

# Синтез и анализ цифровых фильтров с использованием программного пакета MatLab

## Содержание

1. Начало работы . . . . .	1
2. Ввод коэффициентов передаточной функции и построение графиков частотных характеристик аналогового фильтра-прототипа . . . . .	1
3. Нули и полюсы аналогового фильтра-прототипа и их преобразование в коэффициенты передаточной функции . . . . .	3
4. Расчёт аналогового фильтра-прототипа по заданным требованиям к его характеристике затухания . . . . .	4
5. Нахождение передаточной функции цифрового фильтра по аналоговому прототипу методом билинейного z-преобразования и методом инвариантной импульсной характеристики . . . . .	6
6. Структуры цифровых фильтров и соответствующие им алгоритмы цифровой фильтрации . . . . .	9
7. Просмотр характеристик синтезированного цифрового фильтра . . . . .	19
8. Синтез цифрового фильтра с использованием программы fdatool . . . . .	20
9. Исследование влияния квантования сигналов и коэффициентов фильтра на характеристики фильтра . . . . .	23
10. Моделирование работы цифрового фильтра . . . . .	29

## 1. Начало работы

Произведите загрузку системы MatLab. Откроется командное окно (Command Window). Вы будете иметь возможность работы в интерактивном режиме, вводя операторы после значка >>. Кроме того, в командном окне можно ввести указанное в текущем каталоге имя той или иной программы, которой затем будет передано управление. Для очистки командного окна используется команда `clc`.

## 2. Ввод коэффициентов передаточной функции фильтра и построение графиков частотных характеристик аналогового фильтра-прототипа

Передаточная функция аналогового прототипа задаётся в виде:

$$K(p) := \frac{b_0 p^m + b_1 p^{m-1} + \dots + b_m}{a_0 p^n + a_1 p^{n-1} + \dots + a_n}, \quad m \leq n. \quad (1)$$

Для построения графиков АЧХ и ФЧХ нужно ввести векторы **b** и **a** коэффициентов передаточной функции, а затем вызвать подпрограмму расчёта комплексной частотной характеристики и построения графиков. Пусть, например,  $m=1$ ,  $n=2$ ,  $b_0=2$ ,  $b_1=-1$ ,  $a_0=1$ ,  $a_1=3$ ,  $a_2=2.5$ . В интерактивном режиме вводятся следующие операторы:

```
>> b=[2 -1];
>> a=[1 3 2.5];
>> freqs(b,a)
```

Элементы векторов разделяются пробелами и заключаются в квадратные скобки. После ввода вектора ставится точка с запятой, а затем следует нажать клавишу <Enter>. После оператора `freqs(b,a)` точка с запятой не ставится, ввод тоже следует завершить нажатием клавиши <Enter>. Команды запоминаются, и их можно воспроизводить, пользуясь клавишами управления курсором: ↑ и ↓. Будут построены графики АЧХ и ФЧХ (АЧХ в логарифмическом масштабе, но без пересчёта в децибелы, ФЧХ в градусах). По умолчанию выбираются 200 частот, логарифмически равномерно распределённых в диапазоне от 0.1 до 10.

Если нужно построить АЧХ в линейном масштабе, в ином диапазоне частот, ФЧХ в радианах и с устранением скачков на  $2\pi k$  радиан, то следует ввести следующие операторы (пусть, для примера, границы частотного диапазона от  $f_1=100$  Гц до  $f_2=1000$  Гц, число расчётных точек  $N=451$ ):

```
>> f=100:(1000-100)/450:1000;
>> b=[1 0 7.971e6];
>> a=[1 7.427e2 1.501e6 5.536e8];
>> k=freqs(b,a,f*2*pi);
>> subplot(2,1,1)
>> plot(f,abs(k)/max(abs(k))), grid
>> subplot(2,1,2)
>> plot(f,unwrap(angle(k))), grid
```

При вводе строки `subplot(2,1,1)` появится графическое окно **Figure No.1**. Его нужно сместить, чтобы оно не загромождало командное окно, или временно свернуть. В последнем случае, по окончании ввода операторов нужно развернуть графическое окно, чтобы увидеть графики.

График АЧХ аналогового фильтра-прототипа можно использовать для нахождения частоты дискретизации  $F_s$ . Например, может быть задано, что АЧХ на частоте  $F_s/2$  должна достигать определённого уровня. Для детального рассмотрения нужной части графика рекомендуется использовать инструменты изменения масштаба (уменьшение, увеличение), имеющиеся на панели инструментов графического окна.

### 3. Нули и полюсы аналогового фильтра-прототипа и их преобразование в коэффициенты передаточной функции

Если вам известны нули и полюсы аналогового фильтра-прототипа, а не коэффициенты передаточной функции, то прежде нужно ввести нули и полюсы, а затем преобразовать их в коэффициенты, используя специальную функцию преобразования, имеющуюся в MatLab. Далее можно построить графики частотных характеристик. Например, известны полюсы  $p_1=-0.2+0.5i$  и  $p_2=-0.2-0.5i$ , а также нули  $z_1=0.1i$  и  $z_2=-0.1i$ . Осуществляем ввод:

```
>> p=[-0.2+0.5i -0.2-0.5i]';  
>> z=[0.1i -0.1i]';  
>> k=5;
```

Здесь  $k$  - коэффициент, стоящий перед всей дробью передаточной функции (его можно выбрать произвольно, если нет иных соображений). Символ ' (апостроф), поставленный после закрывающей квадратной скобки, означает транспонирование. При этом вектор-строка преобразуется в вектор-столбец. (В MatLab принято полюсы и нули задавать вектор-столбцами, а коэффициенты передаточной функции вектор-строками.)

Далее производим преобразование полюсов и нулей в коэффициенты передаточной функции:

```
>> [b,a]= zp2tf (z,p,k);
```

Здесь  $b$  – строка коэффициентов числителя,  $a$  – строка коэффициентов знаменателя передаточной функции фильтра.

Диаграмму нулей и полюсов можно вывести, если задать следующие команды:

```
>> plot (p,'x')  
>> hold on  
>> plot (z,'o')  
>> hold off  
>> axis equal  
>> axis ([-1 1 -1 1])
```

Последняя команда позволяет задать граничные значения для действительной (первые два числа) и мнимой (следующие два числа) осей. Возможно и обратное преобразование коэффициентов  $b$  и  $a$  в полюсы и нули:

```
>> [z,p,k]= tf2zp (b,a);
```

Если требуется вывести значения каких-либо переменных рабочей области MatLab на монитор, то нужно просто после значка `>>` ввести имя переменной, завершив ввод нажатием клавиши `<Enter>`. Например, если указать

```
>> z
```

то будет выведен вектор нулей. Вслед за оператором

```
>> p
```

будет выведен вектор полюсов. Если не ставить точку с запятой после операторов преобразования, то выходные параметры будут выведены на монитор без дополнительных команд.

#### **4. Расчёт аналогового фильтра-прототипа по заданным требованиям к его характеристике затухания**

Если заданы граничные частоты, допустимые затухания, тип фильтра (ФНЧ, ФВЧ, ППФ, ПЗФ), класс фильтра (Баттерворта, Чебышёва, инверсный Чебышёва, эллиптический, Бесселя), то рассчитать коэффициенты передаточной функции фильтра можно, используя одну из следующих функций MatLab:

- 1) `butter (n, w0, type, 's')` – расчёт фильтра Баттерворта;
- 2) `cheby1 (n, Rp, w0, type, 's')` – расчёт фильтра Чебышёва;
- 3) `cheby2 (n, Rs, w0, type, 's')` – расчёт инверсного фильтра Чебышёва;
- 4) `ellip (n, Rp, Rs, w0, type, 's')` – расчёт эллиптического фильтра;
- 5) `besself (n, w0, type)` – расчёт фильтра Бесселя.

Здесь  $n$  – порядок ФНЧ-фильтра-прототипа;  $R_p$  – неравномерность характеристики затухания в полосе пропускания (в дБ);  $R_s$  – минимальное затухание в полосе задерживания (в дБ);  $w_0$  – скаляр или двухэлементный вектор частот среза (границ полосы пропускания) (в рад/с) (для фильтра Баттерворта частота среза задаётся по уровню 3 дБ). Если для фильтра Баттерворта граничная частота полосы пропускания  $w_p$  задана для

значения затухания  $R_p$  ( $R_p < 3$  дБ), то её нужно пересчитать в частоту среза

$$w_0 := \frac{w_p}{\sqrt[2n]{\frac{R_p}{10^{10} - 1}}}$$

type – строковая константа, задающая тип фильтра. Строка ‘s’ говорит о том, что синтезируется аналоговый фильтр.

При синтезе ФНЧ:  $w_0$  – скаляр, параметр type отсутствует.

При синтезе ФВЧ:  $w_0$  – скаляр, type=’high’.

При синтезе ППФ:  $w_0$  – вектор [ $w_{01}$   $w_{02}$ ], причём  $w_{01} < w_{02}$ , параметр type отсутствует.

При синтезе ПЗФ:  $w_0$  – вектор [ $w_{01}$   $w_{02}$ ], причём  $w_{01} < w_{02}$ , параметр type=’stop’.

Если порядок ФНЧ-прототипа не задан, то его можно определить (имеется в виду минимальный порядок), применяя следующие функции MatLab:

`[n,wn]=buttord (wp, ws, Rp, Rs, ‘s’)`

`[n,wn]=cheb1ord (wp, ws, Rp, Rs, ‘s’)`

`[n,wn]=cheb2ord (wp, ws, Rp, Rs, ‘s’)`

`[n,wn]=ellipord (wp, ws, Rp, Rs, ‘s’)`

Здесь  $n$  – минимальный порядок фильтра;  $w_n$  – частота среза фильтра (для фильтра Баттерворта она определяется по уровню 3 дБ);  $w_p$  – граничная частота полосы пропускания (в рад/с);  $w_s$  – граничная частота полосы задерживания (в рад/с).

Для ППФ и ПЗФ  $w_p$  и  $w_s$  – двухэлементные векторы: [ $w_{p1}$   $w_{p2}$ ] и [ $w_{s1}$   $w_{s2}$ ].

Должны выполняться неравенства:

для ФНЧ:  $w_p < w_s$ ;

для ФВЧ:  $w_p > w_s$ ;

для ППФ:  $w_{s1} < w_{p1} < w_{p2} < w_{s2}$ ;

для ПЗФ:  $w_{p1} < w_{s1} < w_{s2} < w_{p2}$ .

Найденный порядок  $n$  далее используется в функциях расчёта коэффициентов передаточной функции фильтра, описанных выше.

Рассмотрим пример. Пусть требуется найти коэффициенты передаточной функции эллиптического полосно-пропускающего фильтра, для которого  $R_p=1$  дБ,  $R_s=40$  дБ,  $w_{p1}=1000$  рад/с,  $w_{p2}=1100$  рад/с,  $w_{s1}=900$  рад/с,  $w_{s2}=1250$  рад/с. Вводим операторы:

```
>> [n, wn]=ellipord([1e3 1.1e3], [0.9e3 1.25e3], 1, 40, ‘s’);
```

```
>> [b,a]=ellip(n, 1, 40, [1e3 1.1e3], ‘s’);
```

Если требуется определить полюсы и нули, то вместо последнего оператора вводят:

```
>> [z,p,k]= ellip (n, 1, 40, [1e3 1.1e3], 's');
```

Альтернатива: использование функции преобразования:

```
>> [z,p,k]= tf2zp (b,a);
```

## 5. Нахождение передаточной функции цифрового фильтра по аналоговому прототипу методом билинейного z-преобразования и методом инвариантной импульсной характеристики

### 5.1. Билинейное z-преобразование

Это преобразование задаётся путём замены переменных:

$$p := \frac{2}{T} \cdot \frac{z-1}{z+1}, \quad (2)$$

где  $p$  – комплексная частота аналогового фильтра-прототипа,  $z$  – комплексная переменная, от которой зависит передаточная функция цифрового фильтра,  $T$  – интервал дискретизации ( $T=1/F_s$ , где  $F_s$  – частота дискретизации). Достаточно в передаточной функции аналогового прототипа сделать подстановку (2), и будет найдена передаточная функция цифрового фильтра  $K(z)$ :

$$K(z) := \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_M z^{-M}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N}} \quad (3)$$

Если  $a_0 \neq 1$ , то нужно разделить все коэффициенты числителя и знаменателя на  $a_0$ , чтобы знаменатель записывался в виде:

$$1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N}$$

Альтернативным способом нахождения передаточной функции  $K(z)$  методом билинейного  $z$ -преобразования является пересчёт полюсов и нулей аналогового прототипа в полюсы и нули цифрового фильтра по формуле:

$$z := \frac{\frac{2}{T} + p}{\frac{2}{T} - p}$$

Затем осуществляется преобразование нулей и полюсов в коэффициенты фильтра (см. разд.3; функции MatLab, описанные здесь для аналоговых фильтров, годятся и для цифровых фильтров). Если количество полюсов аналогового ФНЧ-прототипа превышает количество его нулей, то возникают также дополнительные нули, так что общее количество нулей и полюсов цифрового фильтра, синтезированного по методу билинейного  $z$ -преобразования, оказывается равным (об этом см. в специальной литературе). Указанные преобразования применимы, если аналоговый фильтр-прототип и цифровой фильтр – это фильтры одинакового типа (оба ФНЧ или оба ФВЧ и т.д.). Значит, если задан ФНЧ-прототип, а его нужно преобразовать в цифровой фильтр другого типа (ФВЧ, ППФ, ПЗФ), то нужно прежде найти передаточную функцию аналогового фильтра этого типа, а затем применить билинейное  $z$ -преобразование. Соответствующее преобразование типов фильтра осуществляется в MatLab операторами:

>> [b1,a1]=lp2hp (b, a, w0);

- преобразование ФНЧ в ФВЧ; w0 – граничная частота ФВЧ (рад/с).

>> [b1,a1]=lp2bp (b, a, w0, Bw);

- преобразование ФНЧ в ППФ; w0 – средняя геометрическая частота полосы пропускания (рад/с) ( $w0 = \sqrt{w1 * w2}$ ); Bw – полоса пропускания (рад/с) ( $Bw = w1 - w2$ ).

>> [b1,a1]=lp2bs (b, a, w0, Bw);

- преобразование ФНЧ в ПЗФ; w0 – средняя геометрическая частота полосы задерживания (рад/с) ( $w0 = \sqrt{w1 * w2}$ ); Bw – ширина полосы задерживания (рад/с) ( $Bw = w1 - w2$ ).

Само билинейное преобразование в MatLab осуществляется следующими операторами:

>> [bz, az]= bilinear (b, a, Fs);

или

```
>> [zz, pz, kz]= bilinear (z, p, k, Fs);
```

Здесь  $b$ ,  $a$ ,  $z$ ,  $p$ ,  $k$  – коэффициенты передаточной функции, нули, полюсы и масштабный коэффициент передаточной функции аналогового фильтра-прототипа.  $bz$ ,  $az$ ,  $zz$ ,  $pz$ ,  $kz$  – соответствующие параметры цифрового фильтра. Векторы  $b$  и  $a$  должны задаваться как векторы-строки,  $z$  и  $p$  – как векторы-столбцы. Преобразование строки в столбец и наоборот осуществляется путём постановки символа ‘ (апостроф) после имени вектора. Например, операция

```
>> d=d’;
```

приводит к транспонированию вектора  $d$ .

Параметр  $F_s$  – это частота дискретизации [Гц].

Отобразить диаграмму полюсов и нулей можно командой

```
>> zplane (z, p)
```

или

```
>> zplane (b, a)
```

В первом случае  $z$  и  $p$  – вектор-столбцы, во втором случае  $b$  и  $a$  – вектор-строки. Полюсы отображаются крестиками, нули – кружками. Отображается также окружность единичного радиуса. Указанную команду можно применять как для цифровых, так и для аналоговых фильтров.

## 5.2. Метод инвариантной импульсной характеристики

Этот метод предполагает, что импульсная характеристика цифрового фильтра совпадает с точностью до постоянного множителя с импульсной характеристикой аналогового прототипа в точках  $t=nT$ , где  $n=0, 1, 2, \dots$ ,  $T=1/F_s$  – интервал дискретизации. Иначе говоря,  $g_d(n)=\alpha g_a(nT)$ , где  $\alpha$  – некоторый коэффициент. Передаточная функция цифрового фильтра записывается в виде:

$$K(z) := \alpha \cdot \sum_{k=1}^N \frac{r_k}{1 - e^{p_k \cdot T} \cdot z^{-1}}, \quad (4)$$

где  $r_k = \text{Res}_{p=p_k} K(p)$  – вычет передаточной функции аналогового прототипа в полюсе  $p_k$ . Общее количество полюсов  $p_k$  равно  $N$  (предполагается, что все полюсы простые).

Вычеты  $r_k$  и полюсы  $p_k$  можно найти, используя следующий оператор MatLab:

```
>> [r, p, k]= residue(b, a)
```



Если не ставить в конце строки точку с запятой и нажать клавишу <Enter>, то на экран монитора будут выведены вектор вычетов  $\gamma$ , вектор полюсов  $\rho$  и вектор коэффициентов целой части  $k$ . Полюсы и вычеты могут быть действительными или образовывать комплексно-сопряжённые пары. Масштабирующий коэффициент  $\alpha$  подбирается таким, чтобы значение  $g_{\alpha}(0)$  было порядка единицы. При этом значения величин  $\alpha k$  по модулю тоже порядка единицы. Если далее сумму дробей (4) привести к одной дроби, то можно получить передаточную функцию в стандартном виде (3). Полюсы цифрового фильтра в методе инвариантной импульсной характеристики связаны с полюсами аналогового прототипа стандартным  $z$ -преобразованием:  $z_k = \exp(\rho_k T)$ .

Синтез цифрового фильтра по методу инвариантной импульсной характеристики осуществляется в MatLab путём ввода оператора:

```
>> [bz, az]=impinvar (b, a, Fs);
```

где  $bz$ ,  $az$  – коэффициенты числителя и знаменателя передаточной функции цифрового фильтра (см. (3));  $b$ ,  $a$  – коэффициенты числителя и знаменателя передаточной функции аналогового прототипа (см. (1));  $F_s$  – частота дискретизации (в Гц).

## **6. Структуры цифровых фильтров и соответствующие им алгоритмы цифровой фильтрации**

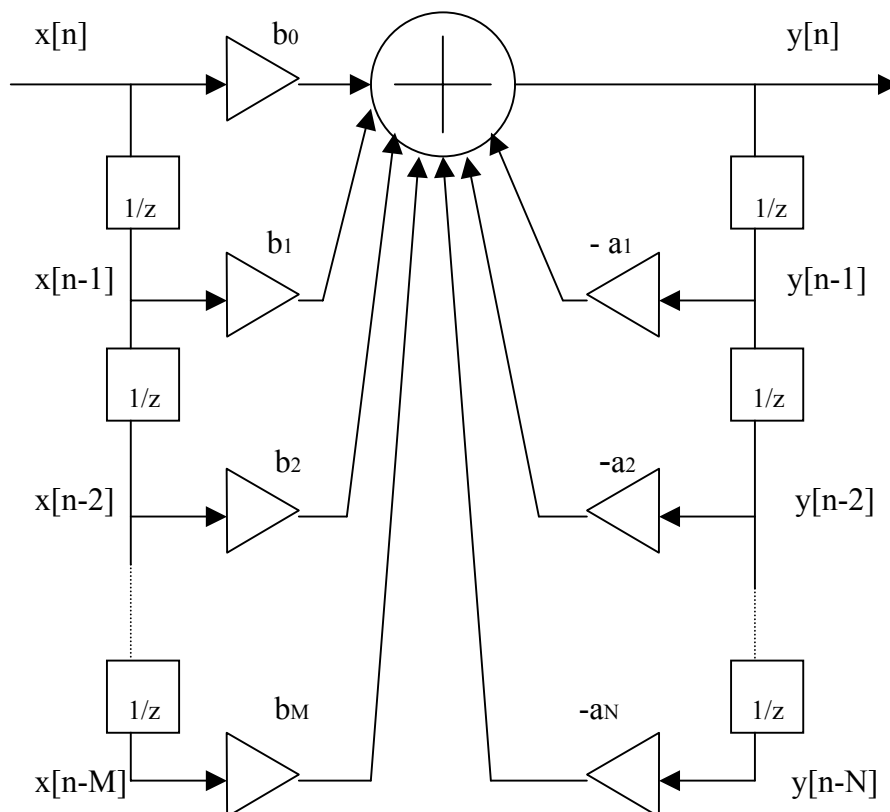
Одной и той же передаточной функции  $K(z)$  цифрового фильтра соответствуют различные формы реализации и разные алгоритмы преобразования отсчётов входного сигнала в отсчёты выходного. Выбор той или иной структуры цифрового фильтра имеет смысл при учёте эффектов конечной разрядности представления коэффициентов фильтра и отсчётов сигнала на входе, в промежуточных точках и на выходе. Дело в том, что дисперсия шума квантования на выходе устройства, чувствительность характеристик к точности представления коэффициентов, быстродействие, необходимый объём памяти и другие характеристики качества цифрового фильтра зависят от алгоритма работы и структуры. Естественно, что предпочтительнее та структура, которая обеспечивает наилучшие характеристики качества при заданных ограничениях. При реализации цифровых фильтров в виде специализированных вычислительных устройств имеет также значение, в какой последовательности выполняются операции, есть ли возможность производить несколько операций параллельно, что позволяет повысить быстродействие; каковы аппаратные затраты при реализации того или иного алгоритма.

### 6.1. Прямая структура рекурсивного фильтра

Передаточной функции

$$K(z) := \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_M \cdot z^{-M}}{a_0 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_N \cdot z^{-N}}$$

соответствует структура вида

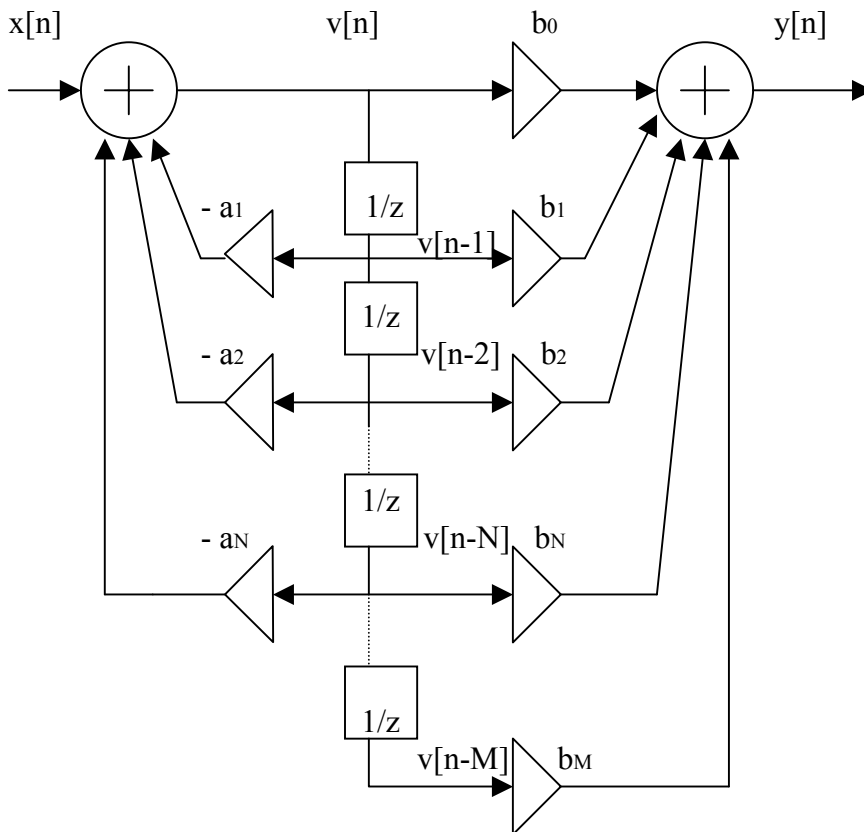


Алгоритм (разностное уравнение) записывается следующим образом:

$$y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] + \dots + b_M x[n-M] - a_1 y[n-1] - a_2 y[n-2] - \dots - a_N y[n-N]$$

### 6.2. Каноническая структура рекурсивного фильтра

Каноническая структура получается из прямой путём разделения сумматора на две части (одна для прямых связей, другая – для обратных связей) с последующей перестановкой левой и правой частей схемы и дальнейшим слиянием параллельных цепочек элементов памяти в одну.



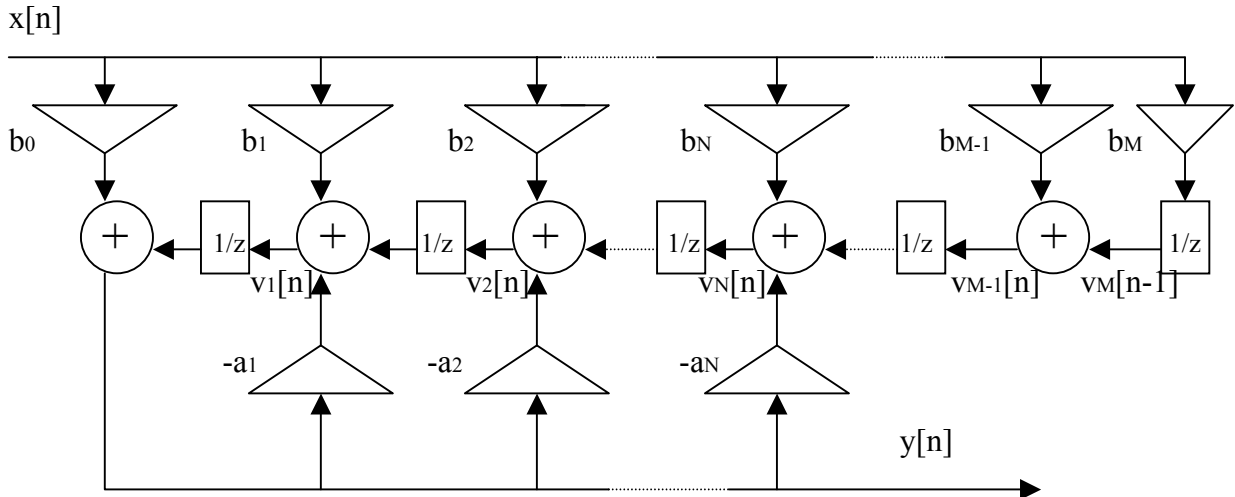
На данной схеме показано, что  $M > N$ , однако это не обязательно; возможны случаи  $M = N$  или  $M < N$ . Наибольшее из чисел  $M$  и  $N$  является порядком фильтра. Алгоритм для канонической структуры записывается в виде двух рекуррентных соотношений:

$$\begin{cases} v[n] = x[n] - a_1 v[n-1] - a_2 v[n-2] - \dots - a_N v[n-N] \\ y[n] = b_0 v[n] + b_1 v[n-1] + b_2 v[n-2] + \dots + b_M v[n-M] \end{cases}$$

Сначала производится вычисление отсчёта сигнала  $v[n]$  в промежуточной точке (на выходе первого сумматора), а затем уже с его использованием – отсчёта выходного сигнала  $y[n]$ . Каноническая форма интересна тем, что в ней, в отличие от прямой структуры, представлена одна последовательность элементов памяти, а не две. Это позволяет экономить память. Однако абсолютные значения отсчётов промежуточного сигнала  $v[n]$  могут превосходить значения отсчётов входного и выходного сигналов, так что может потребоваться увеличенная разрядность ячеек памяти по сравнению с разрядностью регистров для ввода и вывода отсчётов входного и выходного сигналов соответственно.

### 6.3. Транспонированная структура рекурсивного фильтра

Преобразование прямой структуры, связанное с изменением порядка операций задержки и суммирования, приводит к транспонированной структуре.



Алгоритм для транспонированной структуры:

$$\begin{aligned}
 y[n] &= b_0 x[n] + v_1[n-1] \\
 v_1[n] &= b_1 x[n] - a_1 y[n] + v_2[n-1] \\
 v_2[n] &= b_2 x[n] - a_2 y[n] + v_3[n-1] \\
 &\bullet \\
 &\bullet \\
 v_N[n] &= b_N x[n] - a_N y[n] + v_{N+1}[n-1] \\
 &\bullet \\
 &\bullet \\
 v_{M-1}[n] &= b_{M-1} x[n] + v_M[n-1] \\
 v_M[n] &= b_M x[n]
 \end{aligned}$$

Разумеется, возможны случаи  $M=N$  или  $M < N$ . Тогда алгоритм соответствующим образом изменяется. Транспонированная форма имеет то преимущество, что в ней операции умножения отсчётов входного и выходного сигналов, а также операции суммирования можно производить параллельно; при этом повышается быстродействие алгоритма, хотя увеличиваются аппаратные затраты (нужно иметь несколько одновременно работающих перемножителей и сумматоров).

#### 6.4. Каскадная (последовательная) структура

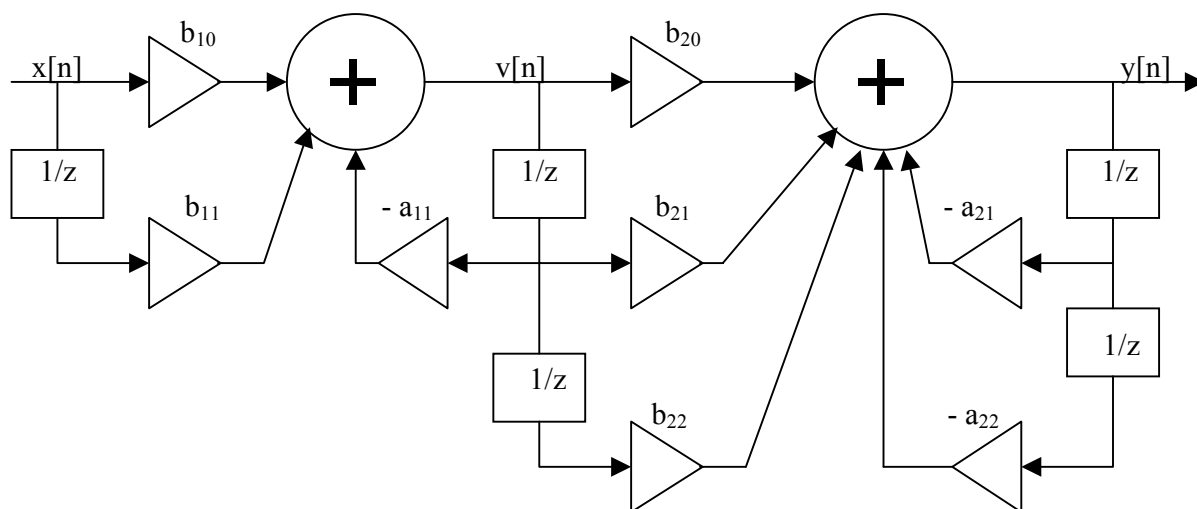
Передаточную функцию  $K(z)$  можно представить в виде произведения передаточных функций (обычно отдельные функции имеют порядок не выше второго):

$$K(z) = K_1(z) K_2(z) \dots K_L(z).$$

Такое представление передаточной функции соответствует каскадному включению цифровых звеньев первого и второго порядка. Пусть, например,

$$K(z) := \frac{b_{10} + b_{11}z^{-1}}{1 + a_{11}z^{-1}} \cdot \frac{b_{20} + b_{21}z^{-1} + b_{22}z^{-2}}{1 + a_{21}z^{-1} + a_{22}z^{-2}}$$

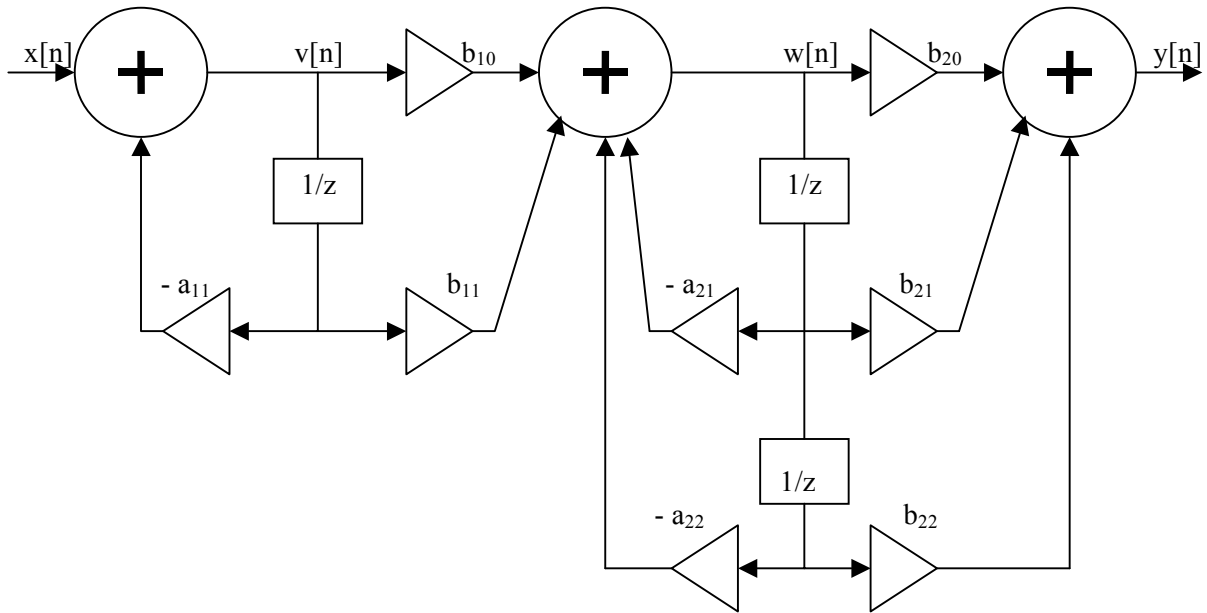
Каскадная структура изображается следующим образом:



Алгоритм:

$$\begin{cases} v[n] = b_{10} x[n] + b_{11} x[n-1] - a_{11} v[n-1] \\ y[n] = b_{20} v[n] + b_{21} v[n-1] + b_{22} v[n-2] - a_{21} y[n-1] - a_{22} y[n-2] \end{cases}$$

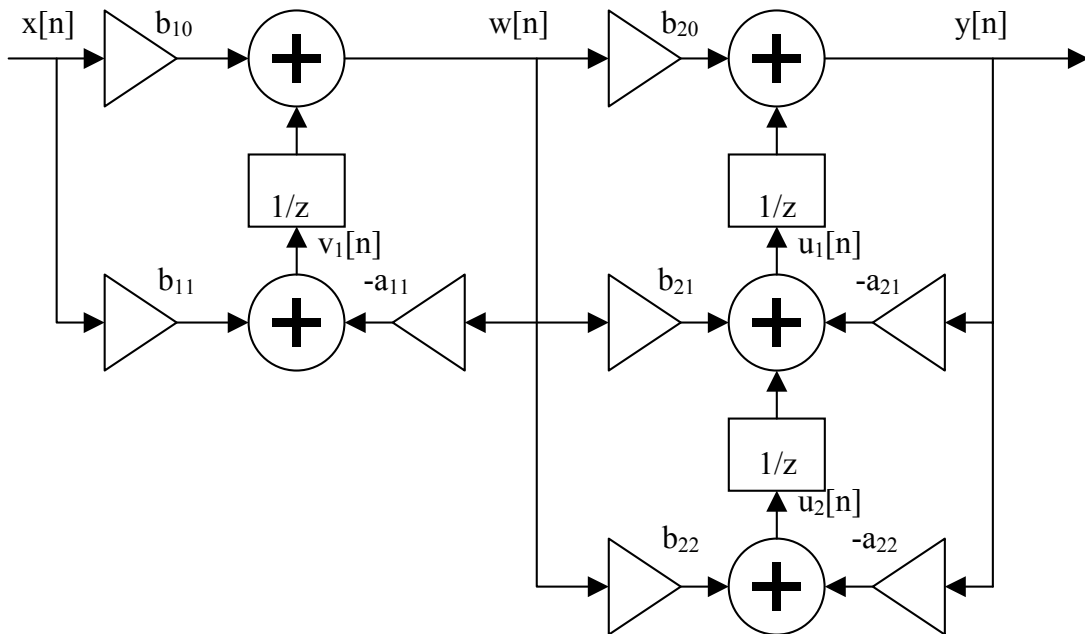
В данной схеме каскады выполнены в виде прямых структур. Возможно их реализовать и в виде канонических форм:



Алгоритм в данном случае содержит три уравнения:

$$\begin{cases} v[n] = x[n] - a_{11} v[n-1] \\ w[n] = b_{10} v[n] + b_{11} v[n-1] - a_{21} w[n-1] - a_{22} w[n-2] \\ y[n] = b_{20} w[n] + b_{21} w[n-1] + b_{22} w[n-2] \end{cases}$$

Можно реализовать каскады и в виде транспонированных структур:



Теперь алгоритм включает в себя пять уравнений:

$$\begin{cases} w[n] = b_{10} x[n] + v_1[n-1] \\ v_1[n] = b_{11} x[n] - a_{11} w[n] \\ y[n] = b_{20} w[n] + u_1[n-1] \\ u_1[n] = b_{21} w[n] - a_{21} y[n] + u_2[n-1] \\ u_2[n] = b_{22} w[n] - a_{22} y[n] \end{cases}$$

Анализ показывает, что каскадная форма рекурсивного фильтра обладает меньшей чувствительностью частотной характеристики цифрового фильтра к точности представления коэффициентов фильтра, чем прямая, каноническая и транспонированная формы. Это означает, что при одинаковой разрядности округлённых коэффициентов  $\mathbf{b}$  и  $\mathbf{a}$  частотная характеристика фильтра в каскадной форме будет в меньшей степени отличаться от расчётной, чем характеристики некаскадных форм.

Для получения коэффициентов передаточных функций для каскадного представления фильтра можно использовать оператор MatLab преобразования фильтра в соединение секций второго порядка (sos – second-order sections):

```
>> [sos,g] = tf2sos (b, a);
```

где  $\mathbf{b}, \mathbf{a}$  – векторы-строки коэффициентов передаточной функции цифрового фильтра,  $\mathbf{sos}$  – шестистолбцовая матрица, каждая строка которой соответствует одной секции и имеет структуру

$$[ b_0 \ b_1 \ b_2 \ 1 \ a_1 \ a_2 ],$$

что соответствует передаточной функции секции вида

$$K(z) := \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

В частном случае какая-либо из секций может иметь первый порядок; тогда соответствующие элементы строки матрицы  $\mathbf{sos}$  будут нулевыми:

$[ b_0 \ b_1 \ 0 \ 1 \ a_1 \ 0 ]$ . Передаточная функция, соответствующая такой строке:

$$K(z) := \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1}}$$

$g$  – масштабный коэффициент, который реализуется в схеме в виде отдельного умножителя либо учитывается путём умножения на него коэффициентов  $\mathbf{b}$  одного из каскадов.

При формировании передаточных функции секций группируются пары комплексно-сопряжённых полюсов с комплексно-сопряжёнными нулями, расположенными наиболее близко к данным полюсам. Действительные полюсы группируются в пары, в которых значения полюсов наиболее близки по модулю. Строки в матрице  $\mathbf{SOS}$  располагаются в порядке приближения полюсов, реализуемых соответствующими секциями, к единичной окружности.

Для вывода рассчитанной матрицы  $\mathbf{SOS}$  на монитор достаточно не ставить точку с запятой в конце оператора или набрать имя матрицы после значка  $\gg$ . Обратное преобразование матрицы  $\mathbf{SOS}$  в коэффициенты  $\mathbf{b}$  и  $\mathbf{a}$  фильтра осуществляется оператором:

```
>> [b, a]= sos2tf (sos, g);
```

### 6.5. Параллельная структура

Представим передаточную функцию  $K(z)$  в виде суммы дробей:

$$K(z) := \frac{r_1}{1-p_1 z^{-1}} + \frac{r_2}{1-p_2 z^{-1}} + \dots + \frac{r_N}{1-p_N z^{-1}} + k_0 + k_1 z^{-1} + \dots + k_{MN} z^{-(M-N)}, \quad (5)$$

где  $r_1, r_2, \dots, r_N$  - вычеты,  $p_1, p_2, \dots, p_N$  – полюсы,  $k_0, k_1, \dots, k_{MN}$  – константы. Последние появляются в разложении, если  $M \geq N$ , то есть прямых связей в структуре фильтра не меньше, чем обратных. Подобное разложение осуществляется в MatLab с использованием оператора `residuez`. После ввода или расчёта коэффициентов системной функции, представленных векторами  $\mathbf{b}$  и  $\mathbf{a}$ , нужно задать:

```
>> [r, p, k]= residuez (b, a)
```

Если не ставить точку с запятой в конце строки и нажать клавишу  $\langle \text{Enter} \rangle$ , то будут выведены значения вектора вычетов  $\mathbf{r}$ , вектора полюсов  $\mathbf{p}$  и вектора коэффициентов  $\mathbf{k}$ . При действительных  $\mathbf{b}$  и  $\mathbf{a}$  значения вычетов и полюсов могут образовывать комплексно-сопряжённые пары или быть действительными. Дроби с комплексно-сопряжёнными значениями вычетов и полюсов нужно объединить в одну дробь второго порядка. Ей будет соответствовать прямая, каноническая или транспонированная структура



второго порядка. Например, применение оператора `residuez` привело к следующему результату:

```
>> [r, p, k]= residuez (b, a)
```

```
r=
```

```
-0.5000 - 0.1000i  
-0.5000 + 0.1000i  
2.1000
```

```
p=
```

```
0.4000 - 0.7000i  
0.4000 + 0.7000i  
0.6000
```

```
k=
```

```
1.5000
```

Для объединения дробей, соответствующих первым двум вычетам и первым двум полюсам в одну дробь второго порядка нужно задать:

```
>> [b1, a1]= residuez (r(1:2), p(1:2), [ ])
```

Здесь функция `residuez` в обратном направлении: объединяет две суммируемые дроби в одну и вычисляет коэффициенты полиномов числителя и знаменателя этой дроби. `[ ]` – символ пустой матрицы (не задаём коэффициент `k`). Будет выведено:

```
b1=
```

```
-1.0000 0.2600
```

```
a1=
```

```
1.0000 -0.8000 0.6500
```

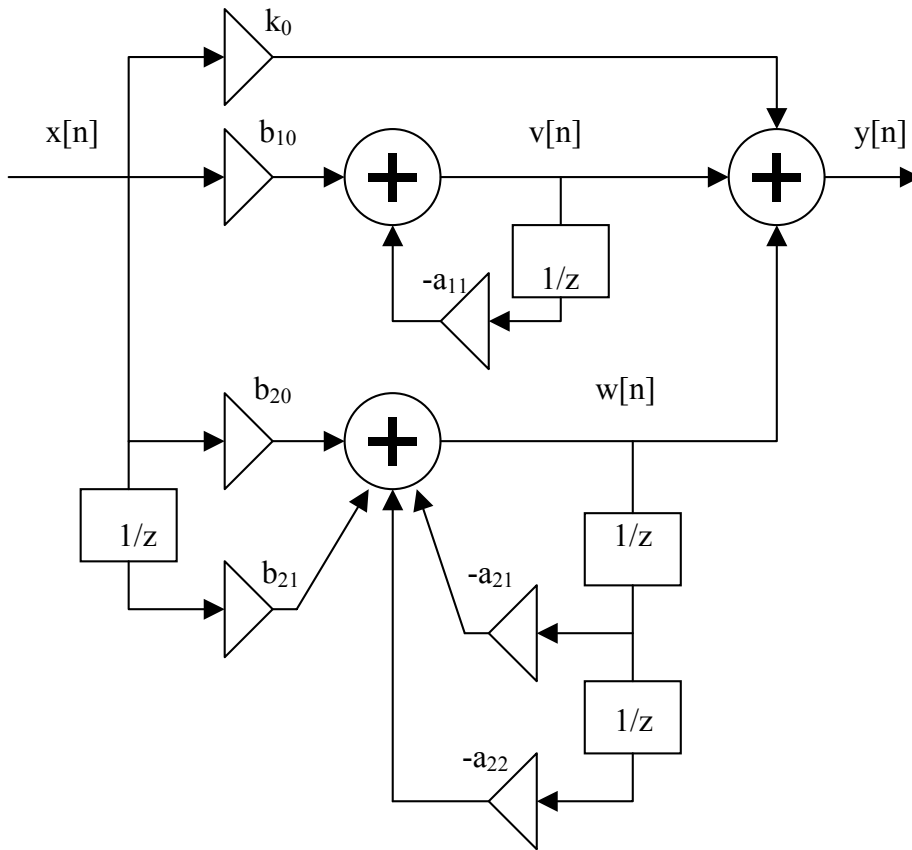
что соответствует дроби

$$\frac{-1 + 0.26z^{-1}}{1 - 0.8z^{-1} + 0.65z^{-2}}$$

Покажем теперь, как изображается параллельная структура и как записать для неё алгоритм. Пусть, например,

$$K(z) := k_0 + \frac{b_{10}}{1 + a_{11}z^{-1}} + \frac{b_{20} + b_{21}z^{-1}}{1 + a_{21}z^{-1} + a_{22}z^{-2}}$$

Такому разложению соответствует схема



Алгоритм:

$$\begin{cases} v[n] = b_{10} x[n] - a_{11} v[n-1] \\ w[n] = b_{20} x[n] + b_{21} x[n-1] - a_{21} w[n-1] - a_{22} w[n-2] \\ y[n] = k_0 x[n] + v[n] + w[n] \end{cases}$$

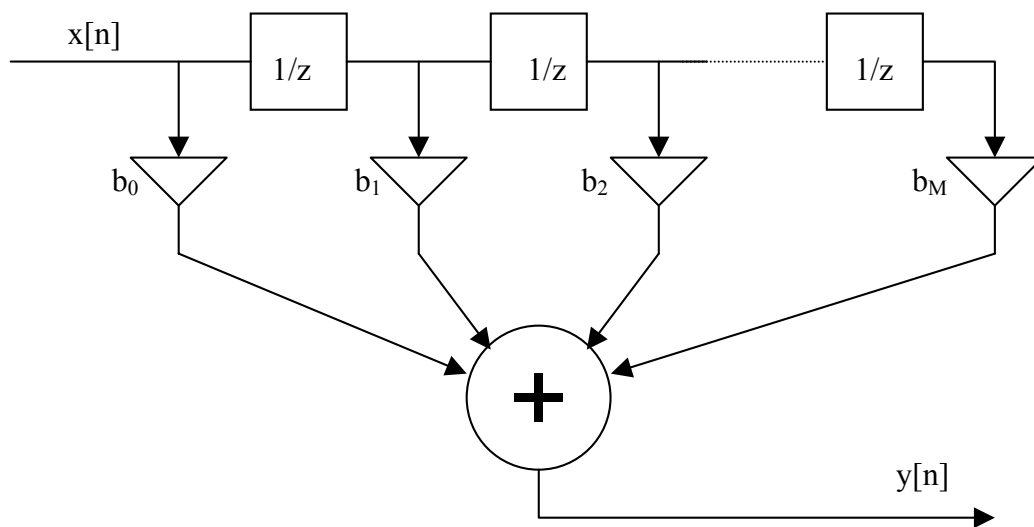
Разумеется, отдельные части схемы можно реализовывать в виде канонических или транспонированных структур (см. подраздел 6.4). Так же как и каскадная форма, параллельная обеспечивает меньшую чувствительность частотных и временных характеристик фильтра к точности представления коэффициентов по сравнению с прямой, канонической и транспонированной структурами. Это позволяет при реализации цифрового фильтра в виде специализированного вычислительного устройства обеспечивать заданные допуски на отклонение характеристик от расчётных при меньшем количестве двоичных разрядов, используемых для представления коэффициентов фильтра.

### 6.6. Нерекурсивный фильтр

Структура нерекурсивного не содержит обратных связей. Значит, все коэффициенты  $a_k$  равны нулю, кроме  $a_0=1$ . Передаточная функция такого фильтра

$$K(z)=b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_Mz^{-M}$$

Схема:



Алгоритм цифровой фильтрации:

$$y[n]= b_0x[n]+b_1x[n-1]+b_2x[n-2]+\dots +b_Mx[n-M].$$

## 7. Просмотр характеристик синтезированного цифрового фильтра.

Для просмотра частотных и временных характеристик синтезированного цифрового фильтра используют оператор

```
>> fvtool (b, a)
```

После задания функции `fvtool` (filter visualization tool) нужно нажать клавишу `<Enter>`. Аргументы функции – коэффициенты числителя и знаменателя передаточной функции, расположенные в порядке возрастания отрицательных степеней  $z$  (см. выражение (3)). Для просмотра характеристик нескольких фильтров одновременно нужно указать несколько пар векторов в списке входных параметров функции `fvtool`:

```
>> fvtool (b1, a1, b2, a2, b3, a3)
```

После вызова функции `fvtool` откроется графическое окно. На панели инструментов в верхней части этого окна имеются значки характеристик фильтра. Подводя курсор к одному из значков и нажимая левую кнопку мыши, можно вывести на экран график желаемой характеристики. Можно просмотреть

- амплитудно-частотную характеристику (АЧХ, *magnitude response*),
- фазочастотную характеристику (ФЧХ, *phase response*),
- АЧХ и ФЧХ совместно,
- характеристику группового времени запаздывания (*group delay*),
- импульсную характеристику (*impulse response*),
- переходную характеристику (*step response*),
- диаграмму полюсов и нулей (*pole/zero plot*),
- коэффициенты фильтра (*filter coefficients*).

Для детального просмотра участков графика используется кнопка `zoom in` панели инструментов. Щёлкнув по этой кнопке, нужно затем подвести курсор мыши к той точке графика, которая должна оказаться в центре увеличенного изображения и нажать левую кнопку мыши. Можно щёлкать левой кнопкой мыши многократно, что будет приводить ко всё большему увеличению. Возврат к исходному состоянию осуществляется щелчком по правой кнопке мыши, что соответствует нажатию кнопки `zoom out` панели инструментов.

## 8. Синтез цифрового фильтра с использованием программы `fdatool`

Расчёт коэффициентов передаточной функции цифрового фильтра при заданных требованиях к частотной характеристике может быть произведён с использованием тех же самых функций, которые применяются для расчёта аналоговых фильтров (см. раздел 4). Отличие заключается в том, что строковый параметр 's' вводить не нужно. Кроме того, все частоты ( $\omega_0$ ,  $\omega_p$ ,  $\omega_r$ ,  $\omega_s$ ) указываются в долях от частоты Найквиста ( $F_s/2$ ), то есть лежат в интервале от 0 до 1. Есть и специальный пакет программ, где собраны многие из функций, рассмотренных нами по отдельности. Он носит название `fdatool` (*filter design & analysis tool*). Этот пакет содержит удобный пользовательский интерфейс и позволяет производить расчёт передаточной функции рекурсивных и нерекурсивных фильтров разнообразными методами синтеза, просматривать характеристики фильтра, анализировать изменение характеристик при квантовании коэффициентов фильтра, отсчётов входного сигнала и результатов промежуточных вычислений. Возможна работа с различными структурами фильтра. Вызов пакета осуществляется путём ввода его имени в командном окне MatLab:

```
>> fdatool
```

После ввода имени с клавиатуры следует нажать клавишу <Enter>.

На экране монитора появится окно программы fdatool. Кроме основного меню и панели инструментов здесь содержится текущая информация о структуре фильтра, его порядке, устойчивости (Current Filter Information); график допусков для АЧХ (в дБ) (Filter Specifications), а также вкладка Design Filter для задания типа фильтра, его класса, метода синтеза, порядка фильтра, частоты дискретизации, граничных частот полос пропускания и задерживания и допустимых затуханий в этих полосах. Кроме того, имеется вкладка для исследования эффектов квантования (Set Quantization Parameters).

### 8.1. Задание требований к АЧХ и расчёт фильтра

На вкладке Design Filter в нижней части окна установите переключатель Filter Type в одно из следующих положений: Lowpass (ФНЧ), Highpass (ФВЧ), Bandpass (ППФ) или Bandstop (ПЗФ). Затем используйте переключатель Design Method. Если выбрать рекурсивный фильтр, иначе БИХ-фильтр (IIR – Infinite Impulse Response), то далее в раскрывающемся списке нужно указать класс фильтра (Batterworth (Баттерворта), Chebyshev Type I (Чебышёва), Chebyshev Type II (инверсный Чебышёва), Elliptic (эллиптический)). При синтезе этих фильтров используется метод билинейного z-преобразования. В случае синтеза нерекурсивного фильтра (КИХ-фильтра, FIR – Finite Impulse Response) возможны методы: Equiripple (метод Ремеза, обеспечивающий равномерные пульсации АЧХ), Least-Squares (обеспечение минимума среднего квадратического отклонения АЧХ от заданной), Window (использование окон в качестве весовых функций при синтезе фильтра) и др.

В разделе Filter Order укажите требуемый порядок фильтра или установите переключатель в положение Minimum order (наименьший возможный порядок).

Далее перейдите к разделам Filter Specifications и Magnitude Specifications. Последовательно подведите курсор мыши к полям ввода параметров и вводите желаемые значения с клавиатуры. Смысл параметров можно понять из расположенного в верхней части окна графика допусков (Filter Specifications). Следует ввести частоту дискретизации  $F_s$ , граничные частоты полосы пропускания и полосы задерживания ( $F_{pass}$  и  $F_{stop}$ ), допустимые затухания в полосе пропускания и в полосе задерживания ( $A_{pass}$  и  $A_{stop}$ ).

После задания всех параметров щёлкните по кнопке Design Filter, расположенной в самом низу. Будет произведён расчёт, после чего можно просмотреть характеристики синтезированного фильтра.

## 8.2 Просмотр характеристик фильтра

Вывести на экран частотные и временные характеристики фильтра, диаграмму полюсов и нулей, коэффициенты фильтра можно точно так же, как это делается в программе fvtool (см. раздел 7). Для перехода к просмотру характеристик удобно вывести на экран специальное окно просмотра путём нажатия кнопки Full View Analysis панели инструментов.

## 8.3 Экспорт и импорт описания фильтра

Выберите команду Export меню File (<Ctrl>+E). В появившемся окне **экспорта** укажите область, куда будут переданы коэффициенты фильтра (Workspace – рабочая область MatLab, Text-file – текстовый файл, MAT-file – MAT-файл, который затем можно загрузить в MatLab командой load). Задайте имена переменных для записи векторов коэффициентов числителя (Numerator) и знаменателя (Denominator) передаточной функции (по умолчанию Num и Den). Если фильтр имеет каскадную структуру (second-order-sections), то экспортируется матрица SOS и коэффициент усиления G. Изменить структуру фильтра можно щёлкнув по кнопке Convert Structure и далее произведя выбор из списка (в версии MatLab 6.1). В версии MatLab 6.5 изменение структуры осуществляется путём выбора соответствующей команды в меню Edit. Если в рабочей области MatLab уже есть переменные с указанными именами, то установите флажок Overwrite existing variables, чтобы их значения были заменены новыми. Щёлкните по кнопке Apply. При экспорте в файл будет запрошено имя файла.

Если программу fdatool нужно использовать не для расчёта, а только для анализа характеристик фильтра, в том числе для анализа их изменения при квантовании коэффициентов и переменных фильтра, то применяется процедура **импорта** описания фильтра. Выберите команду Import Filter (<Ctrl>+ I) меню Filter (в версии MatLab 6.1) или меню File (в версии MatLab 6.5). Вкладка Design Filter окна программы fdatool будет заменена на вкладку Import Filter. Укажите тип структуры (Filter Structure), выбрав нужную строку в раскрывающемся списке. В полях ввода укажите коэффициенты числителя и знаменателя передаточной функции фильтра. Список заключается в квадратные скобки, значения коэффициентов разделяются пробелами. Например, если передаточная функция имеет вид

$$K(z) = \frac{0.2 - 0.42z^{-1} + 0.05z^{-2}}{1 + 0.18z^{-1} - 0.24z^{-2} + 0.081z^{-3}},$$

то в поле Numerator следует ввести [0.2 -0.42 0.05], а в поле Denominator [1 0.18 -0.24 0.081].

В поле Sampling Frequency введите частоту дискретизации  $F_s$ , указав единицу измерения в поле Units.

Если векторы коэффициентов фильтра  $\mathbf{b}$  и  $\mathbf{a}$ , а также частота дискретизации  $F_s$  уже существуют в рабочей области MatLab, то вместо ввода численных значений в соответствующих полях ввода нужно просто указать имена переменных.

Щёлкните по кнопке Import Filter. Далее можно просматривать характеристики фильтра.

Для возврата в режим расчёта фильтров используется команда Design Filter меню Filter (в версии MatLab 6.1) или кнопка с тем же названием, расположенная в вертикальном ряду кнопок у левой границы окна (в версии MatLab 6.5).

## **9. Исследование влияния квантования сигналов и коэффициентов фильтра на характеристики фильтра**

### **9.1. Влияние квантования коэффициентов на АЧХ и другие характеристики, а также на устойчивость фильтра**

Округление коэффициентов  $\mathbf{b}$  и  $\mathbf{a}$  передаточной функции цифрового фильтра, т.е. представление их числами с ограниченным количеством двоичных разрядов, приводит к искажению частотных и временных характеристик фильтров. Искажения могут быть как незначительными, так и более существенными. При проектировании цифровых фильтров желательно установить, какова минимально допустимая разрядность представления коэффициентов фильтра, при которой характеристики не выходят за пределы заданных полей допусков, а дисперсия шума квантования при обработке какого-либо цифрового сигнала данным фильтром не превышает допустимой величины. Если коэффициенты и переменные фильтра представлены числами с меньшим количеством двоичных разрядов, то достигается экономия памяти и повышается быстродействие фильтра, поэтому нахождение наименьшей разрядности – весьма важная задача.

Для исследования эффектов квантования можно использовать программу fdatool, речь о которой уже шла в разделе 8.

Вызовите программу fdatool:

```
>>fdatool
```

Произведите расчёт фильтра или импортируйте его описание (см. разд. 8). Установите флажок Turn quantization on в поле Quantization, расположенном в левой верхней части окна (в версии MatLab 6.1) или нажмите соответствующую кнопку на панели инструментов (в версии MatLab 6.5). Перейдите на вкладку Set quantization parameters (в нижней части окна). Задайте нужную структуру, щёлкнув по кнопке Convert structure,

расположенной в поле Current Filter Information, и далее произведя выбор из раскрывающегося списка (в версии Matlab 6.1), или задайте команды Convert Structure, Convert to Second-Order Sections или Convert to Single Section меню Edit (в версии MatLab 6.5). Возможно исследование структур: Direct form I (прямая), Direct form II (каноническая), Direct form I transposed (транспонированная каноническая), Direct form II transposed (транспонированная прямая), Second-order sections (каскадная). В правой части раздела Set Quantization Parameters приведён набор параметров квантователей, которые используются программой. Изменять нужно только значение в поле Format строки Coefficient (Convert coefficient to) (первая строка, последний столбец). Здесь задаётся формат двоичного представления коэффициентов фильтра. Например, формат [16 15] означает, что всего разрядов 16, а 15 из них отводится под представление дробной части (нужно учесть, что старший разряд – знаковый). Такой формат применим, если коэффициенты представлены в нормализованной форме (их модули не превышают единицы). Просмотрите список коэффициентов, нажав кнопку [b a] панели инструментов, расположенной в верхней части окна, и оцените, сколько двоичных разрядов нужно отводить на представление целой части (с учётом знакового), чтобы правильно представить целую часть коэффициента с наибольшим модулем. Последовательно изменяйте формат, уменьшая количество разрядов (разность между первым и вторым числами формата должна при этом сохраняться неизменной). Для изменения формата подведите курсор мыши к полю ввода (курсор преобразуется в вертикальную черту) и нажмите левую кнопку мыши. Затем, используя клавишу ← и цифровые клавиши, введите новые значения чисел формата. После ввода щёлкните левой кнопкой мыши вне таблицы параметров квантователей, чтобы активизировать кнопку Apply. Щёлкните по кнопке Apply. Просмотрите характеристики фильтра, нажимая соответствующие кнопки панели инструментов. Выводятся сразу две характеристики: исходная (Reference) и полученная в результате квантования (Quantized). Просмотрев критические участки АЧХ в укрупнённом масштабе (используйте кнопки изменения масштаба Zoom In и Zoom Out), определите, не выходит ли АЧХ за пределы заданного поля допусков. Если АЧХ ещё удовлетворяет требованиям, продолжите изменение формата представления коэффициентов. В результате анализа найдите, при каком наименьшем количестве двоичных разрядов АЧХ ещё удовлетворяет предъявляемым к ней требованиям. Посмотрите, как изменяется диаграмма полюсов и нулей, какими стали коэффициенты фильтра. Интересно пронаблюдать и изменения других характеристик (ФЧХ, импульсной, переходной, группового времени запаздывания).

Проведите исследование для различных структур, в том числе и для каскадной. Укажите, для какой структуры разрядность коэффициентов наименьшая.



Если при уменьшении формата каждый раз на единицу характеристики меняются незначительно, используйте более эффективные процедуры, например, метод дихотомии. Так после формата [16 15] задайте формат [8 7]. Если АЧХ укладывается в поле допусков, перейдите к формату [4 3], а если не укладывается, к формату [12 11], и т.д., пока не будет найден формат с минимально допустимым количеством двоичных разрядов.

При малой разрядности коэффициентов возможна потеря устойчивости фильтра, вследствие того что модули некоторых полюсов из-за округления могут превысить единицу (в особенности тех из них, которые, обладая высокой добротностью, располагаются близко к единичной окружности). Информация об устойчивости или неустойчивости фильтра заносится в раздел Current Filter Information основного окна fdatool (Stable: Yes или Stable: No). Проводя исследование эффектов квантования, следует отмечать, сохраняет ли фильтр устойчивость при последовательном уменьшении разрядности его коэффициентов.

## 9.2. Расчёт наименьшей разрядности входного сигнала и выходных регистров умножителей цифрового фильтра.

В качестве исходных данных для расчёта наименьшего количества двоичных разрядов, отводимых для представления отсчётов входного сигнала цифрового фильтра, используются **динамический диапазон сигнала**

$$D = 20 \lg (A_{\max} / A_{\min}) \quad [\text{дБ}],$$

где  $A_{\max}$  и  $A_{\min}$  – максимальная и минимальная амплитуды входного сигнала, и **отношение сигнал/шум квантования** на выходе фильтра

$$R = 10 \lg (P_c / P_{\text{ш}}) \quad [\text{дБ}],$$

где  $P_c$  – минимальная мощность сигнала (в случае синусоидального сигнала  $P_c = A_{\min}^2 / 2$ ),  $P_{\text{ш}} = \sigma_{\text{вых}}^2$  – допустимая мощность (дисперсия) шума квантования на выходе фильтра.

Примем, что разрядность входного сигнала и разрядность умножителей цифрового фильтра одинаковы. Кроме того, будем полагать, что входной сигнал масштабирован, и его максимальная амплитуда  $A_{\max} = 1$ . Масштабированы пусть будут и коэффициенты фильтра, так что максимальное значение коэффициента передачи в полосе пропускания равно единице. В этом случае можно считать, что

$$P_{\text{ш}} = A_{\max}^2 / 2 \cdot 10^{-(R+D)/10} \quad (6)$$

В соответствии с линейной шумовой моделью фильтра на входе фильтра и на выходе каждого умножителя «точно» представленные отсчёты сигналов суммируются с шумом квантования:

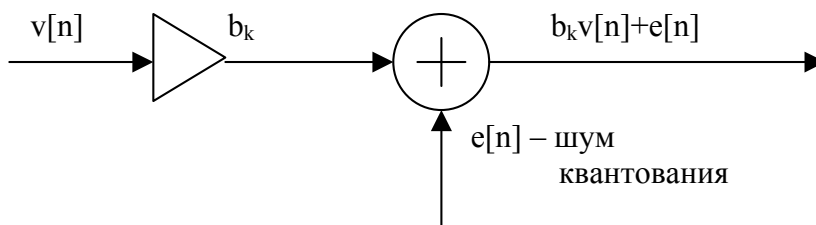


Схема фильтра, таким образом, включает в себя несколько источников шума квантования. Их количество равно числу умножителей плюс единица (учитывается шум квантования входного сигнала). Все источники шума считаются независимыми. В случае округления результатов умножений дисперсия шума квантования равна

$$\sigma^2 = 2^{-2p}/12, \quad (7)$$

где  $p$  – количество разрядов сигнала на выходе умножителя (без учёта знакового). На выходе фильтра каждый источник шума квантования создаёт шум с дисперсией  $\sigma_{\text{вых } i}^2 = \sigma^2 \sum_n (g_i[n])^2$ , где  $g_i[n]$  – импульсная характеристика

части фильтра от  $i$ -го источника шума до выхода. Суммирование квадратов отсчётов импульсной характеристики ведётся для всех номеров  $n$ , при которых значения  $g_i[n]$  существенны (не являются пренебрежимо малыми). В силу независимости источников шума полная дисперсия шума квантования на выходе фильтра равна сумме дисперсий отдельных источников:  $\sigma_{\text{вых}}^2 = \sum_i \sigma_{\text{вых } i}^2$ . В результате анализа, основанного на изложенном подходе,

можно определить, как связаны дисперсии шума квантования на входе и выходе цифрового фильтра для различных структур фильтра.

### 1) Прямая и транспонированная структуры.

$$\sigma_{\text{вых}}^2 = \sigma_{\text{вх}}^2 (\sum (g[n])^2 + (k+m) \sum (g_{\text{рек}}[n])^2), \quad (8)$$

где  $k$  – количество умножителей в обратных связях (с коэффициентами  $a$ ),

$m$  – количество умножителей в прямых связях (с коэффициентами  $b$ ),

$g_{\text{рек}}[n]$  – импульсная характеристика рекурсивной части фильтра.

В случае прямой структуры, как видно из её схемы (см. подраздел 6.1), шум всех умножителей проходит только через рекурсивную часть (умножители с коэффициентами  $\mathbf{a}$ ), в то время как входной шум проходит через весь фильтр. То же самое можно сказать и о транспонированной структуре (см. подраздел 6.3). Дисперсия шума умножителей равна дисперсии входного шума  $\sigma_{\text{вх}}^2$ , поскольку, как указывалось выше, отсчёты сигнала везде представлены одинаковым количеством разрядов  $p$ . Следует отметить, что числа  $k$  и  $m$  необязательно равны количеству коэффициентов  $\mathbf{a}_k$  и  $\mathbf{b}_k$ , т.е. числам  $N$  и  $M+1$  соответственно, так как некоторые из коэффициентов могут быть нулевыми или равняться единице. В этих случаях умножители не применяются.

## 2) Каноническая структура.

$$\sigma_{\text{вых}}^2 = \sigma_{\text{вх}}^2 ((k+1)\sum(g[n])^2 + m). \quad (9)$$

Анализ канонической структуры (см. подраздел 6.2) показывает, что входной шум и шум умножителей рекурсивной части (коэффициенты  $\mathbf{a}$ ) проходят через весь фильтр, а шум умножителей с коэффициентами  $\mathbf{b}$  непосредственно проходит на выход.

## 3) Каскадная структура со звеньями в виде прямых или транспонированных структур.

$$\sigma_{\text{вых}}^2 = \sigma_{\text{вх}}^2 c_1 c_2 \dots c_L, \quad (10)$$

$$\text{где } c_i = \sum(g_i[n])^2 + (k_i + m_i)\sum(g_{\text{рек } i}[n])^2, \quad i=1,2,\dots,L, \quad (11)$$

$L$  – количество каскадов.

Шум квантования, прошедший через первый каскад, характеризуется дисперсией  $\sigma_{\text{вх}}^2 c_1$ . Этот шум является входным для следующего каскада, поэтому дисперсия шума на выходе второго каскада  $\sigma_{\text{вх}}^2 c_1 c_2$  и т.д. Из этого рассуждения становится понятным, каким образом составлено выражение (10).

## 4) Каскадная структура со звеньями в виде канонических структур.

$$\sigma_{\text{вых}}^2 = \sigma_{\text{вх}}^2 c_1 c_2 \dots c_L,$$

$$\text{где } c_i = (k_i + 1) \sum(g_i[n])^2 + m_i, \quad i=1,2,\dots,L \quad (12)$$

## 5) Параллельная структура со звеньями в виде прямых или транспонированных структур.

$$\sigma_{\text{вых}}^2 = \sigma_{\text{вх}}^2 (c_1 + c_2 + \dots + c_L), \quad (13)$$

где  $c_i$  определяются выражением (11),  $L$  – количество параллельно включённых звеньев.

**6) Параллельная структура со звеньями в виде канонических структур.**

$$\sigma_{\text{ВЫХ}}^2 = \sigma_{\text{ВХ}}^2 (c_1 + c_2 + \dots + c_L),$$

где  $c_i$  определяются выражением (12),  $L$  – количество параллельно включённых звеньев.

**7) Нерекурсивный фильтр.**

$$\sigma_{\text{ВЫХ}}^2 = \sigma_{\text{ВХ}}^2 (\sum (g[n])^2 + m) = \sigma_{\text{ВХ}}^2 (\sum b_k^2 + m) \quad (14)$$

(см. рис. на с.19) .

Допустимую дисперсию (среднюю мощность ) шума квантования на выходе можно рассчитать по формуле (6), в которой положить  $A_{\text{max}}=1$ . Затем из формул (8) – (14) выразить дисперсию входного шума квантования  $\sigma_{\text{ВХ}}^2$ , предварительно рассчитав отношение дисперсий для нужной структуры в соответствии с приведёнными выражениями. Далее на основании выражения (7) получаем наименьшее количество двоичных разрядов:

$$p = \text{int} [ 0.5 \log_2 (1/(12\sigma_{\text{ВХ}}^2)) ] + 1, \quad (15)$$

где  $\text{int} [\cdot]$  – операция взятия целой части. С учётом знакового разряда нужно полученное по формуле (15) значение увеличить ещё на единицу.

При работе в среде MatLab для расчёта наименьшей разрядности сигнала и выходных регистров умножителей цифрового фильтра можно применить программу `minubit`. Она вызывается следующим образом:

```
>> minubit (b, a, D, R)
```

Здесь  $\mathbf{b}$  и  $\mathbf{a}$  – векторы коэффициентов передаточной функции фильтра,  $D$  – динамический диапазон входного сигнала [дБ],  $R$  – допустимое отношение сигнал/ шум квантования на выходе фильтра [дБ].

Программа рассчитывает наименьшее количество разрядов (с учётом знакового) для структур перечисленных выше типов. Если фильтр рекурсивный, то производится расчёт для девяти структур (см. с. 26 – 28). Если фильтр нерекурсивный, то для одной структуры (см. с.28 и рис. на с.19). Результаты расчётов выводятся в командное окно по завершении работы программы. Кроме наименьшей разрядности приводятся также значения дисперсии шума квантования на входе и выходе фильтра. Анализируя полученные результаты, можно выбрать оптимальную структуру, обеспечивающую заданное отношение сигнал/ шум квантования на выходе фильтра при заданном динамическом диапазоне и позволяющую установить самую маленькую разрядность отсчётов сигнала по сравнению с другими структурами.

### 9.3. Расчёт дисперсии шума квантования на выходе фильтра при заданной разрядности отсчётов сигнала

Поставим теперь задачу несколько иначе. Пусть разрядность входного сигнала, а также разрядность сигналов на выходах умножителей известна. Нужно рассчитать дисперсию шума квантования на выходах различных структур цифрового фильтра, обладающих одной и той же передаточной функцией  $K(z)$ . Поскольку дисперсия входного шума однозначно определяется количеством разрядов (см. (7)), то она одинакова для всех структур, а так как выражения (8) – (14), связывающие дисперсии шума квантования на входе и выходе различны, получается, что разные структуры будут давать на выходе шум квантования различной средней мощности.

Чтобы произвести расчёт, вызовите программу `quanod`:

```
>> quanod (b, a, p)
```

Здесь **b**, **a** – векторы коэффициентов передаточной функции цифрового фильтра; **p** – разрядность сигнала. Программа рассчитывает и выводит в командное окно дисперсию шума квантования на выходе фильтра для девяти перечисленных выше структур рекурсивного фильтра или для нерекурсивного фильтра, если задан именно он. Выводится также и значение дисперсии шума квантования входного сигнала. На основании анализа полученных результатов можно выбрать оптимальную структуру, обеспечивающую наименьшую дисперсию шума квантования на выходе.

## **10. Моделирование работы цифрового фильтра**

Моделирование работы цифрового фильтра предполагает задание тестового сигнала, использование его отсчётов в качестве входных в алгоритме цифровой фильтрации, нахождение выходного сигнала и сравнение его с входным. Кроме того, полезно рассмотреть спектры входного и выходного сигналов и сопоставить их с частотной характеристикой фильтра.

### 10.1. Задание тестовых сигналов

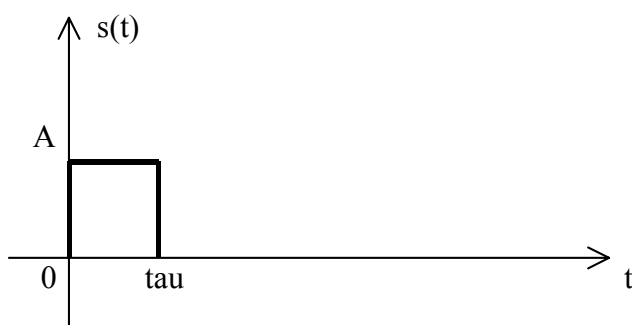
Данная процедура осуществляется в рабочей области MatLab. Сигнал задаётся в виде вектора, сопоставленного с вектором моментов времени. Перед вводом непосредственно модели сигнала нужно указать частоту дискретизации и сформировать вектор-столбец моментов времени. Например,

```
>> Fs = 1e3; t=0:1/Fs:1; t=t';
```

В данном случае введена частота дискретизации 1кГц. Сигнал будет задан на интервале времени 1с (1001 отсчёт). Последний оператор означает преобразование вектора-строки в вектор-столбец ( ' – операция транспонирования матрицы). Не следует забывать ставить точку с запятой в конце каждого оператора, чтобы подавить вывод значений на экран монитора.

Рассмотрим некоторые из возможных сигналов.

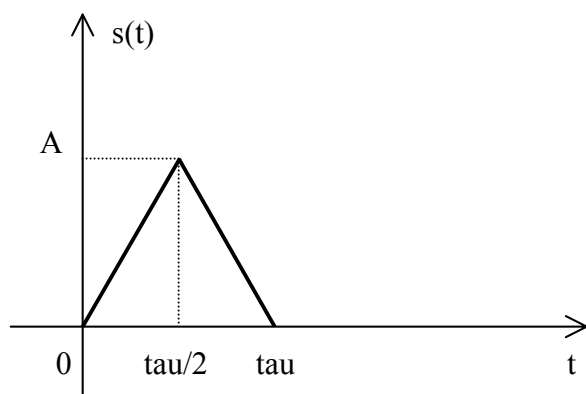
**а) Прямоугольный импульс.**



`>> s=A* rectpuls (t – tau/2, tau);`

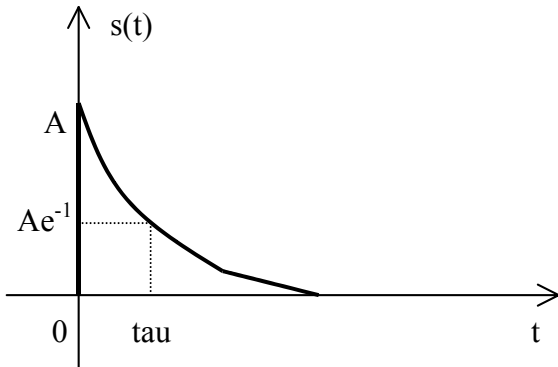
При вводе этого оператора либо нужно предварительно задать значения амплитуды  $A$  и длительности  $\tau$ , либо в самом операторе вместо идентификаторов  $A$  и  $\tau$  поставить численные значения.

**б) Треугольный импульс.**



`>> s= A * tripuls (t – tau/2, tau);`

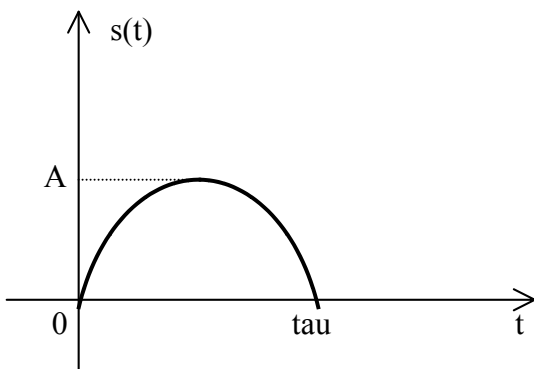
**в) Экспоненциальный импульс.**



$$\gg s = A * \exp(-t / \tau);$$

Подразумевается, что вектор  $t$  задан для моментов времени  $t \geq 0$ .

**г) Синусоидальный импульс.**



$$\gg s = A * \sin(\pi * t / \tau) .* (t \geq 0) .* (t \leq \tau);$$

Здесь используется тот факт, что операции сравнения возвращают 1, если неравенство выполняется, или 0 в противном случае.

**д) Радиоимпульсы.**

Они получаются при умножении видеоимпульса  $s$  на гармоническое колебание:

$$\gg s1 = s .* \cos(2 * \pi * f0 * t + \phi);$$

Предварительно нужно задать значение несущей частоты  $f_0$  и начальной фазы  $\phi$ . Обратите внимание, что операция умножения представлена здесь как  $.*$  (точка перед знаком  $*$ ). Это означает поэлементное умножение

векторов (в противном случае производились бы операция матричного умножения). В тех случаях, когда осуществляется умножение скаляров или матрицы (вектора) на скаляр, можно использовать символ `*`. То же самое относится к операции деления и возведения в степень. Поэлементное деление матриц задаётся оператором `./`, поэлементное возведение в степень `.^`. Число  $\pi$  задаётся в MatLab как `pi`.

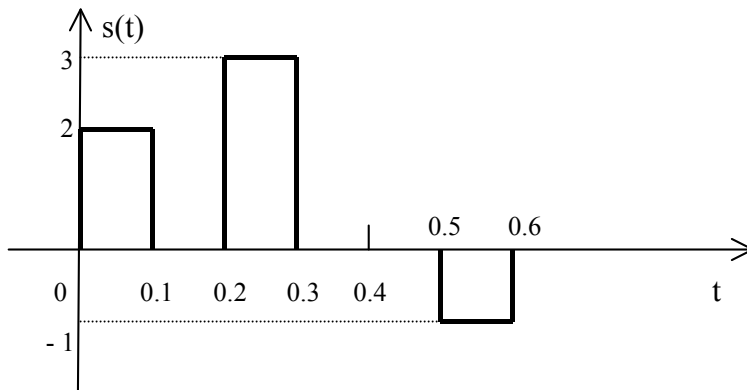
### е) Пачки импульсов.

Для генерации конечной последовательности (пачки) импульсов одинаковой формы с произвольно задаваемыми задержками и амплитудами используется функция `pulstran`. Она вызывается следующим образом:

```
s = pulstran (t, d, 'func', p1, p2 ...)
```

Здесь `t` - вектор значений моментов времени, `d` - вектор задержек и амплитуд импульсов, `'func'` - имя функции, задающей одиночный импульс, например, `'rectpuls'` или `'tripuls'`; `p1, p2 ...` - параметры одиночного импульса, передаваемые функции `func`.

Например, нужно задать следующую последовательность прямоугольных импульсов:



Вводится набор операторов:

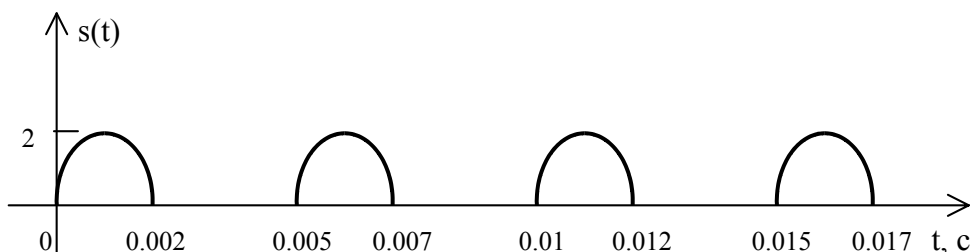
```
>> Fs= 1e3; t= 0:1/Fs:1; t= t';  
>> tau= 0.1;  
>> d(:,1)= [0.05 0.25 0.55]';  
>> d(:,2)= [2 3 -1]';  
>> s= pulstran (t, d, 'rectpuls', tau);
```



Если нужно построить последовательность импульсов произвольной формы причём отсчёты одиночного импульса записаны в векторе  $s1$ , то используют следующую форму задания функции `pulstran`:

$s = \text{pulstran}(t, d, s1, Fs)$ ;

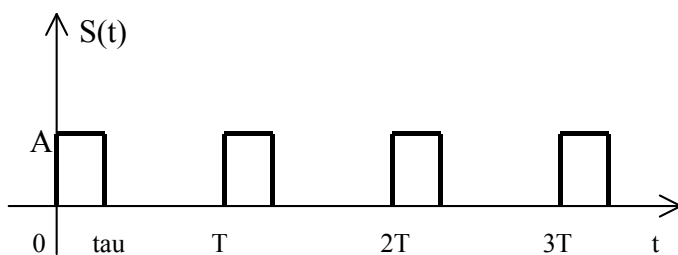
Например, нужно задать пачку из четырёх синусоидальных импульсов:



Вводятся следующие операторы:

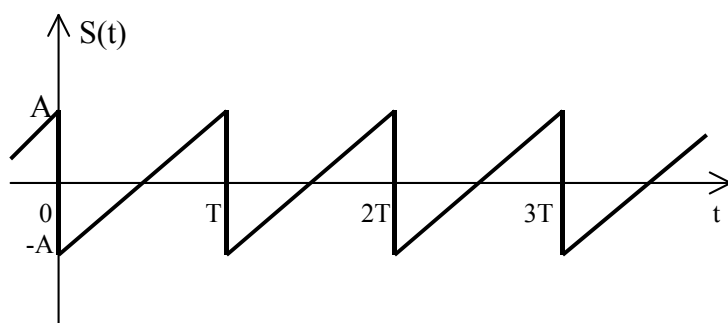
```
>> Fs= 1e4;  
>> t= 0:1/Fs:2e-2; t=t';  
>> tau= 2e-3; A= 2;  
>> s1= sin(pi * t / tau) .* (t<=tau);  
>> d(:,1)= (0:3)' * 5e-3;  
>> d(:,2)= A*ones(4,1);  
>> s= pulstran(t, d, s1, Fs);
```

**ж) Периодическая последовательность прямоугольных импульсов.**



```
>> s= A/2 * (1+square(2*pi*t / T, tau / T*100));
```

**з) Пилообразный сигнал.**



```
>> s = A * sawtooth (2*pi*t / T);
```

**и) Бигармонический сигнал.**

```
>> s = A1 *cos (2*pi*f1*t+phi1) + A2 *cos (2*pi*f2*t+phi2);
```

**к) Амплитудно-модулированный сигнал.**

```
>> s = A*(1+M*cos (2*pi*F*t+psi)).*cos (2*pi*f0*t+phi);
```

**л) Частотно-модулированный сигнал.**

```
>> s = A*cos (2*pi*f0*t + m*cos (2*pi*F*t + psi) + phi);
```

**м) Случайные сигналы.**

Для генерации случайной последовательности с равномерным законом распределения вероятностей служит функция `rand`, с нормальным законом распределения – функция `randn`.

Например, нужно задать нормальный гауссовский белый шум с нулевым математическим ожиданием и дисперсией  $\sigma^2=4$ .

```
>> Fs= 1e3;  
>> t=0:1/Fs:1;  
>> sigma= 2;  
>> n= sigma * randn ( length(t), 1);
```

В дальнейшем можно создавать зашумлённые сигналы:

```
>> sn= s+n;
```

В качестве сигнала `s` можно взять одну из функций, описанных выше.

## 10.2. Построение графика тестового сигнала

График тестового сигнала можно построить, используя функции `plot`, `stem` или `stairs`:

```
>> plot (t, s)  
>> plot (t, s, '.')  
>> stem(t, s)  
>> stairs (t, s)
```

Первая функция строит график сигнала, в котором все отсчётные значения соединены отрезками прямых (получается график соответствующего аналогового сигнала при достаточно малом интервале дискретизации). Вторая функция имеет «точечный» вид (отсчёты сигнала не соединены линиями). Третья функция представляет отсчёты вертикальными линиями с кружком на конце. Четвёртая функция представляет сигнал в виде ступенчатой линии. Можно ограничить диапазон выводимых значений. При этом указывается интервал номеров отсчётов:

```
>> plot (t(1:100), s(1:100))
```

Будут выведены значения сигнала с 1-го по 100-й отсчёт. Можно управлять диапазоном значений, отложенных по осям абсцисс и ординат графика. Для этого используются функции:

**axis** ([xmin xmax ymin ymax]) – задание пределов по оси абсцисс (от xmin до xmax) и по оси ординат (от ymin до ymax); числовые значения в списке разделяются пробелами;

**xlim** ([xmin xmax]) – задание пределов по оси абсцисс;

**ylim** ([ymin ymax]) – задание пределов по оси ординат.

Добавление сетки производится оператором **grid** (**grid on** – включение, **grid off** – выключение). Можно также управлять цветом, типом линий и маркерами точек данных. Для этого используется строковый параметр в списке входных аргументов функций **plot**, **stem**, **stairs**.

Управление цветом:

**b** – синий (blue) (по умолчанию),

**c** – голубой (cyan),

**g** – зелёный (green),

**k** – чёрный (black),

**m** – фиолетовый (magenta),

**r** – красный (red),

**y** – жёлтый (yellow).

Управление типом линий :

– непрерывная (по умолчанию),

- - пунктирная (длинный штрих),

: пунктирная (короткий штрих),

-. штрих-пунктирная.

Управление маркерами данных:

. точка (по умолчанию),

+ знак «плюс»,

\* звёздочка,

- o кружок,
- x крестик,
- s квадрат,
- d ромб,
- v треугольник остриём вниз,
- ^ треугольник остриём вверх.

Например, `plot (t, s, 'r')` – непрерывная линия красного цвета,  
`plot (t, s, ':og')` – пунктирная линия зелёного цвета, точки данных отмечены кружками.

### 10.3.Фильтрация сигнала

Выходной сигнал фильтра можно найти, используя функцию `filter`. Она реализует алгоритм цифровой фильтрации, соответствующий транспонированной структуре фильтра (`direct form II transposed`) (см. подраздел 6.3).

```
>> y= filter (b, a, s);
```

где **b** и **a** – векторы коэффициентов числителя и знаменателя передаточной функции фильтра; **s** – вектор входного сигнала; **y** – вектор выходного сигнала. Рассчитав выходной сигнал, можно построить графики входного и выходного сигналов:

```
>> plot (t, s, 'r', t, y, 'g')
```

Входной сигнал будет отображён непрерывной линией красного цвета, а выходной – непрерывной линией зелёного цвета.

При использовании функции `stem` лучше вывести графики отдельно:

```
>> subplot (2,1,1)  
>> stem (t, s, '.r')  
>> subplot (2, 1, 2)  
>> stem (t, y, '.b')
```

### 10.4. Фильтрация как операция свёртки входного сигнала и импульсной характеристики фильтра

Сначала нужно получить импульсную характеристику фильтра:

```
>> g= impz (b, a);
```

где  $\mathbf{b}$  и  $\mathbf{a}$  – коэффициенты полиномов числителя и знаменателя передаточной функции,  $\mathbf{g}$  – вектор отсчётов импульсной характеристики. Количество отсчётов рассчитывается автоматически и зависит от поведения импульсной характеристики. Для явного задания количества расчётных точек  $N$  нужно ввести оператор:

```
>> g= impz (b, a, n);
```

Для вывода графика импульсной характеристики не нужно указывать выходной параметр и ставить точку с запятой:

```
>> impz (b, a)
```

Рассчитав импульсную характеристику, можно найти выходной сигнал фильтра методом свёртки:

```
>> y= conv (s, g);
```

где  $\mathbf{s}$  – вектор входного сигнала,  $\mathbf{g}$  – вектор импульсной характеристики,  $\mathbf{y}$  – вектор выходного сигнала.

Далее следует построить графики входного и выходного сигналов (см. подразделы 10.2 и 10.3).

#### 10.4. Фильтрация в частотной области с использованием быстрого преобразования Фурье (БПФ)

В соответствии со спектральным методом спектр выходного сигнала равен произведению спектра входного сигнала и комплексной частотной характеристики фильтра. Спектр вычисляется путём применения к сигналу, заданному в виде последовательности отсчётов, дискретного преобразования Фурье (ДПФ). В MatLab алгоритм ДПФ применяется в форме БПФ (FFT – Fast Fourier Transform). Для осуществления фильтрации в частотной области задаются операторы (подразумевается, что сигнал  $s$ , частота дискретизации  $F_s$ , а также коэффициенты фильтра  $\mathbf{b}$  и  $\mathbf{a}$  определены ранее):

```
>> K= freqz (b, a, 512);  
>> S= fft (s, 1024);  
>> Y= S(1:512) .*K;  
>> y= ifft (Y);
```

Нужно следить, чтобы суммарное количество отсчётов сигнала  $s$  и импульсной характеристики  $\mathbf{g} = \text{impz}(\mathbf{b}, \mathbf{a})$  не превышало 1024. В противном случае нужно использовать функцию `fft`, задавая более высокую

размерность преобразования. Тогда большее количество точек  $n$  надо задавать при расчёте частотной характеристики фильтра:  $K = \text{freqz}(b, a, n)$ . ( $n = N/2$ , где  $N$  – размерность БПФ.) Помимо графиков сигналов можно вывести спектры входного и выходного сигналов и частотную характеристику фильтра:

```
>> f = (0:511)/512*(Fs/2);  
>> subplot (3,1,1)  
>> stem (f, abs(S(1:512)))  
>> subplot (3,1,2)  
>> plot (f, abs(K))  
>> subplot (3,1,3)  
>> stem (f, abs(Y(1:512)))
```

Вместо функции `stem` можно использовать функцию `plot`, тогда получим непрерывные спектры. Если имеется необходимость совместить все три графика, то можно использовать операторы `hold on` и `hold off` вместо операторов `subplot`. Для различения графиков нужно задать различный цвет и (или) тип линии (см. подраздел 10.2).

Блочную фильтрацию с помощью БПФ (сигнал разбивается на блоки по  $N$  отсчётов, где  $N$  – размерность БПФ) можно также реализовать, используя функцию `fftfilt`:

```
>>y = fftfilt (g, s, N);
```

где  $g$  – вектор импульсной характеристики фильтра,  $s$  – вектор входного сигнала,  $N$  – размерность БПФ.

### 10.5. Фильтрация сигнала с применением программы `sptool`

Программа `sptool` (signal processing tool) позволяет просматривать графики сигналов и их спектров, производить спектральный анализ сигнала, рассчитывать и анализировать фильтры, а также фильтровать сигналы. Программа имеет удобный графический интерфейс. Фильтрация осуществляется с применением функции `filter` (см. подраздел 10.3). При выполнении спектрального анализа сигнала имеется возможность выбора из довольно широкого набора цифровых параметрических и непараметрических методов. Вызов программы осуществляется из командной строки MatLab:

```
>> sptool
```

После набора имени программы нажмите клавишу <Enter>. Основное окно программы содержит три поля: `Signals`, `Filters` и `Spectra`. В них перечисляются идентификаторы загруженных в программу сигналов, фильтров и спектров.

Под списками расположены кнопки, нажимая на одну из которых можно активизировать выполнение той или иной операции.

Кнопка View – просмотр графика сигнала, спектра или характеристик фильтра. Выводятся графики или характеристики того объекта, который выделен в списке. Для выделения нескольких объектов одновременно нужно при нажатии левой кнопки мыши удерживать клавишу <Ctrl>. При просмотре графиков можно изменять масштаб, укрупняя отдельные участки. Для измерения значений в отдельных точках используются маркеры, которые можно перетаскивать мышью. При просмотре характеристик фильтра имеется возможность совместного или отдельного вывода АЧХ (в логарифмическом или в линейном масштабе), ФЧХ (в градусах или в радианах), характеристики группового времени запаздывания, диаграммы нулей и полюсов, импульсной и переходной характеристик.

Кнопка Create – при её нажатии осуществляется расчёт спектра сигнала, выделенного в списке сигналов. В окне Spectrum Viewer в поле Parameters нужно указать метод спектрального анализа. Имеет смысл испытать несколько методов и сравнить их результаты. Обычно из непараметрических методов предпочтение отдают методу Уэлча (Welch), из параметрических – можно применить авторегрессионный метод Юла-Уолкера (Yule AR). Разумеется, не следует забывать и об обычном ДПФ (FFT). Указав метод, следует щёлкнуть мышью по кнопке Apply. Будет выведен график спектральной плотности мощности. Имеется возможность выводить спектры в линейном или в логарифмическом масштабе (меню Options).

Кнопки New и Edit позволяют осуществить синтез нового фильтра или изменить характеристики уже существующего фильтра. При нажатии одной из этих кнопок появится окно Filter Designer. При проектировании фильтра нужно задать идентификатор фильтра (поле Filter), частоту дискретизации (поле Sampling Frequency), выбрать алгоритм (Algorithm) из раскрывающегося списка. Затем задать параметры частотной характеристики фильтра в разделе Specifications. Щёлкнув по кнопке Apply, получим в центре окна график АЧХ синтезированного фильтра, в разделе Measurements отмечены некоторые параметры АЧХ. Имеется возможность редактировать расположение нулей и полюсов на соответствующей диаграмме, изменять их количество. Для этого в списке Algorithm нужно выбрать строку Pole/Zero Editor. В окне Filter Designer появится z-плоскость с изображёнными на ней полюсами и нулями. Используя мышью и инструменты, значки которых расположены в левой верхней части z-плоскости, нужно скорректировать диаграмму полюсов и нулей. Затем можно перейти в режим просмотра характеристик фильтра (окно Filter Viewer). Удобно одновременно открыть окна Filter Designer и Filter Viewer и разместить их рядом, чтобы следить за тем, как изменения положения полюсов и нулей отражаются на виде АЧХ.

Фильтрация сигнала осуществляется следующим образом.

- 1) Выбор сигнала из списка или загрузка сигнала из рабочей области MatLab. Если сигнал, фильтрацию которого нужно осуществить, имеется в списке (окно SPTool: startup.spt), то его нужно выделить мышью. Имеется возможность загрузить сигнал в программу sptool из рабочей области MatLab. Для этого нужно сначала создать сигнал в рабочей области (см. подраздел 10.1). Затем нужно выбрать команду Import в меню File главного окна программы sptool. Появится окно Import to SPTool. Переключатель Source установите в положение From Workspace. В списке Workspace Contents перечислены переменные, имеющиеся в рабочей области MatLab. В раскрывающемся списке Import As нужно выбрать строку Signal. Далее выбирают в списке идентификатор вектора, содержащего отсчёты сигнала, и нажимают кнопку --> напротив поля Data. Затем аналогичным образом вводят значение частоты дискретизации  $F_s$  в поле Sampling Frequency. Задают имя сигнала в поле Name. Под этим именем он будет помещён в список сигналов sptool. Вслед за этим следует нажать кнопку ОК.
- 2) Расчёт фильтра или загрузка описания фильтра из рабочей области MatLab. Расчёт нового фильтра производится после нажатия кнопки New основного окна sptool (см. выше). Для загрузки описания фильтра из рабочей области MatLab задают команду Import меню File основного окна и далее в окне Import to SPTool в поле Import As выбирают строку Filter. В разделе Source устанавливают переключатель в положение From Workspace. В поле Form указывают форму задания параметров фильтра (Transfer Function – коэффициенты передаточной функции, Zeros, Poles, Gain – нули, полюсы, коэффициент усиления, 2<sup>nd</sup> Order Sections – каскадная форма). В первом случае в полях Numerator и Denominator вводят векторы коэффициентов  $\mathbf{b}$  и  $\mathbf{a}$  передаточной функции фильтра (в квадратных скобках, значения отделяются пробелами) либо имена векторов, если они имеются в рабочей области MatLab. Можно выделить имена в списке Workspace Contents и нажать на кнопку --> . Во втором случае вводят векторы  $z$ ,  $p$  и константу  $k$ , в третьем случае – матрицу  $\mathbf{sos}$ . Вслед за этим нужно указать имя фильтра в поле Name и щёлкнуть по кнопке ОК.
- 3) Фильтрация сигнала. Выделяют нужный сигнал и нужный фильтр в списках основного окна sptool. Затем нажимают кнопку Apply. В появившемся окне Apply Filter задают имя выходного сигнала в поле Output Signal. В раскрывающемся списке Algorithm задают алгоритм фильтрации: Direct Form II Transposed (filter). Затем нужно щёлкнуть по кнопке ОК. Выходной сигнал появится в списке основного окна программы.



- 4) Расчёт спектров сигналов. Выделяют поочерёдно входной и выходной сигналы в списке и нажимают кнопку Create. В окне Spectrum Viewer указывают метод спектрального анализа и его параметры, нажимают кнопку Apply. Будет построен график спектра. В списке спектров появятся имена спектров, соответствующих данным сигналам.
- 5) Просмотр графиков сигналов и спектров. Выделяют оба сигнала одновременно в списке сигналов (для этого совместно с нажатием левой кнопки мыши удерживают клавишу <Ctrl>). Нажимают кнопку View под списком сигналов и просматривают графики входного и выходного сигналов. Имеется возможность изменять цвет и тип линий. Для этого используют соответствующие кнопки панели инструментов окна Signal Browser. Аналогично осуществляется просмотр спектров. Можно также наложить спектр на график АЧХ фильтра. Для этого следует открыть окно Filter Designer (кнопки New или Edit) основного окна sptool), нажать кнопку Overlay Spectrum (предпоследняя на панели инструментов) и далее выбрать нужный спектр из списка в окне Overlay Spectrum, раздел Select a spectrum). Имеет смысл исследовать, как проходит данный сигнал через различные фильтры, а также проанализировать изменение формы выходного сигнала при варьировании его параметров (частоты, длительности и пр.).