

УДК 534.78:681.142:62-501.2:518.62

ПРОГРАММА СИНТЕЗА РЕКУРСИВНЫХ
ЦИФРОВЫХ ФИЛЬТРОВ (РЦФ)

В.С. Лозовский

Пусть ставится задача моделирования на ЦВМ процесса прохождения некоторого сигнала через частотно-селективные цепи. По-видимому, к настоящему времени известны два метода, обеспечивающие минимум затрат времени на фильтрацию:

- метод, использующий вычисление спектра путем "быстрого преобразования Фурье" (алгоритм Кули-Тьюки);
- метод рекурсивных цифровых фильтров (РЦФ).

Первый основывается на вычислении текущего спектра процесса с последующим учетом характеристик фильтрующей системы, после которой подразумевается наличие детектора и фильтра низких частот. Второй метод позволяет, зная определенные параметры системы, непосредственно вычислять значение выходного сигнала для текущих дискретных отсчетов времени. Учитывая указанную особенность, а также то, что при не очень большом числе параллельно работающих фильтров даже "быстрое преобразование Фурье" оказывается громоздким по сравнению с методом РЦФ, ниже речь пойдет только о последнем.

Синтез РЦФ может быть осуществлен различными методами в зависимости от сопутствующих требований и ограничений ([1], [2] и др.). Ниже излагается метод синтеза, принятый в настоящей работе.*

Входной сигнал фильтра представляется дискретными отсчетами с заданным периодом квантования. Сигнал на выходе фильтра опре-

* Используемый метод получил название "Impulse invariant technique" [1].

деляется по следующей рекурсивной формуле:

$$y_t = a_0 x_t + a_1 x_{t-1} + \dots + a_n x_{t-n} - b_1 y_{t-1} - b_2 y_{t-2} - \dots - b_m y_{t-m} \quad (1)$$
 где y_t , y_{t-1} и т.д. - дискретные отсчеты сигнала на выходе фильтра в моменты времени t , $t-1$ и т.д.; x_t , x_{t-1} ... - отсчеты входного сигнала для соответствующих моментов времени; m - порядок фильтра; a_i и b_i - коэффициенты полиномов числителя и знаменателя передаточной функции фильтра в комплексной z -плоскости [3], [4]:

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n}}{1 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_m z^{-m}} \quad (2)$$

Под синтезом РЦФ понимается нахождение коэффициентов a_i и b_i , которыми, очевидно, полностью описывается поведение фильтра.

Используемый здесь подход подразумевает описание фильтрующей системы линейным разностным уравнением с постоянными коэффициентами, для нахождения которых применен хорошо известный аппарат z - преобразования [3], [4].

Прежде чем сосредоточить внимание на определении параметров передаточной функции системы $H(z)$, легко убедиться, что развернутое выражение для выходного сигнала $y_t(I)$ может быть получено из соотношения

$$y(z) = H(z) \cdot x(z) \quad (3)$$

где z имеет смысл переменной или оператора упорядочивания временных последовательностей:

$$H(z) = \sum_{n=0}^{\infty} h(nT) z^{-n} \quad (4)$$

С другой стороны, функция $h(t)$, квантованная по времени с периодом T , может быть записана в следующем виде:

$$h^*(t) = \sum_{n=0}^{\infty} h(nT) \cdot \delta'(t - nT),$$

где $\delta'(t)$ - дельта-функция.

Преобразование $h^*(t)$ по Лапласу имеет вид:

$$\mathcal{L}[h^*(t)] = H^*(s) = \sum_{n=0}^{\infty} h(nT) e^{-nTs} \quad (5)$$

где $s = \sigma + j\omega$ - "комплексная частота".

*) Таким образом, для получения одного отсчета выходного сигнала РЦФ необходимо выполнить $m + m + 1$ операций типа "умножение + сложение". Выигрыш в числе таких операций для РЦФ по сравнению со счетом, например по интегралу свертки, очевиден.

Между выражениями (5) и (4) существует сходство, которое переводится в тождество заменой в (5):

$$z = e^{Ts} \quad (7)$$

Указанное рассуждение приводит к выводу, что одним из возможных способов получения $H(z)$ является применение преобразования Лапласа к системной функции $h(nT)$ [1]. Записав системную функцию в виде:

$$h(nT) = \sum_{i=1}^m A_i e^{s_i nT} \quad (8)$$

и преобразовав (8) по Лапласу, имеем:

$$H(z) = \sum_{n=0}^{\infty} h(nT) z^{-n} = \sum_{i=1}^m A_i \sum_{n=0}^{\infty} e^{s_i nT} z^{-n}$$

Замечая, что в полученном выражении присутствует сумма членов геометрической прогрессии, получаем:

$$H(z) = \sum_{i=1}^m \frac{A_i}{1 - e^{s_i T} z^{-1}} \quad (9)$$

Для весьма обширной сферы применений, в том числе и для целей настоящей работы, в качестве исходной системной функции удобно использовать не её временной отклик $h(nT)$, а передаточную функцию системы в s -области: $H(s)$. При этом придется потребовать, чтобы $H(s)$ представляла собой отношение полиномов:

$$H(s) = \frac{Q(s)}{P(s)}$$

Предполагая наличие в $H(s)$ только простых полюсов s_i , применяем разложение на простые дроби:

$$H(s) = \frac{Q(s)}{P(s)} = \frac{Q(s)}{(s-s_1)(s-s_2)\dots(s-s_m)} = \frac{A_1}{s-s_1} + \dots + \frac{A_m}{s-s_m} = \sum_{i=1}^m \frac{A_i}{s-s_i} \quad (10)$$

где

$$A_i = \frac{Q(s_i)}{P'(s_i)}; \quad P'(s_i) = \left. \frac{dP}{ds} \right|_{s=s_i}$$

Применяя к (10) обратное преобразование Лапласа, получаем:

$$h(t) = \mathcal{L}^{-1} \left(\sum_{i=1}^m \frac{A_i}{s-s_i} \right) = \sum_{i=1}^m A_i e^{s_i t} \quad (11)$$

Потребовав от дискретной системы, чтобы значения её отклика в моменты времени nT равнялись соответствующим величинам для непрерывной системы, описываемой (11), приходим к выражению вида (8). А передаточная функция системы в z -плоскости окончательно будет иметь вид:

$$H(z) = \sum_{i=1}^m \frac{Q(s_i)}{P'(s_i)} \cdot \frac{1}{1 - e^{s_i T} z^{-1}} \quad (12)$$

Таким образом, возникает задача нахождения полюсно-нулевого описания передаточной функции фильтра $H(s)$. Такое описание может быть легко получено (например, [5]) для случаев аппроксимации модуля амплитудно-частотной функции фильтра низких частот по Баттерворту и по Чебышеву (рис. 1).

В первом случае будет получена максимально гладкая аппроксимация $|H(\omega)|$ с хорошей фазовой характеристикой; во втором случае аппроксимирующая функция характеризуется заданным максимальным отклонением от единицы (Δ) в полосе пропускания, большей скоростью затухания в полосе непрозрачности и несколько более неравномерной фазовой характеристикой.

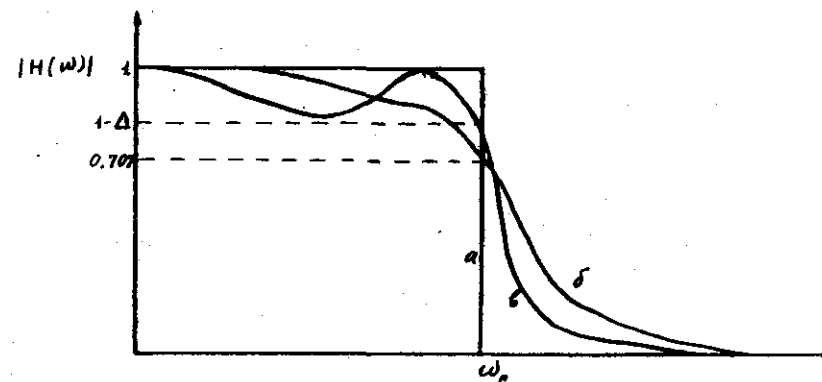


Рис. 1. Частотные характеристики фильтров: а — характеристика идеального ФНЧ; б — аппроксимация по Баттерворту; в — аппроксимация по Чебышеву.

Для фильтров Баттерворта:

$$|H(\omega)| = \frac{1}{\sqrt{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}}} \quad (13)$$

Для фильтров Чебышева:

$$|H(\omega)| = \frac{1}{\sqrt{1 + \delta^2 T_n^2(\omega)}} \quad (14)$$

Здесь n — порядок фильтра;

T_n - полином Чебышева порядка n ;

σ - коэффициент неравномерности; как нетрудно убедиться, σ и Δ (рис. 1) связаны соотношением:

$$\sigma^2 = \frac{1}{(1-\Delta)^2} - 1. \quad (15)$$

Найдем полюса $H(z)$, соответствующие (13). Поскольку $H^2(\omega) = H(j\omega) \times H(-j\omega)$, то, переходя в плоскость z : $j\omega = z$ и полагая $\omega_c = 1$, имеем:

$$H(z)H(-z) = \frac{1}{1+(-1)^n z^{2n}}. \quad (16)$$

Для нахождения полюсов (16), решаем уравнение:

$$1+(-1)^n z^{2n} = 0.$$

Имеем $2n$ корней:

$$z_i = \exp\left(\frac{j\pi(n-1+2i)}{2n}\right); i=1, \dots, 2n. \quad (17)$$

Для $H_b(z)$ используем полюса левой полуплоскости, что соответствует устойчивой минимально-фазовой системе (нули отсутствуют):

$$H(z) = \frac{1}{\prod_{i=1}^n (z - z_i)}, \quad (18)$$

где z_i определяются из (17).

Проведя аналогичные рассуждения [5], можно получить передаточную функцию $H_c(z)$ для ФНЧ с аппроксимацией по Чебышеву. $H_c(z)$ имеет вид (18), но z_i определяются иначе:

$$z_i = \sinh(\psi) \times \sin(\epsilon) + i \cosh(\psi) \times \cos(\epsilon), \quad (19)$$

где

$$\psi = \frac{1}{n} \operatorname{arcsinh}(1/\sigma);$$

$$\epsilon = \frac{\pi}{2n} (2n-1+2i); i=1, \dots, n.$$

Представляет интерес нахождение порядка фильтра заданного типа по ширине полосы на двух уровнях. Пусть $\sigma_1 > \sigma_2$ - два уровня, а δ_1 и δ_2 - соответствующие полосы. В том случае, когда $\sigma_1 \neq 1/\sqrt{2}$ для фильтров Баттерворта или $\sigma_1 \neq 1-\Delta$ для фильтров Чебышева, придется также определять частоту среза ω_c .

Для случая фильтра Баттерворта имеем:

$$\sigma_1 = \frac{1}{\sqrt{1+(\frac{\delta_1}{\omega_c})^{2n}}}; \quad \sigma = \frac{1}{\sqrt{1+(\frac{\delta_2}{\omega_c})^{2n}}};$$

Решая записанную систему уравнений относительно ω_c и n , находим:

$$\omega_c = \frac{\delta_1}{(\frac{1}{\sigma_1^2} - 1)^{\frac{1}{2n}}}; \quad (20)$$

$$n = \frac{1}{2} \frac{\ln\left(\frac{\frac{1}{\sigma_2^2} - 1}{\frac{1}{\sigma_1^2} - 1}\right)}{\ln\left(\frac{\delta_1}{\delta_2}\right)}. \quad (21)$$

Имея $H_b(z)$ или $H_c(z)$ для ФНЧ, путем преобразования частоты [5] можно получить передаточные функции для полосовых фильтров.*)

Имеем:

$$z_H = \frac{\omega_o}{\delta} \left(\frac{z_n}{\omega_o} + \frac{\omega_o}{z_n} \right); \quad (22)$$

здесь $\omega_o = \omega_{c1} \cdot \omega_{c2}$ - резонансная частота;

δ - ширина полосы на уровне $1/\sqrt{2}$;

ω_{c1} и ω_{c2} - верхняя и нижняя границы полосы пропускания на уровне $1/\sqrt{2}$; индексы n и N указывают на принадлежность переменной z к "полосовой" или "низкочастотной" областям. Для удобства задания параметров фильтра вводится дополнительные величины (рис. 2): ω_{cp} , ω_{n1} и $\omega_{n(-1)}$.

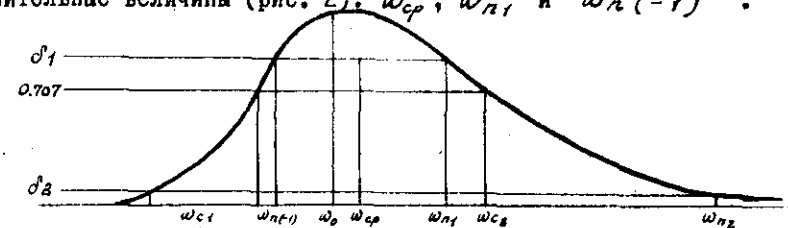


Рис. 2. Параметры задания характеристики ПФ.

При этом $\omega_{cp} = \frac{\omega_{n1} + \omega_{n(-1)}}{2}$; ширина полосы на уровне δ_1 : $\delta_1 =$

$\omega_{n1} - \omega_{n(-1)}$, а на уровне δ_2 : $\delta_2 = 2(\omega_{n2} - \omega_{cp})$.

Итак, пусть найдена $H(z)$ в виде (18) для ФНЧ. Перепишем (22) в виде:

$$z_H = \frac{z_n^2 + \omega_o^2}{z_n \cdot \delta} \quad (22a)$$

и подставим полученное выражение в (18). В этом случае каждый

*) Как легко обнаружить из рассмотрения (12), описываемый метод синтеза позволяет получить $H(z)$ в виде отношения полиномов, у которых степень числителя на единицу меньше степени знаменателя. Таким образом, хотя получить $H(z)$ для ФВЧ и заграждающих фильтров весьма просто, непосредственный перевод её в плоскость z по формуле (12) невозможен.

соотношение (18) примет вид:

$$\frac{1}{3_n - 3_i} \rightarrow \frac{3_n \cdot \theta}{3_n^2 - 3_n \theta 3_i + \omega_0^2}$$

Находим корни знаменателя:

$$3_i'^2 = \frac{\theta 3_i}{2} \pm \sqrt{\left(\frac{\theta 3_i}{2}\right)^2 - \omega_0^2}$$

Удобно нормировать величины:

$$3_n' = \frac{3_n}{\omega_0}; \quad \theta' = \frac{\theta}{\omega_0} \quad (23)$$

тогда нормированные корни примут вид:

$$(3_i'^2)' = \frac{\theta' 3_i'}{2} \pm \sqrt{\left(\frac{\theta' 3_i'}{2}\right)^2 - 1} \quad (24)$$

Выражение для $H(z)$ в случае полосового фильтра запишется так:

$$H(z) = \frac{(3_n')^n \cdot (\theta')^n}{\prod_{i=1}^n (3_n' - (3_i')') (3_n' - (3_i')'')} \quad (25)$$

Здесь $(3_i')'$ и $(3_i')''$ — два нормализованных корня ПФ, на которые расщепился корень 3_i ФНЧ. Преобразование частот по приведенным формулам сопровождается появлением в функции $H(z)$ n — кратного нуля в начале координат плоскости z .

В случае ФНЧ целесообразно использовать шкалу частот, нормированную по ширине полосы θ .

На этом кончается изложение метода синтеза фильтров. Ниже будут приведены описания программ "B-filter" (для ЭВМ типа "М-20") и "Z-filter" (для ЭВМ "БЭСМ-6"), реализующих рассмотренный метод. Предварительно приводится перечень используемых в программах процедур, сопровождаемый необходимыми пояснениями.

Программы написаны на входном языке "Альфа" разработки ВЦ СО АН СССР [6], [7]. Предполагается, что читатель знаком с [8], а также с особенностями программирования на "Альфе" для ЭВМ "БЭСМ-6".

Процедуры к программе "B-filter"
и "Z-filter"

Три нижеследующие процедуры оперируют алгебраическими полиномами вида:

$$A = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n,$$

где a_i — комплексные коэффициенты. Каждый полином задается своей степенью n и одной строкой двумерного комплексного массива коэффициентов a_i . Например, $P[I:m, 0:mp]$.

Процедура *polymult* ($X, i, mx, Y, j, my, z, k, mz, n$) осуществляет умножение полинома, заданного коэффициентами $X[i,]$ степени mx на полином $Y[j,]$ степени my , возведенный в степень n . Результирующие коэффициенты будут получены в массиве $z[k,]$, а переменная mz примет значение степени результирующего полинома.

Процедура *polydif* ($x, i, m1, m2, n$) осуществляет операцию $A := \frac{d^n A}{dx^n}$; $x[i,]$ — коэффициенты исходного

полинома, $m1$ — его степень.

Коэффициенты n раз продифференцированного полинома помещаются на место исходных; переменная $m2$ — степень результирующего полинома.

Процедура *polysum* ($X, i, mx, Y, j, my, z, k, mz$) выполняет сложение полиномов $X[i,]$ степени mx и $Y[j,]$ степени my . Результат помещается в $z[k,]$, а mz — степень результирующего полинома.

Процедура *substit* ($X, i, mx, Y, j, my, z, k, mz$) выполняет подстановку полинома в полином. Например:

$$A(x) = a_0 + a_1 x + a_2 x^2; \quad B(y) = b_0 + b_1 y; \\ A(x) \Big|_{x=B(y)} = A(y) = a_0 + a_1 (b_0 + b_1 y) + a_2 (b_0 + b_1 y)^2 = \dots$$

Здесь $A(x)$ задается коэффициентами $X[i,]$ и степенью mx , $B(y)$ задается коэффициентами $Y[j,]$ и степенью my . Результат помещается в $z[k,]$; mz — степень результата. Из тела процедуры происходит обращение к процедуре *polymult*.

Процедура *print* (θ, e, Y, k) написана для БЭСМ-6 и позволяет выдавать информацию на АЦПУ. Здесь θ и e — простые переменные или переменные с индексами типа "вещественный". На печать выдаются слова, адреса которых лежат в пределах $\langle \theta \rangle + \langle e \rangle$. Y — двумерный массив (целый или вещественный) для формирования информационных слов экстракода печати 064. $Y[k,]$ — вектор, используемый при данном обращении. Граничная пара массива Y по второму измерению должна иметь вид 1:7. Содержательное значение каждого числа в этих семёрках соответствует описанию экстракода 064 [8]. Фиктивные операторы $\theta := \theta$ и $e := e$ в теле процедуры гарантируют при трансляции подстановку этих переменных именем.

Если при обращении к *pgmat* $Y[k,4] = 0$, то процедура фиксирует адреса начала и конца печатаемой информации и формирует одно информационное слово (ИС) разметки. При повторном обращении с $Y[k,4] = 0$ добавляется очередное ИС разметки, и наконец, при обращении с $Y[k,4] = 1$ после добавления очередного ИС происходит печать. Процедура готова к печати новой информации.

Процедура *редак* (x, n, S) написана для БЭСМ-6 и предназначена для редактирования алфавитно-цифровой информации. 48-разрядное слово x делится на шесть слогов по 8 разрядов в каждом. Величина S внутри процедуры записывается в целом виде и помещается в слог n , считая слева. Переменные x, n и S могут быть простыми или с индексами и иметь тип "вещественный" и "целый".

Вещественная процедура *чис* (X) написана для БЭСМ-6 и дает возможность представить переменную $0 \leq X \leq 999999$ в виде буквенной константы для алфавитно-цифровой печати. Перед переводом производится округление X .

Процедура *tape* (m) написана для БЭСМ-6 и при $m < 0$ эквивалентна пустому оператору. В противном случае процедура осуществляет перепись МОЗУ БЭСМ-6, начиная с 0 листа на собственную НМЛ 6-0, начиная с зоны m , печатая попутно информационные слова экстракода 070. Перепись кончается прерыванием по "чужому листу" и печатью программы. В дальнейшем при помощи небольшой ручной программы вызывается основная программа с НМЛ, вводятся дополнительные исходные данные и производится счет или же программа снова записывается на НМЛ, начиная с указываемой зоны. При этом достигается возможность изменения исходных данных с повторным счетом без повторной передачи по каналу связи "М-20" - "БЭСМ-6", а также исключается ввод с ВУ-700 очень больших программ на перфокартах. *)

В программах "B-filter" и "x-filter", схемы которых приводятся ниже, из описаний исключены вышеуказанные процедуры. Места постановки перфокарт с описаниями соответствующих процедур указаны скобками * *

*) Процедура *tape*, а также программа вызова были написаны после консультации автора настоящей работы с Ф.Цангом и Л. Эйфросом (ВЦ СО АН СССР).

Программа предназначена для получения параметров $H(z)$ по полюсовым или низкочастотным РЦФ с аппроксимацией модуля частотной характеристики по Баттерворту. Программа написана на входном языке [6], [7] для машины "М-20". Объем транслированной программы - около 2400₈ команд, а время работы зависит от сложности синтезируемого фильтра и составляет в среднем 12сек.

Основная программа начинает работу вводом исходных данных. Затем печатается номер задачи, порядковый номер синтезируемого фильтра и введенные исходные данные. Далее происходит обращение к процедуре "B-filter", которая и вычисляет параметры $H(z)$. После этого печатаются, а затем перфорируются три массива: паспорт фильтра, показатель степени и коэффициенты полинома числителя $H(z)$, показатель степени и коэффициенты полинома знаменателя $H(z)$. Затем программа обращается к читающему устройству и требует ввода новых исходных данных. Ниже описывается работа процедуры "B-filter".

Исходными данными для программы являются:

- 1) средняя частота фильтра f_0 ;
- 2) уровень δ_1 ;
- 3) ширина полосы пропускания δ_1 на уровне δ_1 (по напряжению);
- 4) уровень δ_2 ;
- 5) ширина полосы пропускания δ_2 на уровне δ_2 ;
- 6) период квантования по времени T ;
- 7) максимально допустимый порядок K низкочастотного прототипа синтезируемого фильтра.

Средней частотой f_0 для ПФ названа частота $f_0 = \frac{f_{c1} + f_{c2}}{2}$, где f_{c1} и f_{c2} - нижняя и верхняя частоты среза. Для ФНЧ f_0 считается равной нулю. по формуле (21) вычисляется требуемый порядок фильтра. Из двух чисел - вычисленного порядка и порядка, заданного в п.7, - выбирается наименьшее, оно и определит порядок синтезируемого фильтра. Условно принято, что если $\delta_2 = \delta_2 = 0$, то синтезируется фильтр с порядком, указанным в п.7.

Частота среза определяется в соответствии с (20).

Далее вычисляется резонансная частота фильтра f'_0 :

$$(f'_0)^2 = (f_0)^2 - (f_c/2)^2,$$

Если $(f'_0)^2 > 0$, окончательно устанавливается, что будет синтезирован ПФ, в противном случае за резонансную час-

тоту принимается нуль (независимо от значения в п.1) и ведется расчет ФНЧ.

Вводится нормализующий коэффициент τ в соответствии с (24) в случае ПФ: $\tau = 2\pi f_{рез}$ или же в случае ФНЧ: $\tau = 2\pi f_c$. Вычисляется нормированный период квантования $T = T_x \tau$, поскольку в выражение (12) введены нормированные полюса.

Определяется число полюсов K функции $H(z)$:

$$K = 2n \quad \text{для ПФ}$$

$$K = n \quad \text{для ФНЧ.}$$

Дальнейшие вычисления ведутся в основном по формуле (12).

В программе предусмотрена возможность контроля положения полюсов передаточной функции путем снятия аналитической частотной характеристики. Включение блока измерителя частотной характеристики производится с пульта машины путем зажатия на РПУ-Ш 45-разряда, а на РПУ-I и II - соответственно начального и конечного значений частоты (в герцах, в двоично-десятичном коде). Программа делит заданный частотный интервал на 100 отсчетов; для каждого значения текущей частоты f вычисляется $z = \exp(j\omega T)$, подставляется в $H(z)$, и модуль полученного числа вместе со значением текущей частоты выводится на печать. Частота меняется по линейному закону.

Получение коэффициентов $H(z)$ не гарантирует устойчивости работы РЦФ. Неустойчивость обуславливается накоплением ошибки при счете по рекурсивной формуле (1) и зависит от погрешностей вычисления, т.е. от длины мантиссы, и проявляется для фильтров высоких порядков, обладающих низкой по отношению к частоте квантования частотой среза (ФНЧ) или низкой f_0 (ПФ). Практически реализовывались фильтры порядка 5-6 (в пересчете на низкочастотный прототип).

Были написаны программы для контроля устойчивости синтезированного фильтра и снятия реальной амплитудно-частотной характеристики, а также для получения временного отклика фильтра на δ - импульс. В обоих случаях счет велся по формуле (1). При снятии частотной характеристики фильтра на вход последнего подавался сигнал, частота которого плавно изменялась в заданных пределах по линейному, косинусоидальному или показательному законам. Выходной сигнал детектировался и сглаживался скользящим ФНЧ.

Программа "x-filter"

Программа предназначена для получения параметров $H(z)$ по лосовых или низкочастотных РЦФ с аппроксимацией по Баттерворту или по Чебышеву. Программа написана для использования на ЭВМ "БЭСМ-6". Программа включает в себя процедуру снятия частотной характеристики синтезируемого фильтра (при этом проверяется устойчивость). Параметры синтезированного фильтра могут быть записаны на магнитную ленту.

Длина программы - порядка 4000₈ ячеек; задача использует 16₁₀ листов МОЗУ, 30₁₀ трактов НМБ и две собственные ленты: одну с параметрами фильтров и вторую - для хранения основной программы. Время синтеза и снятия частотной характеристики одного РЦФ с печатью всех таблиц и графика ЧХ - около 1 мин.

Для записи параметров синтезируемых фильтров на НМЛ принята паспортная система, заключающаяся в следующем.

Запись информации начинается с пятой зоны, четвертая зона - паспортная. Паспортная зона вписывается в массив $PO[0:146, 1:7]$. Компоненты нулевой строки его имеют следующее значение:

- $PO[0,1]$ - номер ленты;
- $PO[0,2]$ - число объектов на ленте;
- $PO[0,3]$ - номер последней записанной зоны;
- $PO[0,4]$ - период квантования δt .

Под объектом в данном случае понимается зона, содержащая параметры некоторой группы фильтров.

Последующие строки паспортного массива имеют отношение к объектам, помещенным на данную ленту. Номер строки соответствует номеру объекта. При этом имеет значение лишь компонента $PO[k,2]$, указывающая номер зоны, где записан объект номер k .

Объект - зона соответствует массиву $store[1:16,1:2,-1:30]$, что позволяет помещать в один объект до 16 фильтров, заданных степенью числителя ($store[k,1,-1]$), числителем ($store[k,1,0:30]$), степенью знаменателя ($store[k,2,-1]$ и знаменателем ($store[k,2,0:30]$) передаточной функции $H(z)$.

В качестве исходных данных вводятся четыре массива и одна переменная:

- y - массив, содержащий алфавитно-цифровую информацию для печати надписей, таблиц, графиков;
- z - массив, определяющий вид информационных слов печати при обращениях к процедуре *print*;
- pt - управляющий массив;

$pI[1] = 0$;
 если $pI[2] = 1$ - печатается положение полюсов $H(s)$;
 если $pI[3] = 1$ - печатается $H(z)$;
 если $pI[4] = 1$ - печатается таблица - частотная характеристика;
 если $pI[5] = 1$ - печатается график частотной характеристики;
 если $pI[6] = 0$ - программа не обращается к ленте с фильтрами, в противном случае $pI[6]$ определяет кодовый номер ленты;
 $pI[7]$ определяет номер объекта на ленте, с которым предполагается обмен.
 $p2$ - массив-спецификация. Его величина позволяет вводить до 16 групп по 11 чисел. Каждая группа определяет режим работы программы над очередным фильтром. Значения компонент массива:
 1) $M1$; если $M1 > 0$ - то это номер фильтра, который необходимо взять с ленты из заданного объекта;
 если $M1 = -1$ - приказ на синтез фильтра заданного порядка (если это возможно; в противном случае - максимально возможного);
 если $M1 = -2$ - приказ на синтез фильтра максимально возможного порядка.
 2) $M2$; если $M2 = 0$ - $H(z)$ на ленту не писать;
 если $M2 > 0$ - $H(z)$ записать на ленту как фильтр номер $M2$.
 3) f_{cp} - средняя частота фильтра (Гц).
 4) $\delta 1$ - ширина полосы на уровне $1/\sqrt{2}$;
 5) $\delta 2$ - второй уровень;
 6) $\delta 2$ - ширина полосы на уровне $\delta 2$;
 7) $T1$ (сек) - период квантования фильтруемого сигнала;
 8) N - максимальный порядок фильтра;
 9) Δ , если $\Delta = 0$, принимается аппроксимация по Баттерворту, в противном случае берется аппроксимация по Чебышеву и Δ определяет максимальное значение провалов на вершине частотной характеристики.
 IO) f_1
 II) f_2 } (Гц) - частотный диапазон снятия АЧХ.
 И, наконец, последняя вводимая величина: ϵ является фактическим параметром при обращении к процедуре *tape*.
 Перед началом счета программа сличает кодовые номера заказанной и поставленной ленты с параметрами фильтров, печатает их и, в случае несовпадения, останавливается.

Поскольку в последующей работе программа не сильно отличается от "B-filler", остановимся лишь на её особенностях.

Для упрощения расчетных формул, как уже отмечалось выше, первый уровень для определения ширины полосы фильтров фиксирован на уровне ~ 0.707 .

Для определения расчетного порядка синтезируемого фильтра при аппроксимации по Чебышеву поступаем следующим образом. Из (14) имеем:

$$\left(\frac{1}{\sqrt{2}}\right)^2 = \frac{1}{1 + \delta^2 \operatorname{ch}^2\left(n \operatorname{arch} \frac{\delta 1}{f_c}\right)};$$

$$(\delta 2)^2 = \frac{1}{1 + \delta^2 \operatorname{ch}\left(n^2 \operatorname{arch} \frac{\delta 2}{f_c}\right)}.$$

Разрешая оба уравнения относительно частоты среза f_c и приравнявая правые части, приходим к записи:

$$\delta 1 \operatorname{ch}\left[\frac{1}{n} \operatorname{arch}\left(\frac{1}{\delta \cdot \delta 2} \sqrt{1 - \delta^2}\right)\right] - \delta 2 \operatorname{ch}\left[\frac{1}{n} \operatorname{arch}\left(\frac{1}{\delta}\right)\right] = 0.$$

Из этого уравнения необходимо найти n . Для этого ищем вещественный корень функции $\operatorname{Ch}(n)$, представляющей собой левую часть (26), методом деления интервала пополам (неописываемая процедура *bis* 01).

После получения $H(z)$ для очередного фильтра осуществляется снятие его частотной характеристики (процедура ИЧХ). Если фильтр неустойчив, а затребован синтез фильтра заданного порядка, порядок фильтра уменьшается на единицу и синтез повторяется.

Если требуется синтезировать фильтр максимально возможного порядка, то синтез начинается с заданного или вычисленного порядка, затем переходят к синтезу фильтров более высокого порядка и т.д. до потери устойчивости. При этом на ленту могут быть записаны параметры последнего из устойчивых фильтров.

Устойчивость

Наиболее серьезным ограничением, налагаемым на использование РЦФ, является потеря устойчивости. На устойчивость отрицательно влияют следующие факторы:

- повышение порядка фильтра;

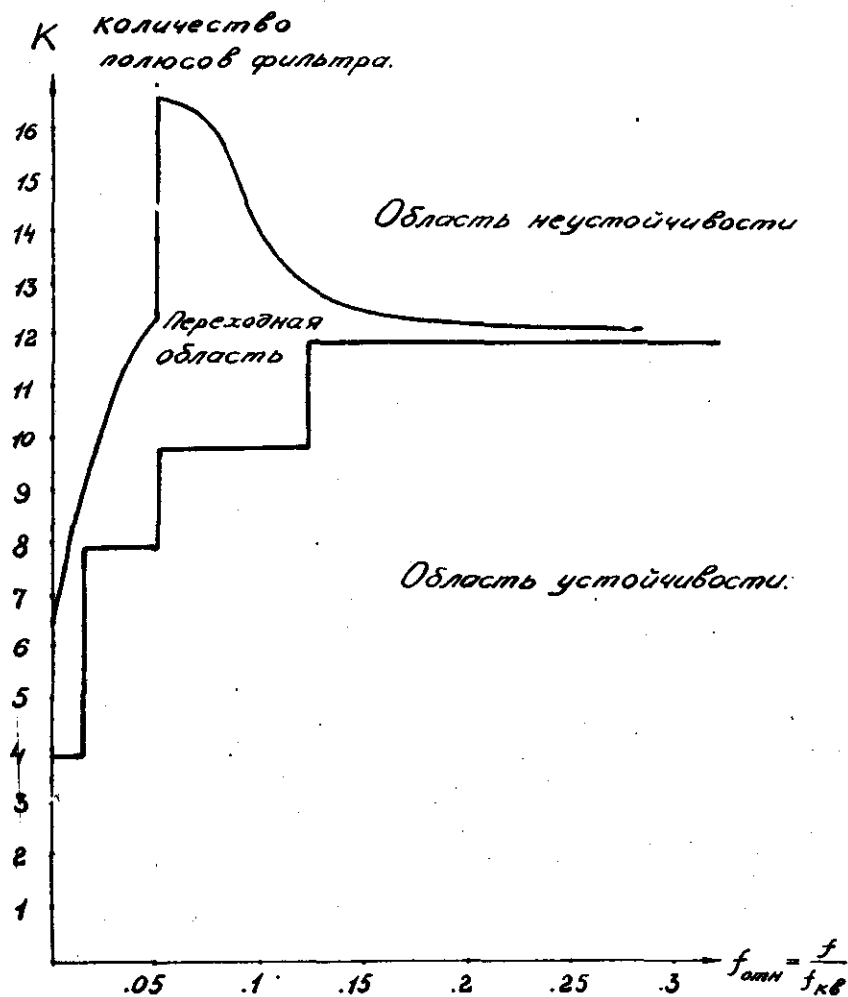


Рис.3 Область устойчивости синтезируемых РЧФ.

- уменьшение относительной ширины $\hat{\delta} = \delta/f$ квант.;
- снижение относительной резонансной частоты ПФ $f_0 = f_0/f$ квант.;
- уменьшение длины мантииссы, т.е. точности счета, проявляющейся при счете по ф. (I).

Для экспериментального исследования границ устойчивости синтезируемых фильтров программа "z-filter" была модифицирована с целью автоматизации перебора вариантов. Результаты проверки приведены на рис. 3. График построен для относительных частот: $f/f_{кв}$, где f - частота среза ФНЧ или резонансная частота ПФ; $f_{кв}$ - частота квантования сигнала.

Число полюсов фильтров K равно порядку n для ФНЧ и равно $2n$ для ПФ.

В переходной области были получены отдельные устойчивые реализации. Полученные данные по устойчивости РЧФ не являются полными и окончательными.

В Приложениях 1, 2 и 3 помещены используемые процедуры и программы "B-filter" и "z-filter". В Приложении 4 приведены в качестве примера результаты синтеза двух фильтров по программе "z-filter".

В заключение автор выражает признательность В.Гусеву за ряд ценных замечаний, сделанных в процессе дискуссий при работе над этой темой.

Альфа-программа проц polymult

```

1 процедура polymult(X,i,mx,y,j,my,z,k,mz,n); значение mx, my; целые mx, my;
2 начало целые p,q; комплексные массивы x[0:mx + my * n], y[0:my], z; n = a + i * b;
3 a = 0; b = 0; для p:= 0,...,mx цикл x[p]:= X[i, p];
4 для p:=0,..., my цикл y[p]:= Y[j, p]; mz := mx + my * n;
5 n раз цикл { для p:= 0,..., mz цикл z[k,p] := s;
6 для p:=0,..., mx цикл для q:=0,..., my цикл z[k,p+q] := z[k,p+q] + x[p] * y[q];
7 mx := mx + my; для p:=0,..., mx цикл x[p] := z[k,p] } конец;
1 процедура polydif (x,i,m1,m2,n);
2 начало целое j; комплексное s; s = a + i * b; a = 0; b = 0; m2 := m1;
3 n раз цикл { для j:=0,..., m2-1 цикл x[i, j] := x[i, j+1] * x(j+1);
4 x[i, m2] := s; m2 := m2-1 } конец;
1 процедура polysum (X,i,mx,y,j,my,z,k,mz);
2 начало целое p; комплексный массив x[0:mx], y[0:my], s; s = a + i * b; a = 0; b = 0;
3 для p:=0,...,mx цикл x[p] := X[i, p];
4 для p:=0,...,my цикл y[p] := Y[j,p];
5 mz := если mx > my то mx иначе my;
6 для p:= 0,..., mz цикл { z[k,p] := s;
7 если p ≤ mx то z[k,p] := z[k,p] + x[p];
8 если n < my то z[k,p] := z[k,p] + y[p] } конец;

```

Альфа- программа "substit" (M-20)

```

1 процедура substit (X,i,mx,y,j,my,z,k,mz); значение mx, my;
2 комплексные массивы X,Y,z; целые mx, my;
3 начало целые p,q,r; комплексные массивы x[0:mx], y[1:1,0:my],
4 z[1:1,0:mx * my], s; s = a + i * b; a := 0; b := 0;
5 для p:=0,...,mx цикл x[p] := X[i,p];
6 для p:=0,...,my цикл y[1,p] := Y[j, p]; mz := mx * my;
7 для p:=1,...,mz цикл z[k,p] := z[k,p] + x[p] * y[1,p] := s;
8 z[k,0] := x[0]; a := 1; z[k,1,0] := s; r := 0;
9 для p:=1,..., mx цикл {polymult(z,z,1,r,y,1,my,zz,1,r,1);
10 для q:=0,..., r цикл z[k,q] := z[k,q] + x[p] * z[k,zz[1,q]] } конец;
1 процедура print(b,e,y,k); начало собственный массив A[1:7];
2 собственный целый массив x[1:20], 1, α; целые q,r; вещественное z;
3 1 = 1; α = 1; c: код (6400,0,0,0); если α = 1 то { q:= 1;
4 для α:= 20,28,40,41,44,52,64 цикл {код (10, α); код (36,27);
5 код (0, z); A[q] := z; q:= q+1; код (10,0); код (0,α) } ;
6 если 1 = 1 то код (14,240,e); код (42,14); код (0, 1);
7 код (14,240,b); код (42,14); код (36,50); код (15,1);
8 код (0, x[1]); 1:=2; на p; b:= b; e:= e; p:= 1; x[1] := α;
9 для q:=1,...,7 цикл {r:=y[k,q]; код (10, r); код (12,c); код (26,A[q]);
10 код (15, x[1]); код (0, x[1]) } ; 1:= 1+1;
11 если y[k,4] = 1 то {код (14,240, x[1]); код (14,64,0); 1:= 1 } конец;

```

Альфа-программа проц редак БЭСМ-6

```

1 процедура редак (x,n,s): начало вещественное q; целое i;
2 c:КОД (0,0,0,377); d:КОД (6400,0,0,0); КОД (14,240, x);
3 i := 16+8 x n; КОД (10,1); КОД (36,27); КОД (0, q); i := s; на р: x := x + x;
4 p:КОД (10,c); КОД (26,q); КОД (14,11,0); КОД (14,12,0); КОД (14,0,0);
5 КОД (10,1); КОД (12,d); КОД (26,q); КОД (14,15, 0);
6 КОД (14,0,0) конеч;

```

Альфа - программа веш проц чис БЭСМ-6

```

1 вещественная процедура чис (X); начало целое i; вещественные x, p,y, z, t;
2 c:КОД (360,7417,360,7400); z := x := X +.5;
3 КОД (10,c); КОД (0,t); если x > .5 то { p:= 105; для i := 1,..., 6 цикл
4 { если x / p ≥ 1 то { y:= entier (z / p); редак (t,1, y); z := z - y x p } ;
5 p:=.1 x p } ; чис: = t конеч;

```

Альфа-программа проц таре БЭСМ-6

```

1 процедура таре (m); начало целый массив x[ 0:0] ,b, c;
2 d : КОД (0,0,0,7777); e: КОД (0,100,0,1); a :КОД (0,0,60,0); на q ;
3 СПО 175 (16, x [ ], x [ ], 48,0);q: b:= m ; если b < 0 то на q ;
4 КОД (10,b ); КОД (11,d ); КОД (15,a ); КОД (0,c); b:= 20; m: КОД (70,c);
5 КОД (230,b ); КОД (11,0); b:= b +1024; СПО 176 (641,c,c,0,0);
6 КОД (10,c): КОД (13,e): КОД (0,c): на m; q : конеч ;

```

Альфа-программа "B- filter" (M-20)

Приложение 2

```

1 начало целое i ;
2 процедура B-filter (f 0, s1, b1, s2, b2, T,n,z ); значения T; вещественное T;
3 начало целые i, j, m, n, q, s, t, u, v, k; получит полудиф substitut получит
4 комплексные массивы p [1:100] ,B, s, D[1:1,0:100] , A [1:2,0:50] , Z[1:100,0:1] ,
5 pк, Bpк [1:1,0:0] , r, x, φ; s[1,j] = sr[1, j] + i si[1,j];
6 p[1] = a[1] + i b[1]; Z[1,j] = q[1,j] + i h[1,j]; x = c + i d; r = α + i β;
7 B[1,j] = Br[1,j] + i Bv[1,j]; D[1,j] = Dr[1,j] + i Dv[1,j];
8 вещественные fc, ω, τ ; собственное вещественное κ ; логическое вр; κ:= 3.1415926535;
9 если s2 = b2 = 0 то {n:= n; на i } ;
10 n:= entier (.5x ln((1/ s2 + 2 -1)/(1/ s1 + 2 -1))/ ln ( b2/ b1)) + 1;
11 если n ≠ 0 & |abs(n) < n то n:= n; если n > 50 то n:=50; n:= n;
12 fc:= exp ( ln ( b1) - 1/ (2 x n) x ln (1/ s1 + 2 -1));
13 τ:= ( f0) t2 - (f c) t 2/4; если τ > то {вр:= истина; κ:= n + n; f0:= sqrt(τ); τ:= 2 x κ f0;
14 ω := 2 x κ x fc/τ } иначе{ вр:= ложь; f0:=0; κ:=n; τ:= 2 x κ x fc; ω:= 1};
15 T:= T x τ; α:=1; β:=0; m:=0; B [1,0]:= r ;
16 для i:=1,...,entier ( n/2+.6) цикл
17 {τ:= κ x ( n- 1+2 x i)/(2 x n); c:= cos(τ ); d:= sin(τ ); если abs(d/c) ≤ 10-6 то d:= 0;
18 Z[1,i] := r; если вр то {φ:= sqrt ((ω x x) t 2/4-1); s:= 4 x i -3;
19 p [s]:= ω x x/2 + φ ; p [ s + 2 ]:= ω x x/2- φ ;
20 p [s+ 1] := cos(p [s] ); p [s + 3 ]:= cos(p [ s + 2])) } иначе

```

АЛФА-программа "в-filter" (M-20)

```

21 {s:=2 x 1 -I; p[s] := x ; p[s+1] := 0om(x)}};
22 s:= I; ДЛЯ i:=1,...,K ЦИКЛ{s:= - s ; z[I,0] := -p[i];
23 polynomial (B,I,m,z,I,i,B,I,m,I); ЕСЛИ s= I ТО ДЛЯ j:= 0,...,m ЦИКЛ B[i,j] := 0 ;
24 o:= d :=0; ЕСЛИ вp ТО { ДЛЯ i:= 0,...,n ЦИКЛ A[i,i] := x ; A[i,n] := x x om t n} ИНАЧЕ
25 { A[i,0] :=x ; n:= 0;
26 polydif (B,I,m,t,I): 2: D[I,0] := t ; N:= 0; s :=I; ДЛЯ i:=1,...,K ЦИКЛ
27 {pk [I,0] := p[i]; g[i,0] :=i; h[i,0] := 0; z[i,I] := - exp (p[i] x T);
28 substit (B,I,t,pk,I,0,Bpk,I,q); polynomial (z,i,I,Bpk,I,0,z,i,I,
29 q,I); polynomial (D,I,N,z,i,I,D,I,N,I); t:=0; s:= - s ;
30 ЕСЛИ s = I ТО ДЛЯ j:=0,...,N ЦИКЛ D[i,I,j] := 0;
31 ДЛЯ j:=0,...,N ЦИКЛ ti:= t + abs(Dr[I,j]); ti:= t/(N+I); ЕСЛИ t < 10-7 ВТ > 107 ТО
32 {g[I,0] := t * (-1/i); h[I,0] := 0; polynomial (A,I,n,z,I,0,A,I,n,I);
33 polynomial (B,I,t,z,I,0,B,I,t,I); n:= 2 };
34 ДЛЯ i:=0,...,100 ЦИКЛ S[i,i] := x ; s :=I; ДЛЯ i:=1,...,K ЦИКЛ
35 {pk [I,0] := p[i]; substit (A,I,n,pk,I,0,B,I,u);
36 ДЛЯ j:=1,...,K ЦИКЛ ЕСЛИ i ≠ j ТО polynomial (B,I,u,z,i,j,I,B,I,u,I);
37 polysum (S,I,u,B,I,u,s,I,u); si:= - s ; ЕСЛИ s= I ТО ДЛЯ j:=0,...,u ЦИКЛ
38 S[i,j] := 0; z[I,-I] := u; z[2,-I] := N;
39 ЕСЛИ вp ТО {o:=0; d:=-I; z[I,0] := exp (x x T)} ИНАЧЕ{g[I,0] := I; h[I,0] := 0;
40

```

АЛФА-программа "в-filter" (M-20)

```

41 x:= (A[I,0] / Br[I,0]) / (B[I,0] / Br[I,0]); g[I,0] :=I/mod (x); h[I,0] := 0;
42 polynomial(s,I,u,z,I,0,s,I,u,I);
43 НАЧАЛО МАСШ U[1:2], f1,f2, ω1, ω2, ω, δω, τ; ЛОГИЧЕСКОЕ a ;
44 ну: КОД (20,3,0,a); ЕСЛИ 7a ТО на out; КОД (20,4,0,a); ЕСЛИ a ТО stop;
45 КОД (20,1,0,f1); КОД (20,2,0,f2); СПООС2 (f1,f1); СПООС2 (f2,f2);
46 ВВОД (313,f1,f2); ti:= 2 x π x (ЕСЛИ вp ТО f0 ИНАЧЕ f0);
47 ωI := 2 x π x f1; ω2 :=2 x π x f2; δω := (ω2 - ωI) x .OI; ω := ωI;
48 ДЛЯ ω := ω + δω ПОКА ω ≤ ω2 ЦИКЛ
49 {o:=0; d:=-I; z[I,0] := exp (x x ω/τ x T);
50 substit (S,I,u,z,I,0,A,I,s); substit (D,I,N,z,I,0,B,I,s);
51 U[i] := ω / (2 x π); U[2] := mod ((A[I,0] / Br[I,0]) / (B[I,0] / Br[I,0]));
52 ВВОД (U)} ;out: КОД (20,4,0,a); ЕСЛИ a ТО на ну КОНЕЦ;
53 ДЛЯ i:= 0,...,u ЦИКЛ z[i,i] := Re ( S[i,i]) / Re (D[i,0]);
54 ДЛЯ i :=0,...,N ЦИКЛ z[2,i] := Re (D[i,i]) / Re (D[i,0]) КОНЕЦ;
55 МАСШ z[i:2,-I:50], z0, bx[i:7];
56 i :=I; in: ВВОД (bx); ВВОД (3265,110,1,bx);
57 в filter (bx[i],bx[2],bx[3],bx[4],bx[5],bx[6],bx[7],z);
58 s0[i] := 2OI; z := i; z0[3] :=bx[7]; z0[4] :=bx[i]; z0[5] := bx[3];
59 s0[6] := bx[6]; z0[7] := 0; ВВОД (s0);
60 СПОГ76 (1153,s[i,-i],s[i,s[i,-i]],0,0);

```

АЛФА-программа "B - filter" (M-20)

```

61      СПО176 (1153, z[2,-1], z[2, -1], 0,0);
62 СПО176(1281, z0 [1], z0 [7], 0,0); СПО176(1281, z[1,-1], z[1, z [1,-1]], 0,0);
63 СПО176 (1281, z[2,-1], z[2, z[2,-1]], 0,0); i := i + 1; на 1m конец z

```

Приложение 3

АЛФА-программа "Z - filter" (БЭСМ-6)

```

1  начало заменить "у[0:мy]" на "у [0:30]", "x[0:мx + my x n]" на "x[0:60]"
2  ", "x [0:мx]" на "x [ 0:30]", "у [1:1,0:мy]" на "у [ 1:1,0:30]"
3  ", "zz [1:1,0:мx x my]" на " zz [ 1:1,0:0]"
4  ; конец отладки
5  начало массив p1[1:15], p2 [1:16,1:11], y[1:100], q[1:15,1:7], z1, z2 [ 0:30],
6  store[1:16,1:2,-1:30], z5[1:30000], p0 [0:146, 1:7]; логическое B; барабан z8 ;
7  вещественные for, r0, b1, b2, f2, t, n, n, d, m1, m2, f1, f2, n, d,
8  t1, f0, oc, e, f, t, y 10; x print, редак, чис, polymult, polysum, polydif, substit, tape;
9  целые i, j, m, qq, s, t, v, n, k, f, f, w, w, v, l, u1, n1, m;
10 комплексные массивы z[1:7,-1:30], z[1:30,0:1], pк[1:2,0:0], one, nil, x, ф ;
11      z[i,j] = zr[i,j] + i z1[i,j]; z[i,j] = zr[i,j] + i z1[i,j];
12      one = or + i oi; nil= nr + i ni; x = zr + i x1;
13 функция Ch ( n ) = ch(1/n x arch(1/(d 2 x d) x sqrt(1 - d 2 1 2))) x b1 -
14      b2 x ch (1/n x arch(1/ d));
15 процедура ИЧХ; начало собственные вещественные p1, p2, dt;
16 целые m, i, j, α, β, γ; массивы z[-30:60000], y1 [-30:0], u, f[1:100];
17 вещественные w1, w2, t, w, dt, p, h;
18      для i:=0,..., v цикл z1 [i] := zr[5, i];
19      для i= 0,..., n цикл z2 [i]:= zr[6,i] ; print coef;
20 если x1 ≠ p1 v i2 ≠ p2 v t1 ≠ dt то {t1 := f1; p2 := f2; dt := t1; w1 := 2 x - x t1;

```

АЛФА-программа "z-filter" (БЭСМ-6)

```

21  w2:=2 x x f2;  w := ( w 2- w1)/(T1 x 3,10^4);  df := ( f2 -f1)x.01;
22  f := f1;  для i :=1,...., 100 цикл {f:= f + df; f[i] := чис ( f )};
23  t := - df;  для i:=1, 1 + 6000 пока i < 3,10^4 цикл {  для j:=1,....,6000 цикл
24  { t := t + df; x[j] := sin (( w i + w /2 x t) x t)};
25  СПОИ77 (1037, x[i], x [6000], 0, x6[i]);};
26  измер: для i:= - 30 ,...., 0 цикл x[i] :=  y1 [i] := 0; u[ ] := 0;  h := t := 0;  α := β :=1; γ:=3;
27  для i:=1, 1 +6000 пока i < 3,10^4 цикл { СПОИ77 (1032, x[i], x[6000], 0, x6[i] )};
28  iter : y10:= y1 [0] :=0;  для j:=0,...., 0 цикл y1 [0] :=  y1[0] + z1 [j] x x[α - j] ;
29  для j:=1,...., N цикл y10:= y10 - z2 [j] x y1[-j];
30  y1 [0] := y1 [0] + y10;  если abs ( y1 [0] ) > h то h:= abs ( y1[0] );
31  t := t + abs ( y1 [0] );  если t ≥ 10^14 то {print( y[55], y[57], q, 2);  n := n -1;
32  если n ≤ 0  v MI > 0 то { print ( y[63], y[65], q, 2 );  на next } ;
33  если MI = -1 то на i;  если MI = -2 то на next } ;
34  для j:= -30,....,-1 цикл y1[j] := y1[j + i] ;  α := α +1;
35  β:= β +1;  если β = 96 то { u[γ] := u[γ] + t ;  u [γ - i] := u [γ - i] + t ;
36  u [γ- 2] := u[γ-2] + t ; t :=0 } ;  если β = 301 то { u [γ] :=  u [γ] + t ;
37  u[γ - i] := u [γ-i] + t; t:=0; β :=1;  γ:= γ +1 ;  если α ≤ 6000 то на iter ;
38  α :=1;  для j := -30,...., 0 цикл x[j] := x[j + 6000 ] } ;
39  t:= u[i];  для i:= 2,...., 100 цикл если u[i] > t  то t:= u[i] ; u[ ]:= u[ ]/t;
40  для i:= 0,...., 0 цикл z1[i] := z1 [i] / h;  print coeff ;

```

АЛФА- программа "z-filter" (БЭСМ-6)

```

41  если p1 [4] = 1 то { print ( y [41], y[43], q, i); x[0] := y[58] ;
42  для i:=1,....,19 цикл x[i] := y [59]; x[20] := y[15];  для i:= 1 шаг 10 до 91 цикл
43  { для j:= 0,...., 9 цикл x[2 x j + i] := f [1 + j] ; print ( x [0], x [20], q, i);
44  print ( u [i] , u [i+ 9], q, 7) ;
45  если p1 [5] = 1 то {для i:= -7,...., -1 цикл x[i] := y [1 +51] ;  для i :=100 шаг - 2 до 0
46  цикл { x [-7] := чис ( i ) ;  если frac (ix.i) ≤ .01 то {x [0] := y [53]; m :=1} иначе m:=0;
47  для j :=1,....,100 цикл если окр (u[j] x 50) = 1/2 то { x[m] := y[52];
48  редак ( x[m] , 5j +8); m := m +1; x[m] := y [51]; print ( x [-7], x[m], q, i )};
49  x[-1] := y[54] x[0] := чис ( f1); m:=1;  для i:= 10 шаг 10 до 100 цикл
50  { x[m] := y[54] ; редак ( x[m] , 6, i +4); x [m +1] := f[i]; m:= m +2};
51  x[21] := y[51]; print ( x[-1], x[21], q, 2)} конец мчх ;
52  процедура print coeff ;  если p1 [3] = 1 то {print ( y[38], y[40], q, i); y[32] :=чис ( v ) ;
53  print( y[32], y[33], q, i); print( z1[0], z1[u] , q, 8);
54  y[32] := чис ( N); print ( y[32], y[33], q, i ) ; print( z2 [0], z2 [N] , q, 8) } ;
55  ввод ( y , q, p1, p2, e ) ; tapo ( e ) ; print( y[i], y[9], q, 2);
56  print ( y[10], y[15], q, 2);  m:= 3.1415926536;  ori:=1; m:= m1 := ol := 0;
57  если p1 [1] = 0 & p1 [6] > 0 то {e:= p1 [6] ; на контроль } ;
58  если p1 [1] ≠ 0 то {e:=p1 [7]; контроль: СПОИ75 (16, p0 [0,1] , p0 [, ], 0,4);
59  y[28] := чис ( e); y[32] := чис (p0 [0,1] ); print ( y[25], y[33], q, 4);
60  i:= p0 [p1 [7], 2] ;  если p0 [0,1] ≠ e то стоп;

```

АЛФА-программа "Z-filter" (БЭСМ-6)

```

61 СПОИ75 (I6, store [I,I-I], store[,,], 0, 1 );
62 ДЛН I:=I6, 1-I пока p2 [I,7] = p2 [I6,I] & 1 ≥ I ЦИКЛ Ф:= 1-I;
63 ДЛН Ф: I,..., Ф ЦИКЛ { MI,M2, f op, bI, s2,b2, TI,n, Δ, fI, f2 | := p2 [Ф,] ;
64 print (y [I6], y[24] , q, I); print (p2 [Ф,I], p2 [Ф,9] , q, 3);
65 если MI > 0 & pI [6] > 0 то {v := store[MI, I, -I];
66 ДЛН i:=0,..., v ЦИКЛ { zr [5,i] := store[MI,I,1]; zi [5,i] := 0 };
67 N := store[MI,2,-I]; ДЛН j := 0,...,N ЦИКЛ { zr [6,i] := store[MI,2,1];
68 zi [6,i] := 0 }; на δI.тх;
69 ε := fop 2 - bI t 2/4; если ε > 0 то { f0:= sqrt (ε) ; B:= истина } иначе
70 {f0:= 0; B:= ложь}; print ( y [60], y[6I], q, I); print ( f0, f0, q, 8);
71 если Δ > 0 то δ := sqrt (I/ (I- Δ) ) t 2 -I); если δ 2 = b2 = 0 то {n:= n ; на I } ;
72 если B то b2 := abs (( f0 t 2 - ( fop + b2/2) t2)/( fop + b2/2));
73 если Δ = 0 то n:= .5 x ln (I/62 t 2 - I)/ ln ( b2/ bI) иначе
74 bIs OI (Ch ,.9,I5,0,.00I,0,.00I,n);
75 print(y[37], y[37], q, I); print ( n, n, q, 5); n:= окр (n);
76 I: если n ≠ 0 & n < n то n := n; 3: если B то {если n > I5 то n:=I5} иначе
77 если n > 30 то n := 30; y [35] := чис (n); print( y[34] , y[36], q, I);
78 fс:= если Δ = 0 то bI иначе bI/ oh (I/n x arch (I/6 ));
79 если B то {k:=n+n; ωс:= fс/ f0; τ := 2 x π x f0 } иначе
80 {κ := n; τI = 2 x π x fс}; τ := τI τ τ ; m := 0; z[4,0] := ....

```

АЛФА-программа "Z-filter" (БЭСМ-6)

```

81 ДЛН i:=I,..., entler ( n/2 +.6) ЦИКЛ { если Δ = 0 то {ε:= π x ( n-I+2 x i)/(2 x n);
82 xτ:= cos ( ε) ; xI := sin (ε) } иначе {ε:= π x(2 x n-I+2 x i)/(2 x n); φ:=I/n x arch(I/6);
83 xτ := sh (φ) x sin(ε); xI:= ch (φ) x cos(ε) } ;
84 если abs (xI/xτ) ≤ I0-6 то xI := 0;
85 если B то {φ := sqrt ((ωс x x) t 2/4-I); s := 4x i-3; z[7,s] := ωс x x/2 + φ ;
86 z[7, s+2] := ωс x x/2 - φ; z[7, s+1] := Con ( z [7, s] );
87 z[7,s+3] := Con(z[7,s+2])} иначе{s:= 2 x i-I; z[7,s] := x; z[7,s+1] := Con(x)};
88 если pI [2] = I то { print( y[62], y[62], q, I); ДЛН i:=I,..., n ЦИКЛ
89 { y[I] := zr [7,i] ; y[2] := zi[7,i] ; print ( y[I], y [2], q, 8) } ; z[I,I] := one ;
90 s := I; ДЛН i:=I,...,n ЦИКЛ {s :=- s ; z[I,0] := - z[7,i] ;
91 polymult ( z,4,m , z ,I,I,z ,4,m ,I); если s = I то ДЛН j:=0,...,m ЦИКЛ zi [4,j] := 0 ;
92 B то { ДЛН j=0,..., n ЦИКЛ z[3,i] := nil ; nn:= n ; z [3,nn] := one x ωсI n} иначе
93 {z[3,0] := one ; nn:= 0} ;
94 polydif (z ,4,m , t,I); 2: z[6,0] := one ; N := 0; s := I; N := 0;
95 ДЛН i:=I,...,n ЦИКЛ {рк [I,0] := z[7,i] ; z [4,0] := one ; z[i,I] := - exp(рк[I,0] x T);
96 substitit ( z,4,t ,рк,I,0,рк,2,qd);
97 polymult ( z,4,I,рк,2,0,z ,i , qq ,I);
98 polymult(z ,6,N ,z ,i ,I,z ,6,N ,I); ε := 0; s :=- s ; εIms = I то
99 ДЛН j:=0,..., n ЦИКЛ zi [6,j] := 0; ДЛН j := 0,...,N ЦИКЛ ε := ε + abs ( zr [ 6,j] );
100 ε := ε /(N+I); если ε < I0-7 v ε > I07 то {zτ[I,0] := ε t (-I/4); zi[I,0] := 0;

```

Параметры фильтров (синтез по "z-filter")

Альба-программа "z-filter" (БЭСМ-6)

```

101  polymult (z, 3, r, z, I, 0, z, 3, m, I);
102  polymult (z, 4, t, z, I, 0, z, 4, t, I); на 2));
103  для i:=0,...,30 цикл z[5,i] := nil ; s:=I;
104  для i:=1,...,к цикл { рк [I,0]:= z[7,i] ; substit (z, 3, m, рк I, 0, z, 4, u);
105  для j:=1,...,к цикл если i ≠ j то polymult (z, 4, u, z, j, I, z, 4, u, I);
106  polymult (z, 5, u, z, 4, u, z, 5, u); s := - s ; если s = I то для j := 0,...,u цикл ai[5,j] := 0);
107  e := zr [6,0]; для i:=0,...,м цикл zr [6,i] := zr [6,i] / e ;
108  для i:=0,...,u цикл zr [5,i] := zr [5,i] / e ;
109  рж:ИЧУ; если M2 > 0 то {store[M2,1,-1]:= u ;
110      для i := 0,..., u цикл store[M2,1,i] := zi[i] ;
111  store[M2,2,-1] := m ; для i:=0,...,м цикл store[M2,2,i] := z2[i]} ;
112  если M1 = -2 то { n := n + I; на если k < 30 то 3 инае next } ; next :);
113  если pI [6] > 0 то { i := p0 [7], 2} ;
114  СПО175 (2I, store[I,1,-1], store[.,.,0, t]) } конец м

```

Тип фильтра: ФНЧ.

Аппроксимация: по Чебышеву, $\Delta = 0.1$.

Ширина полосы: 1000 гц.

Период квантования: $62.5 \cdot 10^{-6}$ сек.

Порядок фильтра: 5

Коэффициенты $H(z)$: $\beta_0 = 1$;

$$a_0 = .105557698964 \cdot 10^{-11} \quad \beta_1 = -.446548469964 \cdot 10^1$$

$$a_1 = .387054452693 \cdot 10^{-4} \quad \beta_2 = .814600211969 \cdot 10^1$$

$$a_2 = .391543818464 \cdot 10^{-3} \quad \beta_3 = -.757759395369 \cdot 10^1$$

$$a_3 = .363948769631 \cdot 10^{-3} \quad \beta_4 = .359132629747 \cdot 10^1$$

$$a_4 = .310747966442 \cdot 10^{-4} \quad \beta_5 = -.693424798749$$

Тип фильтра: полосовой.

Аппроксимация: по Баттерворту.

Центральная частота: 440 гц.

Резонансная частота: 428,5 гц.

Ширина полосы: 200 гц.

Период квантования: $62.5 \cdot 10^{-6}$ сек

Порядок фильтра: 3

Коэффициенты $H(z)$:

$$a_0 = -.288107560245 \cdot 10^{-11} \quad \beta_1 = .576049448833 \cdot 10^1$$

$$a_1 = .216722780674 \cdot 10^{-3} \quad \beta_2 = .139078720352 \cdot 10^2$$

$$a_2 = -.444469101497 \cdot 10^{-3} \quad \beta_3 = -.180125691380 \cdot 10^2$$

$$a_3 = .330700944619 \cdot 10^{-4} \quad \beta_4 = .131983071593 \cdot 10^2$$

$$a_4 = .400375950318 \cdot 10^{-3} \quad \beta_5 = -.518774424558 \cdot 10^1$$

$$a_5 = -.205699724769 \cdot 10^{-3} \quad \beta_6 = .854649512663$$

Массивы y и q (z -filter)

I Б (I73) М ж (I74) и (I75) y_1	25 Б ЗАКАЗА y_2
2 Б (I73) Ц _ _ _ _	26 Б НА _ ЛЕН
3 Б _ _ _ _ _	27 Б ТА : _ _ N
4 Б _ _ _ _ _	28 Б _ _ _ _ _
5 Б _ _ _ _ (I75) _	29 Б (I73) Ч ПОСТ
6 Б _ (I73) Ц В АД	30 Б АВЛЕНА
7 Б А Ч А _ _ 6	31 Б _ ЛЕНТА
8 Б 0 2 0 2 0 _	32 Б 0 0 0 0 0 0
9 Б _ _ 0 I 2 (I72)	33 Б _ _ _ _ _ (I72)
10 Б (I73) X Z _ _ _	34 Б ORDER :
11 Б _ _ F _ _ J	35 Б 0 0 0 0 0 0
12 Б _ _ L _ _ T	36 Б _ _ _ _ _ (I72)
13 Б _ _ В _ _ R y_2	37 Б ORDER (I72) y_2
14 Б (I75)(I73) Y $\equiv$ (I74) ж	38 Б Z _ C O E F
15 Б (I72) _ _ _ _ _	39 Б F J C J E N
16 Б MI (I73) _ M2	40 Б T S : _ _ (I72)
17 Б (I73) ; FCP _	41 Б FREQ . _
18 Б _ _ _ _ _	42 Б RESPON
19 Б (I73) Б BAND	43 Б SE : _ _ (I72)
20 Б W (I73) M L E V	44 Б 0 0 0 0 0 0
21 Б 2 (I73) Ч B A N	45 Б (I73) 8 I (I73) (:
22 Б D W (I73) J T (I73)	46 Б (I73) $\neq$: (I73) ж :
23 Б _ O R D . (I73)	47 Б (I73) P : (I73) M :
24 Б I C H E B (I72)	48 Б (I73) L : (I73) $\leq$:

Массив y и q (z -filter)

49 Б (I73) / : (I73)(I42) : y_1	I z0 z0 z0 zI y_2
50 Б (I73)(I54) : _ _ _	z0 z0 z0
51 Б _ _ _ _ _ (I72)	2 z0 z0 z0 zI
52 Б _ _ _ (I73) _ 0	zI z0 z0
53 Б (I73) 9 _ (I74)(I43) _	3 z3 z0 z8 zI
54 Б _ _ _ _ _ (I73) 4	zI zII zIO
55 Б Ф И Л Ь Т Р	4 z0 z32 z0 zI
56 Б _ Н Е _ П Р	z7 z0 z0
57 Б О Ш Е Л I (I72)	5 z3 zI5 zIO zI
58 Б _ _ _ _ _ (I73) 3	z0 z0 z0
59 Б _ _ _ _ _	6 z3 z3 z9 zI
60 Б F 0 _ C 0 M	z0 zI2 z9
61 Б P U T E Д (I72) y_1	7 z3 z3 z9 zI y_2
62 Б R O O T S (I72)	zI zI2 z9
63 Б U N R E A L	8 z3 z0 zI6 zI
64 Б J Z A B L E	zI z22 z5
65 Б . . . _ _ (I72)	

ЛИТЕРАТУРА

1. W. Rader, B. Gold, Digital Filter Design Techniques in the Frequency Domain, IEEE, February, 1967.
2. H. Holtz and C. T. Leondes, The Synthesis of Recursive Digital Filters. Journal of the ACM, April 1966, v.13, n. 2.
3. I. R. Ragazzini, G. F. Franklin Sampled - Data Control Systems, Mc Graw - Hill B.C., Inc, N.Y., 1958.
4. Л.Т. КУЗИН. Расчет и проектирование дискретных систем управления. Машгиз, М., 1962.
5. Н. БАЛАБАНИН. Синтез электрических цепей. ГЭИ, М.-Л., 1961.
6. А.П. ЕРШОВ, Г.И. КОЖУХИН, Ю.М. ВОЛОШИН. Входной язык для систем автоматического программирования. Новосибирск, 1964.
7. А.П. ЕРШОВ, Г.И. КОЖУХИН, И.В. ПОТТОСИН. Руководство к пользованию системой Альфа. Наука, Новосибирск, 1968.
8. Математическое обеспечение машины "БЭСМ-6". Описание и инструкции. ИТМ и ВТ, ВЦ АН СССР, М., 1967.

Поступила в редакцию
7. IV. 1969г.