

Федеральное государственное автономное образовательное
учреждение высшего образования

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет систем управления и робототехники

Лабораторная работа №4

ЛИНЕЙНАЯ ФИЛЬТРАЦИЯ

Студент: Овчинников П.А.

Поток: ЧАСТ.МЕТ. 1.3

Преподаватель: Перегудин А.А.

Пашенко А.В.

Санкт-Петербург

2024

Содержание

Спектральное дифференцирование	3
Линейные фильтры	4
Фильтр первого порядка	5
Специальный фильтр	7
Сглаживание биржевых данных	10

Спектральное дифференцирование

Зададим $t \in [-25, 25]$ и рассмотрим сигнал $y(t)$, представляющий из себя зашумлённую синусоиду:

$$y(t) = \sin(t) + 0.1(\text{rand}(\text{size}(t)) - 0.5)$$

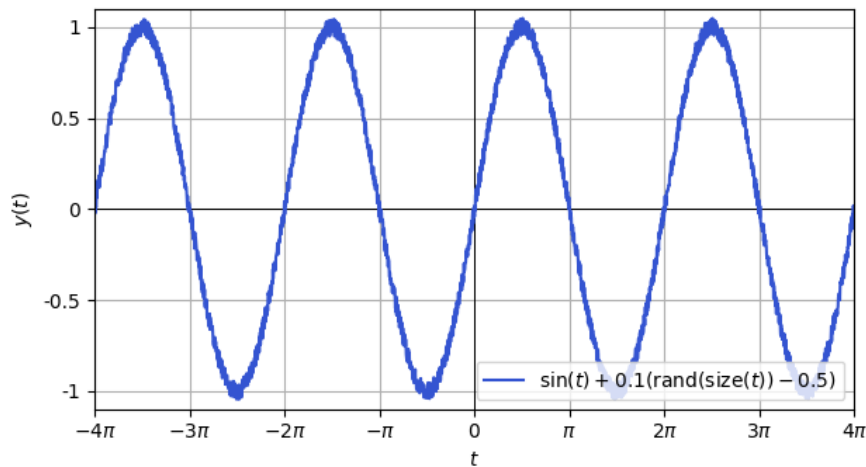


Рис. 1: Сигнал $y(t)$

Теперь найдём численную производную этого сигнала с помощью формулы поэлементного дифференцирования:

$$y'(t) = \frac{y(t+1) - y(t)}{dt}, \quad dt - \text{шаг дискретизации}$$

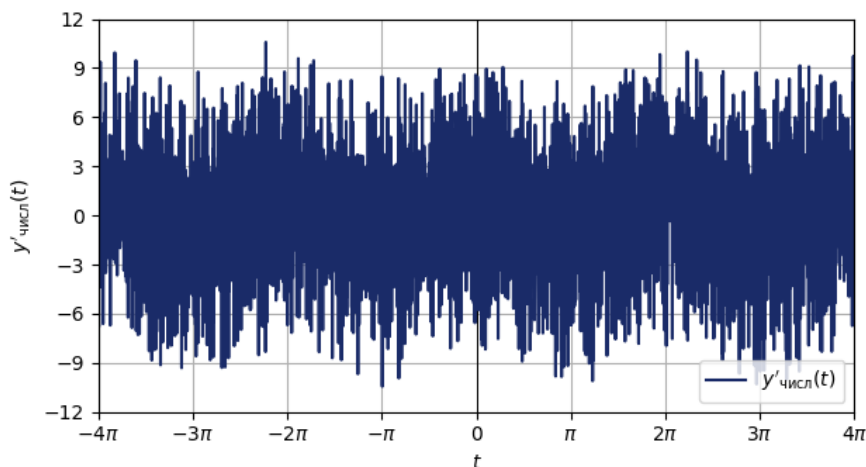


Рис. 2: Численная производная сигнала

Как мы видим, численная производная сигнала имеет колоссальное количество шума. Это связано с резкими скачками функции на каждом шаге дискретизации, вызванными шумом. Конечно, здесь заметна истинная производная сигнала $\cos(t)$, но всё же шум вносит свои неисправимые коррективы.

Теперь найдём спектральную производную сигнала. Для этого применим преобразование Фурье к сигналу $y(t)$, а затем умножим полученный Фурье-образ на $i\omega$. После этого применим обратное преобразование Фурье к полученному произведению — и так мы получим спектральную производную сигнала.

Такая манипуляция основана на свойстве Фурье-образа: $\mathcal{F}\left\{\frac{d}{dt}f\right\} = 2\pi i\nu\mathcal{F}\{f\}$. В зависимости от того, каким преобразованием мы пользуемся, унитарным или неунитарным, нам может потребоваться дополнительный множитель 2π . В этой работе мы пользуемся унитарным преобразованием к угловой частоте, поэтому $i\omega$ достаточно.

Для начала рассмотрим Фурье-образ сигнала и его преобразование после умножения на $i\omega$:

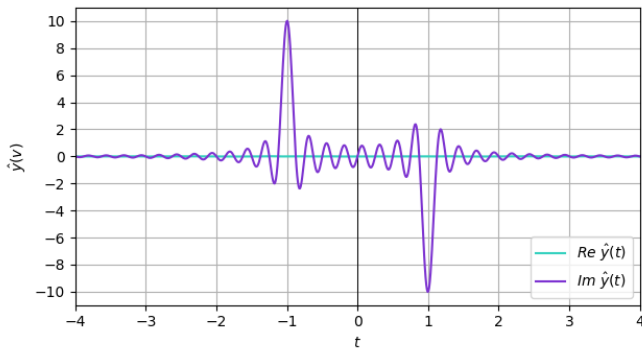
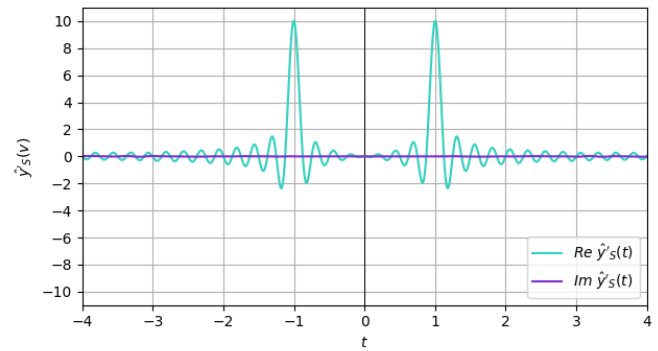


Рис. 3: Фурье-образ сигнала

Рис. 4: Фурье-образ после умножения на $i\omega$

Как мы видим, весь образ оказался в мнимой части. Теперь применим обратное преобразование к получившемуся образу — получим спектральную производную. Так она выглядит в сравнении с истинной производной $\cos(t)$:

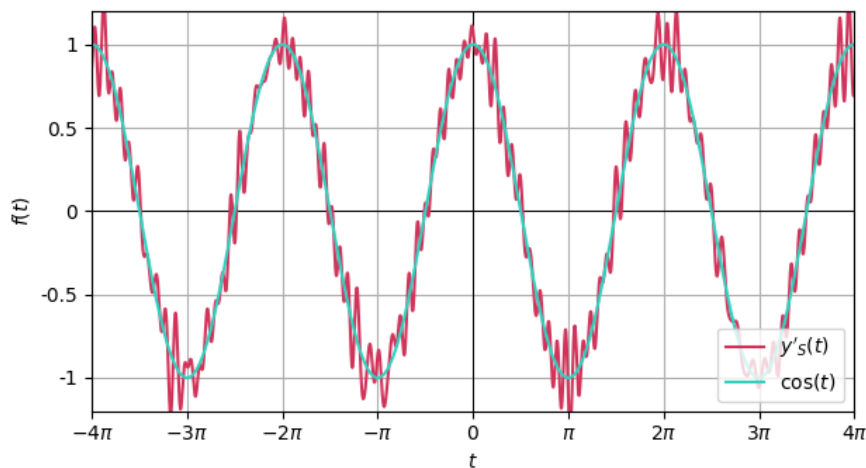


Рис. 5: Сравнение спектральной производной и истинной производной сигнала

Как мы видим, спектральная производная сигнала практически совпадает с истинной производной сигнала, в то время как численная производная сигнала содержит колоссальное количество шума. Поэтому спектральную производную можно рассматривать как способ найти приближённую производную сигнала даже с наличием шумов в нём.

В: Здесь стоит отметить недостаток спектрального дифференцирования. Мы рассматривали промежуток $[-25, 25]$, достаточный для того, чтобы спектральное дифференцирование работало хорошо. Но если мы рассмотрим больший промежуток, например, как в условии задания, $[-100, 100]$, то Фурье-образ будет принимать на себя всё больше и больше шума, что приведёт к ухудшению качества спектральной производной:

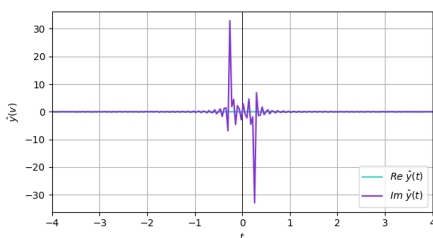


Рис. 6: Фурье-образ сигнала

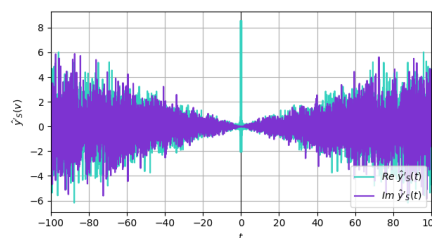
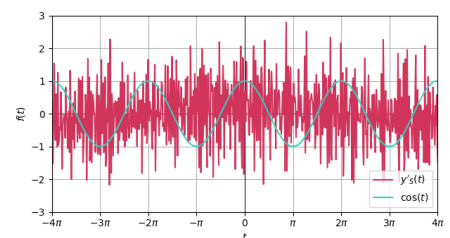
Рис. 7: Фурье-образ $\cdot i\omega$ 

Рис. 8: Сравнение производных

Линейные фильтры

Рассмотрим сигнал, представляющий собой зашумленную прямоугольную волну:

$$u(t) = g(t) + b(\text{rand}(\text{size}(t)) - 0.5) + c \sin(d \cdot t)$$

Фильтр первого порядка

Для выполнения задания с фильтром первого порядка нам понадобится только случайный шум в сигнале, поэтому $c = 0$. Необходимо пропустить сигнал u через фильтр первого порядка, являющийся передаточной функцией:

$$W_1(p) = \frac{1}{Tp + 1}$$

И, изменяя постоянную времени T и параметр a , исследовать их влияние на эффективность фильтрации.

Пусть $a = 5$ — это амплитуда прямоугольной волны, $b = 1$ — амплитуда шума, а касательно постоянной времени предположим $T = 0.5$. Пропустим сигнал с такими параметрами через такой фильтр и посмотрим на результат:

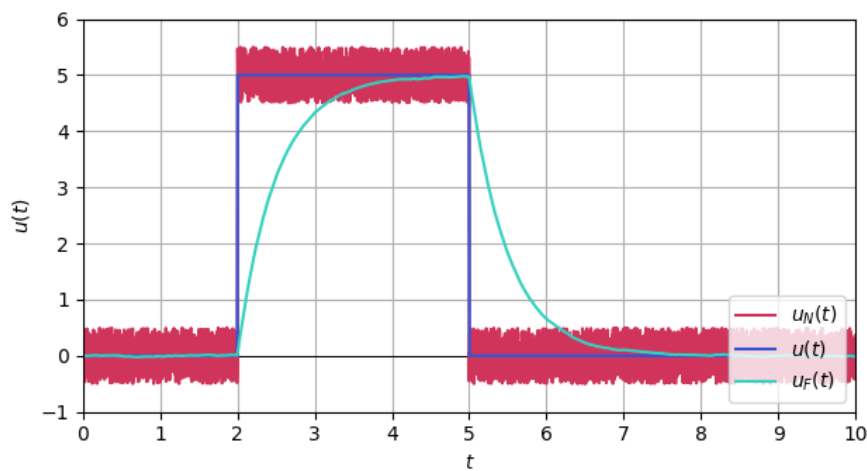


Рис. 9: Графики исходного сигнала $u(t)$, зашумлённого сигнала $u_N(t)$ и отфильтрованного сигнала $u_F(t)$

Фронт и спад прямоугольной волны стали более плавными, а шум уменьшился. Это связано с тем, что фильтр уменьшает высокочастотные компоненты сигнала. Теперь посмотрим на модули Фурье-образов сигналов:

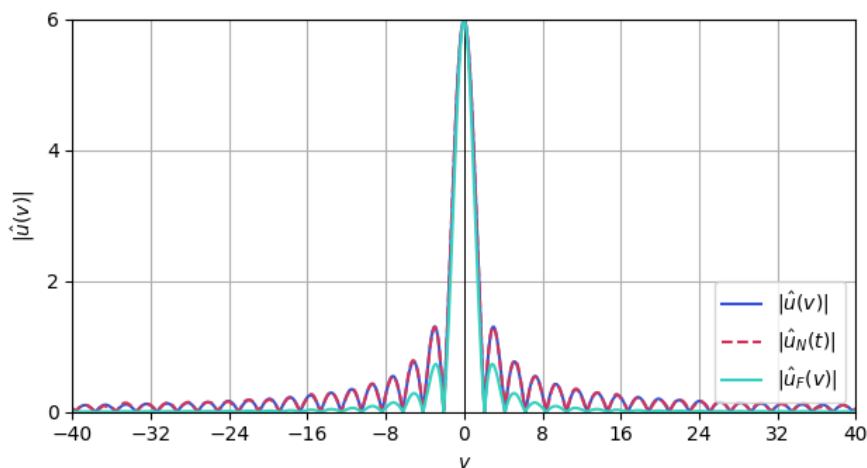


Рис. 10: Графики модулей фурье-образов исходного, зашумлённого и отфильтрованного сигналов

По образу видно, что фильтр плавно уменьшает высокочастотные компоненты, что и привело к сглаживанию.

Посмотрим на АЧХ фильтра:

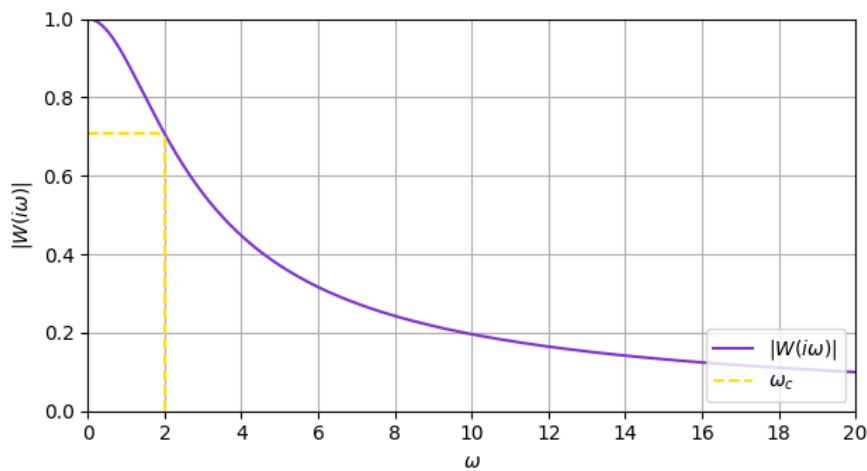


Рис. 11: АЧХ фильтра первого порядка при $T = 0.5$

Здесь и в следующих графиках для наглядности отмечена частота среза ω_c . Для этой частоты выполняется условие $|W_1(j\omega_c)| = 1/\sqrt{2}$. Для фильтра первого порядка эта частота среза равна $\omega_c = 1/T = 1/0.5 = 2$.

Теперь попробуем уменьшить постоянную времени фильтра до $T = 0.1$:

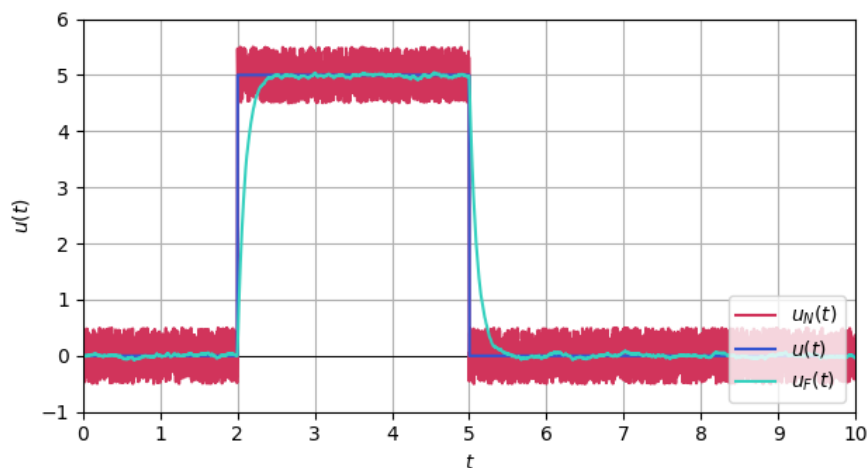


Рис. 12: Графики исходного, зашумлённого и отфильтрованного сигналов при $T = 0.1$

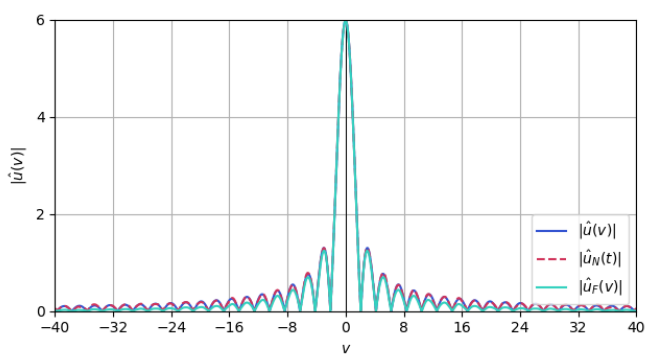


Рис. 13: Модули Фурье-образов сигналов при $T = 0.1$

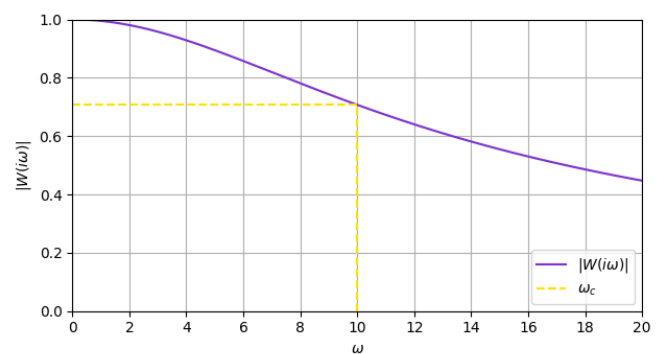


Рис. 14: АЧХ фильтра при $T = 0.1$

Как мы видим, при уменьшении постоянной времени фильтра, фильтрация становится более жёсткой, что уменьшает степень сглаживания. Теперь отфильтрованный сигнал больше похожит на прямоугольную волну, но в нём больше шумов.

Наконец, попробуем при той же постоянной времени увеличить амплитуду прямоугольной волны до $a = 20$:

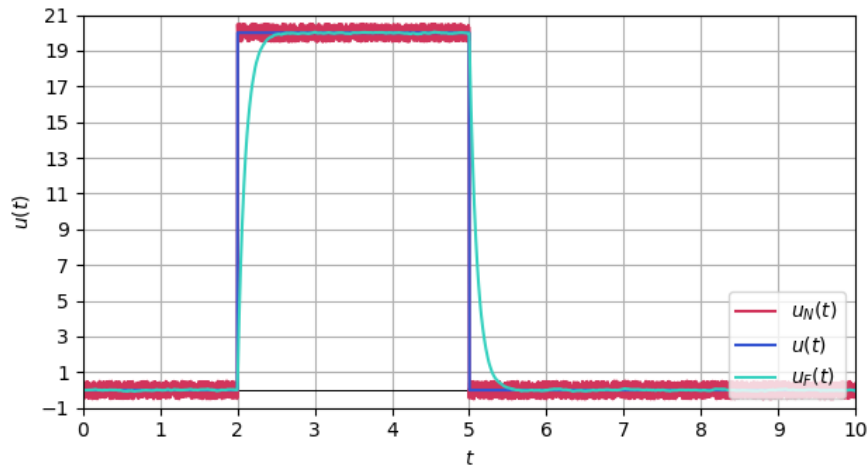


Рис. 15: Графики исходного, зашумлённого и отфильтрованного сигналов при $a = 20$

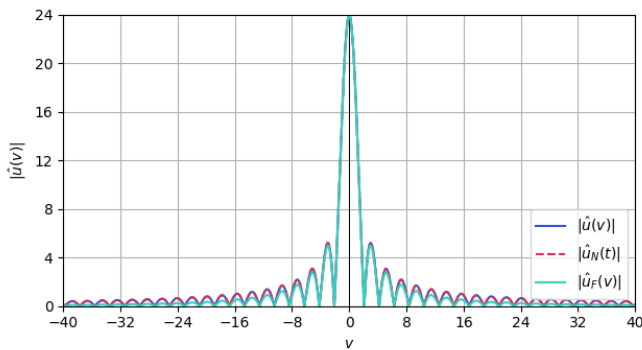


Рис. 16: Модули Фурье-образов сигналов при $a = 20$

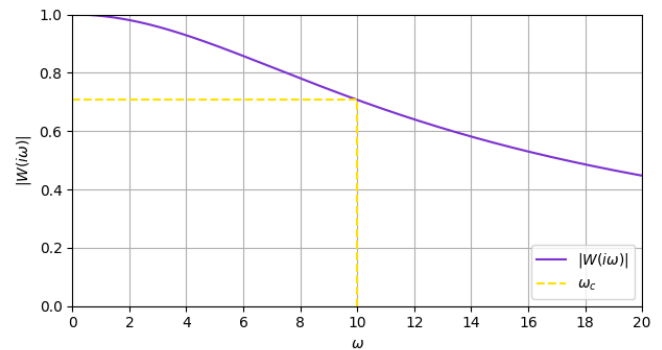


Рис. 17: АЧХ фильтра при $a = 20$

Как мы видим, увеличение амплитуды прямоугольной волны не изменило сглаживание на фронте и спаде, но уменьшило долю шума в отфильтрованном сигнале, потому как отношение амплитуды шума к общей амплитуде прямоугольной волны также уменьшилось. АЧХ фильтра не изменилась, потому как и постоянная времени не изменилась.

Вывод: фильтр первого порядка позволяет уменьшить шум в сигнале, но при этом сглаживает и сам сигнал. Постоянная времени фильтра T влияет на степень сглаживания — чем больше T , тем сильнее сглаживание, а амплитуды прямоугольной волны (параметр a) и шума (параметр b) влияют на долю шума в отфильтрованном сигнале.

Специальный фильтр

Теперь рассмотрим линейный фильтр второго порядка, передаточная функция которого выглядит так:

$$W_2(p) = \frac{(T_1 p + 1)^2}{(T_2 p + 1)(T_3 p + 1)} = \frac{T_1^2 p^2 + 2T_1 p + 1}{T_2 T_3 p^2 + (T_2 + T_3) p + 1}$$

Для выполнения задания понадобится сигнал только с гармоническим шумом, поэтому $b = 0$. Необходимо пропустить такой сигнал u через фильтр, подобрать T_1, T_2, T_3 к различным значениям d так, чтобы фильтр хорошо фильтровал шум и не сильно искажал сигнал, и исследовать как параметр c влияет на фильтрацию.

Мы будем подбирать постоянные времени фильтра следующим образом:

- Параметр T_1 , находясь в числителе передаточной функции, влияет на усиление высоких частот. Так как это нам не требуется, положим $T_1 = 10^{-8}$ — настолько минимальным, насколько это возможно.
- Беря во внимание, что для фильтра первого порядка действует правило частоты среза $\omega_c = 1/T$, подберём T_2 и T_3 так, чтобы они были связаны с частотой гармонического шума d .

Пусть $a = 5$ — амплитуда прямоугольной волны, $c = 0.5$ — амплитуда гармонического шума, $d = 18$ — частота гармонического шума. Подберём постоянные времени фильтра так, чтобы он хорошо убирал гармонический шум (удвоение постоянных связано с тем, что их две в знаменателе):

$$T_1 = 10^{-8} \quad T_2 = T_3 = \frac{2}{d} = \frac{1}{9}$$

Теперь посмотрим на сравнительные графики сигналов и модулей их Фурье-образов, а также на график АЧХ:

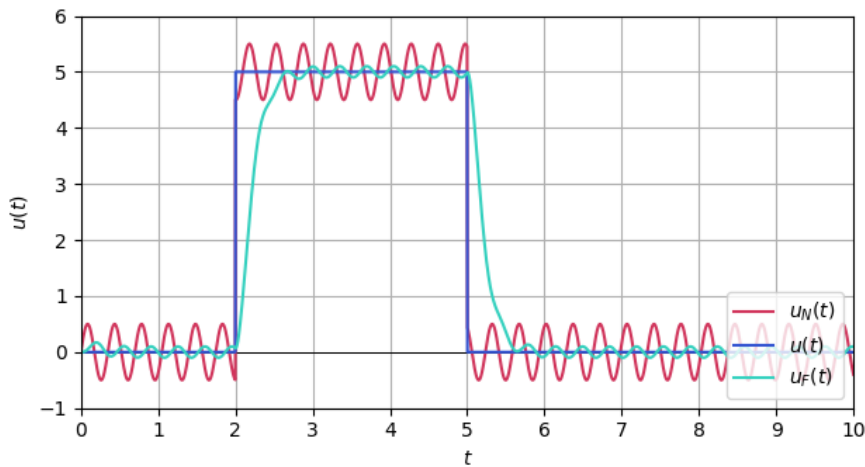


Рис. 18: Графики исходного, зашумлённого и отфильтрованного сигналов при $c = 0.5$ и $d = 18$

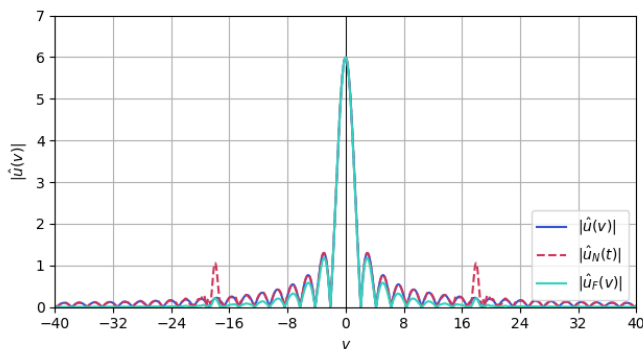


Рис. 19: Модули Фурье-образов при $c = 0.5$ и $d = 18$

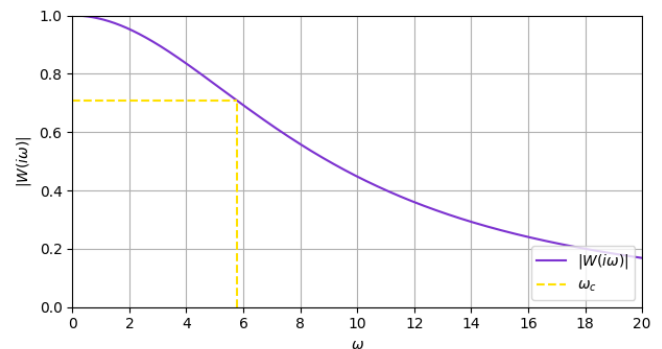


Рис. 20: АЧХ фильтра при $c = 0.5$ и $d = 18$

Мы можем наблюдать, что фильтр второго порядка хорошо убирает гармонический шум и при этом фронт и спад не сильно сглажены. На графике модулей Фурье-образов хорошо видно, что частота синусоидального шума заглушена. Если задать T_2 и T_3 ещё больше, то шум заглушится сильнее, но и в свою очередь фронт и спад прямоугольной волны будут также сглажены ещё сильнее.

Теперь рассмотрим шум с той же амплитудой, но с другой частотой, $d = 30$:

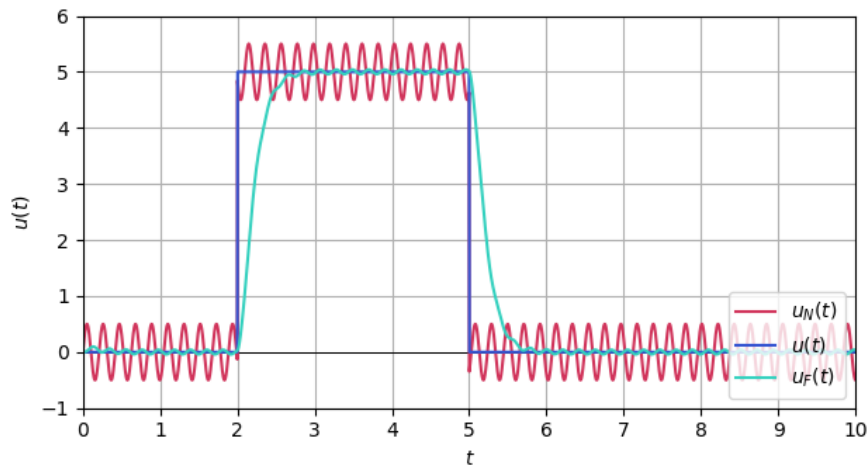


Рис. 21: Графики исходного, зашумлённого и отфильтрованного сигналов при $c = 0.5$ и $d = 30$

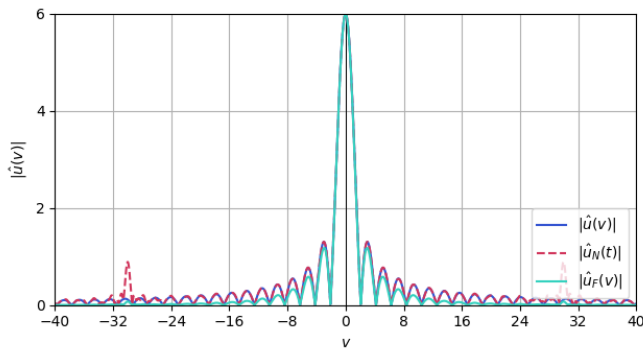


Рис. 22: Модули Фурье-образов при $c = 0.5$ и $d = 30$

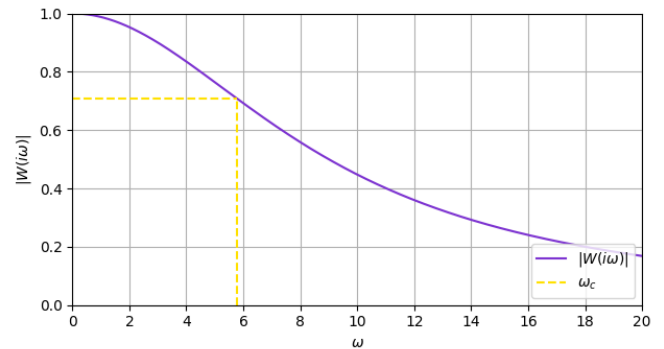


Рис. 23: АЧХ фильтра при $c = 0.5$ и $d = 30$

По АЧХ фильтра видно, что фильтр не изменился. На графике модулей Фурье-образов видно, что частота гармонического шума $d = 30$ также заглушена — т.к. она находится ещё дальше от нулевой частоты, то гармонический шум заглушается ещё сильнее. Именно это мы наблюдаем на сравнительном графике сигналов — амплитуда шума в отфильтрованном сигнале меньше.

Теперь попробуем увеличить амплитуду гармонического шума до $c = 3$, параметр d оставим прежним:

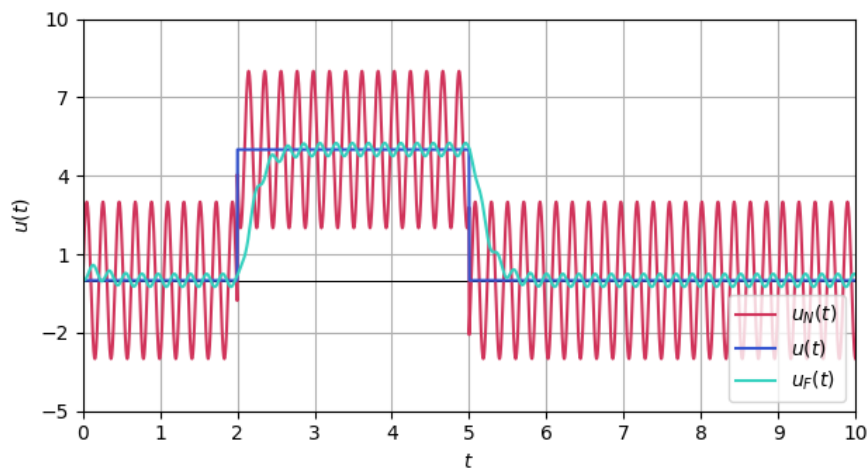


Рис. 24: Графики исходного, зашумлённого и отфильтрованного сигналов при $c = 3$ и $d = 30$

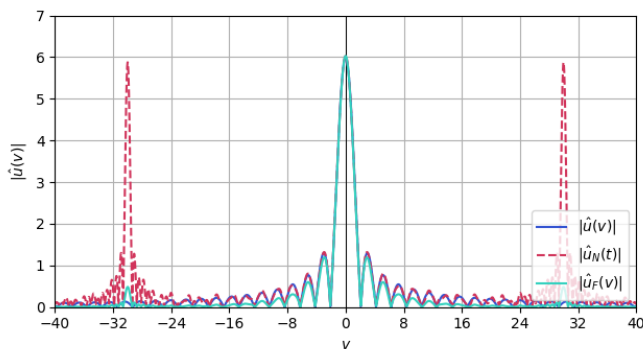


Рис. 25: Модули Фурье-образов при $c = 3$ и $d = 30$

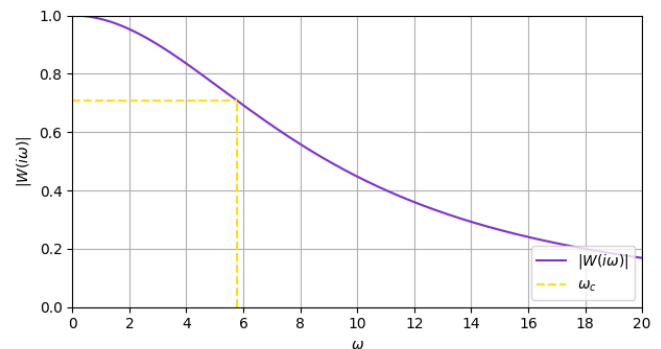


Рис. 26: АЧХ фильтра при $c = 3$ и $d = 30$

Как мы видим, увеличение амплитуды гармонического шума привело к увеличению амплитуды этого шума в отфильтрованном сигнале. На графике модулей Фурье-образов видно, что частота гармонического шума $d = 30$ выросла по амплитуде и при фильтрации заглушилась слабее, в сравнении с предыдущим примером.

Вывод: фильтр второго порядка позволяет убрать гармонический шум из сигнала, не сильно сглаживая сам сигнал. Постоянные времени фильтра T_1 , T_2 , T_3 влияют на степень фильтрации. Амплитуда гармонического шума c влияет на долю шума в отфильтрованном сигнале — чем больше c , тем больше шума в отфильтрованном сигнале.

Сглаживание биржевых данных

Теперь рассмотрим, как можно применить линейные фильтры на практике. К примеру, рассмотрим скользящее среднее — функция сглаживания данных, которая как раз основана на линейном фильтре первого порядка, который сглаживает шумы в графике котировок акции.

Реализуем скользящее среднее со следующими окнами: 1 день, 1 неделя, 1 месяц, 3 месяца, 1 год. Возьмём данные о котировках акции ВКСО, принадлежащих компании VK, за период с 02.07.20 (появление тикета на Мосбирже) по 29.12.23 (последний день торгов 2023 года) с интервалом в 1 день. Таким образом постоянная времени $T = 1$ будет эквивалентна одному дню.

Посмотрим, как работает линейный фильтр первого порядка для всех T :

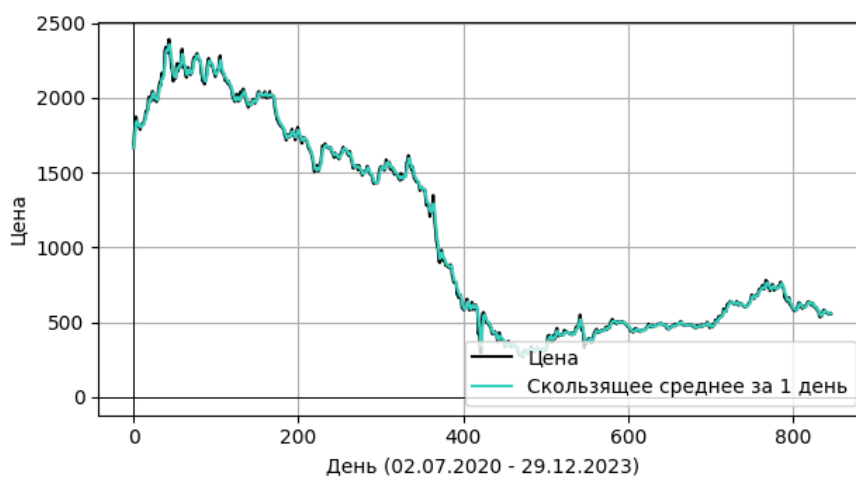


Рис. 27: График цены акции VKCO и скользящее среднее с окном в 1 день ($T = 1$)

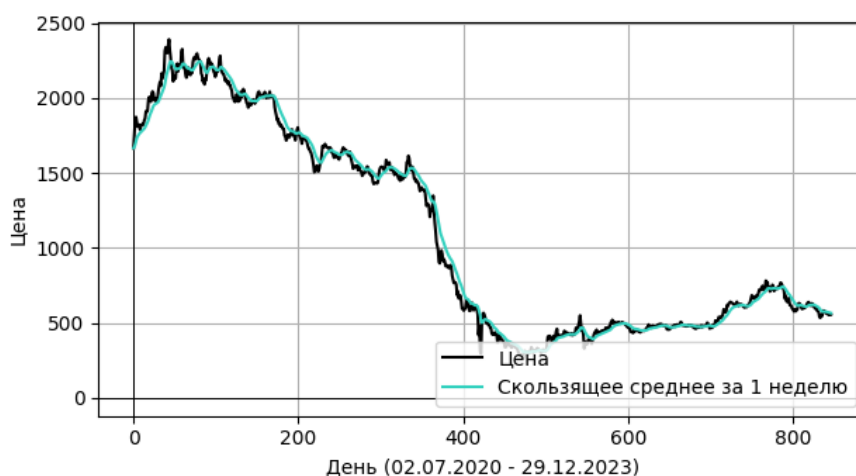


Рис. 28: График цены акции VKCO и скользящее среднее с окном в 1 неделю ($T = 7$)

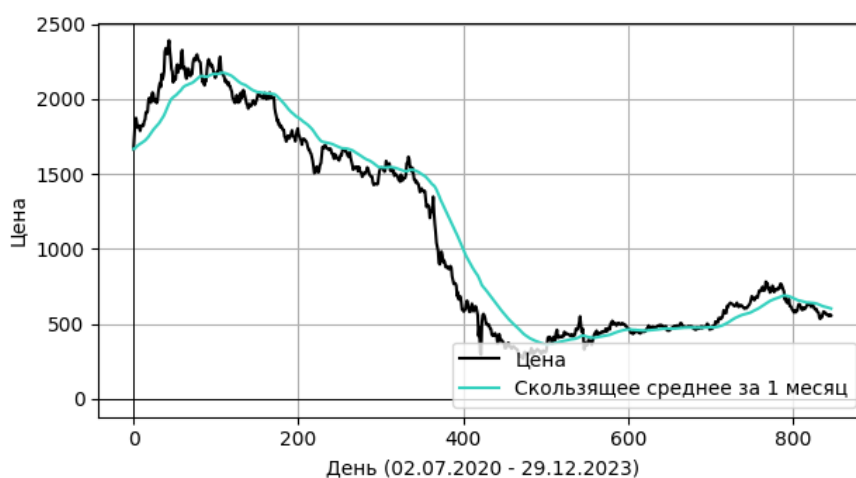
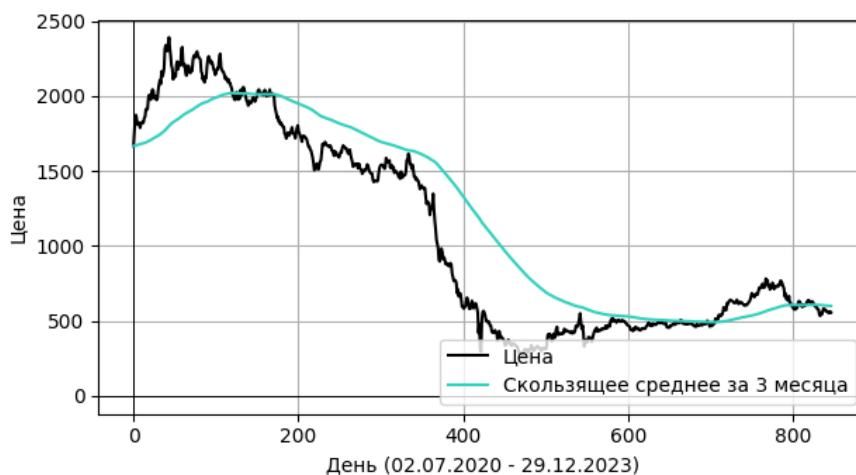
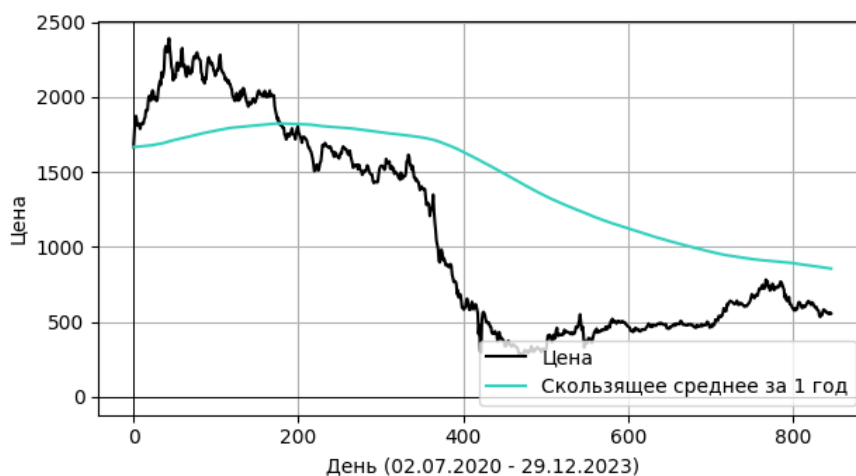


Рис. 29: График цены акции VKCO и скользящее среднее с окном в 1 месяц ($T = 30$)

Рис. 30: График цены акции VKCO и скользящее среднее с окном в 3 месяца ($T = 90$)Рис. 31: График цены акции VKCO и скользящее среднее с окном в 1 год ($T = 365$)

В: при использовании `lsm` в Python фильтрованный сигнал тоже начинается из нуля. Для этого необходимо добавить параметр `X0=prices[0] * T`. После этого фильтрованный сигнал будет начинаться из той же точки, что и график цен.

Вывод: у линейных фильтров есть практическое применение — не только фильтрация шумов в сигнале, но и сглаживание биржевых и в принципе любых других данных.