Алгоритм наименьших квадратов для калибровки Time Interleaved ADC

1. 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 *fs* (интервал дискретизации *Ts* = 1/*fs*). При этом каждый АЦП массива в действительности осуществляет выборку (и преобразование) сигнала на меньшей частоте, *fs*/*M*. [6]

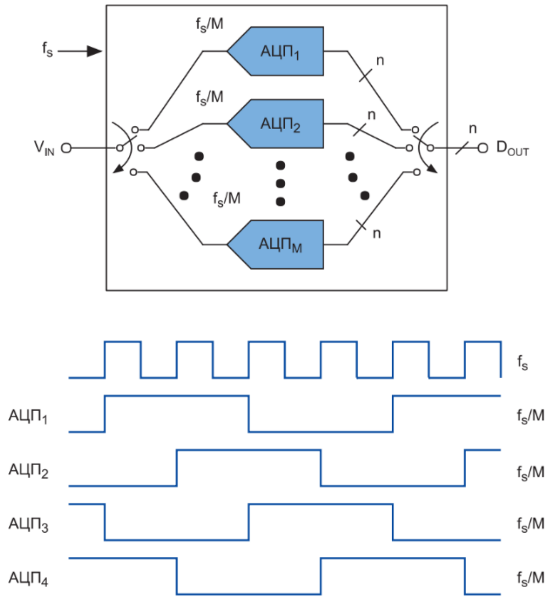

Рисунок 1. Массив из M n-разрядных АЦП с чередованием во времени. (Частота дискретизации каждого АЦП равна fs/M, а общая частота дискретизации всей системы fs)

За референсный (эталонный) сигнал берем выход суб-АЦП0. Тогда для i-го суб-АЦП эталонный сигнал запишется как

$$y_{ri}(n) = y_0(n) * h_{ri}(k), \qquad (15)$$

где $y_0(n)$ – выход АЦП0, $h_{ri}(k)$ – импульсная характеристика фильтра дробной задержки

Частотная характеристика фильтра дробной задержки запишется как:

$$H_{ri}(jw) = e^{-jwT_s\frac{i}{M}}, \qquad (16)$$

Где $\frac{i}{M}$ – дробная часть от частоты дискретизации Fs. 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 Fs, для 2-го суб-АЦП = 2/4 = 0.5 Fs, для 3-го = ¾ = 0.75 Fs.

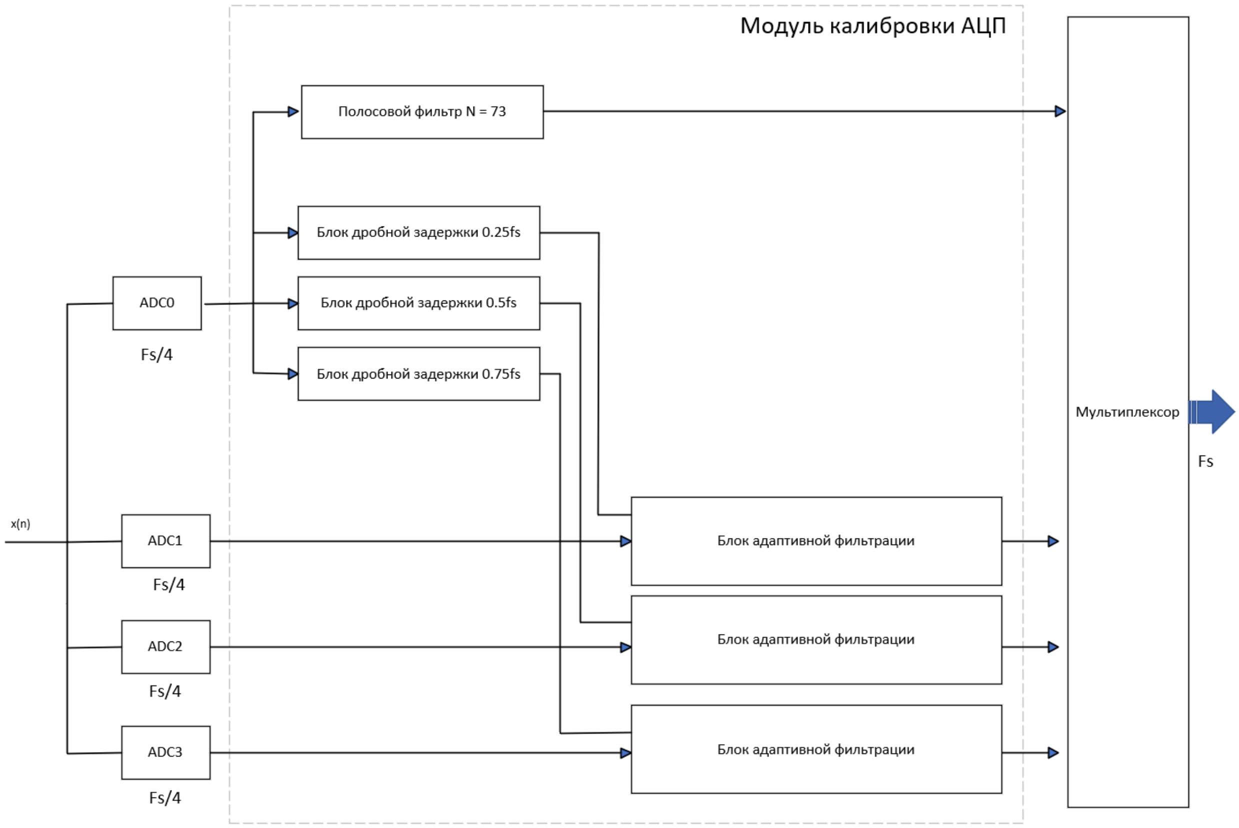

Рисунок 2. Предлагаемая блок-схема алгоритма калибровки, для M = 4

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:

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

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:

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

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

$$J = (y_{ri} - y_{icalo} * h_{\Delta g_i \Delta \tau_i, \Delta w_i})^T * \left(y_{ri} - y_{icalo} * h_{\Delta g_i \Delta \tau_i, \Delta w_i}\right)$$
$$= \ y_{ri}{}^T * y_{ri} - h_{\Delta g_i \Delta \tau_i, \Delta w_i}{}^T * \ y_{icalo}{}^T * y_{ri} - \ y_{ri} * y_{icalo} * h_{\Delta g_i \Delta \tau_i, \Delta w_i} + h_{\Delta g_i \Delta \tau_i, \Delta w_i}{}^T * y_{icalo}{}^T * y_{icalo} * h_{\Delta g_i \Delta \tau_i, \Delta w_i}$$

Найдем производную от $h_{\Delta g_i \Delta \tau_i, \Delta w_i}$, т.е проведем касательную к полученной фигуре:

$$\frac{dj}{dh_{\Delta g_i \Delta \tau_i, \Delta w_i}} \Big|_{h_{\Delta g_i \Delta \tau_i, \Delta w_i} = \widehat{h_{\Delta g_i \Delta \tau_i, \Delta w_i}}} = -2 * y_{icalo}{}^T * y_{ri} + 2 * y_{icalo}{}^T * y_{icalo} * h_{\Delta g_i \Delta \tau_i, \Delta w_i} = 0$$

$$y_{icalo}{}^{T} * y_{icalo} * \widehat{h_{\Delta g_i \Delta \tau_i, \Delta w_i}} = y_{icalo}{}^{T} * y_{ri} \qquad (35)$$

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

$$\widehat{h_{\Delta g_i \Delta \tau_i}}_{,\Delta w_i} = (y_{icalo}{}^{T} * y_{icalo})^{-1} * y_{icalo}{}^{T} * y_{ri} \qquad (36)$$

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:

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

Вектор-столбец y_{ri}запишется как:

$$y_{ri} = \begin{matrix} y_{icalo}\,(1+\frac{N}{2}) \\ \vdots \\ y_{icalo}\,(L_s+\frac{N}{2}) \end{matrix} \qquad (21)$$

Уравнение 36 решается с помощью метода Крамера. Далее приводится кол-во умножений для коррекции сигнала. В данной статье

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

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

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

Для коррекции i-го канала нужно совершить

$$N + (N+1) * (L_s + N!) + (M-1)L * L_s = 128.1\textit{к умножений.}$$

Где L – 73 порядок фильтра дробной задержки, N – 5 порядок адаптивного фильтра

Методом подбора узнаем, что $L_s = 500$

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

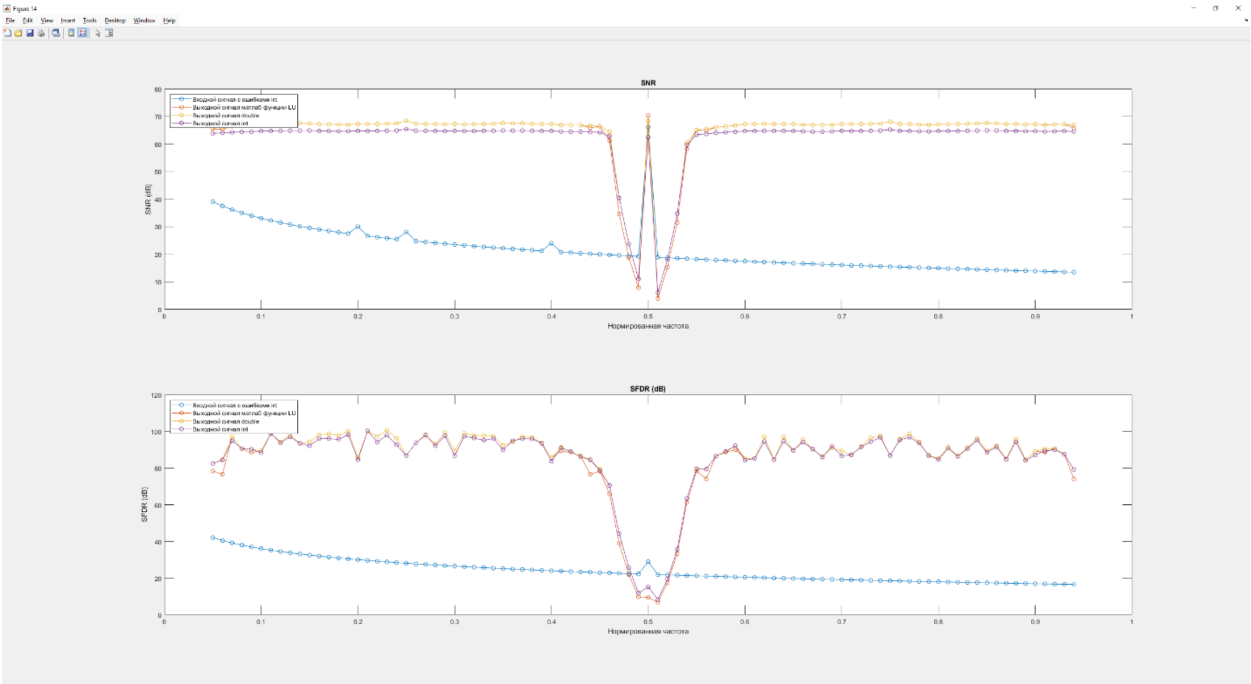

Рисунок 3. Результаты моделирования

1) В целом результаты совпадают с результатами, приведенными в исходной статье. За исключением диапазонов возле границ зон Найквиста – там наблюдается провалы в SNR и SFDR. В статье данные интервалы не моделируют (рисунок 7 (от 50 МГц – 420 МГц), рисунок 8 (550 МГц – 920 МГц), рисунок 9 (1.05 ГГц – 1.42 ГГц)). 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. Соотвественно на частотах, близких к границам зон Найквиста мы давим сигнал.

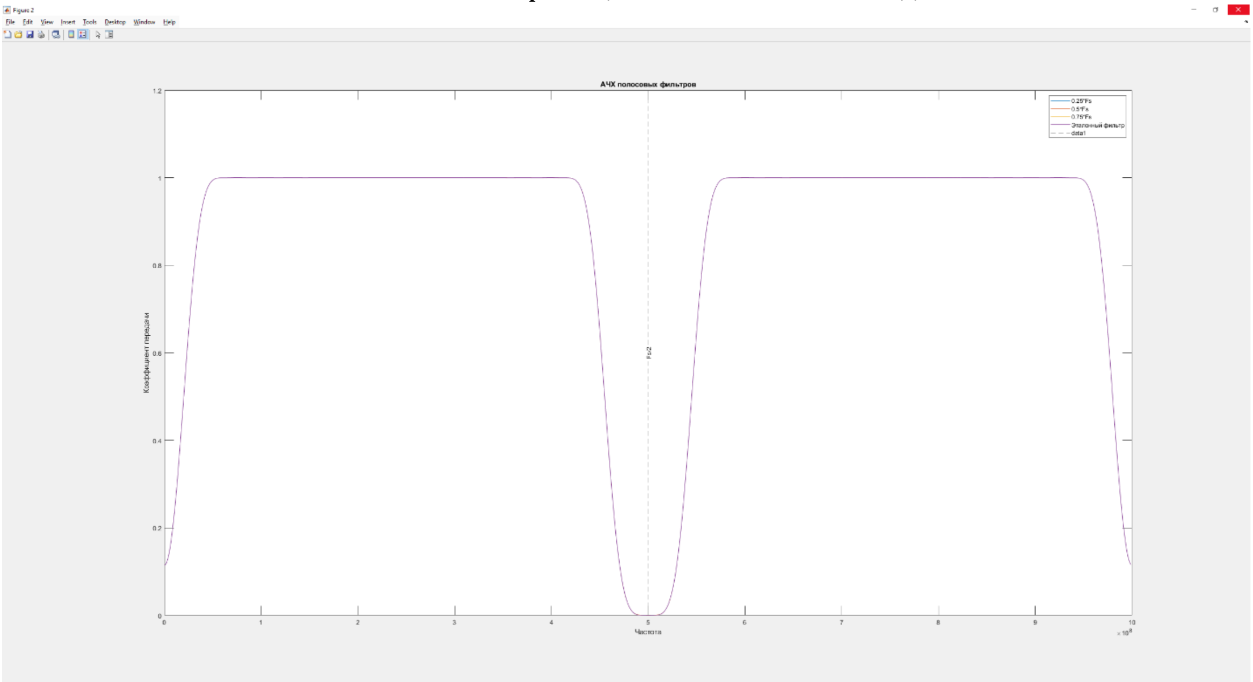

Рисунок 4. АЧХ фильтров дробной задержки

2) Так же есть вопрос по составлении матриц сигнала для адаптивной фильтрации. Правильно ли я понимаю, что адаптивный фильтр можно реализовать двум способами:

2.4 Wiener Filter 37

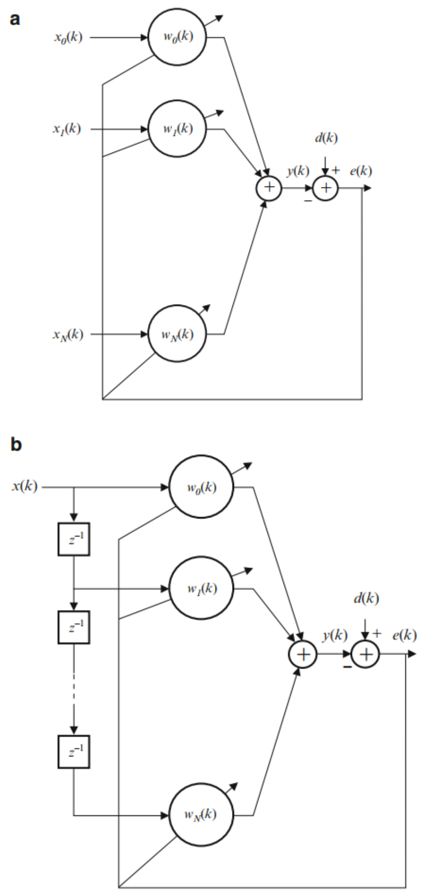

Fig. 2.1 (**a**) Linear combiner; (**b**) Adaptive FIR filter

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

Первый случай:

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

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

Второй случай:

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

$$y(1) = x_2 * \ w_0 + \ x_1 * w_1 + x_0 * w_3$$

We next define the matrix of the observed input samples as

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

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

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

3) И вопрос по определителям. В правиле Крамера необходимо искать определители. Для матрицы ${y_{icalo}}^T * y_{icalo}$ размерностью 5x500 X 500x5 = 5x5 определитель получается слишком большим. Правильно ли я понял размер матрицы, что там действительно $L_s = 500$?
4) Так же не совпадает кол-во умножений в статье для правила Крамера требуется $(N+1)N$ кол-во определителей, у меня же их N+1