Алгоритмы обработки данных с МЭМС-датчиков в ИС

Алгоритмы фильтрации сигналов

Соловьев А. В.

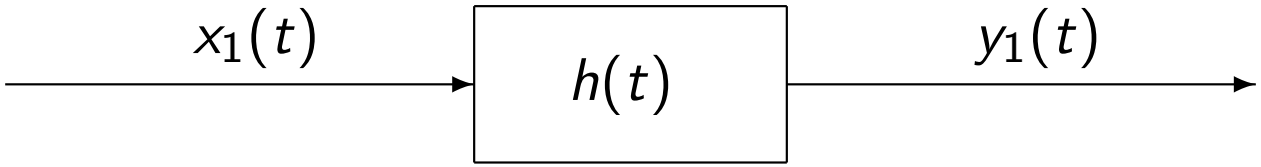
ПетрГУ – КИИСиФЭ

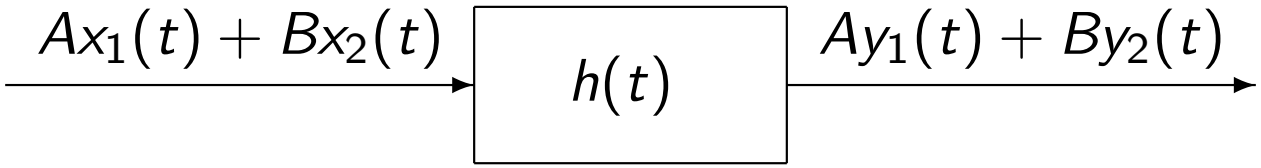
(Rev. 2016 07 09)

Принципы фильтрации сигналов

Линейная система (ЛС)

Отклик системы на сумму воздействий равен сумме откликов на каждое воздействие.





Соотношение входа и выхода ЛС может быть описано ЛДУ:

$$y(t) = \sum_{i=0}^{P} b_i \frac{d^i x(t)}{dt^i} - \sum_{k=1}^{Q} a_k \frac{d^k y(t)}{dt^k}, \qquad (1)$$

коэфф-ты b_i и a_k зависят от характеристик системы (R, L, C).

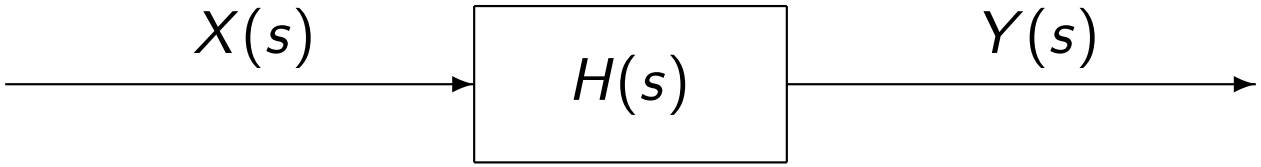
Импульсная характеристика

Импульсная переходная функция (impulse response), или импульсная характеристика (ИХ) — отклик системы на дельта-функцию Дирака.
Если рассматривать ЛС в пространстве времени, сигнал на выходе ЛС $y(t)$ можно рассчитать как свёртку входного сигнала $x(t)$ с импульсной характеристикой $h(t)$:

$$y(t) = \int_{-\infty}^{+\infty} h(\tau)x(t-\tau)d\tau, \quad \text{или} \quad y(t) = h(t) \otimes x(t). \qquad (2)$$

Передаточная функция

В пространстве комплексных частот s ЛС описывается передаточной функцией (transfer function).



Передаточная функция (ПФ) — результат преобразования Лапласа импульсной характеристики ЛС. Выходной сигнал $Y(s)$ является произведением входного сигнала $X(s)$ и передаточной функции $H(s)$:

$$Y(s) = H(s) \cdot X(s), \tag{3}$$

где $X(s) = \mathfrak{L}\{x(t)\}$, $Y(s) = \mathfrak{L}\{y(t)\}$, $H(s) = \mathfrak{L}\{h(t)\}$

АЧХ/ФЧХ. Связь ИХ и ПФ

Для описания ЛС в частотной области (вещественные частоты) рассматривают модуль (АЧХ) и аргумент (ФЧХ) ПФ:

$$\text{АЧХ}(w) = |H(iw)|, \quad \text{ФЧХ}(w) = \operatorname{arctg} \frac{\mathfrak{Im}\{H(iw)\}}{\mathfrak{Re}\{H(iw)\}}. \tag{4}$$

$H(iw)$ – это фурье-образ ИХ, поскольку преобразование Лапласа и преобразование Фурье связаны:

$$\mathfrak{F}\{h(t)\} = \mathfrak{L}\{h(t)\}|_{s=iw}. \tag{5}$$

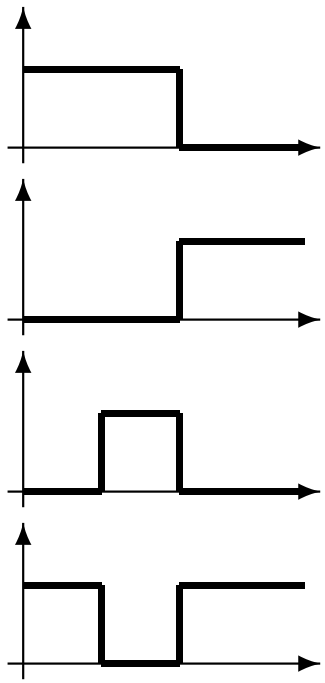
Зная АЧХ/ФЧХ или ПФ, выходной сигнал ЛС можно рассчитать, используя теорему о свёртке:

$$\begin{aligned} y(t) &= \mathfrak{F}^{-1}\{\mathfrak{F}\{h(t)\} \cdot \mathfrak{F}\{x(t)\}\} \text{ или} \\ y(t) &= \mathfrak{L}^{-1}\{\mathfrak{L}\{h(t)\} \cdot \mathfrak{L}\{x(t)\}\}. \end{aligned} \tag{6}$$

Фильтры

Для преобразования сигналов часто используются ЛС, называемые фильтрами, АЧХ которых имеет определённую форму:

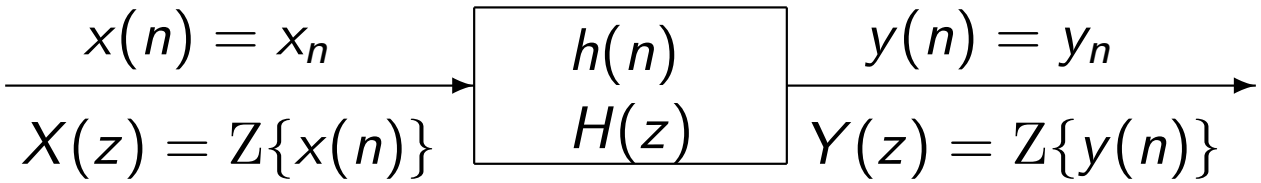
- Фильтр нижних частот – ФНЧ (low-pass filter – LPF).
- Фильтр верхних частот – ФВЧ (high-pass filter – HPF).
- Полосно-пропускающий фильтр – ППФ (band-pass filter – BPF).
- Полосно-заграждающий фильтр (режекторный) – ПЗФ (band-stop filter – BSF).



Линейные дискретные системы (ЛДС)

Дискретная модель ЛС

Применение аналоговых преобразований к дискретным сигналам вычислительно неэффективно. Для обработки дискретных (цифровых) сигналов вводится понятие ЛДС:



ИХ ЛДС h(n) – реакция на единичный скачок.
Соотношение входа и выхода ЛДС описывается формулой дискретной свёртки:

$$y(n) = \sum_{m=0}^{\infty} h(m)x(n-m) \tag{7}$$

Расчёт реакции ЛДС в Python по формуле свёртки (известны отсчёты ИХ, ИХ – конечная):

```
y = scipy.signal.convolve(x, h)
```

Дискретная передаточная функция

ЛДС можно описать в пространстве z-частот при помощи дискретной ПФ $H(z)$ – результат Z-преобразования ИХ ЛДС.

$$H(z) = \mathrm{Z}\{h(n)\} = \sum_{n=0}^{\infty} h(n) z^{-n} \tag{8}$$

Плоскости z и s связаны: $z = e^{s\Delta t}$. Z-преобразование отображает точки iw-оси ($\mathfrak{Re}(s) = 0$) на s-плоскости в окружность единичного радиуса ($|z| = 1$) на z-плоскости.
Дискретная ПФ фильтра может быть записана в виде:

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{i=0}^{P} b_i z^{-i}}{1 + \sum_{k=1}^{Q} a_k z^{-k}}. \tag{9}$$

АЧХ/ФЧХ дискретного фильтра определена для нормированных частот $-\pi < \omega < \pi$, причём π соответствует частоте Найквиста.

Разностное уравнение

Для расчёта реакции ЛДС можно использовать разностное уравнение:

$$y_n = \sum_{i=0}^{P} b_i x_{n-i} - \sum_{k=1}^{Q} a_k y_{n-k}, \tag{10}$$

в котором b_i и a_i – коэффициенты полиномов числителя и знаменателя ПФ, P – порядок входного сигнала, а Q – порядок обратной связи. Обычно $P = Q = N$ и соответствуют порядку фильтра.

Расчёт реакции ЛДС в Python по разностному уравнению (известны коэффициенты ПФ):

```
y = scipy.signal.lfilter(b, a, x)
```

Рекурсивные и нерекурсивные фильтры

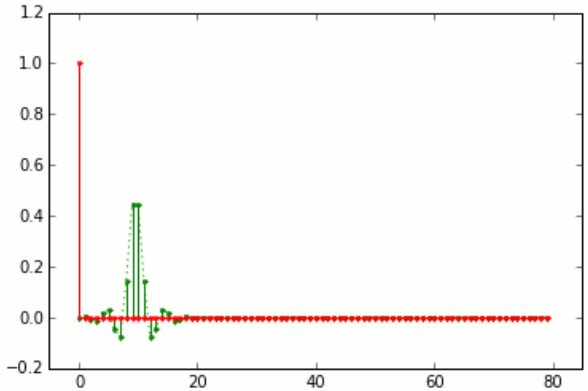
По особенностям разностного уравнения (или ИХ) фильтры можно поделить на два класса:

- Рекурсивные фильтры – фильтры, у которых y_n определяется по x_n и предшествующими значениями входной и выходной последовательностей. Как правило, рекурсивные фильтры имеют бесконечную ИХ $h(n)$, поэтому их ещё называют БИХ-фильтрами.
- Нерекурсивные фильтры – фильтры, у которых y_n определяется текущим и предшествующими значениями только входной последовательности. Нерекурсивные фильтры всегда имеют конечную ИХ $h(n)$, поэтому их ещё называют КИХ-фильтрами.

Достоинства и недостатки КИХ- и БИХ-фильтров

КИХ

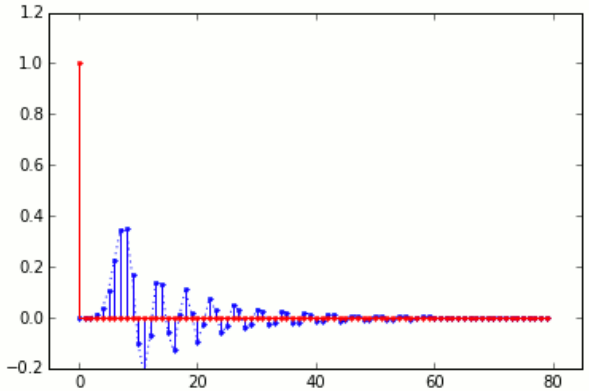
FIR – finite impulse response
(Нерекурсивные фильтры)



- Ошибки не накапливаются. Всегда устойчивы.
- Линейная ФЧХ.
- Порядок больше.

БИХ

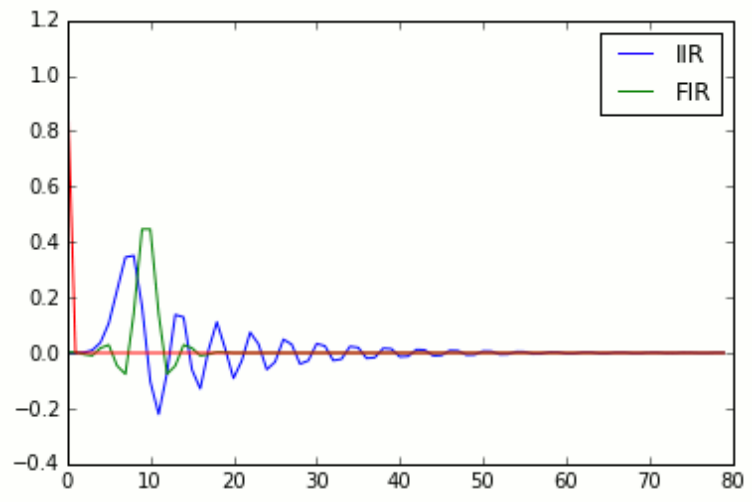
IIR – infinite impulse response
(Рекурсивные фильтры)



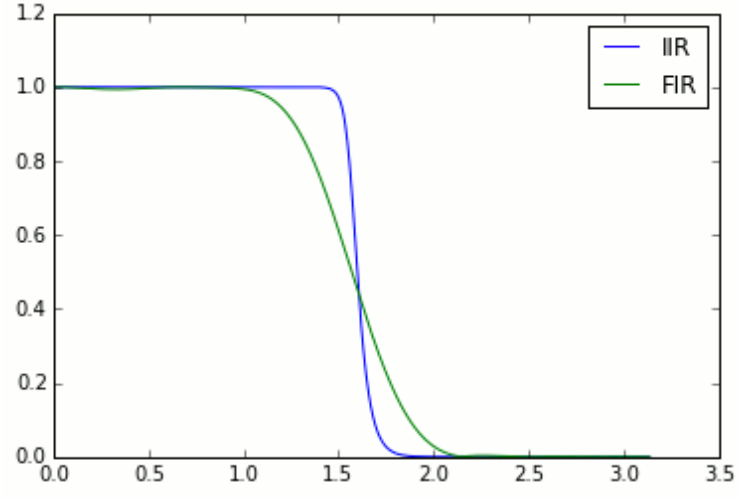
- Требуется проверка на устойчивость.
- Нелинейная ФЧХ.
- Порядок меньше.

КИХ- и БИХ-фильтры (пример)

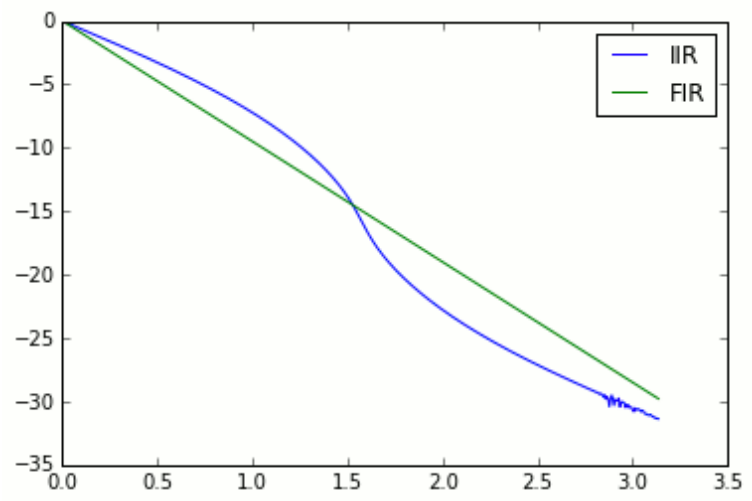
Импульсная характеристика:



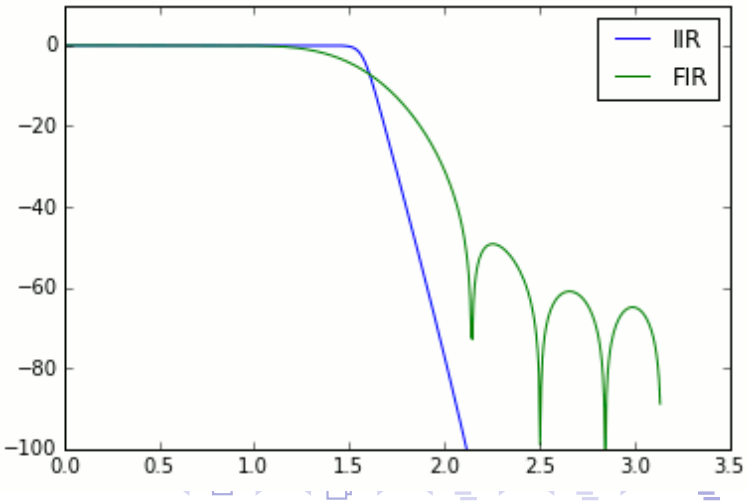
АЧХ:



ФЧХ:



АЧХ (логарифм. масштаб):



Каузальные и некаузальные фильтры

- Каузальный (causal), или «причинный» фильтр – физически реализуемый фильтр, у которого реакция y_n в данный момент времени n НЕ ЗАВИСИТ от значений входного воздействия x_i в последующие моменты $i > n$.
- Некаузальный фильтр – это фильтр, у которого реакция y_n в данный момент времени n зависит от значений входного воздействия x_i в последующие моменты $i > n$. Такой фильтр нельзя физически реализовать в режиме реального времени.

Некаузальные фильтры можно использовать на практике в тех случаях, когда процедура фильтрации происходит не в реальном времени, а выполняется над хранящимися в памяти последовательностями конечной длины.

Некаузальные фильтры

Медианный фильтр (1)

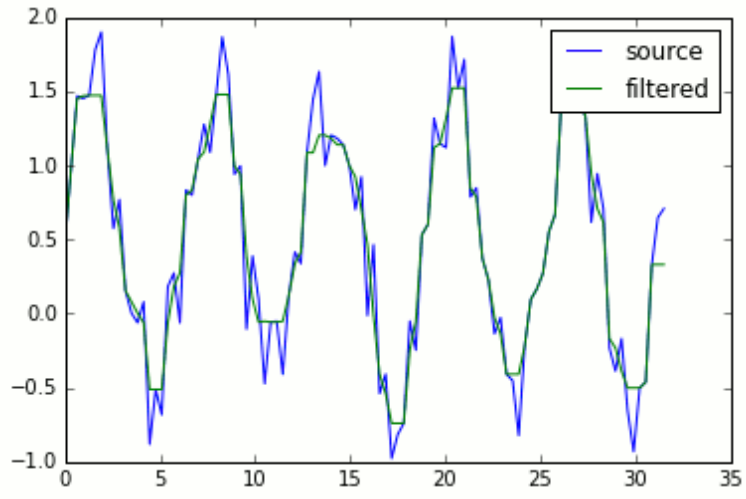
Примером некаузального фильтра является медианный фильтр. Значения отсчётов внутри окна фильтра сортируются в порядке возрастания. На выход фильтра поступает значение, находящееся в середине упорядоченного списка.
Пример фильтра с окном 3:

```
     [0.18, 0.06, 0.49, 0.46, 0.93, 0.87, 0.32, 0.52]
0.18, 0.18, 0.06 -> 0.18
      0.18, 0.06, 0.49 -> 0.18
            0.06, 0.49, 0.46 -> 0.46
                  0.49, 0.46, 0.93 -> 0.49
                        0.46, 0.93, 0.87 -> 0.87
                              0.93, 0.87, 0.32 -> 0.87
                                    0.87, 0.32, 0.52 -> 0.52
                                          0.32, 0.52, 0.52 -> 0.52
     [0.18, 0.18, 0.46, 0.49, 0.87, 0.87, 0.52, 0.52]
```

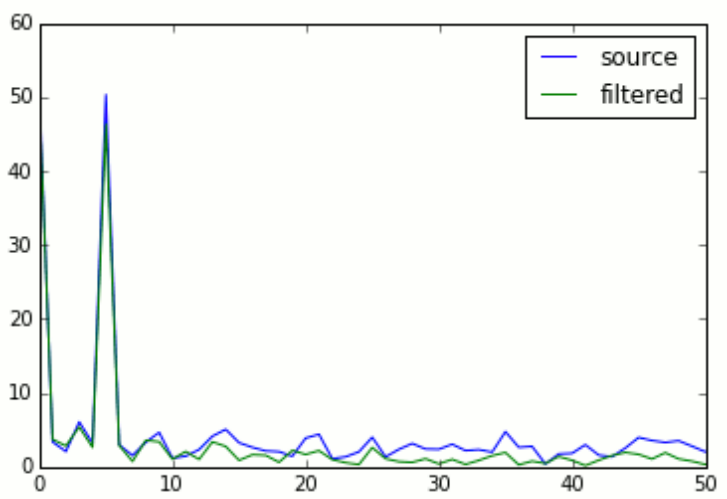
Медианный фильтр (2)

В Python медианный фильтр применяется методом:
`y=scipy.signal.medfilt(x,kernel_size=5)`

Сигнал:



Спектр:

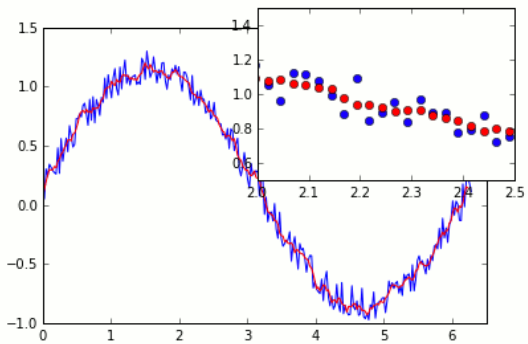


Размер окна должен быть нечётным.

Скользящее среднее

Ещё один пример некаузального фильтра – фильтр скользящего среднего (moving average):

$$y_n = \frac{1}{N} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} x_{n-i} \tag{11}$$



Пример скользящего среднего по пяти точкам:

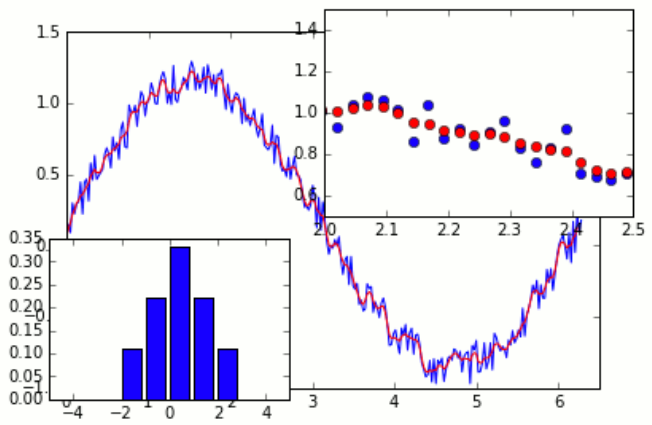
$$y_n = \frac{1}{5}x_{n-2} + \frac{1}{5}x_{n-1} + \frac{1}{5}x_n + \frac{1}{5}x_{n+1} + \frac{1}{5}x_{n+2}$$

Взвешенное скользящее среднее

Иногда, при построении скользящего среднего, некоторые значения исходной функции целесообразно сделать более значимыми.

Взвешенное скользящее среднее:

$$y_n = \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} w_i x_{n-i}, \quad \sum w_i = 1 \tag{12}$$



Взвешенное скользящее среднее по пяти точкам с линейной весовой функцией:

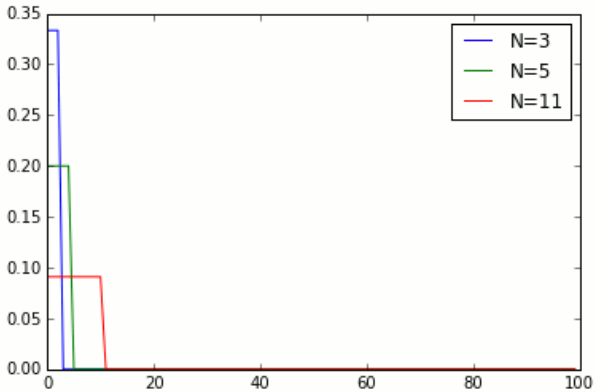
$$y_n = \frac{1}{9}x_{n-2} + \frac{2}{9}x_{n-1} + \frac{3}{9}x_n + \frac{2}{9}x_{n+1} + \frac{1}{9}x_{n+2}$$

Каузальный фильтр скользящего среднего

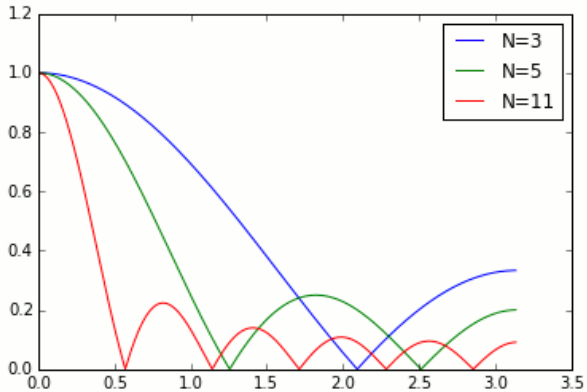
Полученное значение скользящего среднего относится к середине выбранного интервала. Если относить значение к последней точке интервала, фильтр можно считать каузальным:

$$y_n = \frac{1}{N}\sum_{i=0}^{N-1} x_{n-i} \quad \Rightarrow \quad h(n) = \begin{cases} \frac{1}{N}, n=0\ldots N-1 \\ 0 \end{cases}, H(z) = \sum_{i=0}^{N-1} \frac{1}{N} z^{-i}$$

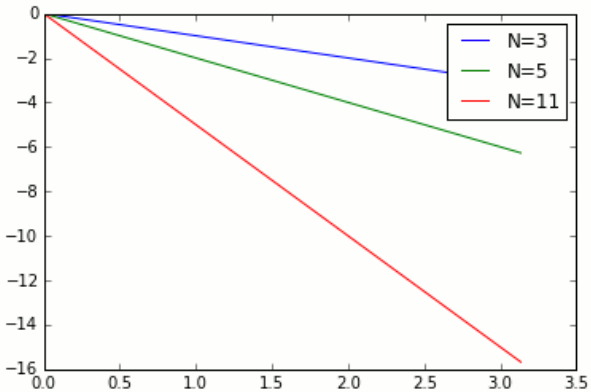
ИХ:



АЧХ:



ФЧХ:

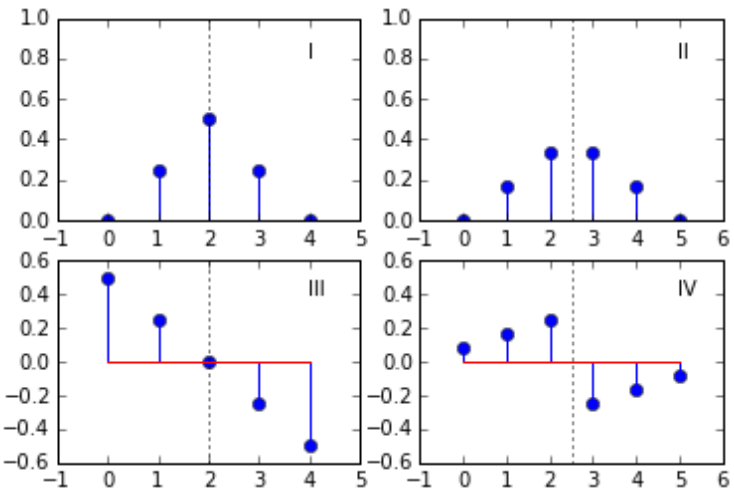


Увеличение числа точек при реализации фильтра сужает основной лепесток, но существенно не уменьшает амплитуду боковых лепестков АЧХ. Фильтры не подходят в том случае, где требуется большое ослабление в полосе задержания.

КИХ-фильтры

КИХ-фильтры с линейной ФЧХ

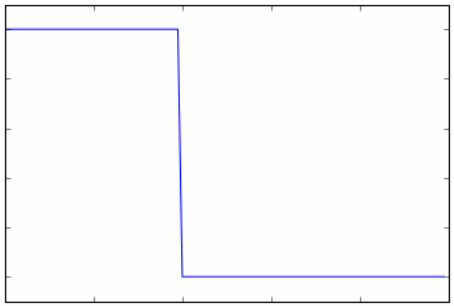
КИХ-фильтр длины N обладает порядком $R = N - 1$.
КИХ-фильтр имеет линейную ФЧХ, если он симметричен или антисимметричен относительно точки $n = \frac{R}{2}$.



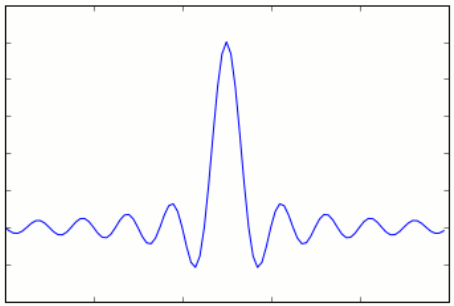
- I – R чётный, $h(n)$ симметричная: любая избирательность
- II – R нечётный, $h(n)$ симметричная: ФНЧ, ППФ
- III – R чётный, $h(n)$ антисимм.: ППФ
- IV – R нечётный, $h(n)$ антисимм.: ФВЧ, ППФ

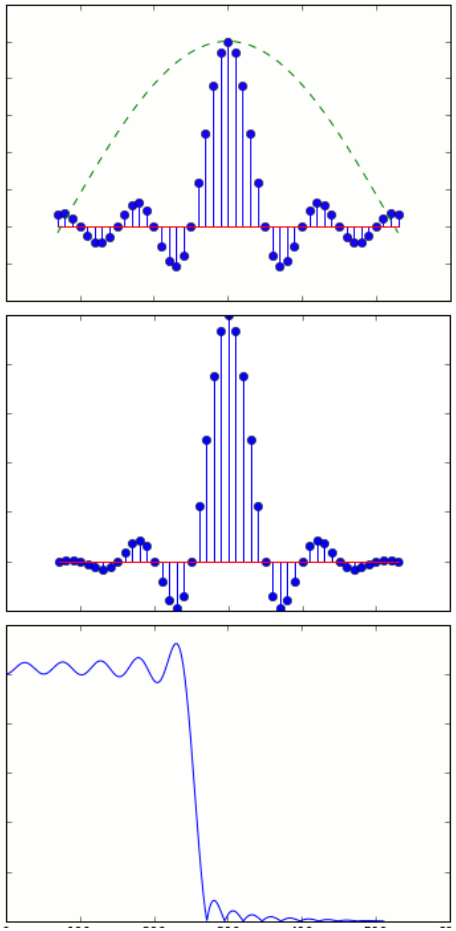
Синтез КИХ на основе идеального ФНЧ

АЧХ идеального ФНЧ



ИХ идеального ФНЧ





1. ИХ идеального ФНЧ усекается, накладывается оконная функция
2. Взвешенная ИХ соответствует КИХ-фильтру I типа
3. АЧХ полученного КИХ-фильтра

Синтез КИХ-фильтра в Python

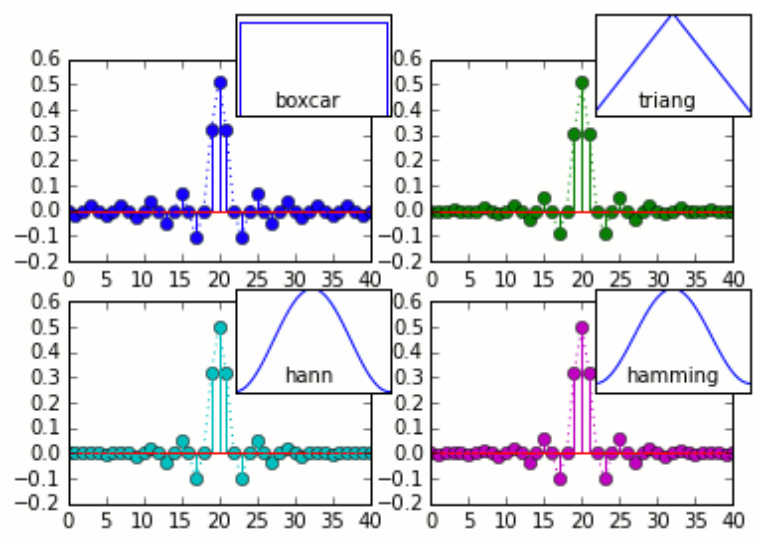
```
h = scipy.signal.firwin(numtaps, cutoff, window='hamming')
```

- *numtaps* – длина фильтра (нечётная – фильтр типа I, чётная – фильтр типа II);
- *cutoff* – частота среза (или список частот);
- *window* – оконная функция (задаётся строкой или кортежем параметров);
- *nyq* – частота Найквиста ($^1/_2$ частоты дискретизации), по умолчанию – 1.0 (остальные частоты задаются в долях).

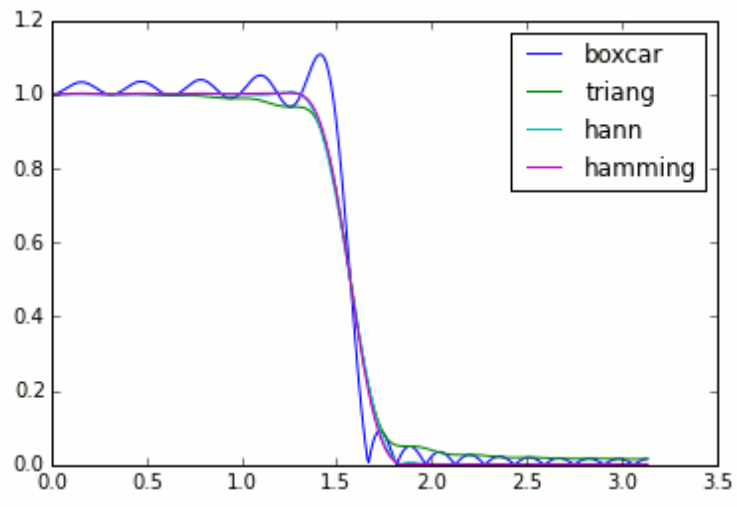
Типы поддерживаемых окон: boxcar, triang, blackman, hamming, hann, bartlett, flattop, parzen, bohman, blackmanharris, nuttall, barthann, kaiser*, gaussian*, general_gaussian*, slepian*, chebwin*
Возвращается *h* – массив отсчётов ИХ.

Синтез КИХ-фильтра в Python (пример)

ИХ:



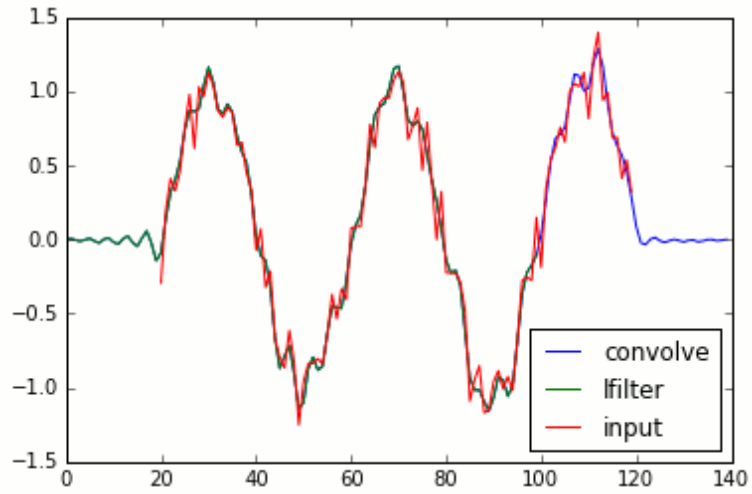
АЧХ:



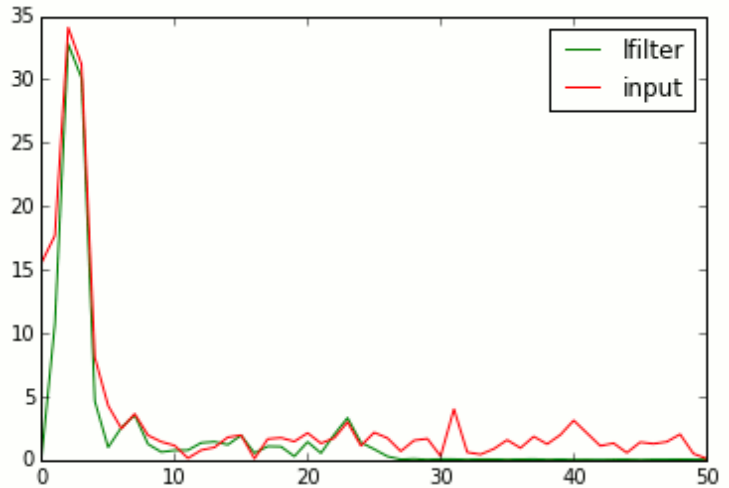

```
b1 = signal.firwin(41, 0.5, window='boxcar')
n1 = signal.get_window('boxcar',41)
w1, h1 = signal.freqz(b1)
```

Применение КИХ-фильтра в Python

Сигнал:



Спектр сигнала:



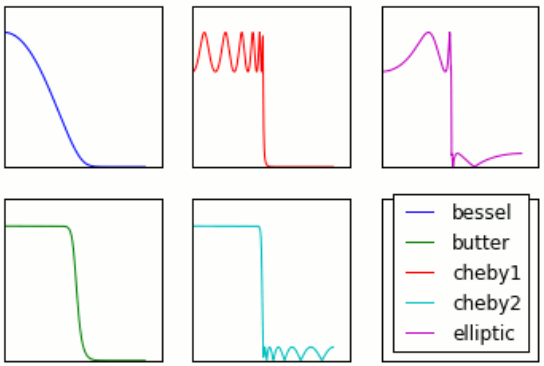

```
z1 = signal.convolve(b1,z)
z2 = signal.lfilter(b1,[1],z)
```

```
# даёт len(b1)+len(z) отсчётов
# даёт len(z) отсчётов
```

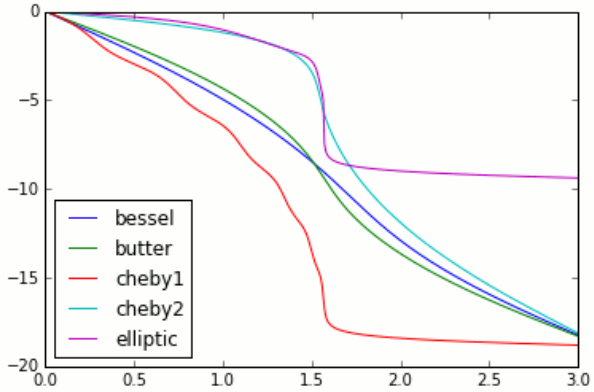
БИХ-фильтры

Синтез БИХ-фильтров на основе аналогового прототипа

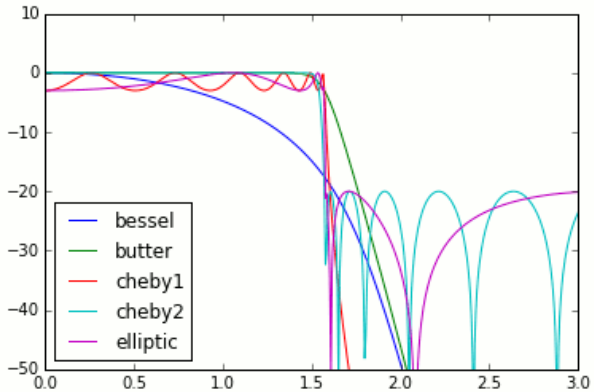
АЧХ:



ФЧХ:



АЧХ (лог. шкала):



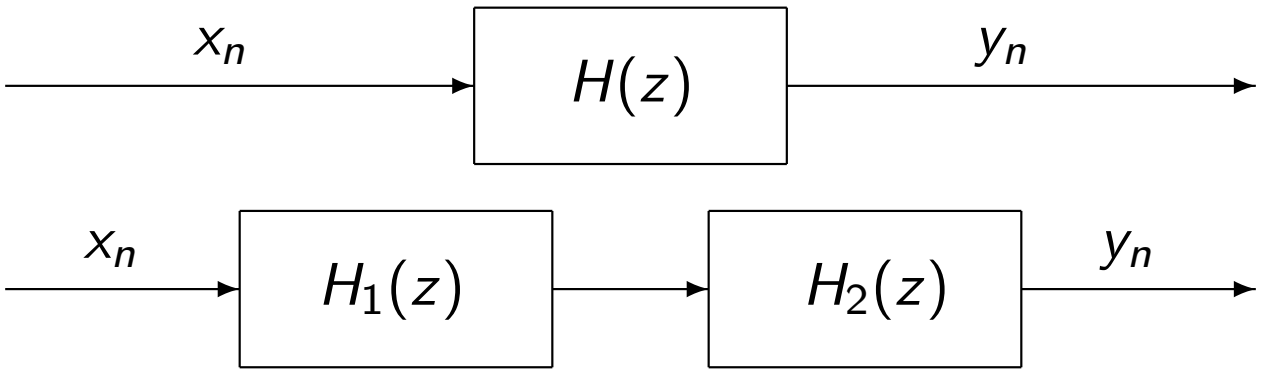

```
b,a=signal.iirfilter(N,Wn,rp=None,rs=None,btype='lowpass',ftype=...)
signal.iirfilter(12,0.5,rp=3,rs=20,btype='lowpass',ftype='elliptic')
```

- *N* – порядок фильтра;
- *Wn* – критическая частота (или массив) в долях *nyq*;
- *btype* – избирательность ('lowpass', 'hipass', 'bandpass', 'bandstop');
- *ftype* – фильтр-прототип ('butter', 'cheby1', 'cheby2', 'elliptic', 'bessel');
- *rp* – амплитуда пульсаций в полосе пропускания (дБ) – для Ч1 и Э;
- *rs* – макс. доп. усиление в полосе задержки (-дБ) – для Ч2 и Э;
- *b* и *a* – коэффициенты числителя и знаменателя ПФ.

Каскадная схема фильтров

Возможна реализация фильтра высокого порядка в виде каскада фильтров более низкого порядка. В случае каскада из двух фильтров входной сигнал сначала поступает на вход 1-го фильтра, а выход 1-го фильтра подаётся на вход 2-го фильтра. Аналитически это соответствует разложению числителя и знаменателя ПФ на множители:

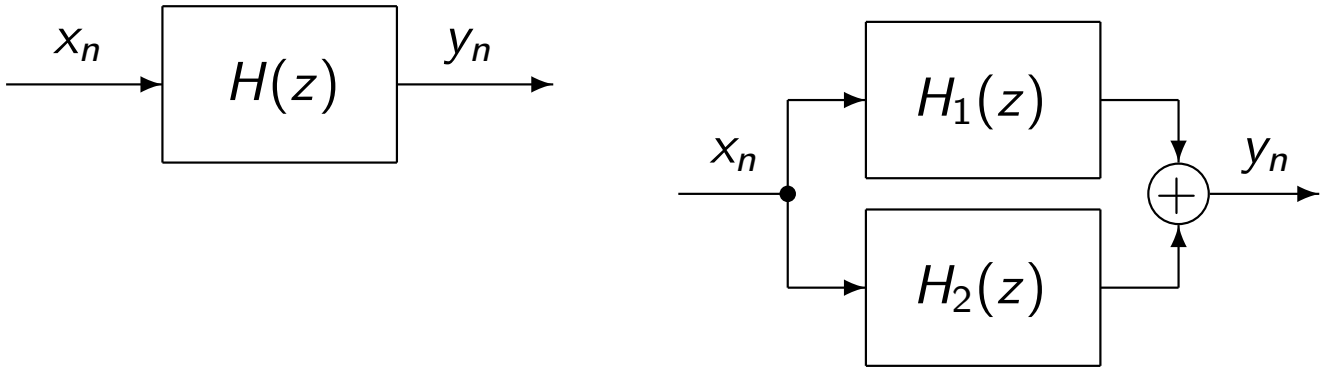
$$H(z) = \frac{B(z)}{A(z)} = \frac{B_1(z)}{A_1(z)} \cdot \frac{B_2(z)}{A_2(z)} = H_1(z) \cdot H_2(z) \qquad (13)$$



Параллельная схема фильтров

Возможна реализация фильтра высокого порядка в виде суммы фильтров более низкого порядка. Результатом является сумма выходных сигналов параллельных фильтров. Аналитически это соответствует разложению ПФ на простые дроби:

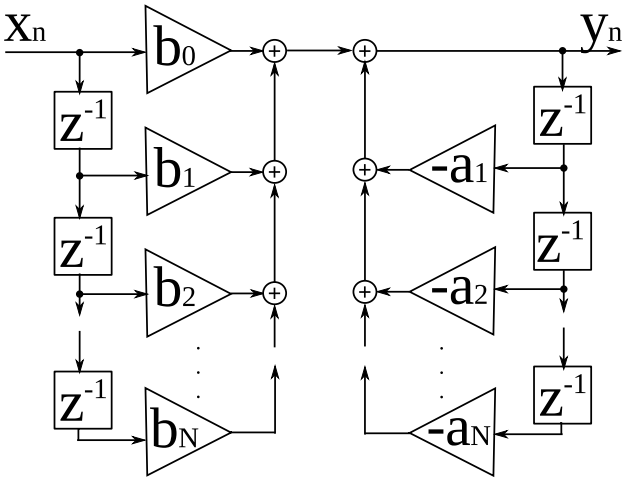
$$H(z) = \frac{B(z)}{A(z)} = \frac{B_1(z)}{A_1(z)} + \frac{B_2(z)}{A_2(z)} = H_1(z) + H_2(z) \qquad (14)$$



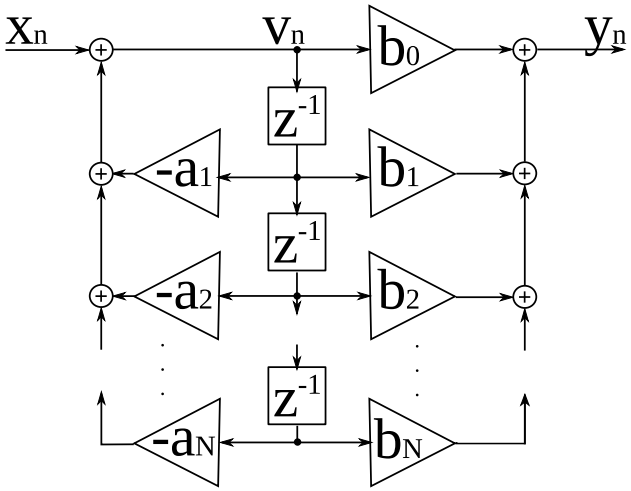
В Python для выполнения разложения на дроби используются функции: `r,p,k=scipy.signal.residue(b,a)` или `r,p,k=scipy.signal.residuez(b,a)`.

Формы реализации фильтров

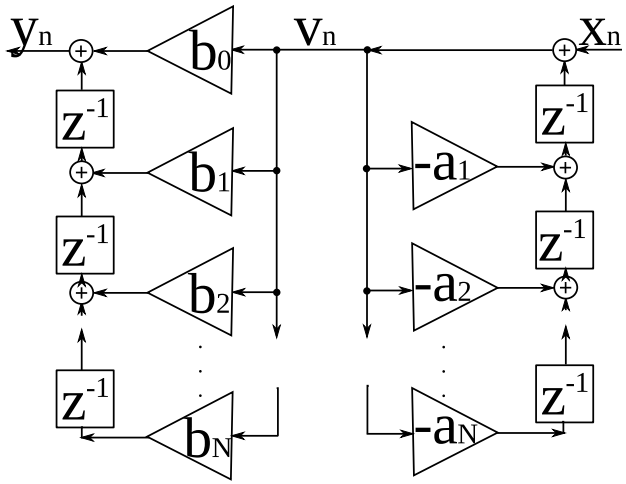
Прямая I



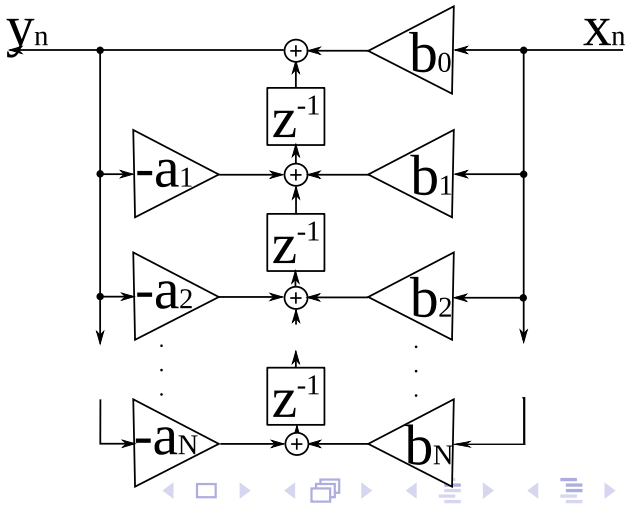
Прямая II (каноническая)



Прямая транспонированная



Каноническая транспонированная



Принципы адаптивной фильтрации

Принципы адаптивной фильтрации

Традиционные КИХ- и БИХ-фильтры можно использовать только для эргодического дискретного сигнала (у которого Mx и Dx не зависят от времени). При этом полосы полезного сигнала и шума должны быть заранее известны и разнесены.

Если стат. характеристики сигнала неизвестны, используют систему, параметры которой адаптируются (подстраиваются) к входному сигналу в процессе работы.

Другой подход к этой проблеме – применение методов теории оценивания.