

Моделирование цифровой обработки сигналов в MATLAB.

Часть 5. Моделирование структур цифровых фильтров с фиксированной точкой программными средствами MATLAB: анализ характеристик БИХ-фильтров

Алла СОЛОНИНА

Настоящая статья продолжает тему, начатую в [9]: моделирование структур цифровых фильтров (ЦФ) с фиксированной точкой (ФТ) программными средствами MATLAB¹. В [9] были перечислены источники ошибок квантования в ЦФ с ФТ, с учетом которых составлена нелинейная модель ЦФ с ФТ, предназначенная для компьютерного моделирования.

Моделирование структуры БИХ-фильтра с фиксированной точкой

В MATLAB весьма широко представлены средства моделирования структур КИХ- и БИХ-фильтров с ФТ в пакетах расширения Filter Design Toolbox и Fixed Point Toolbox. В серии статей, начиная с [9], рассматривается методика моделирования структур ЦФ с ФТ, иллюстрируемая на конкретных примерах. Данная методика согласуется с предлагаемой в MATLAB; с ней можно познакомиться, обратившись к справочной системе в формате HTML, используя поиск по ключевой фразе “Quantized Filters” (квантованные фильтры).

Моделирование структуры БИХ-фильтра с ФТ начинается с описания его исходной структуры (с неквантованными данными) в виде объекта `dfilt`, что можно сделать двумя способами, с которыми мы познакомились ранее [8]:

- синтезировать БИХ-фильтр по заданным требованиям к АЧХ, выбрать необходимую структуру фильтра и описать ее в виде объекта `dfilt`;
- синтезировать БИХ-фильтр непосредственно в виде объекта `dfilt` по требованиям к АЧХ, описанным в виде объекта `fdesign`; в этом случае структура БИХ-фильтра выбирается автоматически, и для ее изменения придется воспользоваться функцией `convert`.

Характерной особенностью исходных структур БИХ-фильтров (объектов `dfilt`) является значение свойства `Arithmetic`: `'double'`. Это значит, что в исходной структуре БИХ-фильтра все данные — коэффициенты передаточной функции, воздействие, результаты выполнения арифметических операций при вычислении реакции и сама реакция — представлены числами максимальной разрядности типа `double` (условно бесконечной).

Для краткости используем терминологию: исходным БИХ-фильтром будем называть исходную структуру БИХ-фильтра, описанную в виде объекта `dfilt` со значением свойства `Arithmetic`: `'double'`.

Для моделирования структур БИХ-фильтров с ФТ в пакете расширения Filter Design Toolbox предусмотрена возможность модификации объекта `dfilt` при значении свойства `Arithmetic`: `'fixed'`.

В дальнейшем используем принятую в MATLAB терминологию: БИХ-фильтром с ФТ (Fixed-Point IIR Filter) будем называть структуру БИХ-фильтра с ФТ, описанную в виде объекта `dfilt` со значением свойства `Arithmetic`: `'fixed'`.

Исходный БИХ-фильтр

В качестве исходного БИХ-фильтра выберем объект `Hf1` — оптимальный БИХ-фильтр Золотарева-Кауэра (Elliptic filter), синтезированный непосредственно в виде объекта `dfilt` с автоматически выбранной каскадной структурой из звеньев 2-го порядка с прямой ка-

нонической структурой — Direct-form II, Second-order sections (пример 8 [8]).

Пример 1

Вывести свойства исходного БИХ-фильтра — объекта `Hf1`:

```
>> load Hf1
>> Hf1
Hf1 =
FilterStructure: 'Direct-Form II, Second-Order Sections'
Arithmetic: 'double'
sosMatrix: [4x6 double]
ScaleValues: [5x1 double]
PersistentMemory: false
```

Выведенные свойства объекта `Hf1` — объекта `dfilt` с `Arithmetic`: `'double'` — комментировались в [8].

БИХ-фильтр с ФТ и его свойства

БИХ-фильтр с ФТ формируется на основе исходного БИХ-фильтра путем присваивания свойству `Arithmetic` значения `'fixed'`.

Если исходный БИХ-фильтр имеет каскадную структуру из звеньев 2-го порядка, то предварительно (до изменения свойства `Arithmetic` на `'fixed'`) в данной структуре для предотвращения или минимизации ошибок переполнения необходимо выполнить следующие действия:

1. Сформировать звенья посредством объединения полюсов с ближайшими нулями, после чего расставить звенья в порядке возрастания радиусов полюсов для минимизации собственных шумов в структуре ЦФ с ФТ.

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

2. Выполнить масштабирование для предотвращения или минимизации переполнений на выходах сумматоров.

Требуемое формирование звеньев и их расстановка в объектах `dfilt` производятся автоматически [5].

Процедура масштабирования программными средствами MATLAB — самостоятельная и весьма важная тема, она подробно рассматривается в [5]. Здесь лишь отметим, что для выполнения масштабирования используется функция (табл. 2 [8]):

```
scale(Hd,norm)
```

Здесь `Hd` — объект с каскадной структурой из звеньев 2-го порядка; `norm` — норма, на основе которой производится масштабирование.

Пример 2

Сформировать БИХ-фильтр с ФТ в виде объекта `Hq1` на основе исходного БИХ-фильтра — объекта `Hf1` (пример 1). Предварительно в объекте `Hf1` выполнить масштабирование, минимизирующее ошибки переполнения в БИХ-фильтре с ФТ (полагая, что требуемое формирование звеньев и их расстановка выполнены по умолчанию).

Создадим объект `Hf1s` — копию объекта `Hf1`, выполним в нем масштабирование на основе нормы `Linf` (это норма L_∞ по умолчанию) с помощью функции `scale` и проверим результат с помощью функции `scalecheck` (табл. 2 [8]). На основе объекта `Hf1s` создадим БИХ-фильтр с ФТ — объект `Hq1` — и сохраним объекты `Hf1s` и `Hq1` на диске:

```
>> load Hf1
>> Hf1s=copy(Hf1);
>> scale(Hf1s)
>> scalecheck(Hf1s,'Linf')
ans =
    1.0000    1.0000    1.0000    1.0000
>> Hq1=copy(Hf1s);
>> set(Hq1,'Arithmetic','fixed')
>> save Hf1s
>> save Hq1
```

Список основных свойств БИХ-фильтра с ФТ, доступных пользователю, выводится по имени объекта `dfilt`. Полный список свойств, включающий основные свойства, а также свойства, при определенных условиях доступные пользователю, выводится с помощью функции:

```
get(<имя объекта>)
```

Пример 3

Для БИХ-фильтра с ФТ — объекта `Hq1` (пример 2) — вывести список основных свойств по его имени (таблица, левый столбец) и полный список свойств с помощью функции `get` (таблица, правый столбец).

Свойства, выделенные полужирным шрифтом (таблица, правый столбец), будут использоваться далее.

Таблица. Список свойств БИХ-фильтра с ФТ

Основные свойства	Полный список свойств
<pre>>> Hq1 Hq1 = FilterStructure: 'Direct-Form II, Second-Order Sections' Arithmetic: 'fixed' sosMatrix: [4x6 double] ScaleValues: [0.0518531799316406;1;1;1;1] PersistentMemory: false CoeffWordLength: 16 CoeffAutoScale: true Signed: true InputWordLength: 16 InputFracLength: 15 SectionInputWordLength: 16 SectionInputAutoScale: true SectionOutputWordLength: 16 SectionOutputAutoScale: true OutputWordLength: 16 OutputMode: 'AvoidOverflow' StateWordLength: 16 StateFracLength: 15 ProductMode: 'FullPrecision' AccumMode: 'KeepMSB' AccumWordLength: 40 CastBeforeSum: true RoundMode: 'convergent' OverflowMode: 'wrap'</pre>	<pre>>> get(Hq1) PersistentMemory: 0 NumSamplesProcessed: 0 FilterStructure: 'Direct-Form II, Second-Order Sections' States: [2x4 embedded.fi] Arithmetic: 'fixed' sosMatrix: [4x6 double] ScaleValues: [5x1 double] CoeffWordLength: 16 CoeffAutoScale: 1 Signed: 1 RoundMode: 'convergent' OverflowMode: 'wrap' InputWordLength: 16 InputFracLength: 15 OutputWordLength: 16 OutputMode: 'AvoidOverflow' OutputFracLength: 11 ProductMode: 'FullPrecision' ProductWordLength: 32 AccumMode: 'KeepMSB' AccumWordLength: 40 CastBeforeSum: 1 NumAccumFracLength: 29 DenAccumFracLength: 29 NumProdFracLength: 29 DenProdFracLength: 29 NumFracLength: 14 DenFracLength: 14 ScaleValueFracLength: 19 StateWordLength: 16 SectionInputWordLength: 16 SectionOutputWordLength: 16 StateFracLength: 15 SectionInputAutoScale: 1 SectionInputFracLength: 14 SectionOutputAutoScale: 1 SectionOutputFracLength: 11</pre>

Назначение свойств фильтров с ФТ дается в [5]. (Отметим, что в версии MATLAB 7.0, описываемой в [5], имеются небольшие расширения в свойствах по сравнению с версией MATLAB 7.6, используемой в данной статье.)

Подробную информацию о свойствах объектов `dfilt` с различными структурами можно получить с помощью справочной системы MATLAB в формате HTML, используя поиск по ключевой фразе “Quantized Filters” и обращаясь к разделам, описывающим объекты `dfilt` с различными структурами.

Необходимые свойства, используемые далее, будут поясняться по мере изложения материала.

Квантование коэффициентов в БИХ-фильтрах с ФТ

Процедуру квантования коэффициентов в БИХ-фильтрах с ФТ поясним на следующих примерах.

Пример 4

Вывести неквантованные коэффициенты передаточной функции исходного БИХ-фильтра (объекта `Hf1s` в примере 2).

В данном случае объект `Hf1s` имеет каскадную структуру [8], поэтому необходимо вывести матрицу с коэффициентами передаточных функций звеньев (свойство `sosMatrix`) и вектор коэффициентов усиления (свойство `ScaleValues`). Присвоим данным свойствам имена `sf1s` и `Gf1s` соответственно:

```
>> load Hf1s
>> sf1s=get(Hf1s,'sosMatrix')
sf1s =
    0.9706   -1.1312    0.9706    1.0000   -0.8915    0.9416
    0.7229    0.2531    0.7229    1.0000    0.0126    0.9345
```

```
    0.4830   -0.7748    0.4830    1.0000   -0.6356    0.8056
    0.8935    0.9781    0.8935    1.0000   -0.2179    0.7950
>> Gf1s=get(Hf1s,'ScaleValues')
Gf1s =
    0.0519
    1.0000
    1.0000
    1.0000
    1.0000
```

Один из коэффициентов числителя по модулю превосходит единицу (в матрице `sf1s` он выделен полужирным шрифтом), следовательно, необходимо нормирование коэффициентов числителя. В MATLAB это делается с помощью функции `normalize` (табл. 2 [9]). С этой целью выполним следующие действия:

1. Создадим объект `Hn` — копию объекта `Hf1s`.
2. В объекте `Hn` выполним нормирование коэффициентов числителя для передаточных функций всех звеньев с помощью функции `normalize`, после чего выведем значения свойств `sosMatrix` и `ScaleValues`, присваивая им имена `sn` и `Gn` соответственно.
3. Сохраним объект `Hn` с неквантованными, но нормированными коэффициентами для дальнейшего использования.
4. Создадим объект `Hqn1` — копию объекта `Hn`.
5. В объекте `Hqn1` установим `Arithmetic: 'fixed'` и выведем значения свойств `sosMatrix` и `ScaleValues`, присваивая им имена `sq` и `Gq` соответственно.
6. Сохраним объект `Hqn1` с квантованными и нормированными коэффициентами на диске:

```
>> load Hf1s
>> Hn=copy(Hf1s);
>> K=normalize(Hn)
K =
    1.1312
    0.7229
    0.7748
    0.9781
>> sn=get(Hn,'sosMatrix')
sn =
    0.8580   -1.0000    0.8580   -0.8915    0.9417
    1.0000    0.3502    1.0000    1.0000    0.0126    0.9345
    0.6234   -1.0000    0.6234    1.0000   -0.6356    0.8056
    0.9135    1.0000    0.9135    1.0000   -0.2179    0.7950
>> Gn=get(Hn,'ScaleValues')
Gn =
    0.0519
    1.0000
    1.0000
    1.0000
    1.0000
>> save Hn
>> Hqn1=copy(Hn);
>> set(Hqn1,'Arithmetic','fixed')
>> save Hqn1
```

Значения квантованных коэффициентов `sq` объекта `Hqn1` можно вывести с помощью функции:

```
>> sq=get(Hqn1,'sosMatrix')
```

Для того чтобы увидеть отличие квантованных коэффициентов от неквантованных, следует установить формат `format long`.

Необходимо иметь в виду, что реакция объектов `Hn` и `Hqn1` будет отличаться от реакции исходного объекта `Hf1s` на нормиру-

ющий множитель GK, равный произведению элементов вектора K:

```
>> GK=1.1312*0.7229*0.7748*0.9781
GK =
    6197
```

Нормирующий множитель GK также следует учитывать при вычислении таких характеристик объектов Hn и Hqn1, как импульсная, переходная, АЧХ и др. (пример 7).

Пример 5

Создать объект Hqn2 — копию объекта Hqn1 (пример 4). Установить в нем требуемые значения свойств, связанных с квантованием коэффициентов, и сохранить объект Hqn2 на диске.

Создадим объект Hqn2 — копию объекта Hqn1 — и выведем полный список свойств объекта Hqn2, установленных по умолчанию, с помощью функции get:

```
>> Hqn2=copy(Hqn1);
>> get(Hqn2)
```

Ради экономии места далее оставлены только свойства, связанные с квантованием коэффициентов (в таблице аналогичные свойства объекта Hq1 выделены полужирным шрифтом):

```
CoeffWordLength:16
CoeffAutoScale:1
Signed:1
NumFracLength:14
DenFracLength:14
ScaleValueFracLength:19
```

Поясним коротко их смысл:

- **CoeffWordLength** — отображает формат представления коэффициентов передаточной функции БИХ-фильтра (1) [8] — слово.
- **NumFracLength** — длина дробной части для коэффициентов числителя передаточной функции БИХ-фильтра в слове **CoeffWordLength**.
- **DenFracLength** — длина дробной части для коэффициентов знаменателя передаточной функции БИХ-фильтра в слове **CoeffWordLength**.
- **CoeffAutoScale** — флаг, при сбросе которого (значении 0) можно произвольно задавать длины дробных частей **NumFracLength** и **DenFracLength**.
- **Signed** — флаг, управляющий знаковыми (при установке) или беззнаковыми (при сбросе) коэффициентами передаточной функции БИХ-фильтра.
- **ScaleValueFracLength** — для БИХ-фильтров с ФТ с каскадной структурой из звеньев 2-го порядка: длина дробной части для коэффициентов усиления; доступно при сбросе (значении 0) флага **CoeffAutoScale**.
В объекте Hqn2 оставим неизменными значения свойств **CoeffWordLength:16** и **Signed:1**, но изменим длину дробных частей **NumFrac-**

Length, **DenFracLength** и **ScaleValueFracLength**, для чего предварительно установим **CoeffAutoScale:0**. Сохраним объект Hqn2 с новыми свойствами для дальнейшего использования:

```
>> load Hqn1
>> Hqn2=copy(Hqn1);
>> set(Hqn2,'CoeffAutoScale',0)
>> set(Hqn2,'NumFracLength',15)
>> set(Hqn2,'DenFracLength',15)
>> set(Hqn2,'ScaleValueFracLength',15)
>> save Hqn2
```

Значения квантованных коэффициентов **sqn2** объекта Hqn2 можно вывести с помощью функции:

```
>> sqn2=get(Hqn2,'sosMatrix')
```

Для того чтобы увидеть отличие квантованных коэффициентов от неквантованных (пример 4), следует установить **format long**.

Пример 6

Создать объект Hqn3 — копию объекта Hqn1 (пример 4). Установить в нем требуемые значения свойств, связанных с квантованием коэффициентов, и сохранить объект Hqn3 на диске:

```
>> load Hqn1
>> Hqn3=copy(Hqn1);
>> set(Hqn3,'CoeffWordLength',8)
>> set(Hqn3,'CoeffAutoScale',0)
>> set(Hqn3,'NumFracLength',7)
>> set(Hqn3,'DenFracLength',7)
>> set(Hqn3,'ScaleValueFracLength',7)
>> save Hqn3
```

Выведем значения квантованных коэффициентов **sqn3** объекта Hqn3:

```
>> sqn3=get(Hqn3,'sosMatrix')
sqn3 =
    0.8594   -1.0000    0.8594    1.0000   -0.8906    0.9453
    0.9922    0.3516    0.9922    1.0000    0.0156    0.9375
    0.6250   -1.0000    0.6250    1.0000   -0.6328    0.8047
    0.9141    0.9922    0.9141    1.0000   -0.2188    0.7969
```

Сравнивая полученные квантованные коэффициенты с неквантованными (пример 4), видим их отличия и без установки **format long**.

Если в исходном БИХ-фильтре (объекте Hf1s в примере 4) с прямой структурой звеньев **Direct-Form I** коэффициент усиления (это первый элемент вектора **ScaleValues** (см. вектор **Gf1s** в примере 4)) превосходит единицу, то его следует нормировать путем деления на нормирующий множитель, обычно равный степени двойки — 2^n . В этом случае значения реакции необходимо умножить на произведение нормирующих множителей 2^n и GK (если производилось нормирование коэффициентов числителей передаточных функций звеньев).

В исходном БИХ-фильтре с прямой канонической структурой звеньев **Direct-Form II** аналогичная процедура нормирования вы-

полняется для последнего элемента вектора **ScaleValues** [5].

Значительно сложнее обстоит дело с нормированием коэффициентов знаменателя a_{1k} передаточных функций звеньев (3) [8]. (Коэффициенты знаменателя a_{2k} устойчивого фильтра меньше единицы по определению). Для этого в MATLAB не предусмотрено формализованного алгоритма. Один из возможных способов заключается в выборе формата данных таким, чтобы в слове **CoeffWordLength** старшие значащие биты, следующие за знаковым, интерпретировались как целая часть числа. Например, если установить в слове **CoeffWordLength:16** следующую длину дробных частей:

```
NumFracLength:14
DenFracLength:14
ScaleValueFracLength:14,
```

то для коэффициентов числителя и знаменателя, а также коэффициента усиления старший (следующий за знаковым) бит будет интерпретироваться как целая часть числа.

В этом случае для правильной интерпретации значений реакции потребуется выполнить их масштабирование, например, используя свойство **Slope** [5], которое мы поясним в следующей статье.

В ЦПОС подобная операция эквивалентна масштабированию данных путем их сдвига, что учитывается в алгоритме вычисления реакции (структуре фильтра). Однако в MATLAB используются типовые структуры **dfilt**, поэтому такие действия для пользователя недоступны.

Анализ характеристик БИХ-фильтров с ФТ

Для анализа характеристик БИХ-фильтров с ФТ воспользуемся функцией **fvtool** [9].

Пример 7

Вывести графики АЧХ:

1. Исходного БИХ-фильтра с неквантованными коэффициентами — объекта Hf1s (пример 4).
 2. БИХ-фильтра с нормированными неквантованными коэффициентами — объекта Hn (пример 4).
 3. БИХ-фильтра с ФТ с 16-разрядными коэффициентами — объекта Hqn2 (пример 5).
 4. БИХ-фильтра с ФТ с 8-разрядными коэффициентами — объекта Hqn3 (пример 6).
- Указать частоту дискретизации 8 кГц (она использовалась при синтезе исходного БИХ-фильтра [7]) и разместить легенду (рисунок):

```
>> load Hf1s
>> load Hn
>> load Hqn2
>> load Hqn3
>> h1=fvtool(Hf1s,Hn,Hqn2,Hqn3);
>> set(h1,'ShowReference','off')
>> set(h1,'Fs','8','legend','on')
>> legend(h1,'Reference IIR','Normalize IIR','IIR 16 bits','IIR 8 bits')
```

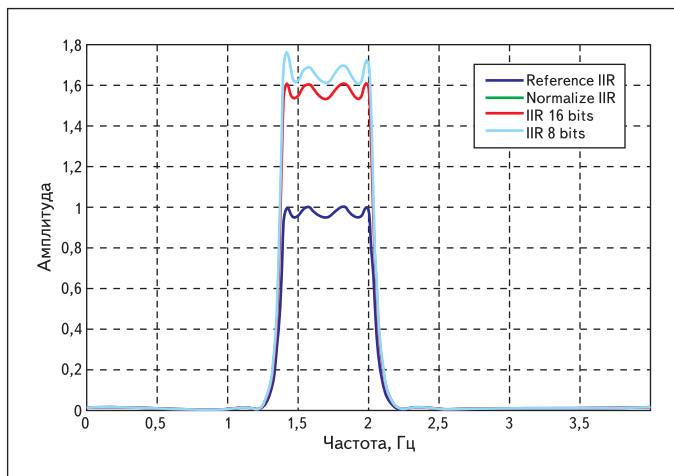


Рисунок. АЧХ БИХ-фильтров с некантованными, нормированными некантованными и квантованными коэффициентами

Для вывода АЧХ, вместо АЧХ (дБ) — характеристики ослабления (5) [6], выводимой по умолчанию, в пункте меню **Analysis** (анализ) была выбрана команда **Analysis Parameters** (параметры анализа) и в раскрывающемся списке **Magnitude Display** (вывод АЧХ) — значение **Magnitude** (АЧХ).

АЧХ исходного БИХ-фильтра и БИХ-фильтра с нормированными коэффициентами отличаются нормирующим множителем $GK=0.6197$ (пример 4). Поэтому при оценке влияния на АЧХ операции квантования коэффициентов имеет смысл сравнить АЧХ БИХ-фильтра с некантованными нормированными коэффициентами (объекта H_n) с АЧХ БИХ-фильтров с квантованными нормированными коэффициентами (объектов H_{qn2} и H_{qn3}).

АЧХ БИХ-фильтра с некантованными нормированными коэффициентами (объекта H_n) и АЧХ БИХ-фильтра с 16-разрядными коэффициентами (объекта H_{qn2}) практически совпали, а с 8-разрядными отличаются.

Выбирая в пункте меню **Analysis** соответствующие команды, можно вывести другие характеристики БИХ-фильтров. При выводе импульсной и переходной характеристик необходимо помнить о нормирующем множителе $GK=0.6197$. ■

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.