

Моделирование цифровой обработки сигналов в MATLAB.

Часть 3. Описание структур КИХ- и БИХ-фильтров в MATLAB

Алла СОЛОНИНА

Разновидности структур КИХ- и БИХ-фильтров

Структура ЦФ отображает алгоритм вычисления реакции, описываемый разностным уравнением (РУ).

Передаточной функции БИХ-фильтра в общем виде:

$$H(z) = \frac{\sum_{i=0}^{N-1} b_i z^{-i}}{1 + \sum_{k=1}^{M-1} a_k z^{-k}}, \quad (1)$$

где $(N-1) \leq (M-1)$, соответствует алгоритм вычисления реакции в виде РУ:

$$y(n) = \sum_{i=0}^{N-1} b_i x(n-i) - \sum_{k=1}^{M-1} a_k y(n-k). \quad (2)$$

Помимо общего вида (1), передаточная функция БИХ-фильтра может быть представлена в других эквивалентных видах, среди которых практический интерес представляют два следующих:

- произведение множителей второго порядка с вещественными коэффициентами:

$$H(z) = G \prod_{k=1}^L \frac{1 + b_{1k} z^{-1} + b_{2k} z^{-2}}{1 + a_{1k} z^{-1} + a_{2k} z^{-2}}; \quad (3)$$

- сумма дробей второго порядка с вещественными коэффициентами:

$$H(z) = \sum_{k=1}^L \frac{B_{0k} + B_{1k} z^{-1}}{1 + a_{k1} z^{-1} + a_{k2} z^{-2}}. \quad (4)$$

Основные этапы проектирования цифровых фильтров (ЦФ) были рассмотрены в [6], а первый этап проектирования — синтез оптимальных КИХ- и БИХ-фильтров программными средствами MATLAB — в [6] и [7] соответственно. В этой статье познакомимся со вторым этапом проектирования — выбором структуры ЦФ и ее описанием в MATLAB¹, а также с новыми средствами MATLAB, позволяющими синтезировать КИХ- и БИХ-фильтры заданной структуры.

Передаточным функциям (3, 4) соответствуют свои эквивалентные виды РУ (2), то есть другие алгоритмы вычисления реакции.

Передаточной функции КИХ-фильтра:

$$H(z) = \sum_{i=0}^{N-1} b_i z^{-i} = \sum_{n=0}^{N-1} h(n) z^{-n} \quad (5)$$

соответствует РУ:

$$y(n) = \sum_{i=0}^{N-1} b_i x(n-i), \quad (6)$$

описывающее алгоритм вычисления реакции.

Таким образом, структура ЦФ, отображающая алгоритм вычисления реакции, определяется видом передаточной функции.

Три основных вида передаточной функции БИХ-фильтров — это:

- общий (дробно-рациональный) (1);
- произведение (3);
- сумма (4).

Они определяют три основные структуры БИХ-фильтров:

- прямую — Direct-Form I и ее модификации (на рис. 1 они представлены для звена 2-го порядка);
- прямую транспонированную структуру — Direct-Form I Transposed;

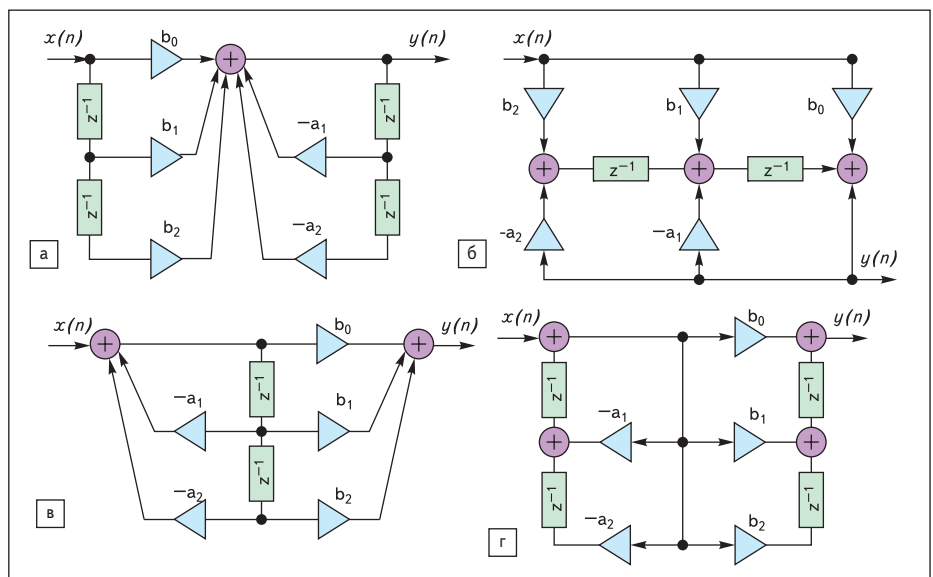


Рис. 1. Структура звена 2-го порядка:

а) Direct-Form I; б) Direct-Form I Transposed; в) Direct-Form II; г) Direct-Form II Transposed

¹ Версия MATLAB R2008a (MATLAB 7.6 Release 2008a).

- прямую каноническую структуру — Direct-Form II;
 - прямую каноническую транспонированную структуру — Direct-Form II Transposed;
 - каскадную из звеньев 2-го порядка с прямой структурой или ее модификацией;
 - параллельную из звеньев 2-го порядка с прямой структурой или ее модификацией.
- Передаточная функция (5) определяет прямую структуру КИХ-фильтра — Direct-Form FIR и ее модификации, среди которых практический интерес представляет прямая приведенная структура, отображающая алгоритм вычисления реакции КИХ-фильтров с линейной ФЧХ (ЛФЧХ) [6]. Она представлена двумя разновидностями:
- с симметричной импульсной характеристикой — Direct-Form Symmetric FIR;
 - с антисимметричной импульсной характеристикой — Direct-Form Antisymmetric FIR.
- Пример прямой приведенной структуры рассматривается в этой статье далее.

Описание структур КИХ- и БИХ-фильтров в MATLAB

Поскольку структура ЦФ определяется видом передаточной функции, описание структуры ЦФ сводится к описанию вида его передаточной функции.

В MATLAB структура ЦФ описывается в виде объекта `dfilt` (от англ. Discrete-time filter object):

```
Hd=dfilt.structure(input1,...)
```

Здесь `Hd` — имя объекта `dfilt`; `dfilt` — тип объекта; `structure` — функция, задающая абстрактную структуру; на ее место ставится функция, задающая конкретную структуру объекта `Hd`; `input1, ...` — список параметров функции `structure`.

Полный список структур можно вывести по команде:

```
help dfilt
```

Основные структуры ЦФ и соответствующие им функции MATLAB приведены в таблице 1. Отметим, что в MATLAB при описании структур по умолчанию подразумевается БИХ-фильтр, поэтому для них в названии структур слово “PIR” отсутствует, а в названии структур КИХ-систем добавлено “FIR”.

Выходным параметром является объект `Hd` (массив записей — `struct array`) со списком свойств (полей), зависящих от структуры.

Вывод списка свойств объекта `Hd` выполняется одним из следующих способов:

- автоматически, если в описании объекта `dfilt` отсутствует точка с запятой;
- по имени объекта:

```
Hd
```

Таблица 1. Основные структуры ЦФ

| Структура ЦФ (structure) | Функция | Параметры функции |
|---|---------------------------|--|
| Direct-Form I (прямая) | <code>df1(b,a)</code> | b, a — векторы коэффициентов передаточной функции (1) |
| Direct-Form I, Second-order sections (каскадная из звеньев 2-го порядка с прямой структурой) | <code>df1sos(s,G)</code> | s, G — матрица коэффициентов и коэффициент усиления передаточной функции (3) |
| Direct-Form I Transposed (прямая транспонированная) | <code>df1t(b,a)</code> | b, a — векторы коэффициентов передаточной функции (1) |
| Direct-Form I Transposed, Second-order sections (каскадная из звеньев 2-го порядка с прямой транспонированной структурой) | <code>df1tsos(s,G)</code> | s, G — матрица коэффициентов и коэффициент усиления передаточной функции (3) |
| Direct-Form II (прямая каноническая) | <code>df2(b,a)</code> | b, a — векторы коэффициентов передаточной функции (1) |
| Direct-Form II, Second-order sections (каскадная из звеньев 2-го порядка с прямой канонической структурой) | <code>df2sos(s,G)</code> | s, G — матрица коэффициентов и коэффициент усиления передаточной функции (3) |
| Direct-Form II Transposed (прямая каноническая транспонированная) | <code>df2t(b,a)</code> | b, a — векторы коэффициентов передаточной функции (1) |
| Direct-Form II Transposed, Second-order sections (каскадная из звеньев 2-го порядка с прямой канонической транспонированной структурой) | <code>df2tsos(s,G)</code> | s, G — матрица коэффициентов и коэффициент усиления передаточной функции (3) |
| Direct-Form Symmetric FIR (прямая приведенная КИХ с симметричной импульсной характеристикой) | <code>dfsymfir(b)</code> | b — вектор коэффициентов передаточной функции (5). Вектор b имеет длину N и его элементы симметричны относительно точки $N/2$ |
| Direct-Form Antisymmetric FIR (прямая приведенная КИХ с антисимметричной импульсной характеристикой) | <code>dfasymfir(b)</code> | b — вектор коэффициентов передаточной функции (5). Вектор b имеет длину N и его элементы антисимметричны относительно точки $N/2$, при этом, если N — четное, то $b(N/2) = 0$ |

- с помощью функции:

```
disp(Hd)
```

Вывод конкретного свойства объекта `Hd` выполняется с помощью функции:

```
V=get(Hd,'name')
```

Здесь `name` — имя свойства; `V` — имя переменной, которой присваивается значение свойства, в противном случае оно присваивается переменной `ans`.

Установка доступных пользователю свойств выполняется с помощью функции:

```
set(Hd,'name1',value1,'name2',value2,...)
```

либо оператора присваивания

```
Hd.name=value
```

Здесь `value` — значение свойства.

Объекты `dfilt` можно сохранять на диске в виде файла данных с расширением `mat` в папке `work` или в собственной папке пользователя.

К сожалению, параллельная структура из звеньев 2-го порядка с передаточной функцией (4) в MATLAB не поддерживается. При необходимости пользователь может создать внешнюю функцию, реализующую данную структуру.

Пример 1

Задана передаточная функция в общем виде (1):

$$H(z) = \frac{0,3 + 0,1z^{-1} + 0,15z^{-2} + 0,01z^{-3} + 0,03z^{-4}}{1 - 1,57z^{-1} + 1,71z^{-2} - 0,99z^{-3} + 0,3z^{-4}} \quad (7)$$

Создать и сохранить на диске объект `Hd1` со структурой Direct-Form I (табл. 1) и вывести список его свойств:

```
>> b=[0.3 0.1 0.15 0.01 0.03];
>> a=[1 -1.57 1.71 -0.99 0.3];
>> Hd1=dfilt.df1(b,a)
Hd1 =
FilterStructure: 'Direct-Form I'
Arithmetic: 'double'
Numerator: [0.3000 0.1000 0.1500 0.0100 0.0300]
Denominator: [1 -1.5700 1.7100 -0.9900 0.3000]
PersistentMemory: false
>> save Hd1
```

Поясним смысл свойств для объектов `dfilt` с прямой структурой:

- `FilterStructure` — отображает структуру ЦФ (табл. 1).
- `Arithmetic` — отображает форму представления данных в структуре ЦФ и принимает значения: 'double' при представлении данных с плавающей точкой и двойной точностью; 'single' — данных с плавающей точкой и одинарной точностью; 'fixed' — данных с фиксированной точкой.
- `Numerator` — хранит коэффициенты числителя передаточной функции (1).
- `Denominator` — хранит коэффициенты знаменателя передаточной функции (1).
- `PersistentMemory` — отображает начальные условия при вычислении реакции и возвращает `true` (истину), если условия ненулевые, и `false` (ложь), если нулевые.

Пример 2

Задана передаточная функция (7). Создать и сохранить на диске объект `Hd5` со структурой Direct-Form I, Second-Order Sections (табл. 1).

Сначала представим передаточную функцию (7) в виде произведения (3) и определим матрицу коэффициентов s и коэффициент усиления G с помощью функции `tf2sos`, после чего создадим объект `Hd5` и выведем список его свойств:

```
>> b=[0.3 0.1 0.15 0.01 0.03];
>> a=[1 -1.57 1.71 -0.99 0.3];
```

```
>> [s,G]=tf2sos(b,a);
>> Hd5=dfilt.df1sos(s,G)
Hd5 =
FilterStructure: 'Direct-Form I, Second-Order Sections'
Arithmetic: 'double'
sosMatrix: [2x6 double]
ScaleValues: [0.3;1;1]
PersistentMemory: false
>> save Hd5
```

Поясним смысл дополнительных свойств `sosMatrix` и `ScaleValues` для объектов с каскадной структурой из звеньев 2-го порядка:

- `sosMatrix` — матрица коэффициентов `s`; Напомним [5], что при использовании функции `tf2sos` коэффициенты `s` передаточной функции (3) представляются в виде матрицы:

$$\begin{bmatrix} 1 & b_{11} & b_{21} & 1 & a_{11} & a_{21} \\ 1 & b_{12} & b_{22} & 1 & a_{12} & a_{22} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & b_{1L} & b_{2L} & 1 & a_{1L} & a_{2L} \end{bmatrix}.$$

- `ScaleValues` — вектор коэффициентов усиления `G`, элементы которого равны:
 - первый элемент — коэффициенту усиления на входе первого звена, то есть на входе структуры;
 - второй — коэффициенту усиления на входе второго звена и т. д.;
 - последний — коэффициенту усиления на выходе последнего звена, то есть на выходе структуры.

Выведем свойства `sosMatrix` и `ScaleValues` объекта `Hd5`, присваивая их значения переменным `s` и `G`:

```
>> s=get(Hd5,'sosMatrix')
s =
1.0000 0.5963 0.4170 1.0000 -1.0260 0.3979
1.0000 -0.2630 0.2398 1.0000 -0.5440 0.7539
>> G=get(Hd5,'ScaleValues')
G =
0.3000
1.0000
1.0000
```

В данном случае вектор-столбец `G` содержит три элемента: 0.3000 — коэффициент усиления на входе первого звена (на входе структуры), 1.0000 — на входе второго звена и 1.0000 — на выходе второго звена (на выходе структуры).

Соответствующая передаточная функция (3) имеет вид:

$$H(z) = 0,3 \times \frac{1+0,5963z^{-1}+0,417z^{-2}}{1-1,026z^{-1}+0,3979z^{-2}} \times \frac{1-0,263z^{-1}+0,2398z^{-2}}{1-0,544z^{-1}+0,7539z^{-2}}. \quad (8)$$

Функции MATLAB для объектов dfilt

Объекты `dfilt` можно применять в качестве входных параметров многих функций MATLAB из пакетов расширения Signal

Таблица 2. Функции MATLAB для объектов `dfilt`

| Функция | Назначение |
|---|---|
| <code>block(Hd)</code> | Создает модель объекта <code>Hd</code> в виде блока для Simulink |
| <code>Hdn=convert(Hd,'newstruct')</code> | Преобразует объект <code>Hd</code> в объект <code>Hdn</code> другой структуры из табл. 1 |
| <code>Hdc=copy(Hd)</code> | Создает новый независимый объект <code>Hdc</code> , сохраняющий все свойства объекта <code>Hd</code> , при этом изменение свойств <code>Hdc</code> не влияет на свойства <code>Hd</code> . При использовании оператора присваивания <code>Hdc=Hd</code> изменение свойств объекта <code>Hdc</code> приводит к автоматическому изменению свойств объекта <code>Hd</code> . |
| <code>filter(Hd,x)</code> | Вычисляет реакцию объекта <code>Hd</code> на воздействие <code>x</code> по разностному уравнению |
| <code>fvtool(Hd)</code> | Открывает графическое окно анализа характеристик ЦФ GUI FVTool с автоматически выведенной АЧХ |
| <code>order(Hd)</code> | Возвращает порядок передаточной функции объекта <code>Hd</code> |
| <code>realizemdl(Hd)</code> | Создает модель объекта <code>Hd</code> в виде подсистемы для Simulink |
| <code>reorder(Hd,dir_flag)</code> | Для объекта <code>Hd</code> с каскадной структурой из звеньев 2-го порядка управляет расстановкой звеньев с помощью параметра <code>dir_flag</code> |
| <code>scale(Hd,'norm')</code> | Для объекта <code>Hd</code> с каскадной структурой из звеньев 2-го порядка выполняет масштабирование согласно норме <code>norm</code> |
| <code>scalecheck(Hq,'norm')</code> | Для объекта <code>Hd</code> с каскадной структурой из звеньев 2-го порядка возвращает единичный вектор, длина которого равна числу звеньев, в том случае, если выполнено масштабирование на основе нормы <code>norm</code> |
| <code>Hds=sos(Hd)</code> <code>Hds=sos(Hd,dir_flag,scale)</code> | Преобразует объект <code>Hd</code> , имеющий прямую структуру, в объект <code>Hds</code> каскадной структуры из звеньев 2-го порядка, где звенья имеют прямую структуру объекта <code>Hd</code> . В расширенном формате выполняет расстановку звеньев с помощью параметра <code>dir_flag</code> и масштабирование с помощью параметра <code>scale</code> . Параметр <code>scale</code> может использоваться только для объекта <code>Hd</code> с прямой структурой <code>df2</code> |

Processing Toolbox и Filter Design Toolbox. Краткий список функций, которые будут использованы в дальнейшем, дается в таблице 2, а полный — может быть выведен по команде:

```
help dfilt
```

Описание структур ЦФ в виде объектов `dfilt` и использование для них функций MATLAB существенно упрощает анализ ЦФ, который в ряде случаев вообще невозможен без такого описания, например, анализ эффектов квантования в ЦФ с фиксированной точкой.

Пример 3

Преобразовать объект `Hd5` (см. пример 2) в объект `Hd6` со структурой Direct-Form I:

```
>> load Hd5
>> Hd6=convert(Hd5,'df1')
Hd6 =
FilterStructure: 'Direct-Form I'
Arithmetic: 'double'
Numerator: [0.3 0.1 0.15 0.009999999999999993 0.03]
Denominator: [1 -1.57 1.71 -0.9900000000000001 0.3]
PersistentMemory: false
```

Созданный объект `Hd6` тождественен объекту `Hd1` (пример 1).

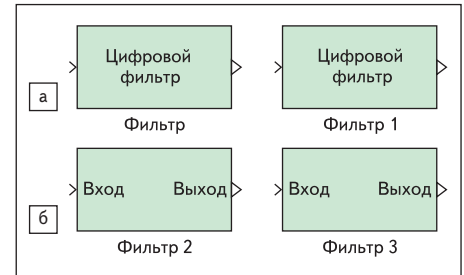


Рис. 2. Модели объектов `Hd1` и `Hd5` для Simulink в виде: а) блоков; б) подсистем

Пример 4

Создать модели объектов `Hd1` и `Hd5` (см. примеры 1 и 2) для Simulink в виде блоков и в виде подсистем (рис. 2):

```
>> load Hd1
>> load Hd5
>> block(Hd1),block(Hd5)
>> realizemdl(Hd1),realizemdl(Hd5)
```

Структура КИХ-фильтра в виде объекта dfilt

Свойства симметрии/антисимметрии импульсной характеристики (ИХ) КИХ-фильтров с ЛФЧХ [6] позволяют преобразовать прямую структуру КИХ-фильтра с РУ (6) в прямую приведенную Direct-Form Symmetric FIR или Direct-Form Antisymmetric FIR (табл. 1), которая получается в результате приведения подобных в РУ.

Например, КИХ-фильтр 1-го типа [6] длины $N = 9$ с симметричной ИХ описывается РУ:

$$y(n) = b_0x(n) + b_1x(n-1) + b_2x(n-2) + b_3x(n-3) + b_4x(n-4) + b_3x(n-5) + b_2x(n-6) + b_1x(n-7) + b_0x(n-8),$$

откуда после приведения подобных получаем РУ в виде:

$$y(n) = b_0[x(n)+x(n-8)] + b_1[x(n-1) + x(n-7)] + b_2[x(n-2)+x(n-6)] + b_3[x(n-3)+x(n-5)] + b_4x(n-4),$$

и алгоритм, отображаемый структурой Direct-Form Symmetric FIR (рис. 3).

Аналогично получают структуру Direct-Form Antisymmetric FIR, которая отличается от изображенной на рис. 3 инверсией знака у соответствующих слагаемых, что символически учитывается знаком минус при стрелке на входе сумматора.

Пример 5

Создать объект `dfilt` с именем `Hd6` — оптимальный КИХ-фильтр ФНЧ порядка $R = 25$ с симметричной ИХ (см. пример 4 в [6]).

Создадим объект `Hd6`, используя коэффициенты передаточной функции `b`:

```
>> load b
>> Hd6=dfilt.df1symfir(b)
```

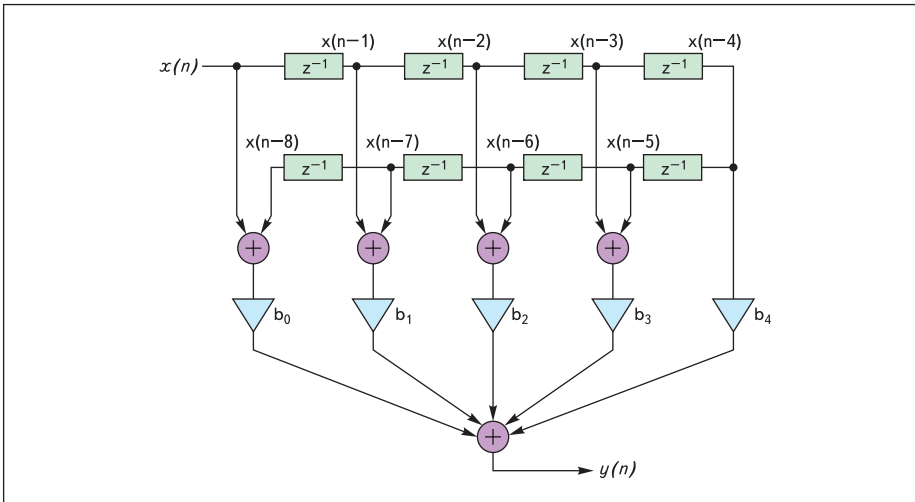


Рис. 3. Структура Direct-Form Symmetric FIR КИХ-фильтра 1-го типа

```
Hd6 =
FilterStructure: 'Direct-Form Symmetric FIR'
Arithmetic: 'double'
Numerator: [1x26 double]
PersistentMemory: false
```

Структура БИХ-фильтра в виде объекта dfilt

Типовые структуры БИХ-фильтров, поддерживаемые в MATLAB, приведены в таблице 1.

Пример 6

Создать объект dfilt с именем He — оптимальный БИХ-фильтр ПФ порядка $R = 4$ (см. пример 3 в [7]) — с каскадной структурой Direct-Form II, Second-Order Sections (табл. 1).

Сначала, используя сохраненные коэффициенты передаточной функции b4, a4, создадим объект Hed с прямой структурой Direct-Form I:

```
>> load b4
>> load a4
>> Hed=dfilt.df1(b4,a4);
```

Затем преобразуем объект Hed в объект He со структурой Direct-Form II, Second-Order Sections:

```
>> He=convert(Hed,'df2sos');
>> get(He,'FilterStructure')
ans =
Direct-Form II, Second-Order Sections
```

Подобным образом можно создать описание структуры БИХ-фильтра в виде объекта dfilt с любой из числа возможных структур (табл. 1).

Описание требований к АЧХ КИХ- и БИХ-фильтров в виде объектов fdesign

В MATLAB в пакете Filter Design Toolbox имеются средства синтеза КИХ- и БИХ-фильтров

непосредственно в виде объекта dfilt. В этом случае требования к АЧХ должны описываться в виде объекта fdesign (от англ. Filter Design Object):

```
Hs=fdesign.type('sp1,sp2,...',sp1,sp2,...[,Fs],[,magunits])
```

Здесь Hs — имя объекта fdesign; fdesign — тип объекта; type — функция, задающая абстрактный тип объекта; на ее место ставится функция, задающая конкретный тип объекта Hs (табл. 3); 'sp1, sp2, ...' — список обязательных параметров функции type (может отсутствовать).

Список обязательных параметров регламентирован и зависит от типа объекта, причем для каждого из них предлагается несколько вариантов списков, которые можно вывести по команде:

```
help fdesign.type
```

если вместо абстрактной функции type указать конкретную функцию из таблицы 3.

Таблица 3. Основные типы объектов fdesign

| Функция | Тип объекта |
|----------|-----------------------|
| lowpass | Lowpass Filter — ФНЧ |
| highpass | Highpass Filter — ФВЧ |
| bandpass | Bandpass Filter — ПФ |
| bandstop | Bandstop Filter — ПФ |

Для синтеза частотно-избирательных ЦФ целесообразно выбрать вариант списка параметров по умолчанию; он представляет собой список требований к АЧХ (дБ) в строго фиксированной последовательности со стандартными обозначениями параметров (табл. 4–6):

- sp1, sp2, ... — значения обязательных параметров; по умолчанию значения граничных частот задаются в шкале нормированных частот $f = f_d(f_d/2)$ в основной полосе [0;1];

Таблица 4. Требования к АЧХ (дБ) для объекта типа lowpass и highpass

| Обозначение в MATLAB | Обозначение в ЦОС |
|----------------------|---|
| Fp | f_p — граничная частота ПП |
| Fst | f_s — граничная частота ПЗ |
| Ap | σ_{max} (дБ) — максимально допустимое затухание в ПП |
| Ast | σ_{min} (дБ) — минимально допустимое затухание в ПЗ |

Таблица 5. Требования к АЧХ (дБ) для объекта типа bandpass

| Обозначение в MATLAB | Обозначение в ЦОС |
|----------------------|---|
| Fst1 | f_{-k} — граничная частота ПЗ1 |
| Fp1 | f_{-l} — левая граничная частота ПП |
| Fp2 | f_k — правая граничная частота ПП |
| Fst2 | f_k — граничная частота ПЗ2 |
| Ast1 | σ_{min}^1 (дБ) — минимально допустимое затухание в ПЗ1 |
| Ap | σ_{max} (дБ) — максимально допустимое затухание в ПП |
| Ast2 | σ_{min}^2 (дБ) — минимально допустимое затухание в ПЗ2 |

Таблица 6. Требования к АЧХ (дБ) для объекта типа bandstop

| Обозначение в MATLAB | Обозначение в ЦОС |
|----------------------|--|
| Fp1 | f_{-l} — граничная частота ПП1 |
| Fst1 | f_{-k} — левая граничная частота ПЗ |
| Fst2 | f_k — правая граничная частота ПЗ |
| Fp2 | f_k — граничная частота ПП2 |
| Ap1 | σ_{max}^1 (дБ) — максимально допустимое затухание в ПП1 |
| Ast | σ_{min} (дБ) — минимально допустимое затухание в ПЗ |
| Ap2 | σ_{max}^2 (дБ) — максимально допустимое затухание в ПП2 |

- Fs — необязательный параметр, используемый для задания частоты дискретизации f_d (Гц); в этом случае все граничные частоты также должны задаваться в шкале абсолютных частот f (Гц);
- magunits — необязательный параметр, устанавливающий единицы измерения допустимых отклонений АЧХ и принимающий значения:

- 'dB' (по умолчанию) — если требования задаются к АЧХ (дБ) (см. (5) в [6]);
- 'linear' — если требования задаются к нормированной АЧХ; в этом случае они вводятся в безразмерных единицах, но выводятся в дБ.

Выходным параметром является объект Hs (массив записей — struct array) со списком свойств (полей), зависящих от типа фильтра.

Вывод списка свойств объекта fdesign выполняется так же, как для объекта dfilt.

Допускается короткий формат описания объекта fdesign:

```
Hs=fdesign.type
```

В этом случае объект Hs создается по умолчанию.

Пример 7

Создать объект Hs типа lowpass по умолчанию и вывести список его свойств:

```
>> Hs=fdesign.lowpass
Hs =
Response: 'Lowpass'
Specification: 'Fp,Fst,Ap,Ast'
Description: {4x1 cell}
NormalizedFrequency: true
    Fpass: 0.45
    Fstop: 0.55
    Apass: 1
    Astop: 60
```

Поясним смысл свойств объекта fdesign:

- Response — отображает тип объекта fdesign (табл. 3);
- SpecificationType — содержит список параметров функции type (в примере это функция lowpass) в заданной последовательности;
- Description — представляет собой массив ячеек (cell array), символьные элементы которого расшифровывают смысл параметров функции type (в примере — функции lowpass):

```
>> get(Hs,'Description')
ans =
'Passband Frequency'
'Stopband Frequency'
'Passband Ripple (dB)'
'Stopband Attenuation (dB)'
```

- NormalizedFrequency — флаг, устанавливающий единицы измерения частот: безразмерные (true) или в герцах (false). Смысл остальных свойств очевиден.

Подобно объекту dfilt, объект fdesign можно сохранять на диске в виде файла данных с расширением mat в папке work или в собственной папке.

Синтез КИХ- и БИХ-фильтров в виде объектов dfilt на основе объектов fdesign

Объект fdesign создается для синтеза ЦФ непосредственно в виде объекта dfilt. С этой целью в пакет расширения Filter Design Toolbox включены дополнительные функции синтеза, основные из которых представлены в таблице 7.

Таблица 7. Основные функции для синтеза фильтров на основе объектов fdesign

| Функция | Метод синтеза и название цифрового фильтра в MATLAB |
|------------|---|
| butter | Метод билинейного Z-преобразования; Butterworth filter — БИХ-фильтр Баттерворта |
| cheby1 | Метод билинейного Z-преобразования; Chebyshev Type I filter — БИХ-фильтр Чебышева I рода |
| cheby2 | Метод билинейного Z-преобразования; Chebyshev Type II filter — БИХ-фильтр Чебышева II рода |
| ellip | Метод билинейного Z-преобразования; Elliptic filter — БИХ-фильтр Золотарева-Кауэра (эллиптический) |
| equiripple | Метод наилучшей равномерной (чебышевской) аппроксимации; Equiripple filter — оптимальный КИХ-фильтр (равноволновый) |
| kaiserwin | Метод окон с использованием окна Кайзера; Filter using a Kaiser window — КИХ-фильтр с окном Кайзера |

Рассмотрим синтез КИХ- и БИХ-фильтров в виде объектов dfilt, при условии, что требования к АЧХ (дБ) описаны в виде объекта fdesign указанным ранее способом.

Синтез БИХ-фильтров выполняется с помощью функции:

```
Hf=function_iir(Hs,'MatchExactly',MATCH)
```

Здесь function_iir — абстрактная функция, вместо которой указывается конкретная функция из таблицы 7 для БИХ-фильтра; Hs — имя объекта fdesign; 'MatchExactly' — флаг, установка которого (присутствие в составе параметров) означает, что требования к АЧХ должны выполняться точно; MATCH — параметр, уточняющий, в какой именно из полос требования к АЧХ должны выполняться точно; при использовании для синтеза БИХ-фильтров функций butter, cheby1 и cheby2 точное выполнение требований к АЧХ гарантируется только для одной из полос — ПЗ или ПП (соответственно, параметр MATCH принимает значения: 'stopband' (по умолчанию) — в ПП, 'passband' — в ПЗ), а при использовании для синтеза БИХ-фильтров функции ellip точное выполнение требований к АЧХ может гарантироваться как в одной из полос (ПЗ или ПП), так и в обеих при значении параметра MATCH, равного 'both'; Hf — имя объекта dfilt, описывающего структуру БИХ-фильтра.

Синтез КИХ-фильтров выполняется с помощью функции:

```
Hf=function_fir(Hs)
```

Здесь function_fir — абстрактная функция, вместо которой указывается конкретная функция из таблицы 7 для КИХ-фильтра; Hs — имя объекта fdesign; Hf — имя объекта dfilt, описывающего структуру КИХ-фильтра.

При синтезе КИХ-фильтров точное выполнение требований к АЧХ гарантируется автоматически в обеих полосах, поэтому параметры 'MatchExactly' и MATCH отсутствуют.

- Синтез ЦФ происходит в два этапа:
1. Создается объект fdesign.
 2. Синтезируется ЦФ в виде объекта dfilt с помощью функции из таблицы 7.

Пример 8

Заданы требования к АЧХ ПФ (см. табл. 4 в [6]). Синтезировать оптимальный БИХ-фильтр Золотарева-Кауэра (эллиптический) в виде объекта dfilt.

Синтез БИХ-фильтра выполним в два этапа:

1. На первом этапе создадим объект fdesign с именем Hs1 с учетом обозначений таблицы 5 и для контроля выведем список его свойств:

```
>> Fs=8000; fk1=1000; ft1=1400; ft2=2000; fk2=2400;
>> Fst1=fk1/(Fs/2); Fp1=ft1/(Fs/2); Fp2=ft2/(Fs/2);...
```

```
Fst2=fk2/(Fs/2);
>> Ast1=40; Ap=0.4455; Ast2=40;
>> Hs1=fdesign.bandpass...
('Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2','Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2')
Hs1 =
Response: 'Bandpass'
Specification: 'Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2'
Description: {7x1 cell}
NormalizedFrequency: true
    Fstop1: 0.25
    Fpass1: 0.35
    Fpass2: 0.5
    Fstop2: 0.6
    Astop1: 40
    Apass: 0.4455
    Astop2: 40
```

2. На втором этапе на основе объекта Hs1 с помощью функции ellip из таблицы 7 синтезируем БИХ-фильтр в виде объекта dfilt с именем Hf1, выведем список его свойств и сохраним на диске для дальнейшего использования:

```
>> Hf1=ellip(Hs1,'MatchExactly','both')
Hf1 =
FilterStructure: 'Direct-Form II, Second-Order Sections'
Arithmetic: 'double'
sosMatrix: {4x6 double}
ScaleValues: {5x1 double}
PersistentMemory: false
>> save Hf1
```

Синтезированный БИХ-фильтр (объект Hf1) по умолчанию имеет структуру Direct-Form II, Second-Order Sections (табл. 1), для преобразования которой в другую можно воспользоваться функцией convert.

Пример 9

На основе созданного в примере 8 объекта fdesign с именем Hs1 синтезировать оптимальный КИХ-фильтр в виде объекта dfilt.

Синтез КИХ-фильтра выполним в два этапа:

1. На первом этапе создается объект fdesign, который уже имеется — Hs1.
2. На втором этапе на основе объекта Hs1 с помощью функции equiripple из таблицы 7 синтезируем КИХ-фильтр в виде объекта dfilt с именем Hf2 и выведем список его свойств:

```
>> Hf2=equiripple(Hs1)
Hf2 =
FilterStructure: 'Direct-Form FIR'
Arithmetic: 'double'
Numerator: {1x41 double}
PersistentMemory: false
```

Синтезированный КИХ-фильтр (объект Hf2) по умолчанию имеет Direct-Form FIR (табл. 1). Однако, по определению, ИХ КИХ-фильтра с ЛФЧХ — симметричная или антисимметричная, что соответствует структуре Direct-Form Symmetric FIR или Direct-Form Antisymmetric FIR (табл. 1).

Для преобразования структуры воспользуемся функцией convert. Предварительно определим, какую именно из данных структур имеет синтезированный КИХ-фильтр, для чего выведем и сравним первый и последний отсчеты ИХ, сохраненные в свойстве Numerator:

```
>> b=get(Hf2,'Numerator');
>> [b(1) b(41)]
ans =
    1.0e-003 *
   -0.7887 -0.7887
```

Имеем симметричную ИХ, следовательно, КИХ-фильтр должен иметь структуру Direct-Form Symmetric FIR. Создадим объект Hf3 с данной структурой, выведем список его свойств и сохраним на диске для дальнейшего использования:

```
>> Hf3=convert(Hf2,'dfsymfir')
Hf3 =
FilterStructure: 'Direct-Form Symmetric FIR'
Arithmetic: 'double'
Numerator: [1x41 double]
PersistentMemory: false
>> save Hf3
```

Синтезированные в виде объектов dfilt КИХ- и БИХ-фильтры будут использованы в следующих статьях для анализа эффектов квантования.

Анализ ЦФ в виде объекта dfilt

Характеристики ЦФ в виде объекта dfilt можно просмотреть в окне **Filter Visualization Tool GUI FVTool**, обращение к которому осуществляется с помощью функции [6]:

```
fvtool (Hd)
```

где Hd — имя объекта dfilt.

При использовании коротких форматов функций синтеза КИХ- и БИХ-фильтров:

```
function_fir(Hs)
function_iir(Hs,'MatchExactly','MATCH')
```

Независимо от того, стоит или нет точка в конце строки, происходит автоматический выход в окно **Filter Visualization Tool**. ■

Литература

1. Ingle V., Proakis J. Digital Signal Processing Using MATLAB. Second Edition — Thomson.
2. Оппенгейм 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.