так продолжается до получения многочлена первой степени. Таким образом находятся

все корни исходного полинома, которые затем переводятся с помощью нелинейного преобразования в частотную область. Зачастую, данная итерационная процедура может быть остановлена после получения многочлена четвертой степени [17], после чего применяются формулы Феррари для точного определения корней [18]. Общей проблемой указанного подхода является то, что численные ошибки, вводимые при делении полиномов на приближенные значения корней, существенно сказываются на точности определения последующих корней. Такое накопление ошибок особенно существенно при работе с полиномами высоких порядков [15], что может привести к неустойчивости используемых АР фильтров [16].

Как видно из вышеизложенного, существующие методы поиска ЛСЧ обладают рядом недостатков. Основной проблемой “сеточных методов” является выбор необходимого размера сетки. При использовании методов последовательного понижения степени накапливающаяся погрешность определения ЛСЧ может приводить к неустойчивости соответствующего АР фильтра, что сказывается на качестве и разборчивости восстанавливаемого речевого сигнала. Вероятность таких сбоев повышается при возрастании порядка используемой АР модели [15].

В следующем разделе предложен универсальный метод решения трансцендентных уравнений, который затем применен к решению уравнений (11) и (14). Кроме того, показано, что свойство упорядоченности ЛСЧ (8) может быть эффективно использовано для сокращения количества вычислений при их поиске.

Отметим, что обратное преобразование от ЛСЧ к АР коэффициентам является существенно более простой процедурой. Прямой способ решения этой задачи – восстановить коэффициенты  $r'_k$  по корням уравнения (14) и произвести действия, обратные формулам (6), (7) (10). Быстрый и эффективный способ проведения этой процедуры представлен в работе [7].

## 2. УНИВЕРСАЛЬНЫЙ МЕТОД РЕШЕНИЯ ТРАНСЦЕНДЕНТНЫХ УРАВНЕНИЙ

В данном разделе предлагается метод полного численного решения произвольных трансцендентных уравнений вида  $f(x) = 0$ . Единственным его ограничением является непрерывная дифференцируемость функции  $f(x)$  и отсутствие у нее кратных корней. Данный алгоритм всегда гарантирует сходимость ко всем корням трансцендентного уравнения и не требует априорной информации об

их расположении. Кроме того, погрешности определения корней независимы друг от друга.

### 2.1. Вспомогательные утверждения

Прежде чем перейти к изложению сути алгоритма, рассмотрим ряд вспомогательных утверждений.

Нас интересует нахождение всех корней уравнения  $f(x) = 0$  на отрезке  $X = [a; b]$ . Предположим, что известна константа  $M_j$ , ограничивающая модуль  $j$ -ой производной данной функции:

$$\sup_{x \in [a; b]} f(x) \leq M_j. \quad (15)$$

Рассмотрим конкретные примеры.

1. Для уравнения (11) ограничение  $j$ -го порядка имеет вид

$$M_j = \sum_{k=0}^M k^j |r_k|. \quad (16)$$

2. Очевидно, для полинома (14) ограничения на отрезке  $[-1; 1]$  могут быть выбраны как

$$M_j = \sum_{k=0}^{M-j} (M-k)(M-k-1) \times \dots \times (M-k-j+1) |r'_k|. \quad (17)$$

Рассмотрим два простых утверждения.

*Лемма 1.* Если на отрезке  $[c; d] \subset X$  для некоторого  $j$  выполняется неравенство

$$f[0.5(c+d)] + \sum_{k=1}^{j-1} \frac{|f^{(k)}[0.5(c+d)]|(d-c)^k}{2^k k!} + \frac{M_j (d-c)^j}{2^j j!} < 0$$

или

$$f[0.5(c+d)] - \sum_{k=1}^{j-1} \frac{|f^{(k)}[0.5(c+d)]|(d-c)^k}{2^k k!} - \frac{M_j (d-c)^j}{2^j j!} > 0,$$

то функция  $f(x)$  не имеет корней на  $[c; d]$ .

*Лемма 2.* Если на отрезке  $[c; d] \subset X$  для некоторого  $j$  выполняется неравенство

$$f'[0.5(c+d)] + \sum_{k=2}^{j-1} \frac{|f^{(k)}[0.5(c+d)]|(d-c)^{k-1}}{2^{k-1} (k-1)!} + \frac{M_j (d-c)^{j-1}}{2^{j-1} (j-1)!} < 0$$

или

$$f'[0.5(c+d)] - \sum_{k=2}^{j-1} \frac{|f^{(k)}[0.5(c+d)]|(d-c)^{k-1}}{2^{k-1}(k-1)!} - \frac{M_j(d-c)^{j-1}}{2^{j-1}(j-1)!} > 0,$$

то функция  $f'(x)$  не имеет корней на  $[c; d]$ .

Доказательство лемм дано в приложении.

С помощью приведенных лемм всегда можно проанализировать поведение функций  $f(x)$  и  $f'(x)$  на произвольном отрезке  $[c; d]$ . При этом для функции  $f(x)$  возможны две ситуации:

- 1)  $f(x)$  не имеет корней на  $[c; d]$ ;
- 2) однозначного ответа о знакопостоянстве  $f(x)$  дать нельзя, т. е. ситуация является неопределенной.

Аналогично имеем две возможных ситуации для производной  $f'(x)$ :

- 1)  $f'(x)$  не имеет корней на  $[c; d]$ , т. е. функция  $f(x)$  монотонна на  $[c; d]$ ,
- 2) ситуация является неопределенной.

Число  $j$ , фигурирующее в условиях лемм 1 и 2, в дальнейшем будем называть порядком аппроксимации. Его выбор зависит от специфики решаемой задачи и, как будет показано далее, является решающим фактором, влияющим на величину вычислительных затрат алгоритма. В данной работе мы ограничимся рассмотрением случая, когда порядки аппроксимации функции и производной совпадают.

### 2.2. Описание алгоритма

С помощью лемм, сформулированных в предыдущем разделе, проанализируем поведение функций  $f(x)$  и  $f'(x)$  на исходном отрезке  $[a; b]$ . Возможны следующие случаи.

1.  $f(x)$  не имеет корней на  $[a; b]$ .
2. Ситуация для  $f(x)$  является неопределенной, но  $f'(x)$  не имеет корней на  $[a; b]$ , то есть  $f(x)$  монотонна на  $[a; b]$ . Тогда наличие корня на отрезке можно определить по знаку выражения  $f(a)f(b)$ . Если  $f(a)f(b) > 0$ , то корней на отрезке  $[a; b]$  нет. В противном случае  $f(x)$  имеет один корень на данном интервале, который может быть определен любым методом поиска единственного корня функции на отрезке.

3. Если ситуация является неопределенной и в случае с  $f(x)$ , и в случае с  $f'(x)$ , то описанные в предыдущем пункте действия выполняются для отрезков  $[a; 0.5(a+b)]$ ,  $[0.5(a+b); b]$  и т. д.

В силу предположения об отсутствии кратных корней (в реальных ситуациях это практически всегда верно), на некотором этапе разбиения не окажется отрезков, где одновременно для функции и ее производной ситуация является неопределенной. Таким образом, рекурсивное разбиение завершится, и будут выделены все отрезки, содержащие по одному корню функции  $f(x)$ . На каждом из таких отрезков точное значение корня определяется с помощью одной из стандартных итерационных процедур [18].

Отметим, что уточнение корня методами, использующими деление на производную функции (например, Ньютона и Ньютона–Рафсона), является гарантированно “безопасным”, поскольку разработанный алгоритм обеспечивает знакопостоянство производной на каждом из выделенных отрезков. Кроме того, наличие вычисленных производных в срединной точке отрезка позволяет без дополнительных затрат эффективно выбрать начальное приближение.

Блок-схема, соответствующая данной процедуре, приведена на рис. 2. Алгоритму решения соответствует дерево конечной длины, в конечных пунктах которого находятся отрезки, для которых ситуации для  $f(x)$  и  $f'(x)$  не являются одновременно неопределенными. Важнейшая характеристика данного алгоритма, указывающая на скорость его работы – количество рекурсивных разбиений исходного отрезка. Эта величина существенно зависит от порядков аппроксимации, выбранных для функции и ее производной.

Существенная особенность выбранного метода состоит в том, что с его помощью находится последовательность корней трансцендентного уравнения в строго монотонном порядке (возрастающим или убывающим – в зависимости от порядка выбора отрезков, получающихся в результате половинного деления). Кроме того, точности определения корней не зависят друг от друга и поэтому могут регулироваться, в зависимости от специфики решаемой задачи. Это означает, что набор ЛСЧ, определяемый с помощью данного алгоритма, является монотонным и, согласно свойству 3 для ЛСЧ из раздела 1, всегда соответствует устойчивому АР фильтру.

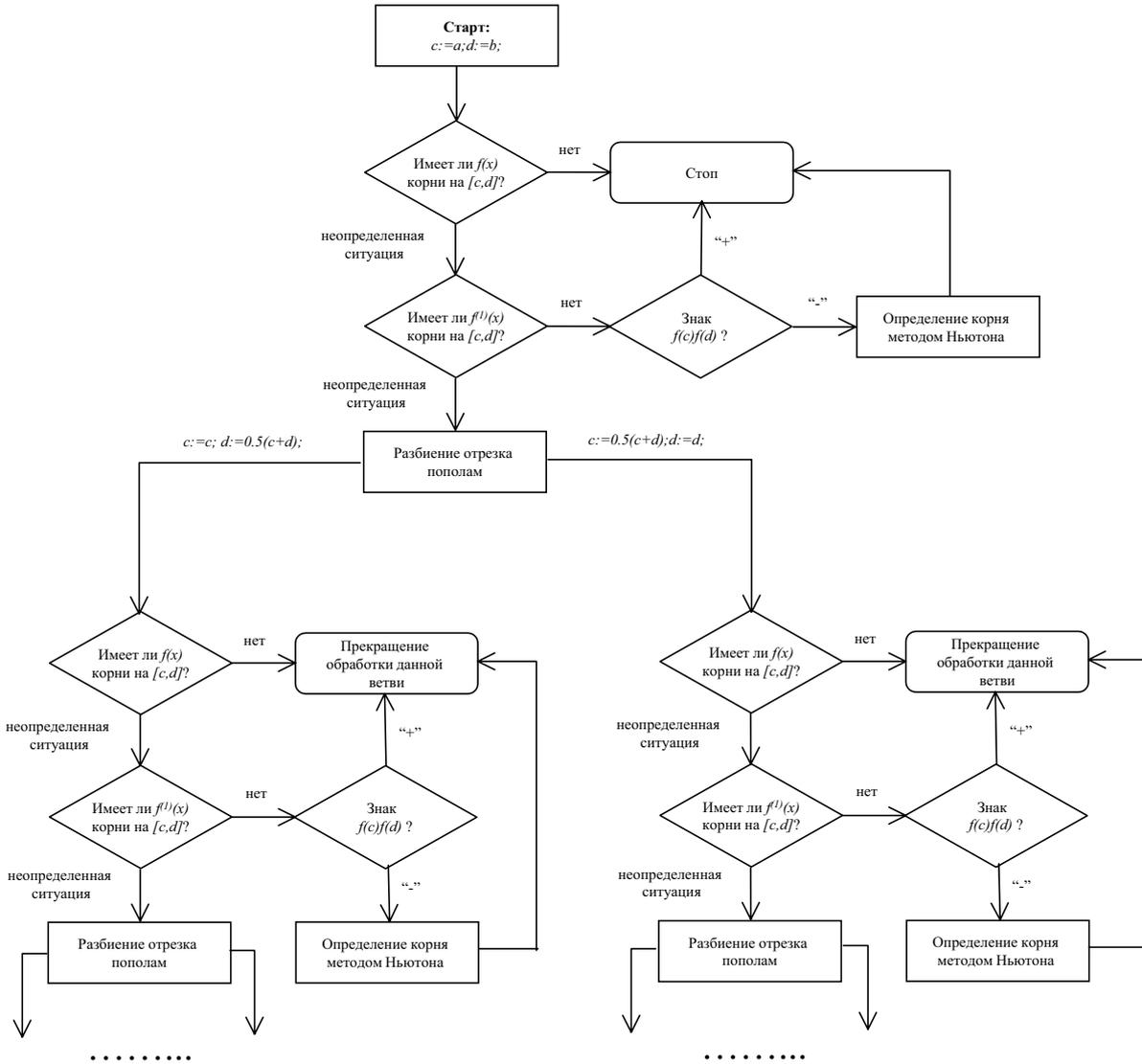


Рис. 2. Блок-схема алгоритма

### 3. ЭКСПЕРИМЕНТАЛЬНЫЕ РЕЗУЛЬТАТЫ

В предыдущих разделах показано, что, применительно к задаче поиска ЛСЧ, разработанный метод решения трансцендентных уравнений обеспечивает произвольно высокую точность вычислений и гарантирует устойчивость АР фильтра. Теперь необходимо проверить эффективность предложенного подхода при работе с реальными речевыми сигналами. При этом будут исследованы модификации данного метода, основанные как на решении уравнений (11), так и на решении полиномиальных уравнений (14).

Эффективность различных модификаций пре-

дложенного метода была оценена для различных четных порядков АР моделей: от 8 до 20. В качестве тестовых сигналов использовались пятиминутные записи шести дикторов (четырёх мужчин и двух женщин), дискретизированные с частотой  $f_s = 8000$  Гц. Для каждого из исследуемых порядков АР модели выполнялась следующая последовательность действий. Речевой сигнал разбивался на отрезки длиной 20 мс (160 дискретных отсчетов). На каждом из них предварительно были определены АР коэффициенты с помощью АКМ. Для каждого из этих наборов затем подсчитывались ЛСЧ. При этом исследовались среднее число элементарных арифметических операций для под-

Табл. 1. Среднее количество операций в секунду, необходимое для вычисления ЛСЧ посредством решения уравнения (11)

Порядок аппроксимации	$p = 8$	$p = 10$	$p = 12$	$p = 14$	$p = 16$	$p = 18$	$p = 20$
2	<b>129160</b>	<b>190820</b>	<b>271390</b>	<b>372020</b>	507860	622480	765620
3	138260	196260	280080	372870	<b>501260</b>	<b>599220</b>	<b>734310</b>
4	159640	223370	317220	415550	556710	661580	808720
5	185620	254840	361660	473510	627930	738850	899740

Табл. 2. Среднее рекурсивных разбиений, необходимое для вычисления ЛСЧ путем решения уравнения (11)

Порядок аппроксимации	$p = 8$	$p = 10$	$p = 12$	$p = 14$	$p = 16$	$p = 18$	$p = 20$
2	13.21	16.33	20.33	25.29	32.09	35.45	40.11
3	10.87	12.57	15.69	18.49	23.16	24.44	27.70
4	10.39	11.98	14.83	17.12	21.24	22.42	25.36
5	10.25	11.68	14.54	16.86	20.55	21.53	24.27

счета всех ЛСЧ и среднее количество рекурсивных разбиений исходного отрезка.

### 3.1. Определение ЛСЧ с помощью решения уравнения (11)

Каждый набор АР параметров  $a_k$ ,  $k=1, 2, \dots, p$  с помощью соотношений (7), (10) был преобразован в коэффициенты  $r_k^{(2)}$ ,  $k=0, \dots, M$ . Для решения уравнения (11) на отрезке  $[0; \pi]$  в соответствии с алгоритмом, описанным в разделе 2, были рассмотрены различные порядки аппроксимации функции и ее производной. При порядке аппроксимации, равном  $j$ , ограничение для модуля  $j$ -ой производной определялось формулой (16). После того, как определены отрезки, содержащие единственный корень функции

$$f(x) = r_0^{(2)} + \sum_{k=1}^M r_k^{(2)} \cos kx,$$

точное значение корня находится с помощью итерационного метода Ньютона [18]. В качестве критерия остановки алгоритма выбрано условие  $|f(x)| < 10^{-6}$ . Таким образом, определяются корни второго уравнения (11), которые являются линейными спектральными частотами  $\omega_2, \omega_4, \dots, \omega_p$ . После этого по формулам (6) и (10) вычисляются коэффициенты уравнения (11), соответствующие полиному  $F_s(z)$ . Заметим, что выполнять поиск отрезков, содержащих единственный корень, нет

необходимости, поскольку таковыми являются интервалы  $[0; \omega_2]$ ,  $[\omega_2; \omega_4]$ ,  $\dots$ ,  $[\omega_{p-4}; \omega_{p-2}]$ ,  $[\omega_{p-2}; \omega_p]$  (в силу свойства упорядоченности (8)). Значения оставшихся частот  $\omega_1, \omega_3, \dots, \omega_{p-1}$  определялись методом Ньютона. В том случае, если на некоторой итерации процедура выходила за пределы исходного интервала (такая ситуация встречалась очень редко), точное значение корня определялось методом ложного положения (результат работы которого на каждой итерации гарантированно остается в пределах анализируемого интервала). Следует отметить, что описанная локализация ЛСЧ с помощью полинома  $F_a(z)$  оказалась более эффективной с точки зрения вычислительных затрат, по сравнению с полиномом  $F_s(z)$ . Это может быть объяснено различным характером распределения ЛСЧ, соответствующих  $F_s(z)$  и  $F_a(z)$  [7, 17].

Указанная процедура выполнялась на всех фреймах тестового речевого сигнала. Средние значения количества операций в секунду приведены в табл. 1, а средние значения количества рекурсивных разбиений исходного отрезка  $[0; \pi]$  – в табл. 2. Для каждого порядка АР модели  $p$  рассмотрены порядки аппроксимации от 2 до 5.

Как видно из табл. 1, оптимальный порядок аппроксимации для значений  $p=8, 10, 12, 14$  равен 2. Для  $p=16, 18, 20$  минимум вычислительных затрат достигается при аппроксимации третьего порядка. Заметим, однако, что хотя дальнейшее повышение порядка аппроксимации влечет за собой

Табл. 3. Среднее количество операций в секунду, необходимое для вычисления ЛСЧ путем решения уравнения (14)

Порядок аппроксимации	$p = 8$	$p = 10$	$p = 12$	$p = 14$	$p = 16$	$p = 18$	$p = 20$
2	71100	166250	337520	655520	1238870	2297890	4240270
3	<b>46340</b>	87490	152220	256110	401240	648310	997360
4	48850	<b>75390</b>	127090	177250	267710	384650	538950
5		79490	<b>124250</b>	<b>172030</b>	<b>235580</b>	<b>314110</b>	422870
6			131110	177210	241920	315440	<b>381240</b>
7				185890	254220	329020	401640
8					265320	344520	422130
9						357580	440160
10							454550

Табл. 4. Сравнение предложенного алгоритма поиска ЛСЧ с методом Кабала–Рамачандрана

Метод	Локализация корней	$\varepsilon = 10^{-3}$	$\varepsilon = 10^{-6}$
Предложенный алгоритм + метод Ньютона	<b>39750</b>	<b>64550</b>	<b>76550</b>
Алгоритм Кабала–Рамачандрана	61050	107500	177000
Алгоритм Кабала–Рамачандрана + метод Ньютона	61050	74980	87420

уменьшение количества разбиений (см. табл. 2), вычислительные затраты при этом начинают медленно возрастать. Это объясняется увеличением количества вычислений в формулах, фигурирующих в условиях лемм 1 и 2.

При наиболее распространенном порядке АР модели  $p = 10$  максимальное зафиксированное число разбиений составило 41, а порог в 25 рекурсивных разбиений превышен лишь в 2.7 % всех случаев. Следовательно, применение данного алгоритма в приложениях реального времени не должно вносить дополнительных задержек.

Из табл. 1 видно, что необходимое количество элементарных операций в секунду составляет от 129160 при  $p = 8$  до 734310 при  $p = 20$ . Таким образом, можно утверждать, что с точки зрения возможностей современных средств сигнальной обработки реализация данного подхода не представляет сложностей. При этом следует помнить, что вычисление ЛСЧ – наиболее ресурсоемкий этап работы большинства современных низкоскоростных речепреобразующих устройств.

Отметим, что при подсчете среднего количества элементарных операций один вызов тригонометрической функции считался как одна элементарная операция. Это объясняется тем, что на стадии выделения отрезков, содержащих единствен-

ный корень, значения функций  $\cos x$  и  $\sin x$  могут браться из фиксированной таблицы (при реализации алгоритма на цифровом сигнальном процессоре операции обращения к памяти и сложения/умножения имеют примерно одинаковый вес). Необходимый размер такой таблицы составляет примерно  $p \sup\{N_d\}$ , где  $\sup\{N_d\}$  – величина, ограничивающая сверху количество разбиений. Отметим, что максимальное число разбиений, зафиксированное для  $p = 20$ , составило 42, т.е. достаточным оказался размер единой для всех порядков таблицы, равный 1000. На стадии же точного вычисления корней количество вызовов тригонометрических функций очень невелико. Например, для  $p = 10$  его среднее значение равно всего 9980 в течение одной секунды, что составляет 5.2 % от общего количества выполненных операций. Таким образом, данный алгоритм выгодно отличается от традиционных методов численного решения уравнения (11), основным недостатком которых являлось именно большое количество вычислений тригонометрических функций [15].

Перечисленные факторы свидетельствуют о том, что алгоритм, основанный на прямом решении уравнения (11) с помощью предложенного метода решения трансцендентных уравнений, может быть применен в системах реального времени.

### 3.2. Определение ЛСЧ с помощью решения полиномиального уравнения (14)

На каждом фрейме тестового речевого сигнала набор АР параметров  $a_k$ ,  $k=1, 2, \dots, p$  преобразовывался с помощью соотношений (7), (10), (12) в коэффициенты  $r_k^{(2)}$  уравнения (14), соответствующего полиному  $F_a(z)$ . Для решения уравнения (14) на отрезке  $[-1; 1]$  использовались ограничения (17). После локализации отрезков, содержащих корни функции

$$f(x) = \sum_{k=0}^M r_k^{(m)} x^{M-k},$$

точные значения корней определялись точно так же, как при решении уравнения (11). В качестве критерия останова алгоритма было выбрано условие  $|f(x)| < 10^{-6}$ . После нахождения всех корней двух уравнений (14), соответствующих  $F_a(z)$  и  $F_s(z)$ , ЛСЧ вычислялись с помощью обратного нелинейного преобразования  $\omega = \arccos x$ .

Средние оценки количества элементарных операций в секунду представлены в табл. 3. Для каждого порядка АР модели  $p$  рассмотрены порядки аппроксимации от 2 до  $p/2$  (производные более высоких порядков здесь равны нулю).

Из табл. 3 видно, что минимум вычислительных затрат для каждого порядка АР модели достигается при разных порядках аппроксимации. Оптимальный порядок аппроксимации монотонно ступенчато возрастает по мере увеличения числа  $p$ . Оптимальное количество элементарных операций, выполняемых для вычисления ЛСЧ на протяжении одной секунды речевого сигнала, составляет от 46340 при  $p=8$  до 381240 при  $p=20$ . Из приведенных данных видно, что алгоритм, основанный на решении уравнения (14), может быть применен в системах реального времени.

Сравнение табл. 1 и 3 показывает, что при  $p=20$  и 8 вычислительные затраты при решении уравнений (14) для соответствующих оптимальных порядков аппроксимации примерно в 1.9 и 2.8 раза меньше, чем в случае применения подхода, основанного на решении (11). Кроме того, преимуществом “полиномиального” подхода является отсутствие вызовов тригонометрических функций.

Теперь сопоставим эффективность предложенного алгоритма с сеточным методом Кабала–Рамачандрана [7] при определении ЛСЧ для порядка АР модели  $p=10$  (именно этот порядок чаще всего используется в алгоритмах средне- и низкоскоростного сжатия речи). Метод Кабала–Рамачандрана является одним из наиболее рас-

пространственных способов поиска ЛСЧ и используется в современных стандартах кодирования речи [11]. Заметим, что его применение требует задания шага сетки, равного минимально возможному расстоянию между корнями уравнений (14). В качестве этого параметра нами было выбрано рекомендуемое для порядка  $p=10$  значение  $\delta=0.02$  [7]. Если два корня одного из полиномов (14) расположены ближе, чем на величину  $\delta$ , алгоритм Кабала–Рамачандрана оказывается неприменимым (очевидно, что уменьшение  $\delta$  приводит к увеличению вычислительных затрат).

Для объективности сравнения необходимо перейти к критерию завершения поиска корней  $|x_k - x_{k-1}| < \varepsilon$ , где  $x_k, x_{k-1}$  – приближенные значения корня, полученные на двух последовательных итерациях. В табл. 4 представлены средние количества операций, необходимых для реализации метода Кабала–Рамачандрана и метода, предлагаемого в данной статье. Поскольку в алгоритме Кабала–Рамачандрана используется уточнение корней с помощью метода половинного деления (для фиксированности количества операций), то рассмотрена также его модификация, соответствующая уточнению корней методом Ньютона (при значениях порога  $\varepsilon$ , равных  $10^{-3}$  и  $10^{-6}$ ), а также вычислительные затраты, необходимые для локализации отрезков, содержащих единственный корень.

Среднее количество операций для локализации корней в методе Кабала–Рамачандрана, равное 61050, объясняется следующим образом. Необходимое количество вызовов полиномиальной функции составляет  $2/\delta + p + 1 = 111$  [7]. Вычисление полинома 5-го порядка требует выполнение 11-ти элементарных операций. Таким образом, на каждом фрейме локализация корней занимает  $111 \times 11 = 1221$  операций. Поскольку на каждом фрейме количество вычислений, требуемое для работы алгоритма Кабала–Рамачандрана, одинаково (это его главное свойство), то на одну секунду приходится  $1221 \times 50 = 61050$  операций.

Из табл. 4 видно, что преимущество предложенного подхода к поиску ЛСЧ перед методом Кабала–Рамачандрана составляет 35 % операций при локализации корней, а также 40 % и 57 % операций при точностях вычисления корней  $\varepsilon = 10^{-3}$  и  $10^{-6}$  соответственно. Также имеет место выигрыш перед комбинацией алгоритма Кабала–Рамачандрана и метода Ньютона. Заметим, однако, что для практической реализации метода в системах реального времени важно также знать максимальное количество требуемых операций в единицу времени (в этом смысле алгоритм Кабала–

Рамачандрана всегда имел преимущество перед другими методами, так как его вычислительная нагрузка фиксирована). Поэтому при тестировании предложенного нами метода были установлены максимальные количества вычислений, приходящиеся на один фрейм, для локализации корней и их вычисления с точностями  $\varepsilon = 10^{-3}$  и  $10^{-6}$ . Эти предельные значения составили 1206, 1798 и 2200 соответственно, что меньше аналогичных величин для алгоритма Кабала – Рамачандрана (1221, 2150 и 3540).

Принципиальный недостаток большинства существующих подходов к задачам, требующим решения уравнений (полиномиальных или трансцендентных), состоит в использовании *фиксированных равномерных* множеств точек, в которых анализируется поведение функции. Преимущество же разработанного в данной работе метода объясняется тем, что он использует *неравномерную* сетку точек, подбираемую *адаптивно* под анализируемую функцию (размер сетки определяется количеством рекурсивных разбиений области определения функции).

## ЗАКЛЮЧЕНИЕ

В статье рассмотрена проблема поиска линейных спектральных частот, являющихся альтернативным представлением авторегрессионной модели.

Для решения этой задачи разработан универсальный метод полного численного определения всех корней произвольного трансцендентного уравнения. Принципиально данный метод состоит из двух частей – выделения отрезков, содержащих единственный корень, и последующего нахождения корней с помощью любого стандартного итерационного алгоритма. Основными параметрами предложенного метода являются порядки аппроксимации функции и ее производной. Важным показателем эффективности работы метода, тесно связанным с его вычислительными затратами, служит количество рекурсивных разбиений исходного отрезка. Данный алгоритм всегда гарантирует сходимость к корням трансцендентного уравнения и не требует никакой априорной информации об их расположении. Точность определения каждого из корней не зависит друг от друга и может регулироваться, в зависимости от специфики решаемой задачи.

Продемонстрировано применение различных модификаций данного метода решения трансцендентных уравнений к задаче поиска ЛСЧ. При этом исследованы два традиционных подхода – ре-

шение трансцендентного уравнения относительно тригонометрических функций и решение полиномиального уравнения с последующим нелинейным преобразованием его корней в ЛСЧ. Показано, что данный метод обеспечивает произвольно высокую точность определения ЛСЧ и гарантирует устойчивость соответствующего им АР фильтра.

В результате экспериментов, проведенных на реальных речевых сигналах, установлены оптимальные порядки аппроксимации для каждого из подходов и их зависимости от порядков используемых АР моделей. Для сокращения вычислительных затрат было эффективно использовано свойство упорядоченности ЛСЧ. Установлено, что вычислительные затраты при “полиномиальном” подходе примерно в  $2 \div 2.5$  раза меньше затрат, чем при “тригонометрическом”. Тем не менее “тригонометрический” подход характеризуется очень малым количеством вызовов тригонометрических функций, что всегда являлось существенным недостатком методов этого типа. Продемонстрировано сокращение вычислений, по сравнению с наиболее широко используемым в настоящее время методом Кабала – Рамачандрана.

Показано, что алгоритмы, развитые на основе обоих обсуждаемых подходов к поиску ЛСЧ, могут быть применены в системах реального времени.

## ПРИЛОЖЕНИЕ

### Доказательство леммы 1

Рассмотрим случай  $j=1$ . Согласно теореме Лагранжа [18], для любой точки  $x$  отрезка  $[c; d]$  выполняется следующее равенство:

$$f(x) = f(0.5(c+d)) + f'(\xi)[x - 0.5(c+d)]. \quad (18)$$

Из соотношения (18) следует, что

$$\begin{aligned} & f(0.5(c+d)) - \\ & - \sup_{\xi \in [c;d]} |f'(\xi)| \sup_{\xi \in [c;d]} |x - 0.5(c+d)| \leq \\ & \leq f(x) \leq f(0.5(c+d)) + \\ & + \sup_{\xi \in [c;d]} |f'(\xi)| \sup_{\xi \in [c;d]} |x - 0.5(c+d)|. \end{aligned} \quad (19)$$

Принимая во внимание, что

$$\sup_{\xi \in [c;d]} |x - 0.5(c+d)| = 0.5(d-c),$$

из неравенства (19) получаем

$$\begin{aligned} f(0.5(c+d)) - 0.5 \sup_{\xi \in [c;d]} |f'(\xi)|(d-c) &\leq \\ &\leq f(x) \leq f(0.5(c+d)) + \\ &+ 0.5 \sup_{\xi \in [c;d]} |f'(\xi)|(d-c). \end{aligned} \quad (20)$$

Сопоставив соотношение (20) с условиями леммы, имеем

$$\forall x \in [c; d] : f(x) < 0 \quad \text{или} \quad \forall x \in [c; d] : f(x) > 0,$$

что и требовалось доказать.

Отметим, что выбор в качестве “опорной” точки  $0.5(c+d)$  обусловлен тем, что для любой другой точки  $y \in [c; d]$  имеет место соотношение

$$\sup_{x \in [c;d]} |x - y| > 0.5(d - c),$$

то есть неравенство (20) стало бы более слабым.

Доказательство леммы в случае  $j \geq 2$  аналогично приведенному выше. При его проведении следует применить разложение функции  $f(x)$  в ряд Тейлора с записью остаточного члена в форме Лагранжа.

Доказательство леммы 2 полностью аналогично доказательству леммы 1.

1. Рабинер Л., Шафер Р. Цифровая обработка речевых сигналов.– М.: Радио и связь, 1981.– 496 с.
2. Маркел Дж., Грей А. Линейное предсказание речи.– М.: Связь, 1977.– С. 26–41.
3. Макхоул Дж., Русос С., Гиш Г. Векторное квантование при кодировании речи // ТИИЭР.– 1985.– 73.– С. 19–61.
4. Buzo A., Gray A. H., Gray R. M., Markel J. D. Speech coding based on vector quantization // IEEE Trans. Acoust. Speech Signal Proces.– 1980.– 28.– P. 562–574.
5. Калужный А. Я., Семенов В. Ю. Экономичный метод очистки речи от шума, основанный на блоч-

ном представлении сигнала в пространстве состояний и векторном квантовании // Акуст. вісн.– 2002.– 4, N 3.– С. 28–34.

6. Itakura F. Line spectrum representation of linear predictive coefficients of speech signals // J. Acoust. Soc. Amer.– 1975.– 57, N 1, Suppl. 1.– P. S35.
7. Kabal P., Ramachandran R. P. The computation of line spectral frequencies using Chebyshev polynomials // IEEE Trans. Acoust. Speech Signal Proces.– 1980.– 28.– P. 562–574.
8. Pan J., Fischer T. R. Vector quantization of speech line spectrum pair parameters and reflection coefficients // IEEE Trans. Speech Audio Proces.– 1998.– 6.– P. 106–115.
9. Campbell J., Welch V., Tremain T. An expandable error-protected 4800 bps CELP coder (US federal standard 4800 bps voice coder) // Proc. IEEE Int. Conf. Acoust. Speech Signal Proces.– Glasgow, 1989.– P. 735–738.
10. McCree A. V., Barnwell T. P. A mixed excitation LPC vocoder model for low bit rate speech coding // IEEE Trans. Speech Audio Proces.– 1995.– 4.– P. 242–249.
11. Salami R. Design and description of CS-ASELP: a toll quality 8 kb/s speech coder // IEEE Trans. Speech Audio Proces.– 1998.– 6.– P. 116–128.
12. Paliwal K. A study of line spectrum pair frequencies for speech recognition // Proc. IEEE Int. Conf. Acoust. Speech Signal Proces.– New York, USA, 1988.– P. 485–488.
13. Liu C., Lin M., Wang W., Wang H. A study of line spectrum pair frequencies for speaker recognition // Proc. IEEE Int. Conf. Acoust. Speech Signal Proces.– Albuquerque, USA, 1990.– P. 277–280.
14. Hansen J., Clements M. Constrained iterative speech enhancement with application to speech recognition // IEEE Trans. Acoust. Speech Signal Proces.– 1991.– 4.– P. 795–805.
15. Schmidt C. E., Rabiner L. R. A study of techniques for finding the zeros of linear phase FIR digital filters // IEEE Trans. Acoust. Speech Signal Proces.– 1977.– 25.– P. 96–98.
16. Rothweiler J. A rootfinding algorithm for line spectral frequencies // Proc. IEEE Int. Conf. Acoust. Speech Signal Proces.– Phoenix, USA, 1999.– P. 661–664.
17. Wu C.-H., Chen J.-H. A novel two-level method for the computation of LSP frequencies using a decimation-in-degree algorithm // IEEE Trans. Speech Audio Proces.– 1997.– 5.– P. 106–115.
18. Корн Г., Корн Т. Справочник по математике для научных работников и инженеров.– М.: Наука, 1977.– 832 с.