

Моделирование цифровой обработки сигналов в MATLAB.

Часть 8. Моделирование цифровой фильтрации средствами программ GUI MATLAB: GUI SPTool

Алла СОЛОНИНА

В предлагаемом цикле «Моделирование цифровой обработки сигналов в MATLAB» предыдущие статьи [6–11] были посвящены моделированию цифровых фильтров (ЦФ), КИХ и БИХ, в том числе с фиксированной точкой (ФТ), программными средствами MATLAB.

На основе программных средств в MATLAB разработаны программы GUI (Graphic User Interface — графический интерфейс пользователя), представляющие собой средства, предназначенные для моделирования путем интерактивного общения без прямого доступа к программным средствам с графическим выводом результатов. В частности, для моделирования ЦФ разработаны две программы GUI — FDATool и SPTool. Проектирование ЦФ средствами FDATool было рассмотрено в [12].

Настоящая статья посвящена моделированию цифровой фильтрации средствами GUI SPTool.

Моделирование цифровой фильтрации средствами GUI SPTool

Программа GUI SPTool (Signal Processing Toolbox — средства обработки сигнала) разработана на основе пакета расширения Signal Processing Toolbox и предназначена для решения следующих задач, связанных с моделированием цифровой фильтрации:

- синтез ЦФ;
- анализ ЦФ;
- импорт входного сигнала;
- моделирование цифровой фильтрации;
- анализ сигналов во временной области;
- анализ сигналов в частотной области;
- экспорт данных из GUI SPTool.

Следует, однако, иметь в виду, что возможности синтеза ЦФ в GUI SPTool существенно ограничены по сравнению с GUI FDATool, в том числе отсутствует возможность моделирования структур ЦФ с ФТ. В [12] описан экспорт в GUI SPTool фильтра, синтезированного в GUI FDATool и сохраненного в виде объекта `dfilt`. Экспортируемый фильтр,

имя которого появляется в списке **Filters**, можно далее использовать для моделирования процесса цифровой фильтрации.

Тем не менее, в ряде случаев удобно оперативно синтезировать ЦФ непосредственно в GUI SPTool. Основные этапы синтеза ЦФ описаны в [12], а технология синтеза легко осваивается самостоятельно по аналогии с GUI FDATool. Подробно она описана в [5]. В примере 1 представлен оперативный синтез КИХ-фильтра в GUI SPTool для его последующего использования при моделировании цифровой фильтрации.

Обращение к GUI SPTool происходит по команде `sptool`, после чего открывается окно **SPTool: startup.spt** (рис. 1) с тремя списками:

- **Signals** («Сигналы»);
- **Filters** («Фильтры»);
- **Spectra** («Спектры»).

Синтез цифровых фильтров

Операции, связанные с цифровой фильтрацией, выполняются с использованием кнопок, расположенных под списком **Filters**:

- **View** («Вид») — анализ ЦФ, имя которого выделено в списке **Filters**.
- **New** («Новый фильтр») — синтез нового ЦФ.
- **Edit** («Редактирование») — изменение требований к ЦФ, имя которого выделено в списке **Filters**, и его повторный синтез без изменения имени.
- **Apply** («Применить») — моделирование процесса фильтрации сигнала, имя которого выделено в списке **Signals**, фильтром, имя которого выделено в списке **Filters**.

В списке **Filters** содержатся имена синтезированных ранее и сохраненных ЦФ; если они отсутствуют, активна только кнопка **New**.

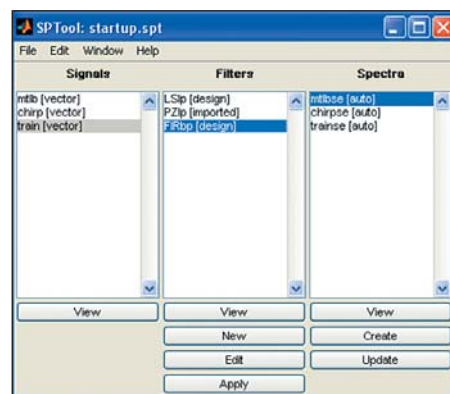


Рис. 1. Окно SPTool: startup.spt

Для синтеза нового ЦФ необходимо нажать кнопку **New**, после чего открывается окно **Filter Designer** («Синтез фильтра») (рис. 2).

Пример 1

В примере 2 [6] был синтезирован программными средствами MATLAB оптимальный КИХ-фильтр ФНЧ по заданным требованиям к АЧХ (табл. 3 [6]) методом чебышевской аппроксимации. Требуется синтезировать тот же фильтр средствами GUI SPTool.

Основные этапы синтеза включают в себя:

1. Выбор типа и метода синтеза ЦФ. В раскрываемом списке **Algorithm** («Алгоритм») выбрать значение **Equiripple FIR** («Метод наилучшей равномерной (чебышевской) аппроксимации»).

2. Задание входных параметров. Флаг **Minimum Order** («Минимальный порядок») установлен, входные параметры отсутствуют [12].
3. Задание требований к АЧХ (табл. 3 [6]):

- Тип избирательности задается в группе **Specifications** («Требования») в раскрываемом списке **Type** («Тип») — **lowpass**.

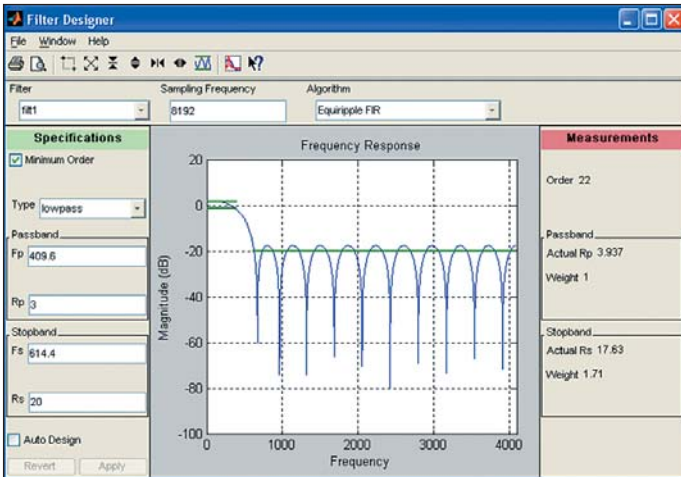


Рис. 2. Окно Filter Designer

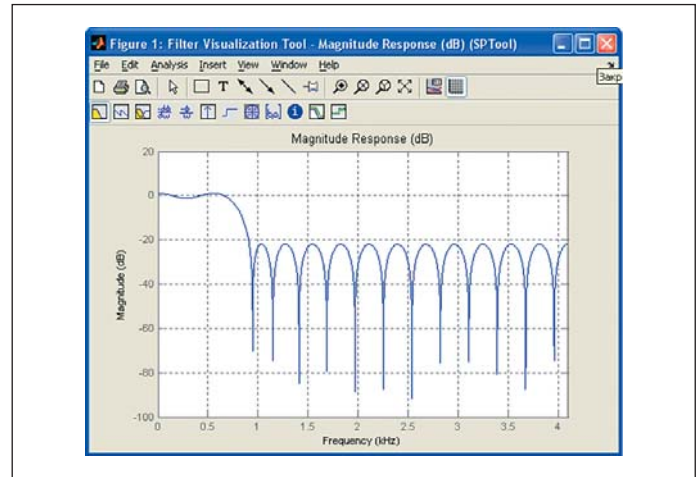


Рис. 3. Окно Figure...: Filter Visualization Tool

- Частота дискретизации и граничные частоты ПП и ПЗ в герцах указываются соответственно в поле ввода **Sampling Frequency** («Частота дискретизации») — 8000 — и в группе **Specifications** в подгруппах:
 - **Passband** («Полоса пропускания») в поле ввода **Fp** — 1000;
 - **Stopband** («Полоса задерживания») в поле ввода **Fs** — 1500.

- Максимально допустимые отклонения АЧХ (дБ) указываются в группе **Specifications** в подгруппах (значения **Rp** (дБ) и **Rs** (дБ) были вычислены в примере 1 в [6]):
- **Passband** в поле ввода **Rp** — 0.4455;
- **Stopband** в поле ввода **Rs** — 40.

4. Синтез фильтра производится после нажатия кнопки **Apply** (при сброшенном флаге **Auto Design**), и по его завершении автоматически выдаются:

- в поле **Frequency Response** («Частотная характеристика») — график АЧХ (дБ) (характеристика ослабления (6) в [6]);
- в группе **Measurements** («Значения»):
 - **Order** — 26;
 - в подгруппе **Passband** действительное отклонение АЧХ (дБ) **Actual Rp** — 0.5697 и вес в ПП **Weight** — 1;
 - в подгруппе **Stopband** действительное отклонение АЧХ (дБ) **Actual Rs** — 37.82 и вес в ПЗ **Weight** — 2.564;
- в раскрывающемся списке **Filter** — имя синтезированного фильтра **filt2**.

Сравнивая действительные значения АЧХ (дБ) **Actual Rp** и **Actual Rs** в группе **Measurements** с заданными пользователем значениями **Rp** и **Rs** в группе **Specifications**, убеждаемся, что требования к АЧХ (дБ) в ПП не выполняются.

Для изменения порядка КИХ-фильтра сбросим флаг **Minimum Order**, в поле ввода **Order** («Порядок») установим порядок 27 и зададим входные параметры по умолчанию:

- в подгруппе **Passband** в поле ввода **Weight** — 1;
- в подгруппе **Stopband** в поле ввода **Weight** — 5.

Весы определены по методике, изложенной в [6], а именно: вес, равный единице, присвоен ПП с наибольшим максимально допустимым отклонением 0,05, а вес в ПЗ рассчитан как отношение 0,05/0,01 = 5.

При повторном синтезе после нажатия кнопки **Apply** (она активизируется после щелчка левой кнопки мыши на поле графика) в группе **Measurements** автоматически выводятся выходные параметры в подгруппах **Passband** и **Stopband**. Сравнивая значения АЧХ (дБ) **Actual Rp** и **Actual Rs** с заданными пользователем в группе **Specifications**, убеждаемся, что требования к АЧХ (дБ) вновь не выполняются.

Увеличивая порядок фильтра, не меняя входные параметры, несложно убедиться, что требования к АЧХ (дБ) будут выполняться при порядке фильтра $R = 31$.

Таким образом, синтезирован оптимальный ФНЧ с ЛФЧХ порядка $R_{opt} = 31$ на базе КИХ-фильтра 2-го типа. (Отметим, что порядок КИХ-фильтра ФНЧ, синтезированного средствами GUI FDATool в примере 1 [12], равен $R_{min} = 25$).

Анализ цифровых фильтров

Для анализа характеристик ЦФ, синтезированного или импортированного в GUI SPTool, необходимо в списке **Filters** окна **SPTool: startup.spt** выделить имя ЦФ и нажать кнопку **View**. После этого происходит автоматическое обращение к GUI FVTool и открывается окно **Figure...: Filter Visualization Tool** (рис. 3). Список команд пункта меню **Analysis** данного окна ничем не отличается от пункта меню **Analysis** окна **Filter Design & Analysis Tool GUI FDATool** [12].

Импорт входного сигнала

Источником сигнала для GUI SPTool может являться:

- рабочее пространство памяти **Workspace**, если сигнал создается непосредственно в командном окне MATLAB;

- диск, если сигнал сохранен как **mat**-файл данных в папке **work** или папке пользователя.

Пример 2

Создать входной сигнал — 1024-точечную последовательность, представляющую собой смесь двух дискретных гармоник:

$$x(n) = \sin(2\pi f_1 nT) + 0,5\sin(2\pi f_2 nT) \quad (1)$$

со следующими значениями частот:

- частота дискретизации $f_d = 8000$ Гц;
- $f_1 = 500$ Гц;
- $f_2 = 2000$ Гц;

с нормальным белым шумом и сохранить его на диске.

Сформируем входной сигнал — вектор **x** — подставив в (1) значения частот:

$$x(n) = \sin(0,125\pi n) + 0,5\sin(0,5\pi n). \quad (2)$$

Результирующий сигнал обозначим как вектор **s** и сохраним на диске:

```
>> N=1024;
>> n=0:(N-1);
>> x=sin(0,125*pi.*n)+0,5*sin(0,5*pi.*n);
>> noise=randn(1,length(x));
>> s=x+noise;
>> save s
```

Для импорта входного сигнала в GUI SPTool необходимо в окне **SPTool: startup.spt** в пункте меню **File** выбрать команду **Import** («Импорт»), после чего открывается окно **Import to SPTool** («Импорт в SPTool»), в котором в группе **Source** («Источник») следует выбрать переключатель **From Workspace** (Из рабочего пространства памяти **Workspace**) или **From Disk** («С диска»).

Если сигнал импортируется **From Workspace**, то необходимо выполнить следующие действия (рис. 4):

1. В группе **Workspace Contents** («Содержимое **Workspace**») курсором выделить имя переменной **s** (пример 2).

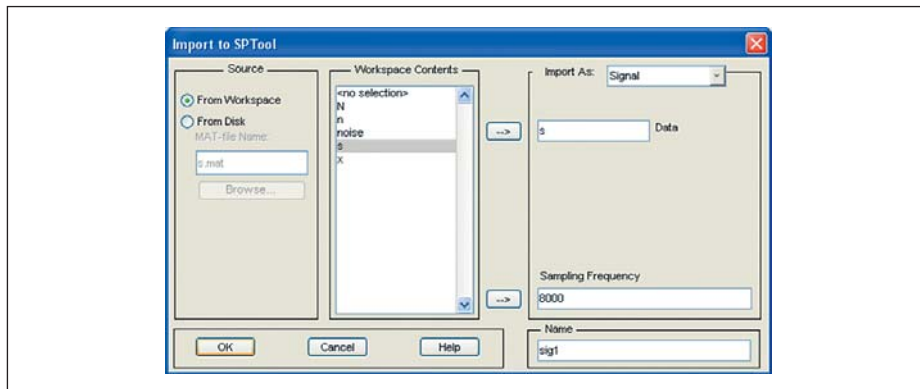


Рис. 4. Окно Import to SPTool при импорте From Workspace

2. Нажать кнопку «>», после чего имя переменной *s* будет отображено в поле ввода **Data** («Данные») — это указание рассматривать переменную *s* как числовой массив.
3. В раскрывающемся списке **Import As** («Импортировать как») выбрать значение **Signal** («Сигнал»).
4. В поле ввода **Sampling Frequency** («Частота дискретизации») указать частоту дискретизации в герцах — 8000.
5. В поле ввода **Name** («Имя») указать имя входного сигнала — предлагаемое по умолчанию **sig1**.
6. Нажать кнопку **OK** и убедиться, что сигнал **sig1** отображился в группе **Signals** окна **SPTool: startup.spt**.

Если сигнал импортируется с диска (**From Disk**), то необходимо в группе **Source** нажать активизированную кнопку **Browse** («Просмотреть»), после чего открывается окно **Select File to Open** («Выбрать файл для открытия»), в котором курсором выделить имя *mat*-файла (у нас — *s*) и нажать кнопку **Открыть**, после чего происходит автоматический возврат в окно **Import to SPTool**. Дальнейшие действия аналогичны описанным выше при импорте **From Workspace**.

Для удаления импортированных сигналов следует в окне **SPTool: startup.spt** в группе

Signals курсором выделить имя удаляемого сигнала и в пункте меню **Edit** выбрать команду **Clear**.

Моделирование цифровой фильтрации

После синтеза ЦФ и импорта входного сигнала можно приступить к моделированию цифровой фильтрации. Для этого в окне **SPTool: startup.spt** необходимо выполнить следующие действия:

1. В группе **Signals** выделить имя сигнала, например, **sig1** (вектор *s* в примере 2).
2. В группе **Filters** выделить имя фильтра, например, КИХ-фильтра ФНЧ **filt2** (пример 1).

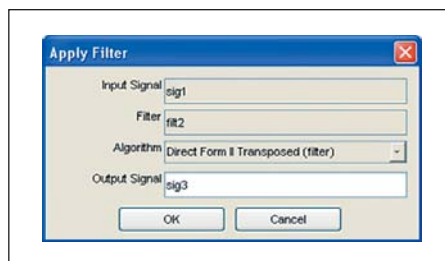


Рис. 5. Окно Apply Filter

3. Нажать кнопку **Apply**, после чего открывается окно **Apply Filter** («Применить к фильтру») (рис. 5) с именами входного сигнала **sig1**, фильтра **filt2** и выходного сигнала **sig3**.
4. Нажать кнопку **OK** и убедиться, что сигнал **sig3** отображился в группе **Signals**.

Анализ сигналов во временной области

Для анализа сигналов во временной области следует в окне **SPTool: startup.spt** в группе **Signals** выделить имя сигнала (одно или несколько) и нажать кнопку **View**, после чего открывается окно **Signal Browser** («Просмотр сигнала»). На рис. 6, 7 представлены графики входного и выходного сигналов — **sig1** и **sig3** соответственно.

В окне **Signal Browser** (рис. 6, 7) автоматически выводятся два графика дискретного сигнала (или нескольких сигналов), изображенные непрерывными линиями:

- нижний **Panner** («Панорама») — общий вид (панорама) сигнала на заданном интервале;
- верхний — общий вид сигнала на заданном интервале с активизированным набором средств для анализа сигнала, которые легко осваиваются самостоятельно.

На верхнем графике входной дискретный сигнал **sig1** представлен на интервале времени $[0; NT] = [0; N/F_s]$ с, где N — длина сигнала, а F_s — частота дискретизации f_d . В нашем случае при $N = 1024$ и $F_s = 8000$ Гц (пример 2) имеем интервал $[0; 0,128]$ с.

Анализ сигналов в частотной области

Для анализа сигналов в частотной области необходимо:

1. В окне **SPTool: startup.spt** в группе **Signals** выделить имя сигнала.
2. В группе **Spectra** нажать кнопку **Create** («Создать»), после чего открывается окно **Spectrum Viewer** («Просмотр спектра»).

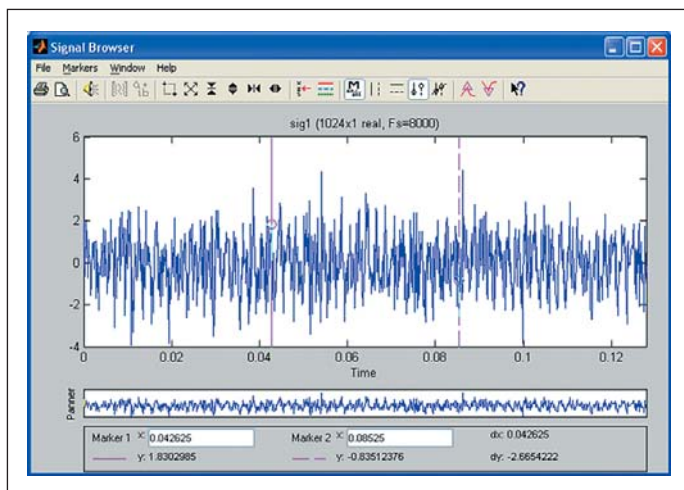


Рис. 6. Окно Signal Browser — входной сигнал sig1

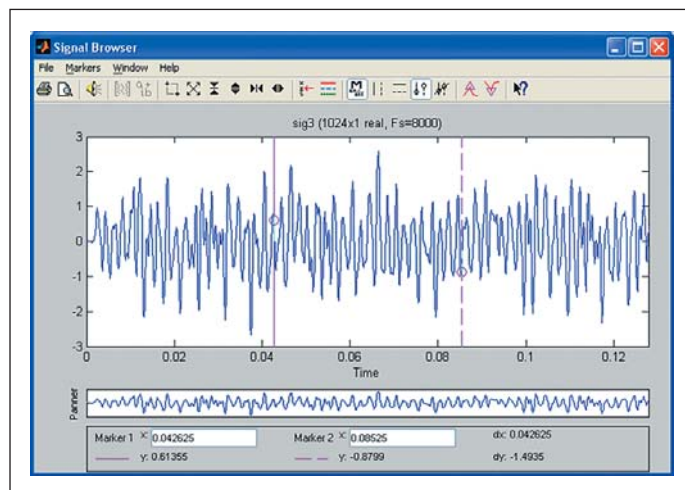


Рис. 7. Окно Signal Browser — выходной сигнал sig3

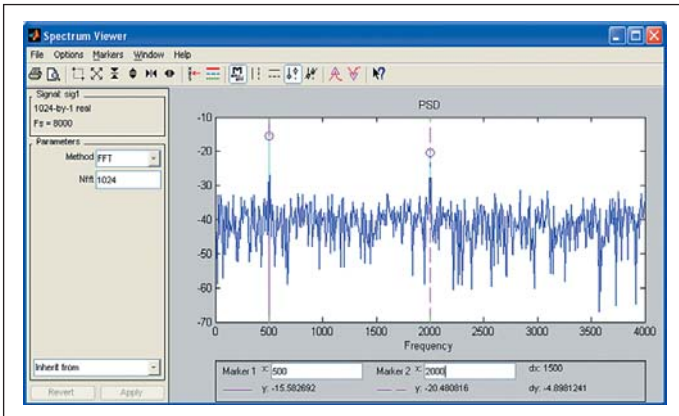


Рис. 8. Окно Spectrum Viewer — PSD spect1 входного сигнала sig1

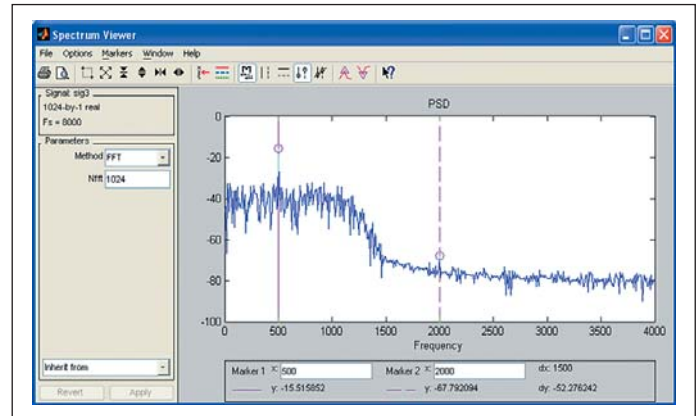


Рис. 9. Окно Spectrum Viewer — PSD spect2 выходного сигнала sig3

В этом окне в группе **Signal** отображается имя анализируемого сигнала, его длина и частота дискретизации в герцах. В группе **Parameters** в раскрывающемся списке **Method** содержатся методы спектрального анализа сигнала [5].

3. Нажать кнопку **Apply**, после чего в окне **Spectrum Viewer** будет выведен график спектральной плотности мощности (PSD) в основной полосе частот $[0; f_n/2]$.

В группе **Signals** последовательно выделим имена входного и выходного сигналов — **sig1** и **sig3** соответственно. Для каждого из них выберем метод спектрального анализа FFT (БПФ), входным параметром которого является задаваемое число точек сигнала — 1024. Графики PSD, соответственно, входного и выходного сигналов **sig1** и **sig3** представлены на рис. 8, 9, где маркеры **Marker 1** и **Marker 2** подведены к частотам сигнала 500 и 2000 Гц (пример 2), а переменные **y1** и **y2** фиксируют соответствующие значения PSD (дБ/Гц). Вычисленным PSD автоматически присваиваются имена **spect1**, **spect2** и т. д., у нас PSD входного сигнала имеет имя **spect1**, а выходного — **spect2**.

В GUI SPTool при использовании метода FFT спектральная плотность мощности PSD (дБ/Гц) рассчитывается по формуле:

$$P(k) = 10 \lg \left(\frac{1}{NF_s} |X(k)|^2 \right),$$

при $k = 0, 1, \dots, (N-1)$, (3)

где F_s — частота дискретизации f_n ; N — размерность ДПФ; $X(k)$ — N -точечное ДПФ N -точечной последовательности $x(n)$ (см. вычисление ДПФ в MATLAB [5]); k — дискретная нормированная частота; $P(k)$ — PSD (дБ/Гц).

Пример 3

Рассчитать в MATLAB значения спектральной плотности мощности $P(k)$ (дБ/Гц) (3) входного сигнала **s** (пример 2) на частотах 500 и 2000 Гц и сравнить их с выведенным в окне **Signal Browser** (рис. 9).

С помощью ДПФ рассчитаем спектральную плотность мощности $P(k)$ (дБ/Гц) (3) (вектор PDB) сигнала **s**:

```
>> load s;
>> Fs=8000;
>> N=1024;
>> S=fft(s);
>> MODS=(abs(S));
>> P=(1/(N*Fs)).*(MODS.^2);
>> PDB=10.*log10(P);
```

Определим значения спектральной плотности мощности (вектора PDB) на частотах 500 и 2000 Гц. Для этого найдем соответствующие дискретные нормированные частоты k_1 и k_2 с помощью внешней функции `fft_e1` (см. вычисление ДПФ в MATLAB [5]):

```
>> [max(P) min(P)]
ans = 0.0277 2.0302e-007
>> e1=0.05;
>> [MODm,m]=fft_e1(N,P,e1)
MODm = 0.0277 0.0090 0.0090 0.0277
m = 65 257 769 961
```

С учетом того, что нижняя граница индексов в MATLAB равна единице, в основной полосе дискретных нормированных частот $0 \leq k \leq 511$ имеем $k_1 = 64$ и $k_2 = 256$, а в MATLAB — $k_1 = 65$ и $k_2 = 257$.

Выведем значения спектральной плотности мощности $P(k_1)$ и $P(k_2)$ (дБ/Гц):

```
>> [PDB(65) PDB(257)]
ans = -15.5827 -20.4808
```

Полученные значения практически совпадают с выведенными в окне **Spectrum Viewer** (рис. 8).

На рис. 9 представлена спектральная плотность **spect2** выходного сигнала **sig3** КИХ-фильтра ФНЧ **filt2**. В данном случае одна из спектральных составляющих входного сигнала **sig1** с частотой 500 Гц оказалась в полосе пропускания ФНЧ, а другая, с частотой 2000 Гц — в полосе задерживания (пример 1).

Для удаления рассчитанных PSD следует в окне **SPTool: startup.spt** в группе **Spectra** курсором выделить имя удаляемого PSD и в пункте меню **Edit** выбрать команду **Clear**.

Экспорт данных из GUI SPTool

Если результаты моделирования в GUI SPTool предполагается в дальнейшем использовать в MATLAB, необходимо осуществить их экспорт. Для этого в окне **SPTool: startup.spt** в пункте меню **File** следует выбрать команду **Export** («Экспорт»), после чего открывается окно **Export from SPTool** («Экспорт из SPTool») (рис. 10), в котором необходимо:

1. В группе **Export List** («Список имен экспортируемых данных») выделить имя (или имена) экспортируемых данных, например, выделим имена: выходного сигнала **sig3**, синтезированного КИХ-фильтра ФНЧ **filt2** и спектральной плотности мощности PSD выходного сигнала **spect2**.
2. С помощью соответствующих кнопок указать, куда экспортируются данные (**Export to Disk** или **Export to Workspace**). При экспорте данных в **Workspace** они автоматически сохраняются с теми же именами, как в GUI SPTool. Экспорт данных на диск осуществляется последовательно, при этом каждый раз в автоматически открывающемся окне **Export to Disk** («Экспорт на диск») следует указать имя **mat**-файла (без расширения), которое может совпадать или не совпадать с именами в GUI SPTool, после чего нажать кнопку **Save** («Сохранить»), и дан-

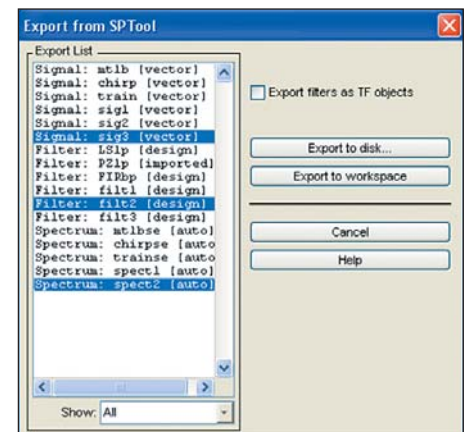


Рис. 10. Окно Export from SPTool

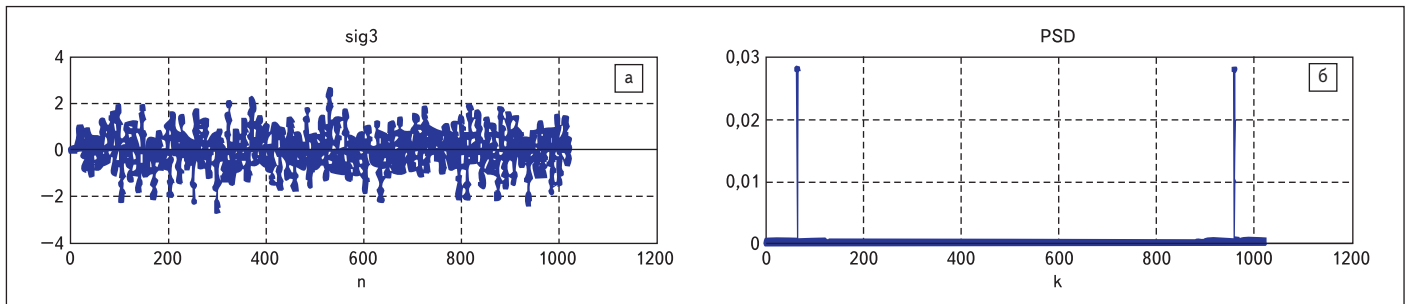


Рис. 11. а) Выходной сигнал sig3; б) PSD

ные сохраняются в папке work или в предварительно открытой папке пользователя. Загрузка mat-файлов с диска в **Workspace** производится, как обычно, по команде load. Для использования в приведенных далее примерах экспортируем сигнал **sig3**, фильтр **filt2** и PSD **spect2** в **Workspace** с теми же именами.

Сохраняемые данные представляются в виде массивов записей struct array [5].

Пример 4

Определить содержимое файла **sig3**, экспортируемого из GUI SPTool в **Workspace**:

```
>> load sig3
>> sig3
sig3 =
    data: [1024x1 double]
      Fs: 8000
    type: 'vector'
  lineinfo: [1x1 struct]
 SPTIdentifier: [1x1 struct]
    label: 'sig3'
```

Поле data хранит значения выходного сигнала и может быть использовано, например, для построения его графика (рис. 11а):

```
>> s=sig3.data;
>> n=0:1023;
>> subplot(2,1,1), stem(n,s,'fill','MarkerSize',3),...
grid, xlabel('n'), title('sig3')
```

Поле label хранит имя сигнала в GUI SPTool, а поля lineinfo и SPTIdentifier — вспомогательную информацию.

Пример 5

Определить содержимое файла **filt2**, экспортируемого из GUI SPTool на диск:

```
>> load filt2
>> filt2
filt2 =
    tf: [1x1 struct]
    ss: []
    zpk: []
    sos: []
    imp: []
    step: []
    t: []
    H: []
    G: []
      f: []
    specs: [1x1 struct]
      Fs: 8000
    type: 'design'
  lineinfo: []
 SPTIdentifier: [1x1 struct]
    label: 'filt2'
```

Поле tf хранит коэффициенты передаточной функции КИХ-фильтра ФНЧ и, в свою очередь, содержит поля num и den, хранящие коэффициенты числителя и знаменателя передаточной функции КИХ-фильтра — рациональной функции:

```
>> filt2.tf
ans =
    num: [1x32 double]
    den: 1
```

В массиве записей filt2 зарезервирован ряд пустых полей [] с фиксированными именами для хранения других представлений передаточной функции, а также импульсной характеристики, переходной характеристики и др. После необходимых расчетов этим полям могут быть присвоены соответствующие значения. Например, по коэффициентам передаточной функции вычислим импульсную характеристику КИХ-фильтра с помощью функции imp и сохраним ее значения в поле impz:

```
>> N=32;
>> filt2.imp=impz(filt2.tf.num,filt2.tf.den,N);
```

Поле label хранит имя фильтра в GUI SPTool, а поля lineinfo и SPTIdentifier — вспомогательную информацию.

Пример 6

Определить содержимое файла **spect2**, экспортируемого из GUI SPTool в **Workspace**:

```
>> spect2
spect2 =
      P: [1536x1 double]
      f: [1536x1 double]
    confid: [1x1 struct]
    specs: [1x1 struct]
    signal: [1024 1 0]
  signalLabel: 'sig3'
      Fs: 8000
    type: 'auto'
  lineinfo: [1x1 struct]
 SPTIdentifier: [1x1 struct]
    label: 'spect2'
```

Поле f хранит массив значений частот, соответствующих PSD в виде:

$$\tilde{P}(k) = \frac{1}{NF_s} |X(k)|^2, \quad (4)$$

при $k = 0, 1, \dots, (N-1)$,

а массив P — сами значения $\tilde{P}(k)$ (4).

Сравнивая соотношения (3) и (4), получаем взаимосвязь между $\tilde{P}(k)$ (4) и $P(k)$ (3):

$$\tilde{P}(k) = 10^{-P(k)/10}. \quad (5)$$

Как видим, поле f содержит 1536, а не 1024 точки, как ожидалось. Определим частотный диапазон, в котором выведена PSD (4), для чего выведем первую и последнюю частоты поля f (обозначим fbeg и ffin), а также интервал между соседними значениями частот поля f (обозначим delta_f):

```
>> fbg=spect2.f(1)
fbeg = -4000
>> ffin=spect2.f(1536)
ffin = 7.9922e+003
>> delta_f=spect2.f(2)-spect2.f(1)
delta_f = 7.8125
```

Интервал между соседними значениями частот (delta_f) равен: $\Delta f = f_n/N = 8000/1024 = 7,8125$ Гц, а диапазон частот: $[f_n/2; f_n - \Delta f] = [-4000; 7992,2]$ Гц.

Построим график (4) в шкале дискретных нормированных частот k, используя 1024 значения, соответствующих частотному диапазону: $[0; f_n - \Delta f] = [0; 7992,2]$ Гц, где частоте 0 Гц соответствует дискретная нормированная частота $K = ((f_n/2)/\Delta f) + 1$ (рис. 11б):

```
>> K=4000/delta_f+1
K = 513
>> spect2.f(513)
ans = 0
>> k=K:(K+1023);
>> kk=0:1023;
>> subplot(2,1,2), stem(kk,spect2.P(k),'fill','MarkerSize',3),...
grid, xlabel('k'), title('PSD')
```

Выведем значение $\tilde{P}(K+64)$:

```
>> spect2.P(K+64)
ans = 0.0281
```

Это значение совпадает с $P(k)$ (дБ/Гц) (3) на той же частоте (значение PSD на рис. 9, отмеченное маркером **Marker 1** с учетом взаимосвязи (5):

```
>> 10.^(-1.5515852)
ans = 0.0281
```

Поле label хранит имя PSD в GUI SPTool, поле signal — информацию о длине выход-

ного сигнала sig3 (первый элемент вектора), а поля `confid`, `specs`, `lineinfo` и `SPTIdentifier` — вспомогательную информацию.

Выход из GUI SPTool

При выходе из программы GUI SPTool автоматически появляется запрос, требуется ли сохранять данные (сигналы, фильтры, спектральные плотности мощности). В случае положительного ответа нажимается кнопка **Save** («Сохранить»), данные сохраняются в файле `startup.spt` в папке `work` или папке пользователя и могут быть использованы при следующих сеансах работы в GUI SPTool. ■

Литература

1. Ingle V., Proakis J. Digital Signal Processing Using MATLAB. Second Edition. Thomson.
2. Опенгейм А., Шафер Р. Цифровая обработка сигналов. М.: Техносфера, 2006.
3. Сергиенко А. Б. Цифровая обработка сигналов, 2-е изд. СПб.: ПИТЕР, 2006.
4. Солонина А. И., Улахович Д. А., Арбузов С. М., Соловьева Е. Б. Основы цифровой обработки сигналов. 2-е изд. СПб.: БХВ-Петербург, 2005.
5. Солонина А. И., Арбузов С. М. Цифровая обработка сигналов. Моделирование в MATLAB. СПб.: БХВ-Петербург, 2008.
6. Солонина А. Моделирование цифровой обработки сигналов в MATLAB. Часть 1. Синтез оптимальных (по Чебышеву) КИХ-фильтров программными средствами MATLAB // Компоненты и технологии. 2008. № 11.
7. Солонина А. Моделирование цифровой обработки сигналов в MATLAB. Часть 2. Синтез оптимальных БИХ-фильтров программными средствами MATLAB // Компоненты и технологии. 2008. № 12.
8. Солонина А. Моделирование цифровой обработки сигналов в MATLAB. Часть 3. Описание структур КИХ- и БИХ-фильтров в MATLAB // Компоненты и технологии. 2009. № 1.
9. Солонина А. Моделирование цифровой обработки сигналов в MATLAB. Часть 4. Моделирование структур цифровых фильтров с фиксированной точкой программными средствами MATLAB: анализ характеристик КИХ-фильтров // Компоненты и технологии. 2009. № 2.
10. Солонина А. Моделирование цифровой обработки сигналов в MATLAB. Часть 5. Моделирование структур цифровых фильтров с фиксированной точкой программными средствами MATLAB: анализ характеристик БИХ-фильтров // Компоненты и технологии. 2009. № 3.
11. Солонина А. Моделирование цифровой обработки сигналов в MATLAB. Часть 6. Моделирование структур цифровых фильтров с фиксированной точкой программными средствами MATLAB: квантование воздействия и вычисление реакции // Компоненты и технологии. 2009. № 4.
12. Солонина А. Моделирование цифровой обработки сигналов в MATLAB. Часть 7. Моделирование цифровых фильтров средствами программ GUI MATLAB: GUI TDATool // Компоненты и технологии. 2009. № 5.