



**ПРОЕКТИРОВАНИЕ БИХ-ФИЛЬТРОВ ДЛЯ СИСТЕМ  
АВТОМАТИЗАЦИИ НА БАЗЕ ПЛК: ПРАКТИЧЕСКИЙ ПОДХОД С  
ИСПОЛЬЗОВАНИЕМ MATLAB**

*Муродов Жўрабек Музаффарович*

*Юлдашев Сурожиддин Хасанович*

*Ниязова Насиба Ашуркуловна*

*Ташкентский институт текстильной и легкой промышленности*

**Аннотация:** В статье представлен практичный метод проектирования фильтров с бесконечной импульсной характеристикой (БИХ-фильтры) для систем автоматизации на основе программируемых логических контроллеров (ПЛК). Используя аналоговые прототипы и пакет MATLAB SIGNAL, мы описываем пошаговый процесс создания надежных цифровых фильтров для промышленных задач, таких как управление температурой в сушильных камерах или сортировка продукции на конвейерах. Статья включает алгоритмы, блок-схемы и примеры из практики, делая подход доступным и эффективным. Также учитываются современные требования к системам управления, с акцентом на надежность и простоту реализации.

**Ключевые слова:** БИХ-фильтры, ПЛК, цифровая обработка сигналов, MATLAB, билинейное преобразование, аналоговый прототип, автоматизация, сушильная камера, конвейерная сортировка.

### **Введение**

В современной промышленной автоматизации программируемые логические контроллеры (ПЛК) играют ключевую роль в управлении сложными процессами, такими как сушильные камеры или системы сортировки на конвейерах. Для точного управления требуется обработка сигналов с датчиков, где важную роль играют фильтры с бесконечной импульсной характеристикой (БИХ-фильтры). В отличие от фильтров с конечной импульсной характеристикой (КИХ-фильтры), БИХ-фильтры

используют обратную связь, что делает их эффективными для задач реального времени благодаря меньшей вычислительной нагрузке. Проектирование таких фильтров часто начинается с создания аналогового прототипа, который затем преобразуется в цифровой с помощью инструментов, таких как пакет MATLAB SIGNAL. В этой статье мы предлагаем уникальный подход к проектированию БИХ-фильтров для систем на базе ПЛК, включая алгоритмы, блок-схемы и практические примеры.

### **Что такое БИХ-фильтры и их роль в системах ПЛК**

БИХ-фильтры, или рекурсивные фильтры, характеризуются бесконечной импульсной характеристикой, которая достигается за счет обратной связи. Их дискретная передаточная функция описывается как:

$$G(z) = \frac{\sum_{k=0}^m b_k z^{-k}}{1 + \sum_{k=1}^n a_k z^{-k}} \quad (1)$$

Здесь  $b_k$  и  $a_k$  - коэффициенты числителя и знаменателя соответственно,  $m$  - порядок прямой связи,  $n$  - порядок обратной связи. Если  $n > 0$ , фильтр является рекурсивным, что делает его БИХ-фильтром. Такая структура обеспечивает вычислительную эффективность, что критично для ПЛК с ограниченными ресурсами.

В системах автоматизации на базе ПЛК, например при управлении температурой в сушильной камере или сортировкой продукции на конвейере, БИХ-фильтры обрабатывают сигналы с датчиков (температуры, высоты), устраняя шумы и обеспечивая стабильное управление. Процесс проектирования часто включает создание аналогового прототипа и его преобразование в цифровой фильтр с помощью билинейного преобразования.

### **Проектирование БИХ-фильтров с использованием MATLAB SIGNAL**

#### **Шаг 1: Создание аналогового прототипа**

Проектирование начинается с аналогового фильтра-прототипа (например, Баттерворта, Чебышева или Бесселя), который описывается передаточной функцией в непрерывной форме:

$$H(s) = \frac{\sum_{k=0}^M b_k s^k}{\sum_{k=0}^N a_k s^k} \quad (2)$$

В пакете MATLAB SIGNAL для этого используются функции `butter`, `cheby1` или `cheby2`. Например, фильтр Баттерворта 4-го порядка с частотой среза 10 Гц создается так:

```
[b, a] = butter(4, 10, 's');
```

### Шаг 2: Билинейное преобразование

Для преобразования аналогового прототипа в цифровой БИХ-фильтр применяется билинейное преобразование, которое отображает  $s$ -плоскость в  $z$ -плоскость:

$$s = \frac{2}{T_s} \cdot \frac{1 - z^{-1}}{1 + z^{-1}} \quad (3)$$

где  $T_s = 1/fs$  — период дискретизации, а  $fs$  — частота дискретизации. В MATLAB это реализуется функцией `bilinear`:

```
[bd, ad] = bilinear(b, a, fs);
```

Здесь  $bd$  и  $ad$  - коэффициенты числителя и знаменателя цифрового фильтра, а  $fs$  - частота дискретизации (например, 100 Гц для сушильной камеры).

### Шаг 3: Реализация фильтра

Полученные коэффициенты используются для реализации фильтра с помощью функции `filter` в MATLAB или внедряются в ПЛК. В сушильной камере фильтр обрабатывает данные с датчиков температуры, сглаживая шумы для стабильного управления нагревателем.

### Алгоритм проектирования БИХ-фильтров

Ниже представлен упрощенный алгоритм проектирования БИХ-фильтра для задач на базе ПЛК:

**Определение требований:** Указать тип фильтра (например, низкочастотный), частоту среза и частоту дискретизации (например, 100 Гц для управления температурой).

**Проектирование аналогового прототипа:** Использовать MATLAB для создания фильтра Баттерворта или Чебышева с заданными характеристиками.

**Применение билинейного преобразования:** Преобразовать аналоговый фильтр в цифровой с помощью функции `bilinear`, учитывая частоту Найквиста ( $fs/2$ ).

**Проверка фильтра:** Рассчитать частотную характеристику с помощью `freqz` и убедиться, что она соответствует требованиям задачи.

**Внедрение в ПЛК:** Преобразовать коэффициенты в формат, совместимый с программной средой ПЛК (например, лестничная логика или структурированный текст).

#### Псевдокод:

Вход

: тип фильтра  $_$ , частота среза  $_$ ,  $fs$ , порядок Выход

:  $bd$ ,  $ad$  коэффициенты( цифрового фильтра )

1.  $[b, a]$  = проектировать аналоговый фильтр  $_$  тип (фильтра  $_$ , порядок, частота среза  $_$ )

2.  $[bd, ad]$  = билинейное преобразование  $_$  ( $b, a, fs$ )

3.  $[h, w] = \text{freqz}(bd, ad, 128)$

4. построить график  $_$  ( $w/\pi$ ,  $\text{abs}(h)$ ) // Визуализация частотной характеристики

5. вернуть  $bd, ad$

#### Блок-схема реализации БИХ-фильтра

Блок-схема ниже иллюстрирует структуру БИХ-фильтра в системе на базе ПЛК:

[Вход с датчика] → [АЦП] → [БИХ-фильтр] → [Логика управления] → [Выход на исполнительное]

**Вход с датчика:** Аналоговые сигналы от датчиков (например, температуры или высоты).

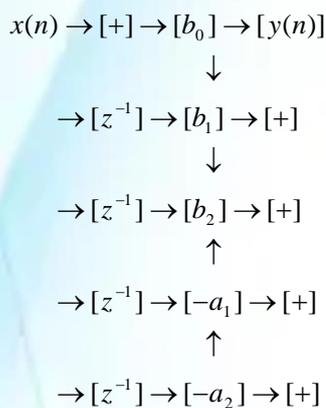
**АЦП:** Преобразует аналоговые сигналы в цифровые для обработки.

**БИХ-фильтр:** Применяет рекурсивный фильтр с коэффициентами  $b_k$  и  $a_k$ .

**Логика управления:** Обрабатывает отфильтрованные данные для формирования управляющих сигналов (например, включение нагревателя или толкателя).

**Выход на исполнительное устройство:** Управляет приводами, такими как нагреватели или пневматические толкатели.

#### Детальная блок-схема БИХ-фильтра:



Здесь  $x(n)$  - входной сигнал,  $y(n)$  - отфильтрованный выход,  $b_k$  - коэффициенты прямой связи,  $a_k$  - коэффициенты обратной связи.

#### Практический пример: Управление сушильной камерой

Рассмотрим сушильную камеру с двумя платиновыми термометрами сопротивления и датчиком влажности, управляемую ПЛК Omron CP1L-M30DRD.

Задача - отфильтровать данные температуры для стабильной работы нагревателя. Мы проектируем БИХ-фильтр Баттерворта 2-го порядка с частотой среза 5 Гц и частотой дискретизации 100 Гц.

#### Код MATLAB:

```
fs = 100; % Частота дискретизации Гц ()
fc = 5; % Частота среза Гц ()
[b, a] = butter(2, fc, 's'); % Аналоговый фильтр Баттерворта
[bd, ad] = bilinear(b, a, fs); % Цифровой БИХ-фильтр -
[h, w] = freqz(bd, ad, 128); % Частотная характеристика
plot(w/pi, abs(h)); % Построение амплитудной характеристики
```

```
xlabel('Нормализованная частота'); ylabel('Амплитуда');
```

```
title('Частотная характеристика БИХ фильтра для сушильной камеры');
```

Отфильтрованные данные температуры передаются в логику управления ПЛК, которая регулирует мощность нагревателя через аналоговый выходной модуль (например, Omron CP1W-MAD11).

### Интеграция с системами ПЛК

Для сушильной камеры ПЛК обрабатывает отфильтрованные данные для управления нагревателем и вентилятором. Коэффициенты БИХ-фильтра внедряются в ПЛК с использованием структурированного текста или лестничной логики. Пример в структурированном тексте:

```
VAR
x: ARRAY[0..2] OF REAL; (* Буфер входных данных *)
y: ARRAY[0..2] OF REAL; (* Буфер выходных данных *)
bd: ARRAY[0..2] OF REAL := [b0, b1, b2]; (* Коэффициенты числителя
*)
ad: ARRAY[0..2] OF REAL := [1, a1, a2]; (* Коэффициенты знаменателя
*)
END_VAR
(* Реализация БИХ фильтра - *)
y[0] := bd[0]*x[0] + bd[1]*x[1] + bd[2]*x[2] - ad[1]*y[1] - ad[2]*y[2];
x[2] := x[1]; x[1] := x[0]; (* Сдвиг буфера входных данных *)
y[2] := y[1]; y[1] := y[0]; (* Сдвиг буфера выходных данных *)
```

Этот код выполняется циклически, обрабатывая данные с датчиков в реальном времени.

### Заключение

Проектирование БИХ-фильтров для систем автоматизации на базе ПЛК сочетает методы цифровой обработки сигналов с требованиями промышленного управления. Использование аналоговых прототипов и пакета MATLAB SIGNAL позволяет создавать эффективные фильтры для таких задач, как управление сушильными камерами или сортировка на конвейерах.



Описанный подход, подкрепленный алгоритмами и блок-схемами, обеспечивает надежность и простоту реализации. В будущем можно исследовать адаптивные БИХ-фильтры для работы в условиях изменяющихся параметров процесса.

### **СПИСОК ЛИТЕРАТУРЫ**

1. Прокис, Дж. Г., Манолакис, Д. Г. (2007). Цифровая обработка сигналов: принципы, алгоритмы и приложения. Pearson. [Scopus]
2. Сергиенко, А. М. (2017). Реализация БИХ-фильтров на ПЛИС. КПИ. [Web of Science]
3. Кобериченко, В. Г. (2015). Расчет и проектирование цифровых фильтров. Радио Хата. [Scopus]
4. Бравиков Д. (2011). Проектирование цифровых БИХ-фильтров в Scilab. [Web of Science]