

ТЕХНИКА ЯДЕРНОГО  
ЭКСПЕРИМЕНТА

УДК 53.07

МЕТОДЫ ПРЕОБРАЗОВАНИЯ ДВУХЭКСПОНЕНЦИАЛЬНЫХ И  
ТРЕХЭКСПОНЕНЦИАЛЬНЫХ ИМПУЛЬСОВ В ТРАПЕЦЕИДАЛЬНЫЕ  
С ПОМОЩЬЮ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ

©2026. М. А. Голубев\*, Т. И. Глушкова, В. А. Соловей

*Петербургский институт ядерной физики им. Б.П. Константинова  
Национального исследовательского центра “Курчатовский институт”  
Россия, 188300, Гатчина Ленинградской обл., Орлова роща, 1*

*\*e-mail: golubev\_ma@npi.nrcki.ru*

Поступила в редакцию 10.09.2025 г.

После доработки 03.02.2026 г.

Принята к публикации 05.03.2026 г.

Рассмотрен реализованный в программируемой логической интегральной схеме (ПЛИС) цифровой фильтр для преобразования двухэкспоненциальных импульсов в трапецеидальные. Такие сигналы характерны для детекторов частиц, а преобразование необходимо для того, чтобы избежать наложения плавно затухающих импульсов для более точного измерения их относительных амплитуд. Обнаружено, что в одноплатном компьютере ADC-SoC, совмещающем аналого-цифровой преобразователь (АЦП) и ПЛИС, входная схема АЦП добавляет третью временную константу к двухэкспоненциальным импульсам, искажая их, поэтому был аналитически рассчитан и реализован в ПЛИС цифровой фильтр для преобразования импульсов, названных нами трехэкспоненциальными, в трапецеидальные. Найдены условия устойчивости фильтра. Проведены измерения амплитудного спектра с использованием искусственных двухэкспоненциальных импульсов.

## 1. ВВЕДЕНИЕ

Последовательность значений сигнала  $x[n]$ , полученного через равные промежутки времени, например от аналого-цифрового преобразователя (АЦП), можно преобразовать с помощью дискретного фильтра с конечной импульсной характеристикой (КИХ)  $h[n]$  или бесконечной импульсной характеристикой (БИХ) к последовательности  $y[n]$ . В случае БИХ-фильтра последовательность  $h[n]$ , как правило, задается с помощью рекурсивной формулы. Математическое применение дискретного фильтра – это операция дискретной свертки

$$y[n] = h[n] * x[n] = \sum_{i=-\infty}^{\infty} h[n-i]x[n]. \quad (1)$$

Единичным сигналом мы называем дискретную дельта-функцию  $\delta(n)$ , подразумевая алгебраический смысл свертки дискретного сигнала и импульсной характеристики дискретного фильтра

$$\delta[n] = 1, n = 0; \quad \delta[n] = 0, n \neq 0. \quad (2)$$

Действительно,

$$h[n] * \delta[n] = \sum_{i=-\infty}^{\infty} h[n] \delta[n - i] = h[n] \delta[0]_{i=n} = h[n]. \quad (3)$$

Разностное уравнение дискретного фильтра, содержащее в себе рекурсивную формулу, имеет вид

$$\sum_{i=0}^M a_i y[n - i] = \sum_{j=0}^N b_j x[n - j]. \quad (4)$$

В случае КИХ-фильтра  $M = 0$ , тогда рекурсивная формула в разностном уравнении не содержится. Далее мы будем выражать интересующие нас фильтры с конечными и бесконечными импульсными характеристиками их разностными уравнениями.

Дискретная функция Хэвисайда  $u[n]$  является импульсной характеристикой или откликом на единичный сигнал сумматора:

$$u[n] = 1, n \geq 0; \quad u[n] = 0, n < 0. \quad (5)$$

Сумматор является неустойчивым фильтром с рекурсивной формулой

$$u[n] = \delta[n] + u[n - 1]. \quad (6)$$

Для того чтобы гарантировать линейность

$$\forall f \in \mathbb{R}: f y[n] = h[n] * (f x[n]), \quad (7)$$

стационарность

$$\forall \Delta n \in \mathbb{Z}: y[n - \Delta n] = h[n] * x[n - \Delta n] \quad (8)$$

и детерминированность решения разностного уравнения, а следовательно, его единственность, на последовательности  $x[n], h[n]$  накладывается условие состояния покоя в начальном положении, т. е.

$$x[n] = 0, \quad h[n] = 0, \quad n < 0. \quad (9)$$

Изложенные выше и прочие базовые вопросы цифровой фильтрации разобраны в работе [1].

В схемотехнике односторонний экспоненциальный сигнал получается при пропуске прямоугольных импульсов через дифференцирующую CR-схему.

Двухэкспоненциальный сигнал характерен для электроники ионизационных детекторов частиц. Он содержит две временные константы – одна из них определяется временем затухания заряда  $\tau_0$  в детекторе, а вторая определяется RC-схемой, необходимой для интегрирования сигнала и уменьшения шума [2]. В нашей работе исследовались двухэкспоненциальные сигналы, полученные дифференцированием прямоугольных импульсов с помощью CR-схемы и последующим интегрированием с помощью RC-схемы.

## 2. ВОССТАНОВЛЕНИЕ ЕДИНИЧНОГО ДИСКРЕТНОГО СИГНАЛА ИЗ ДВУХЭКСПОНЕНЦИАЛЬНОГО

Односторонний экспоненциальный сигнал можно описать следующей формулой:

$$A(n) = u(n)e^{-nT/\tau_0}, \quad (10)$$

где  $T$  – период дискретизации,  $\tau_0$  – временная константа. Здесь  $A(0) = 1$ , формула не обращается в нуль при  $n \geq 0$  и обращается в нуль для  $n < 0$ , что соответствует условию состояния покоя в начальном положении.

Формула для реализации рекурсивного фильтра, преобразующего единичный сигнал  $\delta(n)$  в односторонний экспоненциальный сигнал, получается следующим образом:

$$A(n) = u(n)e^{-nT/\tau_0} = u(n)e^{-(n-1)T/\tau_0}e^{-T/\tau_0} = u(n)e^{-T/\tau_0}A(n-1). \quad (11)$$

Здесь мы выразили каждое следующее значение импульсной характеристики через предыдущее. Введем следующее обозначение:

$$a = e^{-T/\tau_0}. \quad (12)$$

Подставив в равенство (11) в формулу (6), получим

$$A(n) = aA(n-1) + a^n\delta(n). \quad (13)$$

Любая функция, умноженная на  $\delta(n)$ , равна произведению  $\delta(n)$  и значению этой функции в нуле, поскольку остальные значения обращаются в нуль. Получаем рекурсивное уравнение

$$A(n) = \delta(n) + aA(n-1). \quad (14)$$

Двухэкспоненциальный импульс представляет собой (с точностью до коэффициента и горизонтального сдвига) разность двух односторонних экспоненциальных импульсов с разными временными константами и в то же время их свертку. Его можно описать формулой

$$F(n) = u(n) \frac{(e^{-(n+1)T/\tau_0} - e^{-(n+1)T/\tau_1})}{(e^{-T/\tau_0} - e^{-T/\tau_1})}, \quad (15)$$

где  $T$  – период дискретизации,  $\tau_0, \tau_1$  – временные константы. Здесь следует отметить, что мы выбрали формулу такого вида, что в аргументах экспонент стоит  $n + 1$ , а в знаменателе дроби выражение, определяемое только временными константами. Дело в том, что разность экспонент с аргументом нуль обращается в нуль, а мы ожидаем ненулевое значение при  $n = 0$ , поэтому мы сдвигаем разность на одно значение влево. Значение в знаменателе дроби нормирует значение в нуле на единицу; действительно,  $F(0) = 1$ .

Обозначим константу для второго одностороннего экспоненциального сигнала следующим образом:

$$b = e^{-T/\tau_1}. \quad (16)$$

Обозначим два односторонних экспоненциальных сигнала с разными временными константами следующим образом:

$$\begin{aligned} A(n) &= u(n)e^{-nT/\tau_0} = u(n)a^n, \\ B(n) &= u(n)e^{-nT/\tau_1} = u(n)b^n. \end{aligned} \quad (17)$$

Можно показать, что двухэкспоненциальный сигнал – это свертка двух односторонних экспоненциальных сигналов. Для этого преобразуем свертку двух односторонних экспоненциальных сигналов следующим образом:

$$\begin{aligned} A(n) * B(n) &= \sum_{i=0}^n (A(i)B(n-i)) = \dots \\ \dots &= \sum_{i=0}^n (u(i)u(n-i)a^i b^{n-i}) = \sum_{i=0}^n (a^i b^{n-i}) = \frac{a^{n+1} - b^{n+1}}{a-b} = \dots \\ \dots &= \frac{A(n+1) - B(n+1)}{a-b} = \frac{u(n+1)e^{-(n+1)T/\tau_0} - u(n+1)e^{-(n+1)T/\tau_1}}{e^{-T/\tau_0} - e^{-T/\tau_1}} = \dots \\ \dots &= (u(n) + \delta(n+1)) \frac{e^{-(n+1)T/\tau_0} - e^{-(n+1)T/\tau_1}}{e^{-T/\tau_0} - e^{-T/\tau_1}} = \dots \\ \dots &= F(n) + \delta(n+1) \frac{e^{-(n+1)T/\tau_0} - e^{-(n+1)T/\tau_1}}{e^{-T/\tau_0} - e^{-T/\tau_1}} = F(n). \end{aligned} \quad (18)$$

Предельный переход при  $n \rightarrow \infty$  мы рассматривать не будем.

Тогда для простоты, применив  $z$ -преобразование, а не преобразование Фурье, как в работе [2], можно записать передаточную функцию  $H(z)$  для фильтра, преобразующего единичный сигнал  $\delta(n)$  в двухэкспоненциальный, в виде произведения двух передаточных функций  $H_a(z), H_b(z)$  для преобразования  $\delta(n)$  к односторонним экспоненциальным сигналам:

$$H_a(z) = \frac{1}{(1+az^{-1})}, H_b(z) = \frac{1}{(a+bz^{-1})}, \quad (19)$$

где константы  $a, b$  выражены в формулах (12) и (16),

$$H(z) = H_a(z)H_b(z) = \frac{1}{(1+az^{-1})(1+bz^{-1})}. \quad (20)$$

Чтобы преобразовать двухэкспоненциальный сигнал обратно в единичный, следует воспользоваться обратным фильтром с передаточной функцией

$$\frac{1}{H(z)} = (1 + az^{-1})(1 + bz^{-1}). \quad (21)$$

Обозначим две передаточные функции для фильтров первого порядка следующим образом:

$$H_\alpha(z) = (1 + az^{-1}), \quad H_\beta(z) = (1 + bz^{-1}), \quad \frac{1}{H(z)} = H_\alpha(z)H_\beta(z). \quad (22)$$

Уравнение для реализации обратного фильтра второго порядка имеет следующий вид:

$$\Phi(n) = \delta(n) - (a + b)\delta(n - 1) + ab\delta(n - 2). \quad (23)$$

Его можно представить в виде свертки (последовательного применения) двух фильтров первого порядка:

$$\Phi(n) = \alpha(n) * \beta(n), \quad (24)$$

где

$$\begin{aligned} \alpha(n) &= \delta(n) - a\delta(n - 1), \\ \beta(n) &= \delta(n) - b\delta(n - 1). \end{aligned} \quad (25)$$

Поскольку аргументы экспонент в выражениях для корней и полюсов представленных выше передаточных функций отрицательны, очевидно, что представленные выше фильтры устойчивы.

Преобразования дискретных двухэкспоненциальных сигналов описаны в работах [2, 3], однако приведенных выше выкладок в них нет, так как мы вывели их самостоятельно.

### 3. ПРЕОБРАЗОВАНИЕ ЕДИНИЧНОГО СИГНАЛА В ТРАПЕЦЕИДАЛЬНЫЙ

Если сравнить первую, вторую и третью производные дискретного трапецеидального сигнала, то становится ясно, почему в работе [2] он получен именно такой комбинацией однородной второй производной трапеции и двух однородных сумматоров (интегралов), – для этого требуется меньше сложений и вычитаний и не требуется умножение.  $A, B$  – целые числа, обозначающие соответственно величины ширины ребра и меньшего основания равнобедренной трапеции по дискретной шкале времени.

Под взятием производной мы подразумеваем применение фильтра с импульсной характеристикой,

$$d(n) = \delta(n) - \delta(n - 1), \quad (26)$$

обратного обычному сумматору:

$$u(n) * d(n) = \delta(n). \quad (27)$$

Нетрудно записать выражение для трапецеидального сигнала, следуя логике схемы из работы [2]:

$$T(n) = \sum_{i=0}^A (i\delta(n - i)) + \sum_{i=A}^{A+B} (a\delta(n - i)) + \sum_{i=A+B}^{2A+B} ((2A + B - i)\delta(n - i)), \quad (28)$$

его первую дискретную производную,

$$d[n] * T(n) = \sum_{i=0}^A \delta(n - i) - \sum_{i=A+B}^{2A+B} \delta(n - i), \quad (29)$$

вторую дискретную производную,

$$d[n] * d[n] * T(n) = \delta(n) - \delta(n - A) - \delta(n - A - B) + \delta(n - 2A - B), \quad (30)$$

и третью дискретную производную,

$$\begin{aligned} d[n] * d[n] * d[n] * T[n] = & \delta(n) - \delta(n - 1) - \delta(n - A) + \delta(n - A - 1) - \dots \\ \dots - & \delta(n - A - B) + \delta(n - A - B - 1) + \delta(n - 2A - B) - \delta(n - 2A - B - 1). \end{aligned} \quad (31)$$

Под интегрированием мы подразумеваем применение фильтра с импульсной характеристикой  $u(n)$  из формул (5) или (6), т. е. применение сумматора.

Таким образом, единичный сигнал  $\delta(n)$  можно преобразовать к трапецеидальному сигналу следующими тремя последовательными преобразованиями, задаваемыми формулами (5) и (30), или их сверткой:

$$u(n) * \{d[n] * d[n] * T(n)\} * u(n) * \delta(n) = T(n). \quad (32)$$

Свертка коммутативна, это значит, что последовательность применения данных фильтров важна только для минимизации негативных последствий квантования чисел, над которыми проводятся вычисления.

#### 4. МОДЕЛИРОВАНИЕ ПРЕОБРАЗОВАНИЯ ДВУХЭКСПОНЕНЦИАЛЬНЫХ ИМПУЛЬСОВ В ТРАПЕЦЕИДАЛЬНЫЕ

Представленные выше фильтры можно смоделировать следующим образом на языке программирования Python 3 с использованием библиотеки NumPy:

```
N=400
def gen_delta():
    result = np.zeros(N)
    result[N//2] = 1
```

```

    return result
def uniexp(x, a):
    result = np.zeros(len(x))
    for i in range(1,len(x)):
        result[i] = x[i] + a*result[i-1]
    return result
def doubleexp(x, a, b):
    return uniexp(uniexp(x, a), b)
def deuniexp(x, a):
    result = np.zeros(len(x))
    for i in range(1,len(x)):
        result[i] = x[i] - a*x[i-1]
    return result
def ddtrapezoid(x,a,b):
    result = np.zeros(len(x))
    for i in range(2*a+b,len(x)):
        result[i] = x[i] - x[i-a] - x[i-a-b] + x[i-2*a-b]
    return result
def acc(x):
    result = np.zeros(len(x))
    for i in range(1,len(x)):
        result[i] = x[i] + result[i-1]
    return result
def gen_noise():
    return np.random.rand(N) - 0.5.

```

Далее с помощью библиотеки Matplotlib можно построить на графиках различные последовательности применения этих цифровых фильтров к сформированному функцией gen\_delta единичному сигналу: сначала последовательность, приводящую к нарастанию ошибки (рис. 1), а затем более предпочтительную (рис. 2).

**Рис. 1.**

**Рис. 2.**

На нижнем графике на рис. 1в базовая линия приобретает наклон, в отличие от базовой линии на рис. 2в, что на практике демонстрирует, что даже для оперирования числами с плавающей точкой с ограниченной точностью следует сначала применять интегрирующие фильтры, а потом дифференцирующие (рис. 2), а не наоборот (рис. 1), о чем было упомянуто в работе [2]. Как отмечено в работе [3], схему с интегрирующим фильтром следует сбрасывать при переполнении чисел конечной разрядности, поскольку сумматор неустойчив.

Мы смоделировали реализованную нами схему на Verilog, подав на вход тестового модуля значения нескольких двухэкспоненциальных сигналов с шумом 10% от амплитуды большего. Входные сигналы получены аналогичным образом с помощью языка программирования Python 3. Входной сигнал представляет собой последовательность чисел по 14 бит, вычисления и результат оперируют числами по 32 бита, результат приведен на рис. 3. Схема приведена на рис. 4, здесь значения для генерации входных импульсов по формулам (12) и (16) вносятся в регистры  $d$  и  $k$  с 16-ю младшими дробными битами.

**Рис. 3.**

**Рис. 4.**

Следующий фрагмент кода на Verilog демонстрирует реализацию одного из фильтров первого порядка схемы:

```

assign mult_d[MULT_WIDTH-1:0] = {{32{d[BUS_WIDTH-1]}}, d[BUS_WIDTH-1:0]}*{{32{acc2[1][BUS_WIDTH-1]}}, acc2[1][BUS_WIDTH-1:0]};
assign afterdw[BUS_WIDTH-1:0] = acc2[0] — mult_d[BUS_WIDTH+15:16];
always @(posedge clk or posedge overflow_rst) begin
    if (overflow_rst) begin
        afterd[0][BUS_WIDTH-1:0] <= {BUS_WIDTH{1'b0}};
        afterd[1][BUS_WIDTH-1:0] <= {BUS_WIDTH{1'b0}};
    end else begin
        afterd[0][BUS_WIDTH-1:0] <= afterdw[BUS_WIDTH-1:0];
        afterd[1][BUS_WIDTH-1:0] <= afterd[0][BUS_WIDTH-1:0];
    end
end.

```

## 5. ЭКСПЕРИМЕНТ ПО ПРЕОБРАЗОВАНИЮ ДВУХЭКСПОНЕНЦИАЛЬНЫХ ИМПУЛЬСОВ В ТРАПЕЦЕИДАЛЬНЫЕ С ПОМОЩЬЮ АЦП И ПЛИС ADC-SoC

Двухэкспоненциальные импульсы мы получаем, пропустив меандр со скважностью 50% (т. е. прямоугольные импульсы с равноудаленными друг от друга восходящими и ниспадающими фронтами) от генератора импульсов через интегрирующую и дифференцирующую  $CR$ - и  $RC$ -схемы с разными временными константами  $R_1C_1, R_2C_2$ . Результат представлен на рис. 5.

**Рис. 5.**

Чтобы получить значения констант  $a, b$ , мы подгоняем двухэкспоненциальный импульс (рис. 6) идеальной функцией с помощью библиотеки SciPy на языке программирования Python 3 следующим образом:

```

from scipy.optimize import curve_fit
def imp(x, *p):
    A, tau, rc, t0 = p
    return A*np.greater_equal(x-t0, np.zeros(len(x))) * (np.exp(-(x-t0)/rc) - np.exp(-(x-t0)/tau))
ind = list(adc).index(adc[100:-100].max())
p0 = [2500., 20., 10., 200]
coeff, var_matrix = curve_fit(imp, np.array(range(2000)), adc[ind-200:ind+1800], p0=p0)
a = math.exp(-1/coeff[1])
b = math.exp(-1/coeff[2]).

```

### Рис. 6.

Записав полученные значения  $a, b$  в регистры  $d$  и  $k$  нашей прошивки, проверим полученный трапецидальный сигнал (рис. 7).

### Рис. 7.

Мы наблюдаем, что, во-первых, подогнанная функция не идеально совпадает с формой импульса, а во-вторых, результирующая трапеция искажена. Это происходит потому, что входная схема нашего АЦП добавляет еще одну временную константу к нашему сигналу. Попробуем приблизить форму входного сигнала формулой

$$G(n) = u(n)(e^{-nT/\tau_0} + e^{-nT/\tau_1} - e^{-nT/\tau_2}), \quad (33)$$

где

$$\tau_0 < \tau_1 < \tau_2. \quad (34)$$

Будем называть такие импульсы трехэкспоненциальными, потому что они состоят из линейной комбинации трех экспонент с разными временными константами. Мы внесли требуемые изменения в код на Python 3 для подгонки формы входного сигнала и поиска временных констант (рис. 8).

### Рис. 8.

Мы решили остановиться на функции такого вида, потому что это наиболее простое выражение, наилучшим образом приблизительно описывающее реальные входные сигналы АЦП в нашем эксперименте из опробованных нами.

## 6. ВОССТАНОВЛЕНИЕ ЕДИНИЧНОГО СИГНАЛА ИЗ ТРЕХЭКСПОНЕНЦИАЛЬНОГО

Мы рассмотрим свертку трехэкспоненциального импульса с тремя фильтрами, обратными к односторонним экспоненциальным сигналам с совпадающими константами  $a, b, c$  из следующей формулы:

$$\begin{aligned}
a &= e^{-T/\tau_0}, \quad \alpha(n) = \delta(n) - a\delta(n-1), \quad \alpha(n) * u(n)e^{-nT/\tau_0} = \delta(n), \\
b &= e^{-T/\tau_1}, \quad \beta(n) = \delta(n) - b\delta(n-1), \quad \beta(n) * u(n)e^{-nT/\tau_1} = \delta(n), \\
c &= e^{-T/\tau_2}, \quad \gamma(n) = \delta(n) - c\delta(n-1), \quad \gamma(n) * u(n)e^{-nT/\tau_2} = \delta(n),
\end{aligned} \tag{35}$$

где

$$0 < a < b < c < 1 \tag{36}$$

в силу неравенств (34).

Упростим выражение для свертки трехэкспоненциального импульса  $G(n)$  с этими тремя фильтрами  $\alpha(n), \beta(n), \gamma(n)$ :

$$\begin{aligned}
&\alpha(n) * \beta(n) * \gamma(n) * G(n) = \dots \\
\dots &= (\alpha(n) * u(n)e^{-nT/\tau_0}) * \beta(n) * \gamma(n) + \alpha(n) * (\beta(n) * u(n)e^{-nT/\tau_1}) * \gamma(n) - \dots \\
&\dots - \alpha(n) * \beta(n) * (\gamma(n) * u(n)e^{-nT/\tau_2}) = \dots \\
&\dots = \beta(n) * \gamma(n) + \alpha(n) * \gamma(n) - \alpha(n) * \beta(n).
\end{aligned} \tag{37}$$

Далее распишем одно из получившихся слагаемых:

$$\begin{aligned}
\alpha(n) * \beta(n) &= (\delta(n) - a\delta(n-1)) * (\delta(n) - b\delta(n-1)) = \dots \\
\dots &= \delta(n) - a\delta(n-1) - b\delta(n-1) + ab\delta(n-2).
\end{aligned} \tag{38}$$

Теперь подставим аналогичные выражения для каждого слагаемого и упростим выражение:

$$\begin{aligned}
&\beta(n) * \gamma(n) + \alpha(n) * \gamma(n) - \alpha(n) * \beta(n) = \dots \\
\dots &= \delta(n) - (b+c)\delta(n-1) + bc\delta(n-1) + \delta(n) - (a+c)\delta(n-1) + ac\delta(n-2) - \dots \\
&\dots - \delta(n) + (a+b)\delta(n-1) - ab\delta(n-2) = \dots \\
&\dots = \delta(n) - 2c\delta(n-1) + (bc+ac-ab)\delta(n-2).
\end{aligned} \tag{39}$$

Мы получили сигнал, отличный от единичного. Найдем рекурсивное уравнение обратного фильтра к полученному сигналу. Для этого запишем передаточную функцию,

$$H(z) = 1 - 2cz^{-1} + (bc + ac - ab)z^{-2}, \tag{40}$$

и обратную к ней,

$$\frac{1}{H(z)} = \frac{1}{1 - 2cz^{-1} + (bc + ac - ab)z^{-2}}. \tag{41}$$

Обратный БИХ-фильтр описывается рекурсивным уравнением

$$R(n) = \delta(n) + 2cR(n-1) - (bc + ac - ab)R(n-2). \quad (42)$$

Получается, что

$$\alpha(n) * \beta(n) * \gamma(n) * R(n) * G(n) = \delta(n). \quad (43)$$

Найдем полюса и условия устойчивости фильтра  $R(n)$ , которым мы собираемся дополнять  $\alpha(n) * \beta(n) * \gamma(n)$ , чтобы получить комбинированный обратный фильтр к трехэкспоненциальному сигналу. Полюса передаточной функции – это решение квадратного уравнения

$$z^2 - 2cz + (bc + ac - ab) = 0. \quad (44)$$

Дискриминант

$$D = c^2 - bc - ac + ab = (c - a)(c - b) \quad (45)$$

положителен в силу неравенств (36), а значит, оба корня действительны:

$$z = c \pm \sqrt{(c - a)(c - b)}. \quad (46)$$

Обозначим меньший и больший корни соответственно

$$\begin{aligned} z_1 &= c - \sqrt{(c - a)(c - b)}, \\ z_2 &= c + \sqrt{(c - a)(c - b)}. \end{aligned} \quad (47)$$

В силу неравенств (36) меньший корень, очевидно, меньше единицы, поэтому условие устойчивости фильтра  $R(n)$  сводится к неравенству

$$c + \sqrt{(c - a)(c - b)} < 1. \quad (48)$$

Из этого неравенства выразим  $c$  через остальные константы:

$$\sqrt{(c - a)(c - b)} < 1 - c, \quad (49)$$

$$c^2 - (a + b)c + ab < 1 - 2c + c^2, \quad (50)$$

$$(2 - a - b)c < 1 - ab, \quad (51)$$

$$c < \frac{1 - ab}{2 - a - b}. \quad (52)$$

Как оказалось, устойчивость искомого фильтра определяется только соотношением трех временных констант,  $\tau_0, \tau_1, \tau_2$ , или однозначно определяемых ими величин  $a, b, c$ .

Для наглядности и простоты реализации требуемого нам фильтра разложим его на пять фильтров первого порядка:

$$\alpha(n) = \delta(n) - a\delta(n-1), \quad (53)$$

$$\beta(n) = \delta(n) - b\delta(n - 1), \quad (54)$$

$$\gamma(n) = \delta(n) - c\delta(n - 1), \quad (55)$$

$$\epsilon(n) = \delta(n) + z_1\epsilon(n - 1), \quad (56)$$

$$\zeta(n) = \delta(n) + z_2\zeta(n - 1). \quad (57)$$

Все сказанное выше подразумевает следующее равенство:

$$\alpha(n) * \beta(n) * \gamma(n) * \epsilon(n) * \zeta(n) * G(n) = \delta(n). \quad (58)$$

## 7. ЭКСПЕРИМЕНТ ПО ПРЕОБРАЗОВАНИЮ ТРЕХЭКСПОНЕНЦИАЛЬНЫХ ИМПУЛЬСОВ В ТРАПЕЦЕИДАЛЬНЫЕ С ПОМОЩЬЮ АЦП И ПЛИС ADC-SoC

Мы модифицировали нашу цифровую схему на Verilog. Теперь она состоит из трапеции и пяти фильтров первого порядка. Здесь величины  $a, b, c, z_1, z_2$  соответствуют величинам  $d, k, l, m, n$  на графической схеме (рис. 9).

**Рис. 9.**

Константы  $a, b, c$  мы нашли, подгоняя форму входного сигнала с помощью трехэкспоненциального импульса (рис. 8), а  $z_1, z_2$  вычислили по формуле (47)- Мы решили, что в тех случаях, когда больший полюс  $z_2$  достигает единицы, вместо него подставлять значение, меньшее единицы, в шестнадцатом бите после запятой, чтобы обеспечить устойчивость фильтра  $\zeta(n)$  из формулы (57). Для нашего сигнала больший полюс  $z_2$  нашего фильтра оказался больше единицы только в пятом десятичном знаке после запятой, поэтому подмена значения не должна привести к сильному искажению трапеции. На крупном плане получившегося импульса (рис. 10) мы не видим искажений, однако они заметны на общем плане (рис. 11).

**Рис. 10.**

**Рис. 11.**

## 8. ДЕТЕКТИРОВАНИЕ МОДИФИЦИРОВАННЫХ ИМПУЛЬСОВ

Метод, описанный в работе [4], мы дополнили восстановлением базовой линии из работы [5]. Разность двух скользящих средних по  $B$  значений с задержкой  $A + B$  между ними преобразует трапецеидальные импульсы к биполярным импульсам с базовой линией около нуля. В какой-то момент эта разность будет разностью между средним вершины (малого основания трапеции) импульса и средним базовой линии перед трапецеидальным импульсом. Максимальное положительное значение такого биполярного импульса и будет искомой

амплитудой. Однако для простоты реализации скользящее среднее мы заменили скользящей суммой.

Таким образом, мы дополнили схему [4] сдвиговым регистром на  $2B + A$  значения сигнала трапецеидальных импульсов, а также сдвиговым регистром на  $A + B$  значений разности соответствующих сумм (скользящих средних), вычисленных на первом сдвиговом регистре. Искомое значение в момент перехода триггера “dff” в нулевое значение – это максимальное значение, хранящееся во втором сдвиговом регистре.

## 9. ВЫВОДЫ

В результате выполненных работ был построен амплитудный спектр (рис. 12) собранных значений амплитуды, представляющий собой гистограмму с шагом, равным единице целых значений скользящих сумм амплитуд трапецеидальных сигналов, подогнано нормальное распределение и получено соответствующее ему значение  $\sigma$ . Отношение  $\sigma$  к положению вершины пика по горизонтальной оси равно 0.0013926, т. е. полная ширина пика на полувысоте составляет 0.25852% от средней амплитуды преобразованных импульсов.

**Рис. 12.**

## ФИНАНСИРОВАНИЕ

Работа выполнена при поддержке Министерства науки и высшего образования Российской Федерации (соглашение №075-10-2025-153 от 05 июня 2025 г.).

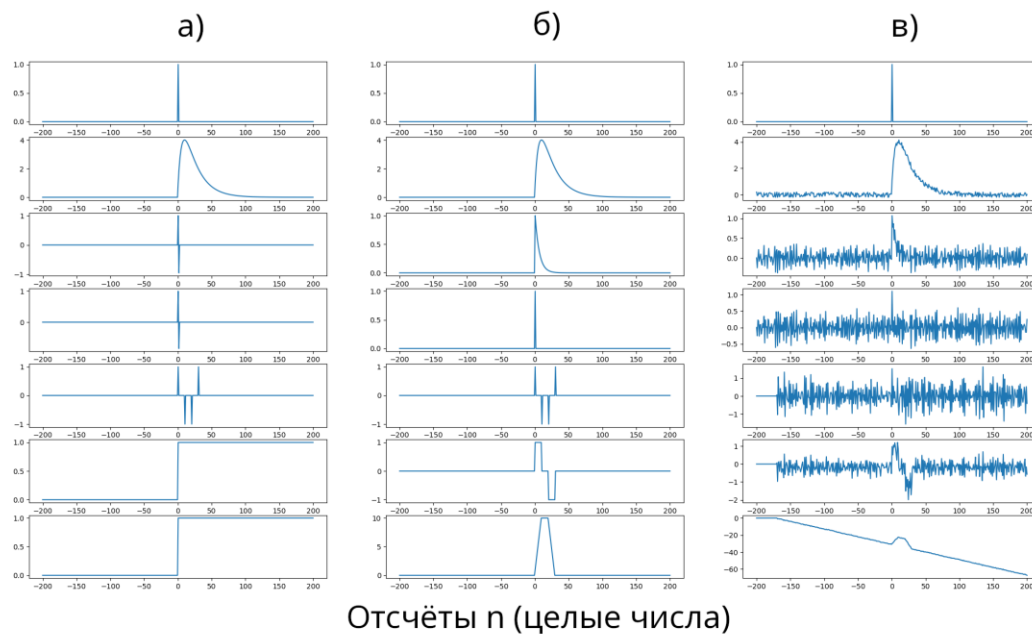
## СПИСОК ЛИТЕРАТУРЫ

1. *A. Оппенгейм, P. Шафер.* Цифровая обработка сигналов. Москва: Техносфера, 2006.
2. *Guo-Qiang Zeng, Jian Yang, Ming-Fu Yu, Kai-Qi Zhang, Qing Ge, Liang-Quan Ge.* // Chin. Phys. C. 2017. V. 41. № 1. P. 016102. <https://doi.org/10.1088/1674-1137/41/1/016102>
3. *Valentin T. Jordanov.* // Nucl. Instrum. Methods Phys. Res., Sect. A. 2016. V. 805. P. 63. <https://doi.org/10.1016/j.nima.2015.07.040>
4. *Jordanov V.T., Hall D.L.* // IEEE Xplore. 2002. P. 140. <https://doi.org/10.1109/NSSMIC.2002.1239285>
5. Lab 4 - Baseline Restorer. <https://gitlab.sci-compiler.com/examples/simulation/lab-4-baseline-restorer>

## ПОДПИСИ К РИСУНКАМ

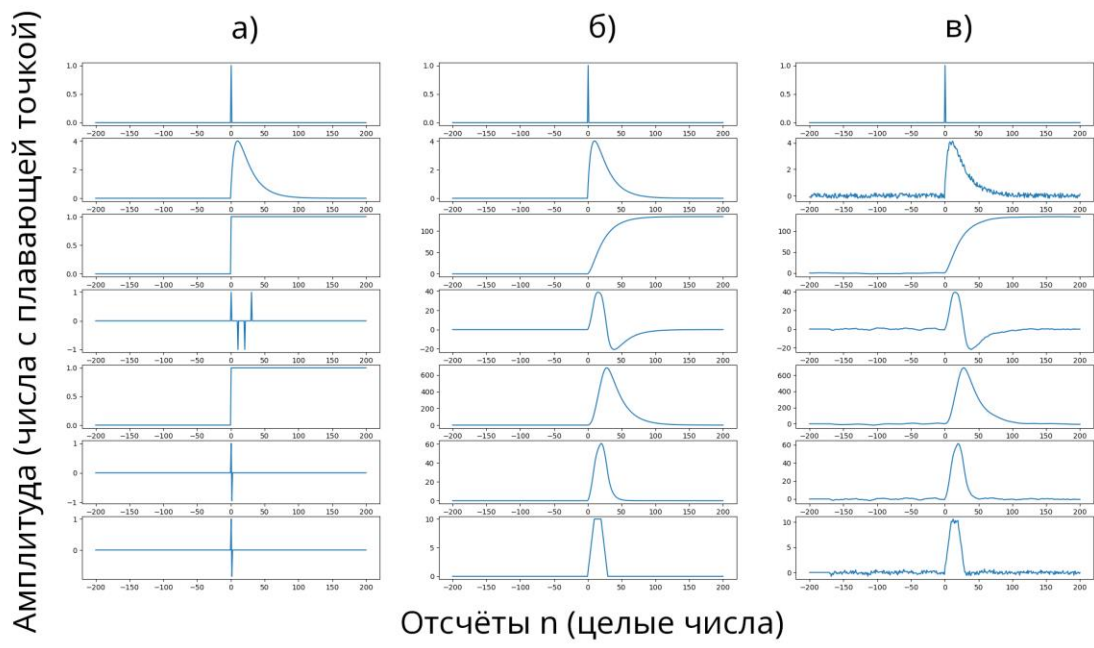
- Рис. 1.** Три серии графиков для случая численно-неустойчивой последовательности: **а** – импульсные характеристики последовательности преобразований, **б** – последовательность результатов применения соответствующих фильтров из предыдущей серии, **в** – в отличие от предыдущей серии, на втором графике к двухэкспоненциальному сигналу добавлен случайный шум величиной 10% от амплитуды пика.
- Рис. 2.** То же, что и на рис.1, но для случая численно-устойчивой последовательности.
- Рис. 3.** Снимок экрана результата моделирования цифровой схемы: *1* – серия входных двухэкспоненциальных импульсов, уменьшающихся вдвое один за другим с шумом 10% от большего, *2* – результат применения сумматора, *3* – результат последующего применения фильтра  $d[n] * d[n] * T[n]$ , *4* – результат последующего применения второго сумматора, *5* – результат последующего применения фильтра  $\alpha(n)$ , *6* – результат последующего применения фильтра  $\beta(n)$ .
- Рис. 4.** Графическое представление цифровой схемы на Verilog
- Рис. 5.** Сигнал, регистрируемый АЦП.
- Рис. 6.** Подгонка входного сигнала с помощью двухэкспоненциального импульса с целью нахождения временных констант.
- Рис. 7.** Трапецеидальный сигнал, полученный после применения соответствующих фильтров для двухэкспоненциального сигнала.
- Рис. 8.** Подгонка входного сигнала с помощью трехэкспоненциального импульса с целью нахождения временных констант.
- Рис. 9.** Графическое представление модифицированной цифровой схемы на Verilog
- Рис. 10.** Трапецеидальный сигнал, полученный применением соответствующих пяти фильтров первого порядка для трехэкспоненциального сигнала.
- Рис. 11.** Трапецеидальный сигнал, полученный применением соответствующих пяти фильтров первого порядка для трехэкспоненциального сигнала, общий план.
- Рис. 12.** Амплитудный спектр (синий), подогнанное нормальное распределение (оранжевый).

Амплитуда (числа с плавающей точкой)

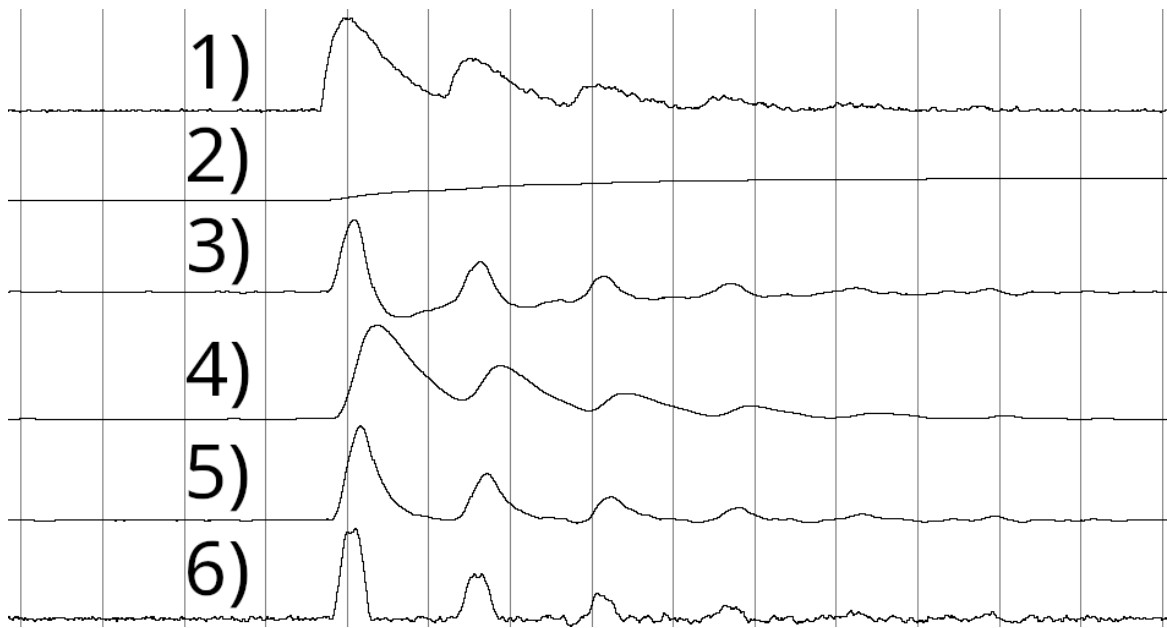


Отсчёты n (целые числа)

Рис. 1.



**Рис. 2.**



**Рис. 3.**

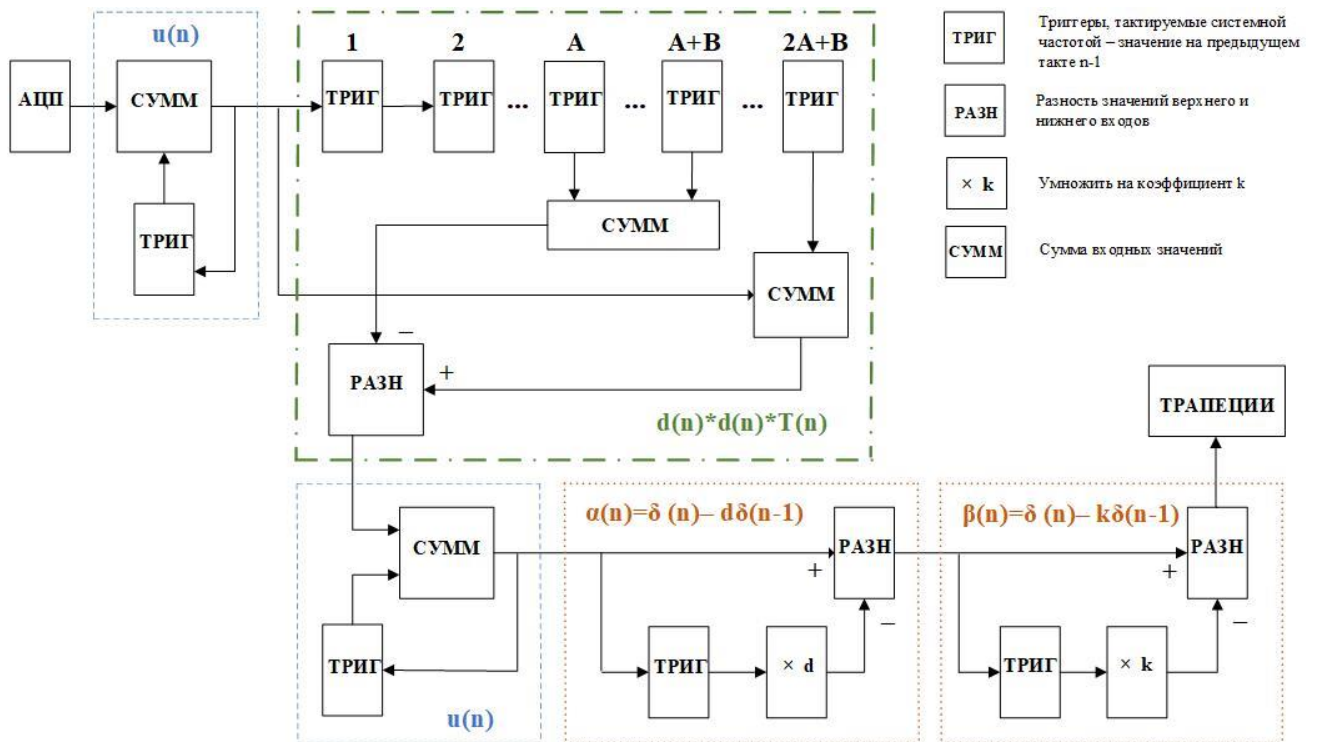
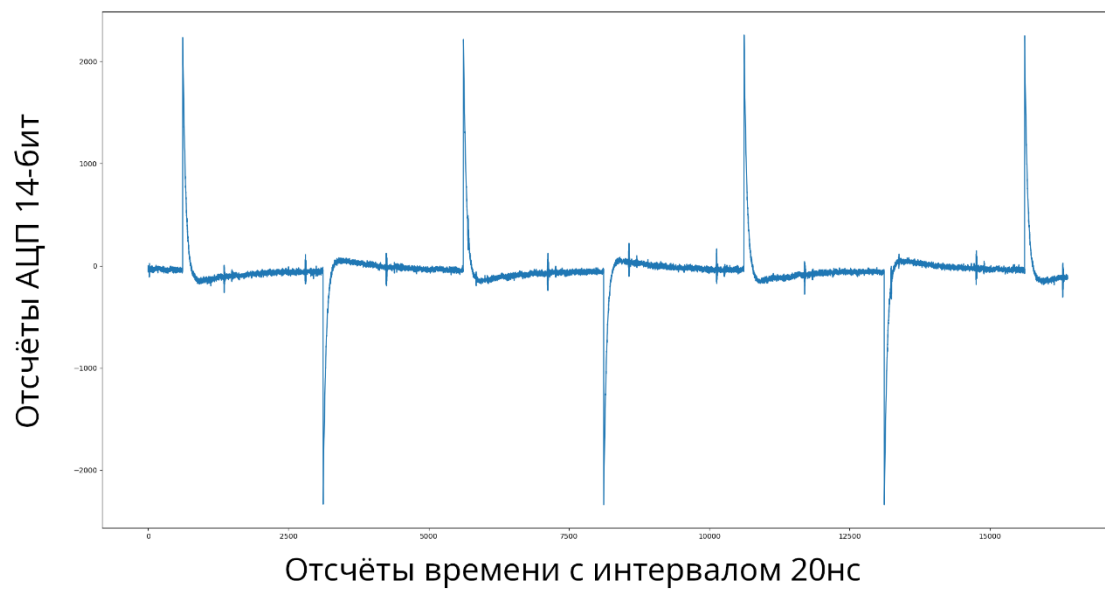
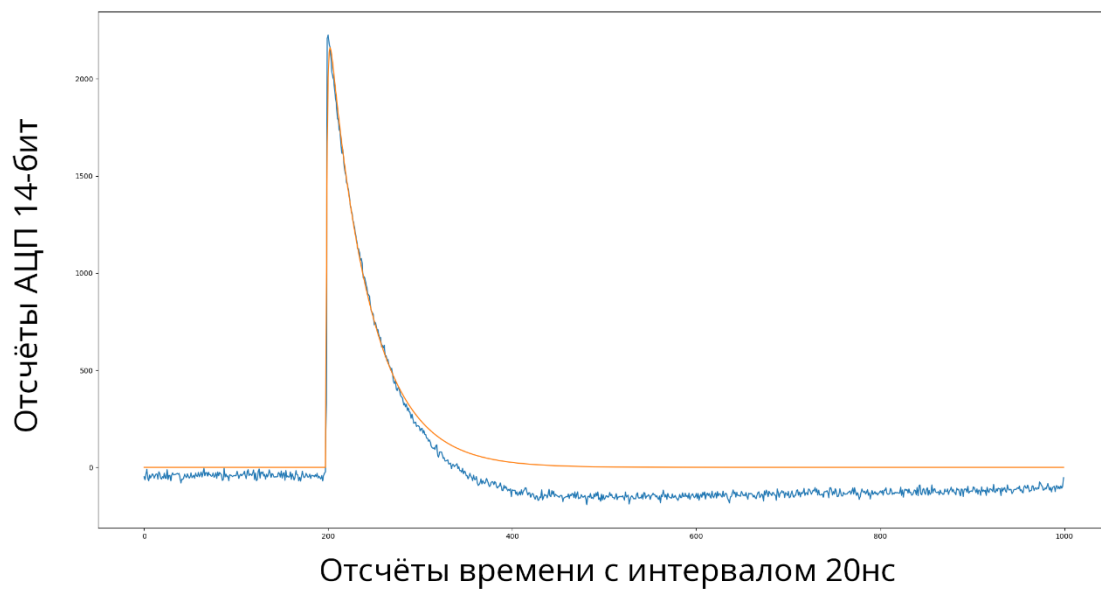


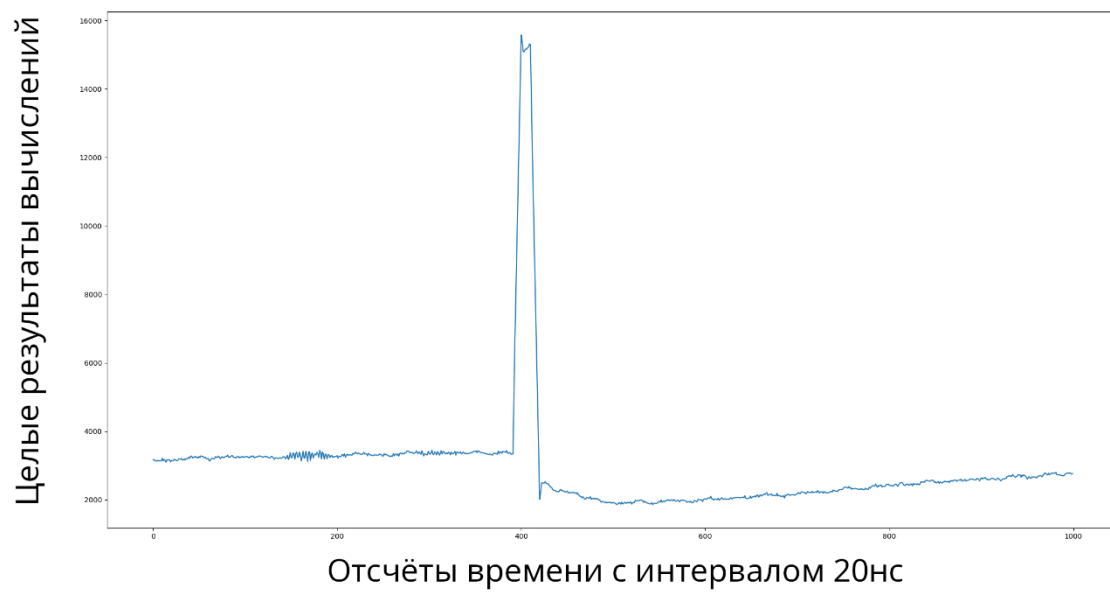
Рис. 4.



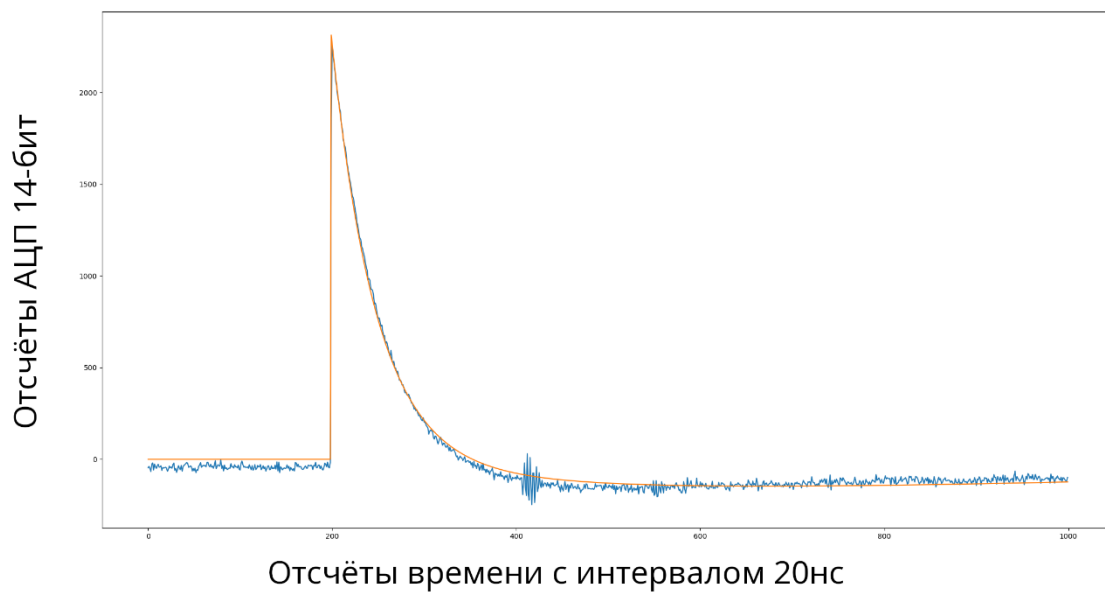
**Рис. 5.**



**Рис. 6.**



**Рис. 7.**



**Рис. 8.**

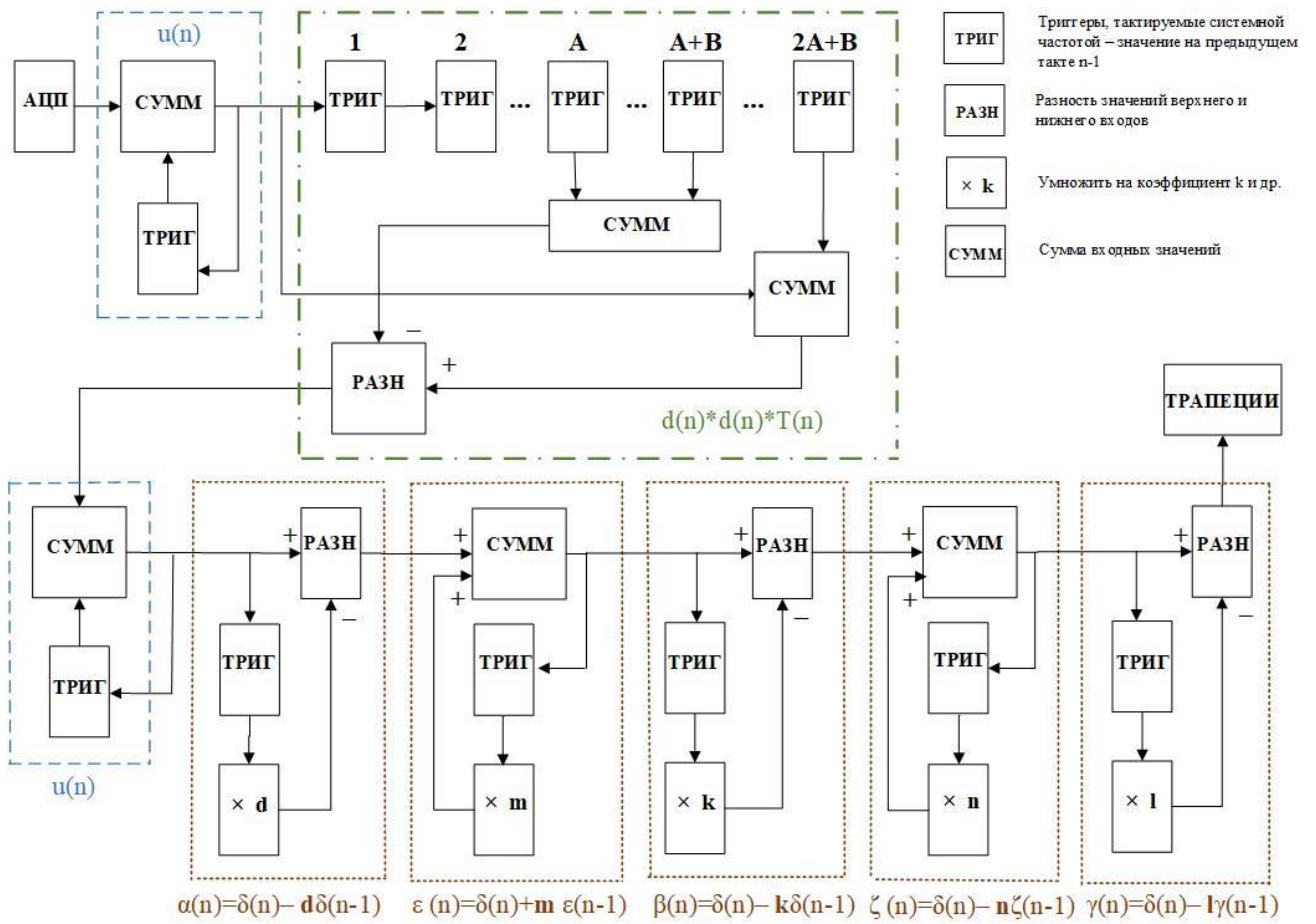
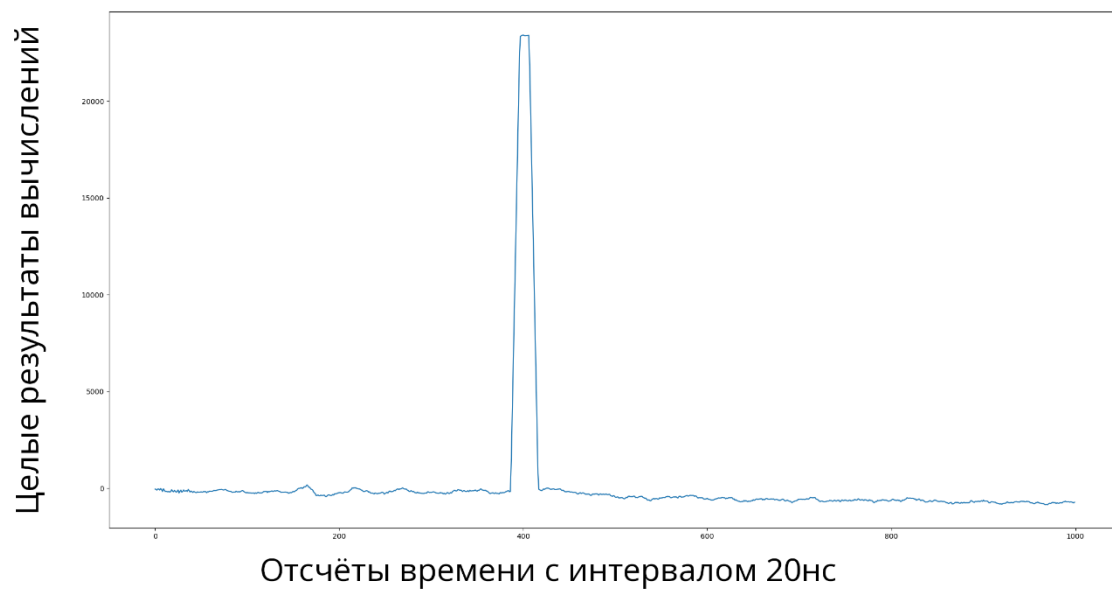
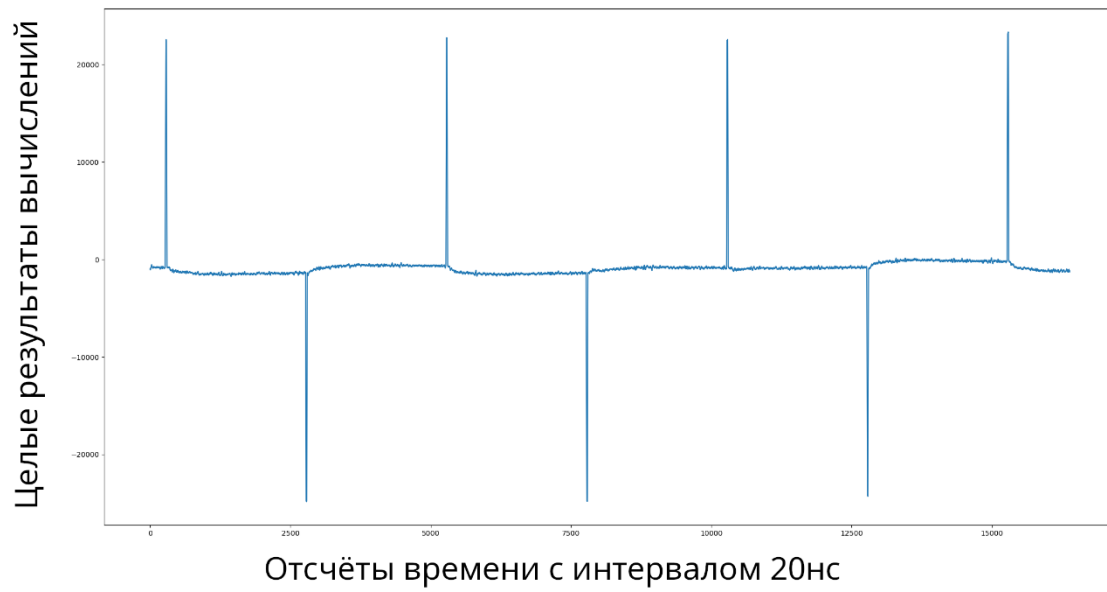


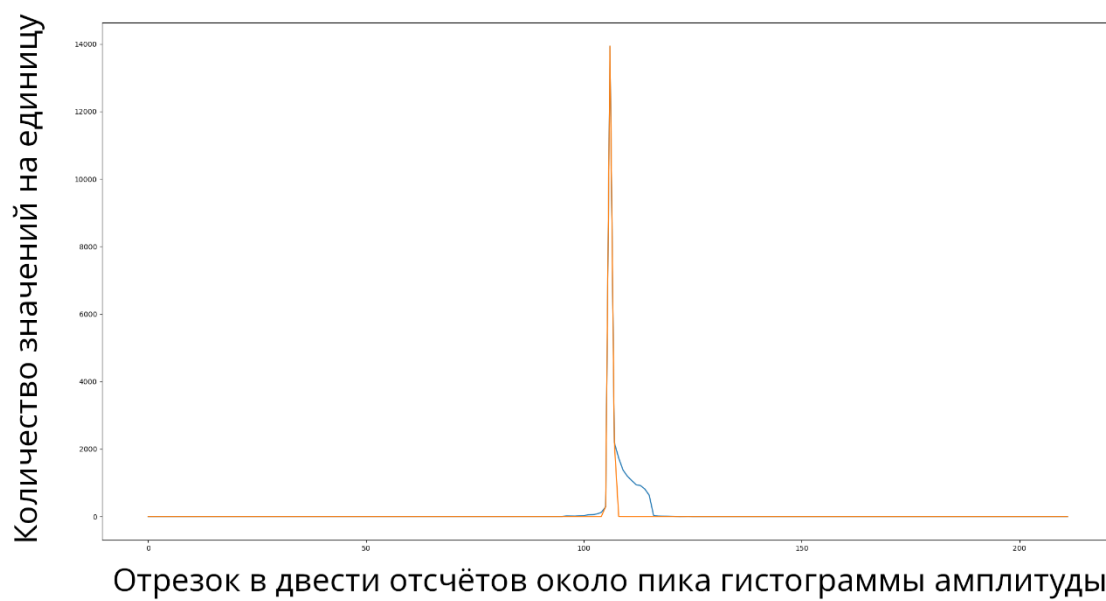
Рис. 9



**Рис. 10.**



**Рис. 11.**



**Рис. 12.**