

Лекция №5: Введение в теорию цифровых фильтров

1. Что такое фильтр

Фильтр – это система (устройство или алгоритм), которая изменяет спектральный состав сигнала, пропуская одни частотные компоненты и задерживая (подавляя) другие.

Основные задачи фильтрации:

- Выделение полезного сигнала на фоне шумов и помех.
- Подавление нежелательных частотных составляющих (например, сетевой наводки).
- Формирование сигналов с заданными спектральными свойствами (эквалайзеры, синтезаторы звука).
- Сглаживание данных (удаление высокочастотных флуктуаций).
- Обнаружение границ или перепадов (например, в обработке изображений).

Примеры из повседневной жизни:

- Регулировка тембра в музыкальном проигрывателе (бас/высокие частоты).
- Размытие фотографий (фильтр нижних частот).
- Выделение контуров на изображении (фильтр верхних частот).

2. Основные характеристики фильтров

1. Амплитудно-частотная характеристика (АЧХ)

Показывает, как коэффициент передачи (отношение амплитуды выходного сигнала к амплитуде входного) зависит от частоты.

$$|H(f)| = \frac{|Y(f)|}{|X(f)|}$$

- АЧХ – вещественная, неотрицательная функция.
- Часто выражается в децибелах: $20 \log_{10} |H(f)|$.

2. Фазо-частотная характеристика (ФЧХ)

Показывает фазовый сдвиг, вносимый фильтром на каждой частоте.

$$\phi(f) = \arg(H(f)) = \arctan \frac{\Im(H(f))}{\Re(H(f))}$$

- Для сохранения формы сигнала желательна **линейная ФЧХ** (постоянное групповое время задержки).

3. Импульсная характеристика $h[n]$ – реакция фильтра на единичный импульс $\delta[n]$

- Для дискретных систем выходной сигнал вычисляется как **свёртка**:

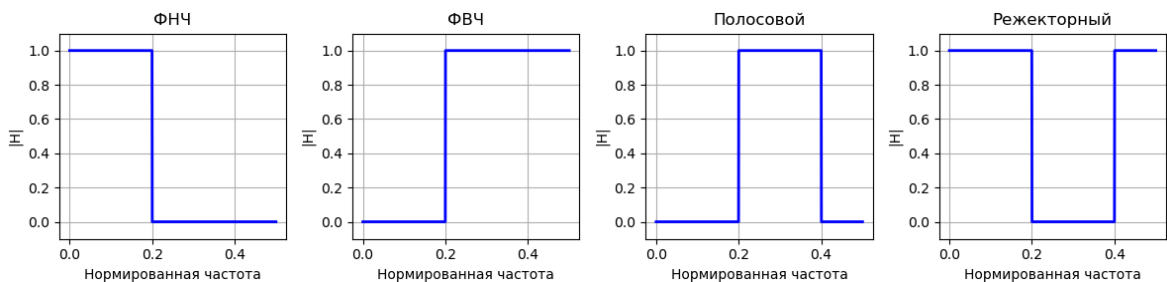
$$y[n] = (x * h)[n] = \sum_{k=-\infty}^{\infty} x[k] h[n - k]$$

4. Переходная характеристика – реакция на единичный скачок (функцию Хевисайда). Показывает, как фильтр реагирует на резкое изменение сигнала.

3. Классификация фильтров по типу АЧХ

В зависимости от того, какие частоты пропускаются, различают четыре основных типа фильтров.

- **Фильтр нижних частот (ФНЧ)** – пропускает частоты ниже частоты среза f_c , подавляет высокие.
- **Фильтр верхних частот (ФВЧ)** – пропускает частоты выше f_c .
- **Полосовой фильтр** – пропускает частоты в диапазоне $[f_{c1}, f_{c2}]$.
- **Режекторный фильтр (заградительный)** – подавляет узкую полосу частот.



Идеальные фильтры (прямоугольная АЧХ) физически нереализуемы. Реальные фильтры имеют переходную полосу и пульсации.

4. Аналоговые и цифровые фильтры: сравнение

Характеристика	Аналоговые фильтры	Цифровые фильтры
Сигнал	Непрерывный во времени	Дискретные отсчёты
Элементная база	RLC-цепи, операционные усилители	Алгоритмы на DSP, ПЛИС, микропроцессорах
Точность	Ограничена допусками компонентов, температурный дрейф	Высокая стабильность, не зависят от температуры
Перестраиваемость	Требует замены компонентов	Программное изменение параметров

Характеристика	Аналоговые фильтры	Цифровые фильтры
Фаза	Нелинейная (обычно)	Можно реализовать строго линейную фазу (КИХ)
Характеристика	Аналоговые фильтры	Цифровые фильтры
Сложность	Простые фильтры – дешево, высокие порядки – дорого	Вычислительные затраты, но легко масштабировать
Недостатки	Дрейф, старение, низкая точность	Необходимость АЦП/ЦАП, ограничение по частоте (теорема Найквиста), шум квантования

Вывод: цифровые фильтры предпочтительны в задачах, требующих высокой точности, перестройки и стабильности. Однако для очень высоких частот (гигагерцы) аналоговые фильтры остаются единственным решением.

5. Классификация цифровых фильтров по длине импульсной характеристики

5.1. Фильтры с конечной импульсной характеристикой (КИХ-фильтры, FIR – Finite Impulse Response)

Разностное уравнение:

$$y[n] = \sum_{k=0}^M b_k x[n - k]$$

- Импульсная характеристика имеет конечную длину $M + 1$ (коэффициенты b_k).
- Выход зависит только от прошлых входных отсчётов (нет обратной связи).
- **Всегда устойчивы** (конечная сумма ограниченных величин).
- Могут обладать **строго линейной фазой**, что важно для сохранения формы сигнала (например, в обработке ЭКГ, аудио).
- Недостаток: для достижения крутого среза требуется высокий порядок M (много коэффициентов).

Пример: скользящее среднее с $M = 2$: $y[n] = (x[n] + x[n - 1] + x[n - 2])/3$.

5.2. Фильтр с бесконечной импульсной характеристикой (БИХ-фильтры, IIR – Infinite Impulse Response)

Разностное уравнение:

$$y[n] = \sum_{k=0}^M b_k x[n - k] - \sum_{k=1}^N a_k y[n - k]$$

- Присутствует обратная связь по выходу (рекурсия).
- Импульсная характеристика теоретически бесконечна.
- **Могут быть неустойчивы** (неправильный выбор коэффициентов a_k).
- Более экономичны: для заданной крутизны среза требуют меньшего порядка, чем КИХ.
- Нелинейная ФЧХ (можно компенсировать, но сложнее).

Пример: экспоненциальное сглаживание: $y[n] = 0.1x[n] + 0.9y[n - 1]$.

6. Фазовая характеристика фильтра и групповая задержка

6.1. Почему фаза важна?

АЧХ показывает, какие частоты фильтр пропускает или подавляет. Но **форма сигнала** зависит не только от амплитуд гармоник, но и от их **фазовых соотношений**.

6.2. Линейная фаза и её свойства

Определение

Фильтр имеет **линейную фазу**, если его ФЧХ является линейной функцией частоты:

$$\phi(f) = -2\pi f\tau_0$$

где τ_0 — постоянная задержка (в секундах или отсчётах).

6.3. Групповая задержка — количественная мера искажений

Групповая задержка $\tau_g(f)$ — это скорость изменения фазы с частотой:

$$\tau_g(f) = -\frac{1}{2\pi} \frac{d\phi(f)}{df}$$

Физический смысл: групповая задержка показывает, на сколько задерживается **огibaющая** узкополосного сигнала на частоте f .

Связь с формой сигнала

- Если $\tau_g(f) = \text{const}$ для всех частот сигнала → **линейная фаза** → форма сигнала сохраняется.
- Если $\tau_g(f)$ меняется с частотой → разные частотные компоненты задерживаются по-разному → **фазовые искажения** → форма сигнала изменяется.

6.4. КИХ-фильтры с линейной фазой

Условие линейной фазы для КИХ

КИХ-фильтр имеет линейную фазу, если его импульсная характеристика симметрична или антисимметрична:

$$h[n] = \pm h[M - n], \quad n = 0, 1, \dots, M$$

Типы симметрии:

- **Симметричная** (чётная): $h[n] = h[M - n]$ — линейная фаза, сдвиг $\tau = M/2$
- **Антисимметричная** (нечётная): $h[n] = -h[M - n]$ — линейная фаза + сдвиг на $\pi/2$

Вывод: для КИХ-фильтров легко обеспечить линейную фазу, просто выбрав симметричную импульсную характеристику

6.5. БИХ-фильтры и их фазовая проблема

Почему у БИХ нелинейная фаза?

БИХ-фильтры имеют обратную связь, поэтому их импульсная характеристика не может быть симметричной (кроме тривиальных случаев). Как следствие, ФЧХ нелинейна, а групповая задержка зависит от частоты.

Как бороться с фазовыми искажениями БИХ?

1. Дополнительная коррекция фазы

После БИХ-фильтра можно включить **фазовый корректор** (всепропускающий фильтр), который выравнивает групповую задержку.

2. Фильтрация с обращением времени (**forward-backward** фильтрация)

Пропустить сигнал через фильтр дважды: сначала в прямом направлении, затем в обратном. Результат имеет нулевую фазу (фильтр с нулевой задержкой).

В Python: `scipy.signal.filtfilt(b, a, x)`



3. Использование КИХ вместо БИХ

Если фаза критична (ЭКГ, обработка изображений, аудио высокого разрешения), предпочитают КИХ-фильтры.

7. От преобразования Фурье к Z-преобразованию

Ограничения преобразования Фурье для анализа фильтров

- Преобразование Фурье существует только для сигналов, удовлетворяющих условию абсолютной интегрируемости (или суммируемости): $\sum |x[n]| < \infty$.
- Многие важные сигналы (например, единичный скачок $u[n]$, экспоненциально растущие последовательности) не имеют преобразования Фурье в классическом смысле.
- Преобразование Фурье не позволяет анализировать устойчивость системы (реакцию на сигналы, имеющие начало).

Z-преобразование обобщает преобразование Фурье, вводя комплексную переменную $z = re^{j\omega} = e^{\sigma} e^{j\omega} = e^{\sigma + j\omega}$.

7.1. Определение Z-преобразования и его связь с преобразованием Лапласа

Двустороннее Z-преобразование (для дискретных сигналов)

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}, \quad z \in \mathbb{C}$$

Связь с преобразованием Лапласа

Преобразование Лапласа для непрерывных сигналов:

$$X(s) = \int_{-\infty}^{\infty} x(t)e^{-st} dt, \quad s = \sigma + j\Omega$$

Идеализированная дискретизация:

$x[n] = x(nT)$, T — период дискретизации. Тогда

$$X(z) \approx \sum_{n=-\infty}^{\infty} x(nT)z^{-n}.$$

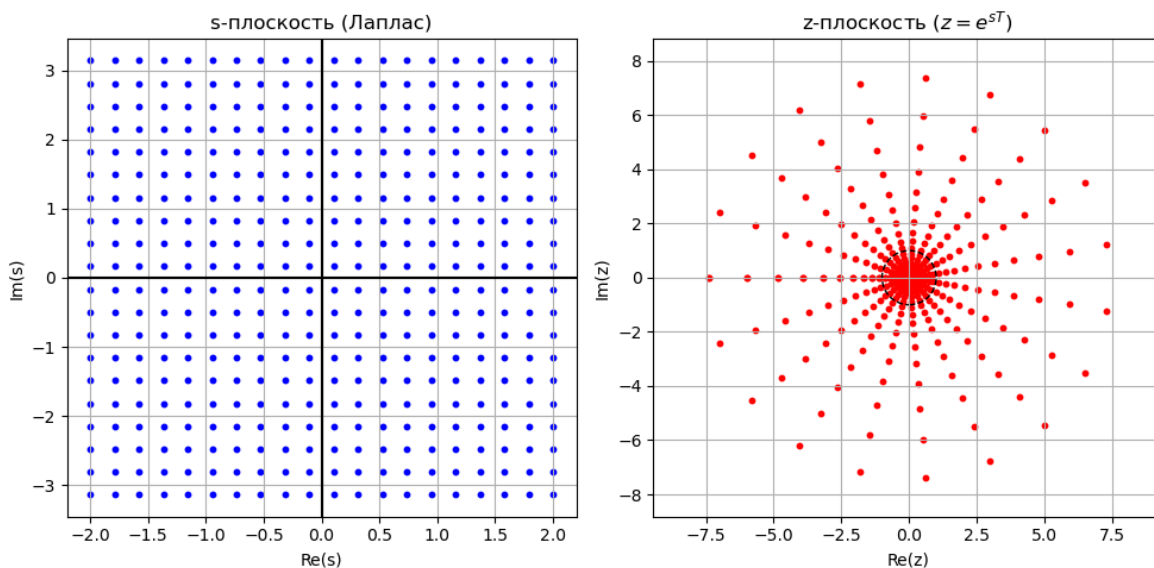
Подстановка $z = e^{sT}$ даёт связь:

$$X(z)|_{z=e^{sT}} = \sum_{n=-\infty}^{\infty} x(nT)e^{-snT},$$

что является приближением преобразования Лапласа для дискретизированного сигнала.

Физический смысл:

- s — комплексная частота (рад/с), $\sigma > 0$ — экспоненциально нарастающие сигналы, $\sigma < 0$ — экспоненциально затухающие сигналы.
- z — комплексная переменная, отображающая мнимую ось ($s = j\Omega$) на единичную окружность ($|z| = 1$), левую полуплоскость ($\sigma < 0$) — внутрь единичного круга ($|z| < 1$), правую полуплоскость ($\sigma > 0$) — вне единичного круга ($|z| > 1$).



7.2. Зачем нужно Z-преобразование вместо преобразования Фурье?

Преобразование Фурье

Ограничено сигналами с конечной энергией

Анализирует поведение на единичной окружности

Z-преобразование

Работает для более широкого класса сигналов (включая растущие экспоненты)

Анализирует всю комплексную плоскость

Преобразование Фурье	Z-преобразование
Не даёт прямой информации об устойчивости	Устойчивость определяется по расположению полюсов
Удобно для спектрального анализа	Удобно для анализа разностных уравнений и синтеза фильтров

Ключевое преимущество: Z-преобразование превращает разностные уравнения в алгебраические уравнения, позволяя легко находить передаточную функцию $H(z)$ и исследовать её свойства.

7.3. Передаточная функция и её связь с разностным уравнением

Рассмотрим систему, описываемую разностным уравнением:

$$y[n] = \sum_{k=0}^M b_k x[n-k] - \sum_{k=1}^N a_k y[n-k],$$

применим Z-преобразование и используем свойство сдвига:

$$Y(z) = \sum_{k=0}^M b_k z^{-k} X(z) - \sum_{k=1}^N a_k z^{-k} Y(z).$$

Отсюда получаем передаточную функцию:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^M b_k z^{-k}}{1 + \sum_{k=1}^N a_k z^{-k}}.$$

Частотная характеристика получается подстановкой $z = e^{j\omega T}$ (или $z = e^{j\omega}$ для нормированной частоты).

7.4. Полюса, нули и устойчивость

- **Нули** – корни числителя $B(z) = 0$.
- **Полюса** – корни знаменателя $A(z) = 0$ (т.е. знаменатель = 0).

Условие устойчивости:

Все полюса передаточной функции должны лежать **внутри единичной окружности** $|z| < 1$.

Почему?

Что такое устойчивость фильтра с точки зрения времени?

Фильтр (линейная стационарная система) устойчив, если при любом ограниченном входном сигнале выходной сигнал остаётся ограниченным. Самый простой тест: **подать короткий импульс (дельта-функцию)** и посмотреть, что будет на выходе.

Выход на импульс называется **импульсной характеристикой** $h(t)$.

- **Устойчивый фильтр:** $h(t) \rightarrow 0$ при $t \rightarrow \infty$. Отклик затухает.
- **Неустойчивый фильтр:** $h(t)$ не стремится к нулю (растёт, колеблется с постоянной амплитудой или стремится к константе).

Как преобразование Лапласа связывает полюсы и $h(t)$?

Возьмём передаточную функцию $H(s)$. Это дробно-рациональная функция:

$$H(s) = \frac{N(s)}{D(s)}$$

Полюсы — это корни знаменателя: $D(s) = 0$.

Ключевой факт: Разложение $H(s)$ на простые дроби (в случае простых полюсов) даёт:

$$H(s) = \sum_k \frac{A_k}{s - p_k}$$

Каждому полюсу p_k соответствует слагаемое $\frac{A_k}{s - p_k}$.

Обратное преобразование Лапласа этой дроби (для $t \geq 0$) даёт:

$$\mathcal{L}^{-1} \left\{ \frac{A_k}{s - p_k} \right\} = A_k \cdot e^{p_k t}$$

Следовательно, импульсная характеристика фильтра — это **сумма экспонент**, порождённых полюсами:

$$h(t) = \sum_k A_k \cdot e^{p_k t}$$

Почему знак действительной части полюса решает всё?

Полюс — комплексное число: $p_k = \sigma_k + i\omega_k$.

Тогда:

$$e^{p_k t} = e^{\sigma_k t} \cdot e^{i\omega_k t}$$

- **Мнимая часть** $\omega_k \rightarrow$ даёт колебания ($\cos(\omega_k t) + i \sin(\omega_k t)$). Она не влияет на рост или затухание амплитуды, только на то, как быстро осциллирует сигнал.
- **Действительная часть** $\sigma_k \rightarrow$ определяет **огibaющую** экспоненты $e^{\sigma_k t}$.

Смотрим на поведение при $t \rightarrow \infty$:

Положение полюса σ_k		Поведение $e^{\sigma_k t}$	Устойчивость
Левая полуплоскость $\sigma_k < 0$	Отрицательная	$e^{-\ \sigma_k\ t} \rightarrow 0$	Устойчив (затухает)
Мнимая ось $\sigma_k = 0$	Ноль	$e^0 = 1$ (или t если кратный)	Гранично устойчив (не затухает, но и не растёт)
Правая полуплоскость $\sigma_k > 0$	Положительная	$e^{+\ \sigma_k\ t} \rightarrow \infty$	Неустойчив (взрывается)

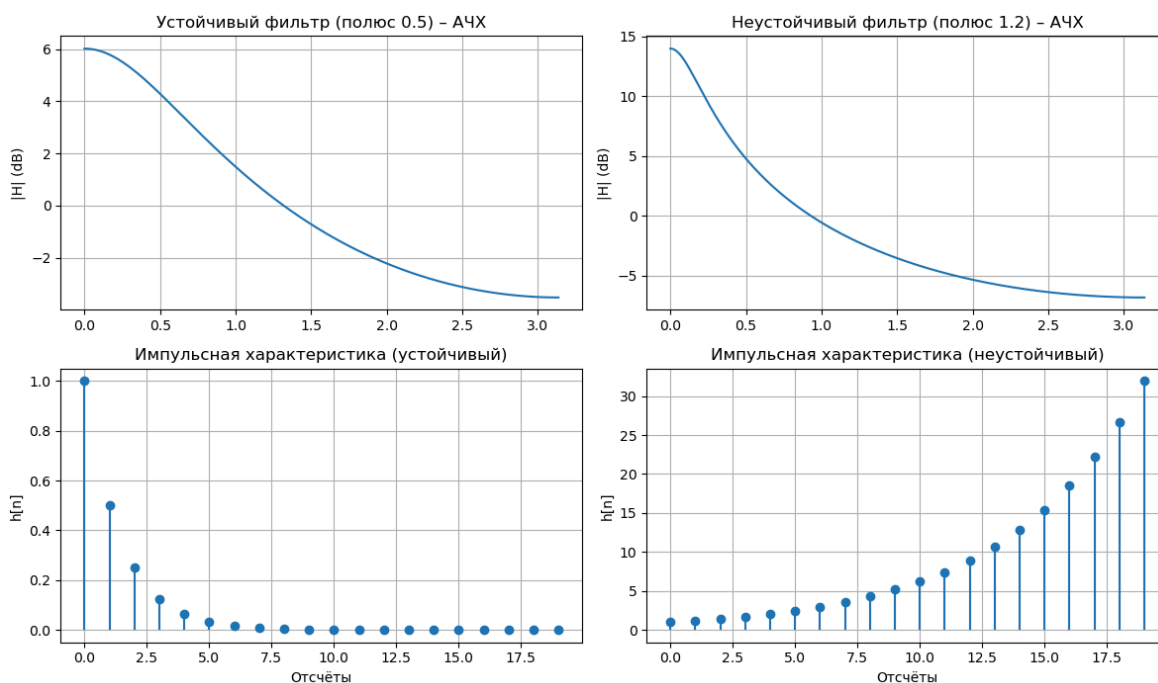
Вывод: Если **хотя бы один полюс** имеет положительную действительную часть (находится в правой полуплоскости), соответствующий член $e^{p_k t}$ будет расти неограниченно. Сумма таких членов тоже будет расти — фильтр неустойчив.

На что влияют нули передаточной функции?

Нули — это корни числителя $N(s)$. Они влияют на **амплитуды** A_k в разложении (коэффициенты перед экспонентами), но не на сами показатели p_k .

Нули могут сделать так, что коэффициент A_k обнулится для какого-то полюса (сокращение полюса-нуля), но это редкий и хрупкий случай. В общем случае:

Полюсы определяют, есть ли рост. Нули — только насколько сильно каждый режим возбуждён.



7.5. Практическое использование Z-преобразования

1. Анализ устойчивости:

Найти полюса передаточной функции; проверить, все ли они внутри единичной

окружности.

2. Расчёт частотной характеристики:

$$H(e^{j\omega}) = H(z)|_{z=e^{j\omega}}.$$

3. Синтез фильтров:

По заданной АЧХ определить коэффициенты b_k, a_k .

4. Решение разностных уравнений:

Перейти в z -область, решить алгебраическое уравнение, выполнить обратное Z -преобразование.

Обратное Z -преобразование обычно выполняется с помощью разложения на простые дроби или вычетов. В Python это делает `scipy.signal.residuez` и `scipy.signal.invresz`.

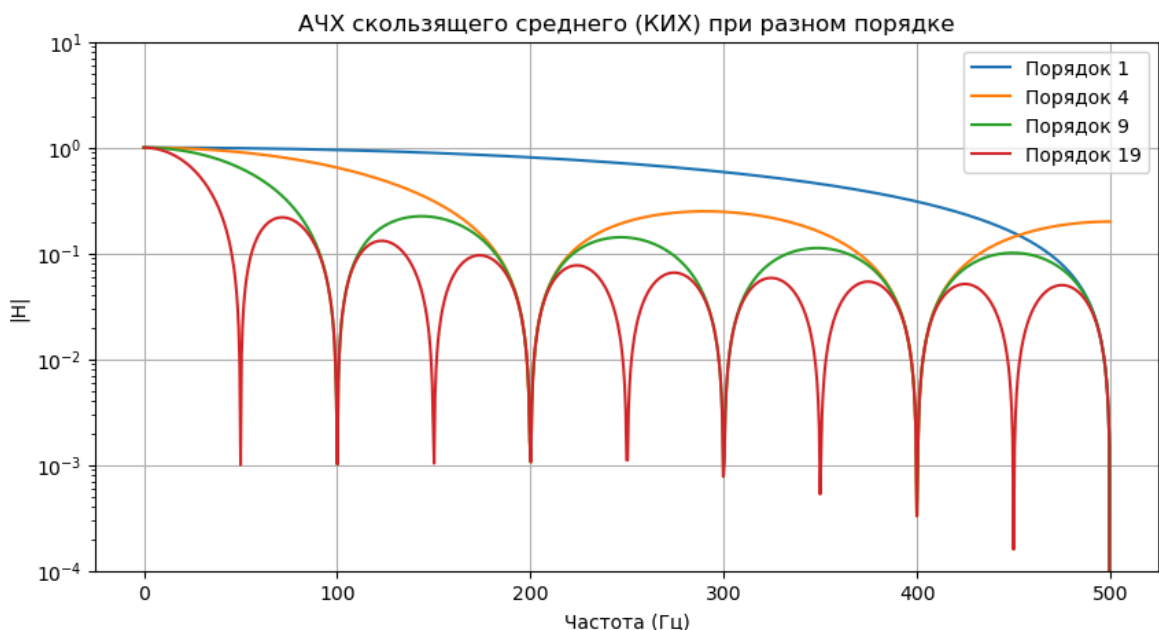
8. Порядок фильтра и его влияние на крутизну среза

Порядок фильтра – максимальная задержка, используемая в разностном уравнении:

- Для КИХ: порядок = M (количество коэффициентов минус 1).
- Для БИХ: порядок = $\max(M, N)$.

Влияние порядка на АЧХ:

Чем выше порядок, тем круче спад АЧХ в переходной полосе, но тем больше вычислительная сложность и задержка сигнала.



Вывод: увеличение порядка улучшает избирательность, но требует больше ресурсов.

9. Простейшие базисные фильтры (примеры)

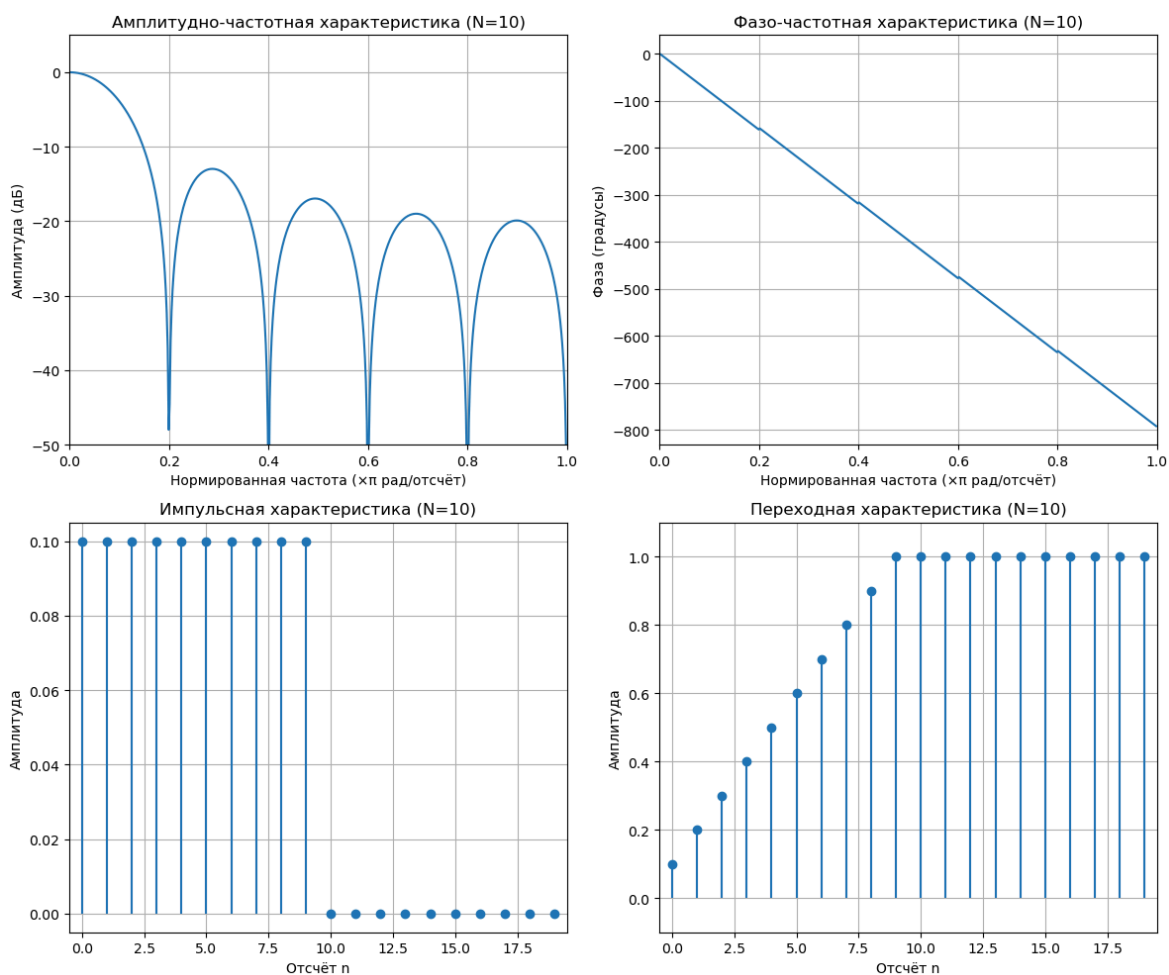
9.1. Скользящее среднее (КИХ, ФНЧ)

$$y[n] = \frac{1}{M} \sum_{k=0}^{M-1} x[n - k]$$

АЧХ имеет вид

$$\left| H \left(e^{2\pi f / f_s} \right) \right| = \left| \frac{\sin(\pi f M / f_s)}{M \sin(\pi f / f_s)} \right|$$

Подавляет высокие частоты, сглаживает шум.



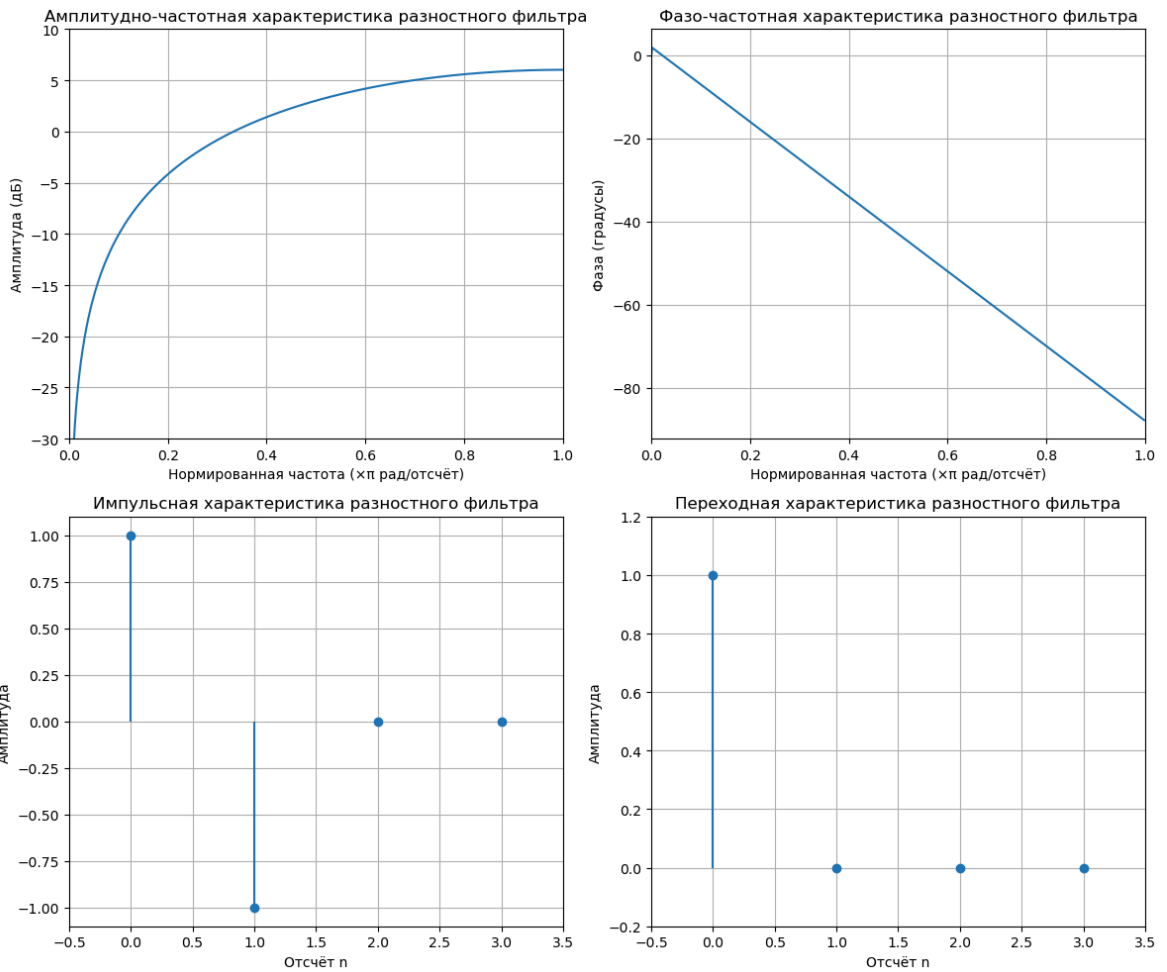
9.2. Разностный фильтр (КИХ, ФВЧ)

$$y[n] = x[n] - x[n - 1]$$

АЧХ фильтра:

$$\left| H \left(e^{j2\pi f/f_s} \right) \right| = 2 \left| \sin(\pi f/f_s) \right|$$

Выделяет перепады сигнала, подчёркивает высокие частоты. Часто используется для обнаружения границ.



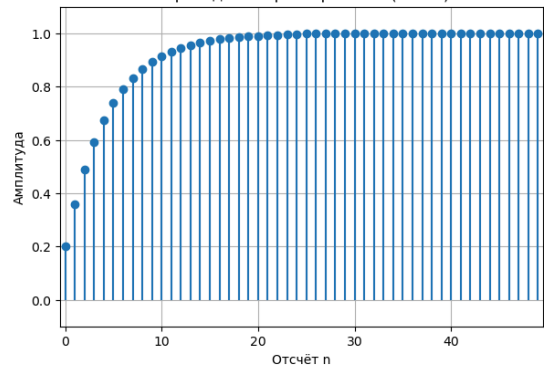
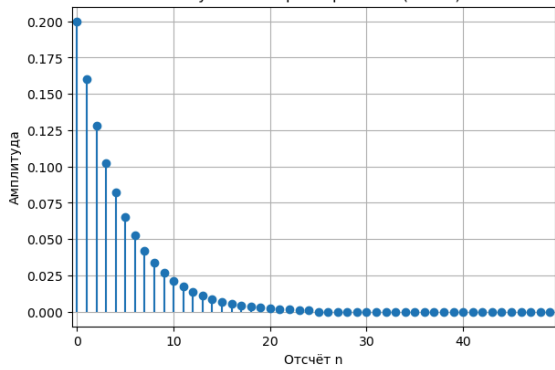
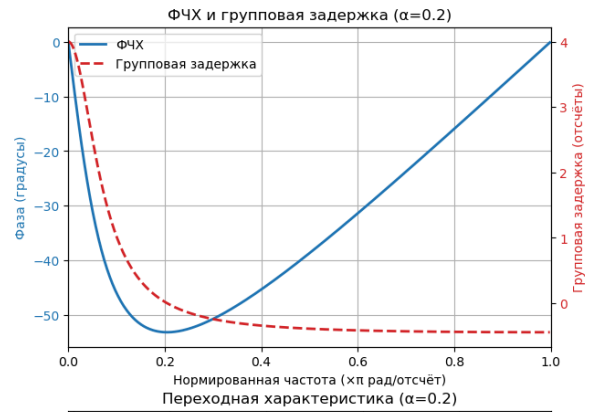
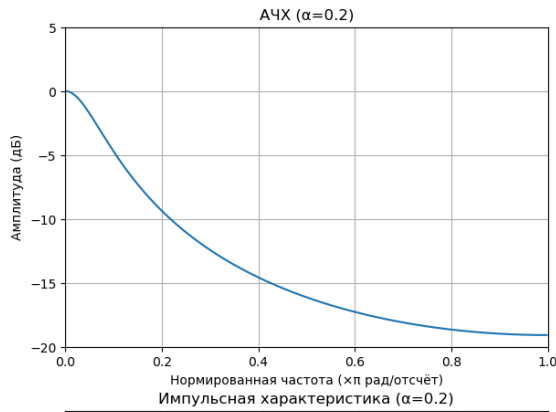
9.3. Экспоненциальное сглаживание (БИХ, ФНЧ)

$$y[n] = \alpha x[n] + (1 - \alpha)y[n - 1], \quad 0 < \alpha < 1$$

Передаточная функция

$$H(z) = \frac{\alpha}{1 - (1 - \alpha)z^{-1}}$$

Имеет единственный полюс в точке $z = 1 - \alpha$ (внутри единичной окружности, устойчив).



9.4. Сравнение экспоненциального фильтра и скользящего среднего

Критерий	Экспоненциальный фильтр	Простое скользящее среднее
Потребление памяти	$O(1)$ — требуется хранить только предыдущее сглаженное значение	$O(N)$ — нужно хранить N последних значений (использовать циклический буфер)
Вычислительная сложность	$O(1)$ на шаг — одна операция умножения и сложения	Или $O(N)$, или $O(1)$, но с памятью $O(N)$
Реакция на старые данные	плавное экспоненциальное затухание — старые данные не исчезают скачком, но никогда не «забывает» полностью	жёсткое окно — после N шагов точка данных полностью пропадает, что может вызвать скачок
Настройка чувствительности	один параметр α (0..1) — плавная настройка	только целочисленный параметр N — дискретные шаги настройки
Критерий	Экспоненциальный фильтр	Простое скользящее среднее
Поведение на резких фронтах	экспоненциальный отклик — быстро реагирует в начале фронта, плавно замедляясь, но теоретически бесконечное время установления	линейный отклик за N шагов — предсказуемо, но долго
Работа с пропусками данных	классический алгоритм ломается — нужно модифицировать формулу	проще — берём доступные точки в окне (среднее по фактическим значениям)

Критерий	Экспоненциальный фильтр	Простое скользящее среднее
Контроль фазового сдвига	нелинейная фазовая характеристика, сложнее предсказать запаздывание	линейная фаза — предсказуемая задержка
Устойчивость к выбросам	реакция на выброс зависит от α — при малом α хорошо сглаживает, при большом — пропустит, но выброс влияет на все последующие значения	влияние выброса длится ровно N шагов, затем исчезает полностью, но если N мало — выброс сильно исказит среднее
Где лучше применять	поточковые данные, встроенные системы (Arduino, PLC), трейдинг, телеметрия, где критичны память и скорость	офлайн-анализ, фиксированные временные окна (например, скользящее среднее за 7 дней), частотная фильтрация с известной задержкой