

## Алгоритм наименьших квадратов для калибровки Time Interleaved ADC

### 1. Описание алгоритма

В очень упрощенном приближении чередование во времени заключается в мультиплексировании во времени выходов параллельного массива из  $M$  идентичных АЦП (как показано на рис. 1) для достижения более высокой суммарной частоты дискретизации  $f_s$  (интервал дискретизации  $T_s = 1/f_s$ ). При этом каждый АЦП массива в действительности осуществляет выборку (и преобразование) сигнала на меньшей частоте,  $f_s/M$

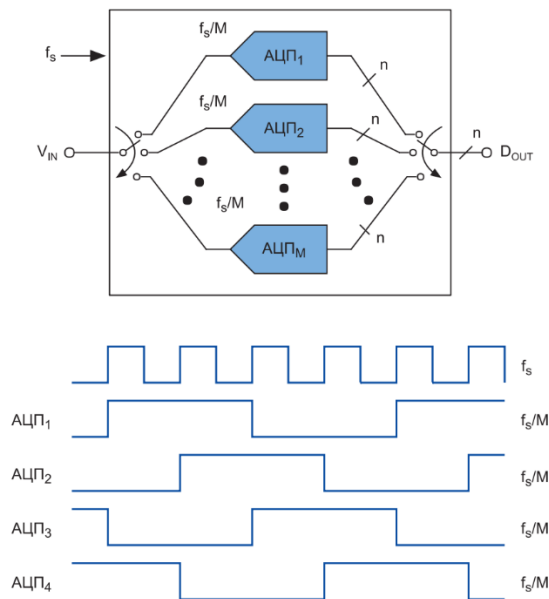


Рисунок 1. Массив из  $M$   $n$ -разрядных АЦП с чередованием во времени. (Частота дискретизации каждого АЦП равна  $f_s/M$ , а общая частота дискретизации всей системы  $f_s$ )

За референсный (эталонный) сигнал берем выход суб-АЦП<sub>0</sub>. Тогда для  $i$ -го суб-АЦП эталонный сигнал запишется как

$$y_{ri}(n) = y_0(n) * h_{ri}(k), \quad [1](15)$$

где  $y_0(n)$  – выход АЦП<sub>0</sub>,  $h_{ri}(k)$  – импульсная характеристика фильтра дробной задержки

Частотная характеристика фильтра дробной задержки запишется как:

$$H_{ri}(j\omega) = e^{-j\omega T_s \frac{i}{M}}, \quad [1](16)$$

Где  $\frac{i}{M}$  – дробная часть от частоты дискретизации  $F_s$ .  $M$  – количество суб-АЦП,  $i$  – номер суб-АЦП. Соответственно для системы, состоящей из 4 суб-АЦП, задержка для 1-го суб-АЦП равна  $1/4 = 0.25 F_s$ , для 2-го суб-АЦП  $= 2/4 = 0.5 F_s$ , для 3-го  $= 3/4 = 0.75 F_s$ .

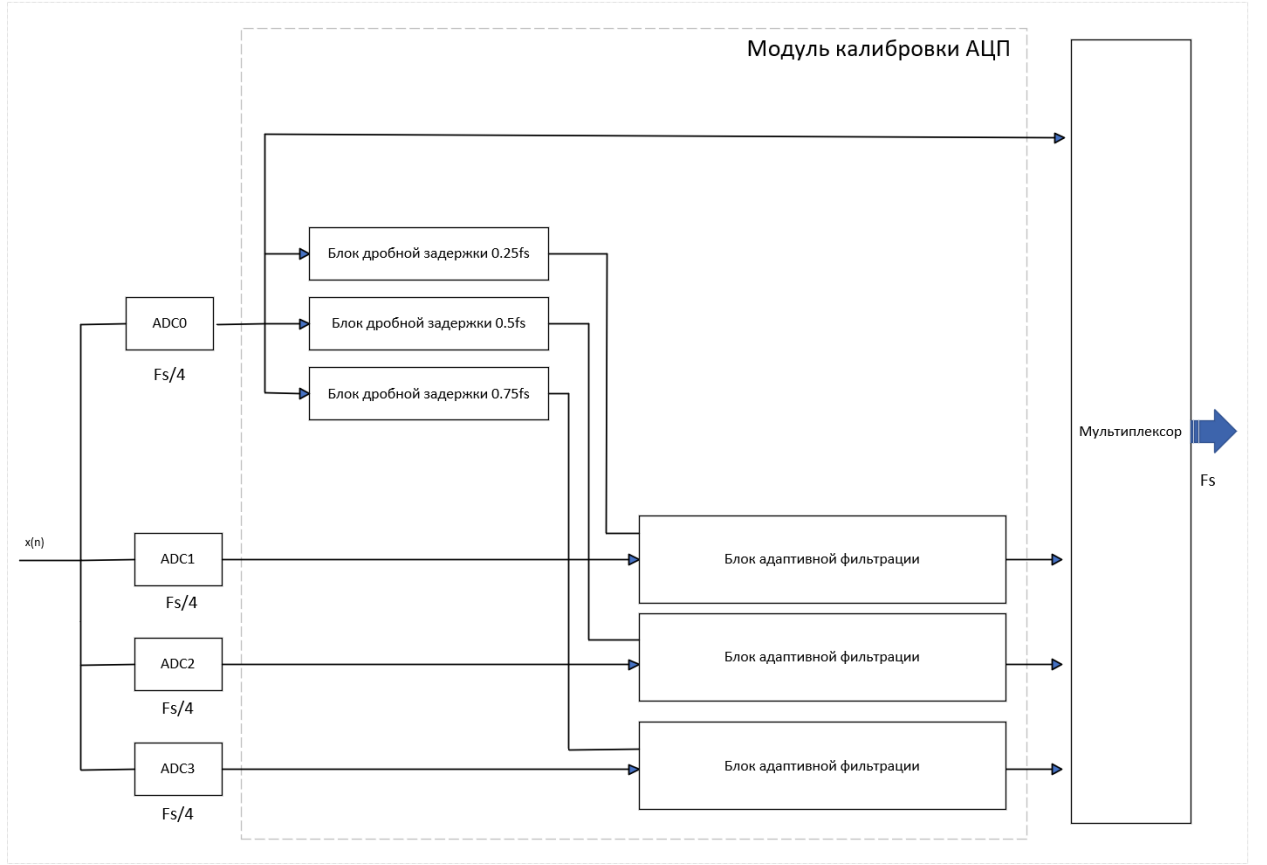


Рисунок 2. Предлагаемая блок-схема алгоритма калибровки, для  $M = 4$

Разностное уравнения для фильтра дробной задержки:

$$y_{ri}(n) = \sum_{k=1}^L h_{ri}(k) * y_0(n),$$

Где  $L = 73$  – порядок фильтра дробной задержки

Ошибка между эталонным сигналом (выходом фильтра дробной задержки) и выходом адаптивного фильтра равна:

$$e_i(n) = y_0(n) * h_{ri}(k) - y_{icalo} * h_{\Delta g_i \Delta \tau_i \Delta w_i}, \quad [1](17)$$

Где  $h_{\Delta g_i \Delta \tau_i \Delta w_i}$  – коэффициенты адаптивного фильтра,  $y_{icalo}$  – входные отсчеты с  $i$ -го суб-АЦП.

Перепишем уравнение 17:

$$e_i(n) = y_{ri}(n) - \sum_{k=1}^N y_{icalo}(n - k + 1 + \frac{N}{2}) * h_{\Delta g_i \Delta \tau_i \Delta w_i} \quad [1](18)$$

Далее найдем значение  $\sum e_i(n)^2$  с помощью метода наименьших квадратов. Целевая функция запишется как  $J = \sum e_i(n)^2 = e^T e$

$$\begin{aligned} J &= (y_{ri} - y_{icalo} * h_{\Delta g_i \Delta \tau_i \Delta w_i})^T * (y_{ri} - y_{icalo} * h_{\Delta g_i \Delta \tau_i \Delta w_i}) \\ &= y_{ri}^T * y_{ri} - h_{\Delta g_i \Delta \tau_i \Delta w_i}^T * y_{icalo}^T * y_{ri} - y_{ri} * y_{icalo} * h_{\Delta g_i \Delta \tau_i \Delta w_i} + h_{\Delta g_i \Delta \tau_i \Delta w_i}^T \\ &\quad * y_{icalo}^T * y_{icalo} * h_{\Delta g_i \Delta \tau_i \Delta w_i} \end{aligned}$$

Найдем производную от  $h_{\Delta g_i \Delta \tau_i \Delta w_i}$

$$\begin{aligned} \frac{dj}{dh_{\Delta g_i \Delta \tau_i \Delta w_i}} \Big|_{h_{\Delta g_i \Delta \tau_i \Delta w_i} = \widehat{h_{\Delta g_i \Delta \tau_i \Delta w_i}}} &= -2 * y_{icalo}^T * y_{ri} + 2 * y_{icalo}^T * y_{icalo} * h_{\Delta g_i \Delta \tau_i \Delta w_i} = 0 \\ y_{icalo}^T * y_{icalo} * \widehat{h_{\Delta g_i \Delta \tau_i \Delta w_i}} &= y_{icalo}^T * y_{ri} \end{aligned} \quad (35)$$

Выразим  $h_{\Delta g_i \Delta \tau_i \Delta w_i}$

$$\widehat{h_{\Delta g_i \Delta \tau_i \Delta w_i}} = (y_{icalo}^T * y_{icalo})^{-1} * y_{icalo}^T * y_{ri} \quad [1](36)$$

Где  $y_{icalo}$  – матрица входного сигнала  $i$ -го суб-АЦП,  $y_{ri}$  – вектор-столбец эталонного сигнала,

$h_{\Delta g_i \Delta \tau_i \Delta w_i}$  – вектор коэффициентов адаптивного фильтра

Матрица  $y_{icalo}$  запишется как:

$$y_{icalo} = \begin{bmatrix} y_{icalo} (1 + \frac{N}{2}) & \cdots & y_{icalo} (N + \frac{N}{2}) \\ \vdots & \ddots & \vdots \\ y_{icalo} (L_s + \frac{N}{2}) & \cdots & y_{icalo} (L_s + N + \frac{N}{2}) \end{bmatrix} \quad [1](20)$$

Вектор-столбец  $y_{ri}$  запишется как:

$$y_{ri} = \begin{bmatrix} y_{icalo} (1 + \frac{N}{2}) \\ \vdots \\ y_{icalo} (L_s + \frac{N}{2}) \end{bmatrix} \quad [1](21)$$

Уравнение (36) решается с помощью метода Крамера. Далее приводится кол-во умножений для коррекции сигнала [1] стр 12.

$y_{icalo}^T * y_{icalo}$  – требует  $N L_s N$  кол-ва умножений

$y_{icalo}^T * y_{ri}$  – требует  $N L_s$  кол-ва умножений

Правило Крамера требует  $N! (N + 1) N$  кол-ва умножений

Тогда количество умножений для коррекции  $i$ -го канала рассчитывается как:

$$\begin{aligned} N! (N + 1) N + N L_s N + N L_s + (M - 1) L L_s &= N L_s (N + 1) + N! (N + 1) N + (M - 1) L L_s \\ &= N + 1 (N L_s + N! N) + (M - 1) L L_s = \\ &= N + (N + 1) * (L_s + N!) + (M - 1) L * L_s = 128.1 \times 10^3 \quad [1](\text{стр. 12}) \end{aligned}$$

Где  $L$  – 73 порядок фильтра дробной задержки,

$N$  – 5 порядок адаптивного фильтра [1] стр 12.

Методом подбора узнаем, что  $L_s = 500$

$$5 + (5 + 1) * (500 + 120) + (4 - 1) * 73 * 500 = 128100$$

Следовательно размер матрицы входного сигнала  $y_{icalo}$  в выражении [1](20) имеет размер 500x5. Взяв исходную формулу расчета кол-ва умножений:

$$\begin{aligned} N! (N + 1) N + N L_s N + N L_s + (M - 1) L L_s \\ &= 5! * (5 + 1) * 5 + 5 * 500 * 5 + 5 * 500 + (4 - 1) * 73 * 500 \\ &= 3600 + 12500 + 2500 + 109500 \end{aligned}$$

Занесем эти данные в таблицу 1.

Таблица 1. Количество умножений, необходимое для корректировки  $i$ -го канала

Вид операции	Кол-во умножений
Решение уравнений методом Крамера	3600
Умножение матриц $y_{icalo}^T * y_{icalo}$	12500
Умножение матрицы на вектор $y_{icalo}^T * y_{ri}$	2500
Получение отсчетов эталонного сигнала	109500

## 2. Результаты моделирования алгоритма

Результаты моделирования алгоритма приведены на рис. 3 для системы АЦП 12 бит, 4 канала, частота дискретизации 4 ГГц. SNR входного сигнала 65 дБ. Ошибки time skew, gain приведены в таблице 2.

Таблица 2. Ошибки для системы Time-interleaving АЦП

Номер суб-АЦП	Offset	Gain	Time skew
1	0	0	0
2	0	1.4	0.2
3	0	1.4	0.3
4	0	1.4	0.4

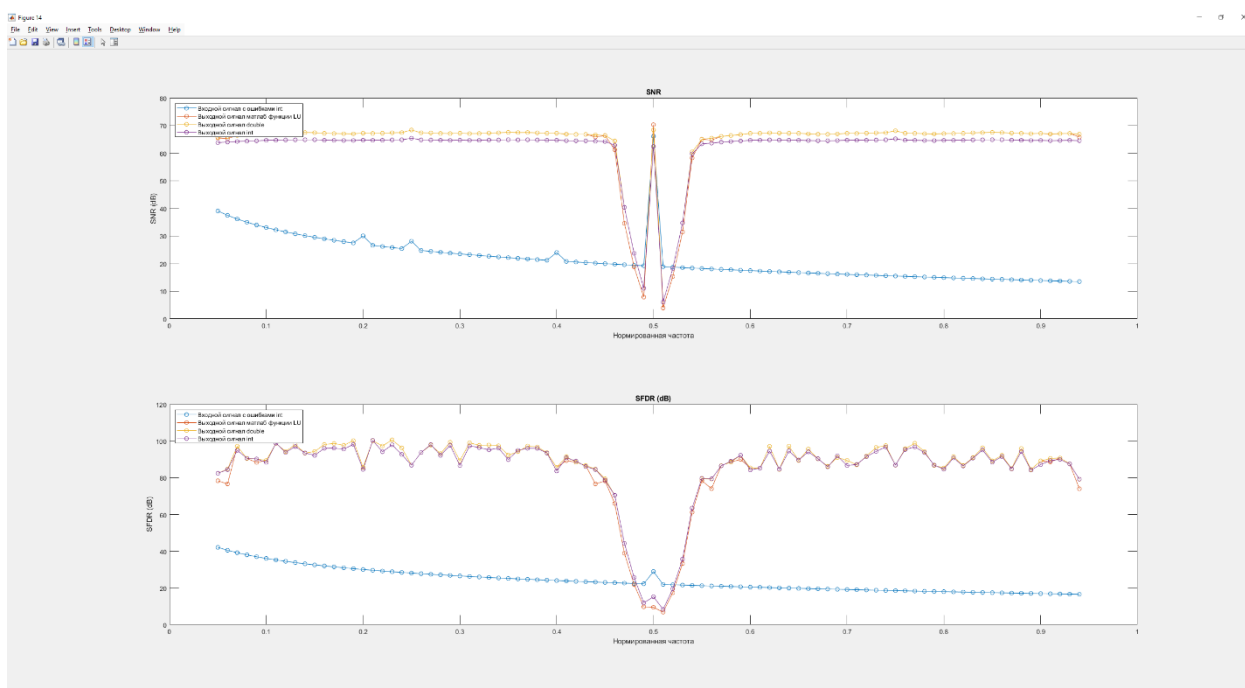


Рисунок 3. Результаты моделирования

В статье [1] SNR входного сигнала равен 60 дБ, а после калибровки SNR выходного сигнала равно 65 дБ [1] стр.11, то есть SNR дополнительно улучшается на 5дБ. Данный алгоритм не улучшает SNR, а возвращает исходное отношение сигнал/шум. Может в статье урезали полосу входных суб-АЦП и тем самым получили прирост в SNR, т.к моделируются частоты (рисунок 7 (от 50 МГц – 420 МГц), рисунок 8 (550 МГц – 920 МГц), рисунок 9 (1.05 ГГц – 1.42 ГГц)), то есть частоты возле границ зон Найквиста не моделируются. В данном алгоритме образуются провалы в SNR и SFDR возле границ зон Найквиста. Я так понимаю это происходит из-за ограничения эталонного сигнала, т.к его мы получаем с помощью фильтра дробной задержки, который работает в 1-ой зоне Найквиста.

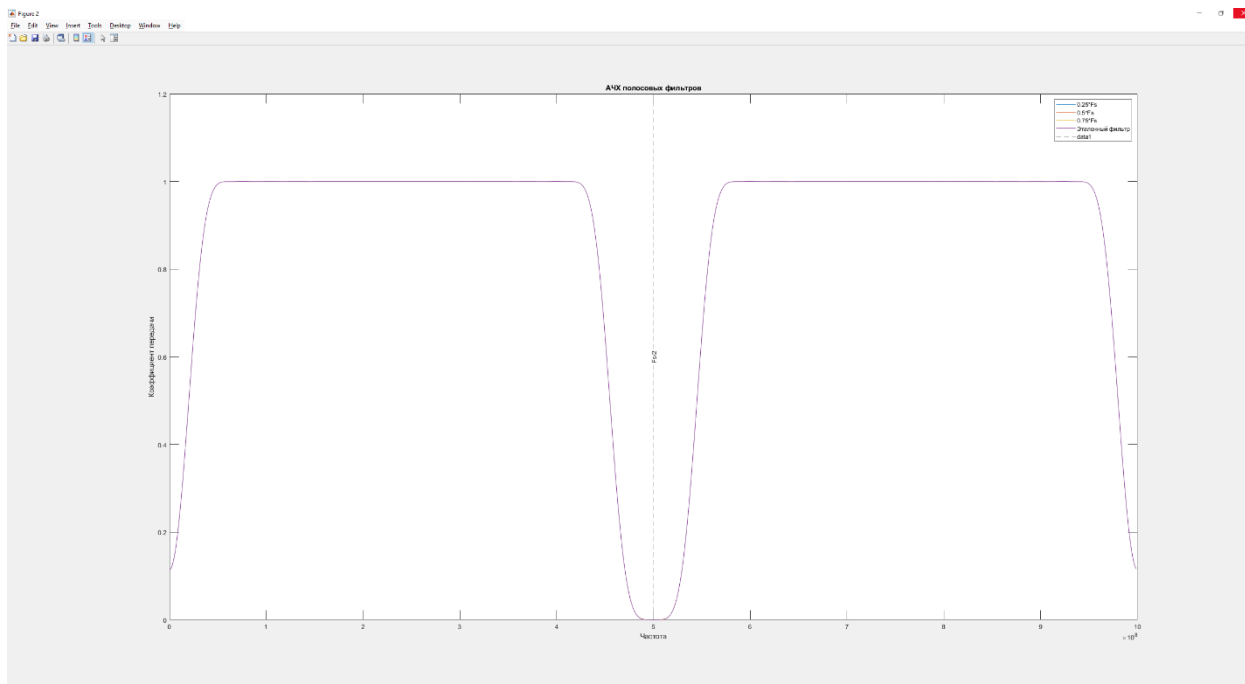


Рисунок 4. АЧХ фильтров дробной задержки

Как видно из рисунка 3, результаты SNR по форме схожи с АЧХ фильтров дробной задержки.

Параметры моделирования алгоритма указаны не полностью в таблице 4 [1]. Известно, что нормированная частота составляет 0.167 fs и 4-ёх канальный суб-АЦП. Так же указано улучшение по SNR и SFDR. Поэтому мои результаты могут быть как лучше, так и хуже.

Таблица 3. Сравнение результатов

	Улучшение SNR, (дБ)	Улучшение SFDR, (дБ)
Алгоритм статьи	38,36	55,75
Мой алгоритм	36.5	50.1



Рисунок 5. Результаты моделирования на 0.167 fs

Длина сигнала 2к семплов, нормированная частота 0.167fs. Результаты моделирования занесены в таблицу 3.

### Про размер матрицы

Цель авторов [1] пропустить через систему 2к семплов, чтобы достичь максимального SNR и SFDR. Об этом говорится в их новой статье [3] и статье, на которую они ссылаются [4]

TABLE IV  
THE SNDR AND SFDR AFTER FOR THE DIFFERENT VALUE OF  $L$

The value of $L$	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384
SNDR(dB)	2.07	7.18	14.23	15.52	16.57	23.99	28.70	34.22	42.29	44.52	66.75	66.72	66.77	66.69
SFDR(dB)	5.23	10.52	17.27	18.59	19.57	27.03	31.74	37.20	45.39	47.43	69.28	69.21	69.24	69.21

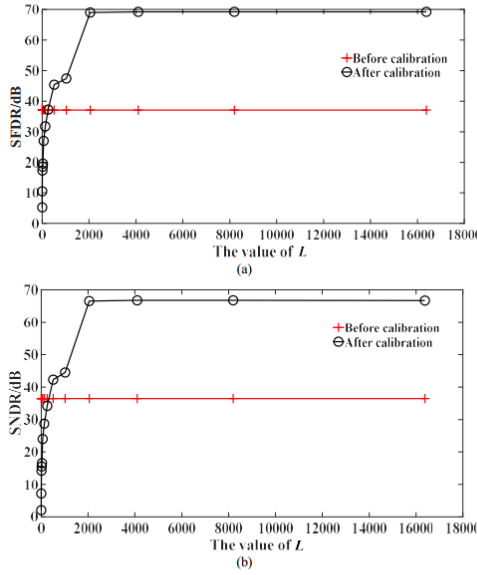


FIGURE 8. The performance of the TI-ADCs with the different values of  $L$ . (a) the SNDR before and after calibration; (b) the SFDR before and after calibration.

window function as the first-order FIR derivative filter for time skew calibration.

Our statistical knowledge-based calibration method exhibits larger and uncertain errors with insufficient samples. Therefore, selecting an appropriate sample size is a crucial issue. Derivative calculations are performed using digital filters and the interleaved and reorganized output  $y[n]$ . Therefore, the input signal frequency is constrained within the range of  $(0, 0.388f_s]$ . Fig. 7 shows the SNDR and SFDR before and after calibration for the different number of samples when the number of filter taps is 21 and the frequency of input is  $0.388f_s$ . The calibration algorithm adopted is the method proposed in Section III-B.

Additionally, TABLE IV presents the calibrated SNDR and SFDR under various values of  $L$ . It can be observed that when condition  $L \geq 512$  is met, the algorithm begins to exhibit a certain calibration effect. Furthermore, when condition  $L \geq 2048$  is satisfied, the calibrated SNDR stabilizes above 66 dB, and the SFDR stabilizes above 69 dB. Therefore, selecting  $L = 2048$  is a good choice.

### C. VERIFICATION OF THE PROPOSED TIME SKEW CALIBRATION TECHNIQUE

The time skew calibration technique of Section III-A is used to calibrate the four-channel TI-ADCs. The number of samples in each channel is 2048, and  $\mu$  is 0.00001.

Рисунок 6. Пример из [3]

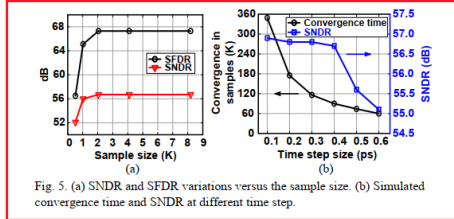


Fig. 5. (a) SNDR and SFDR variations versus the sample size. (b) Simulated convergence time and SNDR at different time step.

For a given order length ( $M$ ) and interpolation vector ( $m$ ), the coefficients in each sub-FIR filter are fixed.

#### B. Error Analysis of the Lagrange Differentiator

The performance of the proposed calibration method primarily hinges upon that of the differentiator. Therefore, it is necessary to analyze the effectiveness of the Lagrange differentiator. The frequency response of the ideal first-order differentiator is expressed as [8]

$$H_{id}(\omega) = j\omega \quad (9)$$

To find the optimal performance of the Lagrange differentiator, an error function is defined by

$$J = \sum_{i=1}^3 \int_0^{\pi} (|H_i(\omega)| - \omega)^2 d\omega \quad (10)$$

where  $H_i(\omega)$  is the transfer function of the  $i$ -th filter, whose impulse response is shown as (8). As three channels are set to be calibrated, the error function is the sum of these errors and defines as the total deviations to the ideal transfer function in the first Nyquist zone. Fig. 4 (a) compares the error function  $J$  for various order length ( $M$ ) of the sub-FIR filter. Clearly, when  $M=14$ , the designed Lagrange differentiator exhibits the lowest error. Oscillation would occur with  $M>14$ , known as the Runge phenomenon, resulting in a sharp increase of  $J$ . Fig. 4 (b) shows the magnitude responses of the designed Lagrange differentiators in the first Nyquist zone with  $M=14$ , where  $m=6, 7$ , and  $8$  are for channel 2, 3, and 4 respectively. Note that the response of  $m=6$  is nearly the same with  $m=8$  in all frequency scope due to symmetry. In low frequency scope, all Lagrange responses fit the ideal one perfect. However, when  $\omega>2.8$ ,  $J$  would be large enough so that the proposed calibration method would be invalid. Such bandwidth limitation is a common issue in all of the derivative-based calibration methods [8-10].

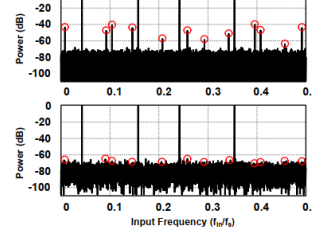


Fig. 7. Output spectrum before and after calibration for a multitone input.

One possible way to deal with this problem is to use complicated IIR filters which may exhibit better performance in high frequency. But design IIR filters have the potential stabilization issues. A more practical way is to slightly improve the sampling frequency of the TI-ADC for some applications. By doing this, the bandwidth will cover the desired input frequency.

#### C. Characteristics of the calibration method

As the convergence time of the proposed calibration method mainly depends on two factors, i.e. sample size used in each calibration cycle and the length of time step (defined as  $\mu$  in Fig. 3). With larger sample size, the value of  $D_1$  reveals smaller variation while results in slower convergence. For a smaller value of  $\mu$ , the time accumulator in Fig. 3 could achieve better accuracy while sacrifices the convergence speed. There exist an optimized sample size and  $\mu$ .

A behavioral model of a 4-channel 600-MS/s TI-ADC based on the proposed timing skew calibration is designed in MATLAB to find the proper sample size and  $\mu$ . In this simulation, it is assumed that there are no offset and gain mismatches among channels. The standard deviation of timing skew is set as  $0.01T_s$ , and the input frequency is 200 MHz. Fig. 5 (a) shows the sample size versus the SNDR and SFDR after calibration. When the sample size is larger than 2048, the SNDR and SFDR will no longer increase. Therefore, the sample size is chosen as 2048. The simulated time step size versus convergence time and SNDR is shown in Fig. 5 (b). It suggests the optimal value of  $\mu$  is around 0.4 ps, where the SNDR is larger than 56.5 dB and the convergence time is also small. Fig. 6 shows the histograms for SNDR and SFDR of 1000 Monte-Carlo simulations, where the timing skews are

Рисунок 7. Пример из [4]

Чтобы получить минимальное количество умножений для того, чтобы пропустить 2к семплов через систему необходимо для каждого суб-АЦП в системе из 4-ех суб-АЦП накопить матрицу из 500 семплов.  $500 \times 4 = 2000$

$$N!(N + 1)N + NL_sN + NL_s + (M - 1)LL_s = 128.1 * 10^3$$

Допустим, для матрицы из 50 семплов, чтобы пропустить 2к семплов необходимо сделать:

$$10 * (N!(N + 1) * N) + 10 * N * 50 * N + N * 50 + (M - 1)LL_s = 160.5 * 10^3$$

Что уже больше чем исходный вариант, поэтому авторы статьи [1] таким образом добились снижения кол-ва умножений.



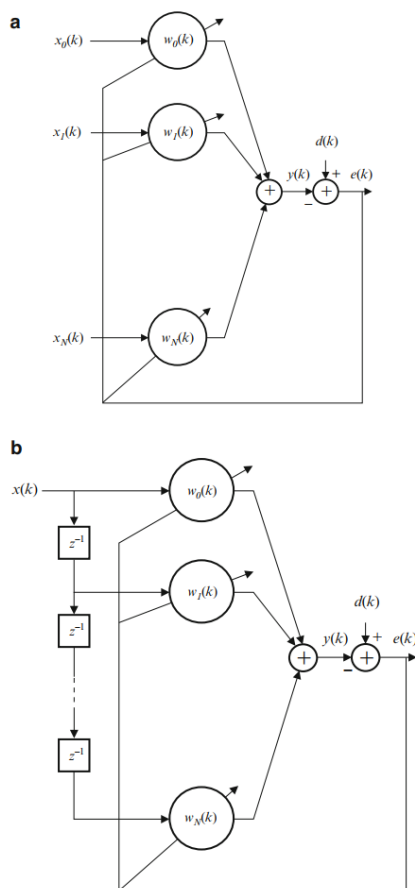
### 3. Вопросы

1. Вопрос по составлению матриц сигнала для адаптивной фильтрации. Правильно ли я понимаю, что адаптивный фильтр можно реализовать двумя способами:

2.4 Wiener Filter

37

**Fig. 2.1** (a) Linear combiner; (b) Adaptive FIR filter



Соответственно и матрицы входного сигнала для них будут отличаться. В первом случае отсчеты будут идти слева направо, а во втором случае справа налево?

Первый случай:

$$A = \begin{matrix} & x_0 & x_1 & x_2 \\ x_0 & & & \\ x_1 & & & \\ x_2 & & & \end{matrix} \quad \mathbf{x} = \begin{matrix} x_0 \\ x_1 \\ x_2 \end{matrix}$$

$$y(1) = x_0 * w_0 + x_1 * w_1 + x_2 * w_2$$

Второй случай:

$$A = \begin{matrix} & x_2 & x_1 & x_0 \\ x_3 & & & \\ x_4 & & & \end{matrix} \quad \mathbf{x} = \begin{matrix} x_0 \\ x_1 \\ x_2 \end{matrix}$$

$$y(1) = x_2 * w_0 + x_1 * w_1 + x_0 * w_2$$

We next define the matrix of the observed input samples as

$$\mathbf{X}^T = \begin{bmatrix} x_1(1) & x_1(2) & \cdots & x_1(N) \\ x_2(1) & x_2(2) & \cdots & x_2(N) \\ \vdots & \vdots & & \vdots \\ x_M(1) & x_M(2) & \cdots & x_M(N) \end{bmatrix} \quad \begin{matrix} \downarrow \\ \text{snapshots} \end{matrix} \rightarrow \text{data records } (M \times N) \quad (9.2.4)$$

where we assume that  $N > M$ . This defines an over-determined least-squares problem.

For the case in which we have one dimensional input signal, as shown in Figure 9.2.1b, the data matrix takes the form

$$\mathbf{X}^T = \begin{bmatrix} x(M) & x(M+1) & \cdots & x(N) \\ x(M-1) & x(M) & \cdots & x(N-1) \\ x(M-2) & x(M-1) & \cdots & x(N-2) \\ \vdots & \vdots & & \vdots \\ x(1) & x(2) & \cdots & x(N-M+1) \end{bmatrix} \quad (9.2.5)$$

2. Вопрос по определителям. В правиле Крамера необходимо искать определители. Для матрицы  $y_{icalo}^T * y_{icalo}$  размерностью  $5 \times 500 \times 500 \times 5 = 5 \times 5$  определитель получается слишком большим. Правильно ли я понял размер матрицы, что там действительно  $L_s = 500$ ?

Список литературы:

1. Hu, M.; Yi, P., "Digital Calibration for Gain, Time Skew, and Bandwidth Mismatch in Under-Sampling Time-Interleaved System." *Appl. Sci.* 2022, 12, 11029, doi:10.3390/app122111029
2. Jun, J. A condensed Cramer's rule for the minimum norm least-squares solution of linear equations. *Linear Algebra Its Appl.* **2012**, 437, 9.
3. Hu, M.; Yi, P., "Novel Multi-Stage Feedback Technique for Time Skew Calibration in Time-Interleaved ADCs"
4. Dengquan Li, Zhangming Zhu; "A 10-bit 600-MS/s Time-Interleaved SAR ADC With Interpolation-Based Timing Skew Calibration"