

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

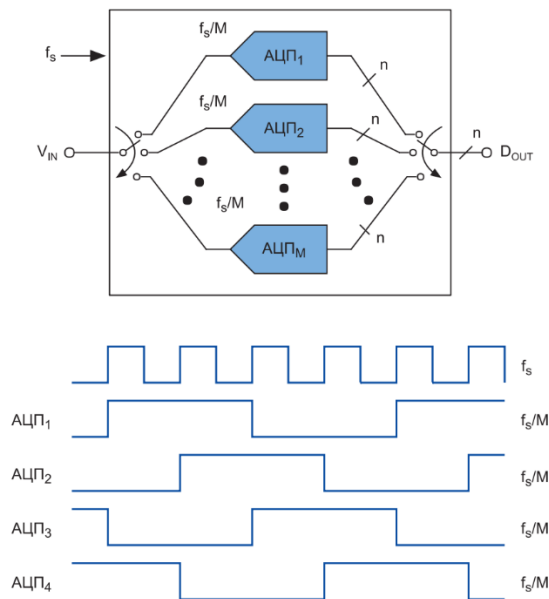


Рисунок 1. Массив из M n -разрядных АЦП с чередованием во времени. (Частота дискретизации каждого АЦП равна f_s/M , а общая частота дискретизации всей системы f_s)

За референсный (эталонный) сигнал берем выход суб-АЦП₀. 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 АЦП₀, $h_{ri}(k)$ – 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 $= 3/4 = 0.75 F_s$.

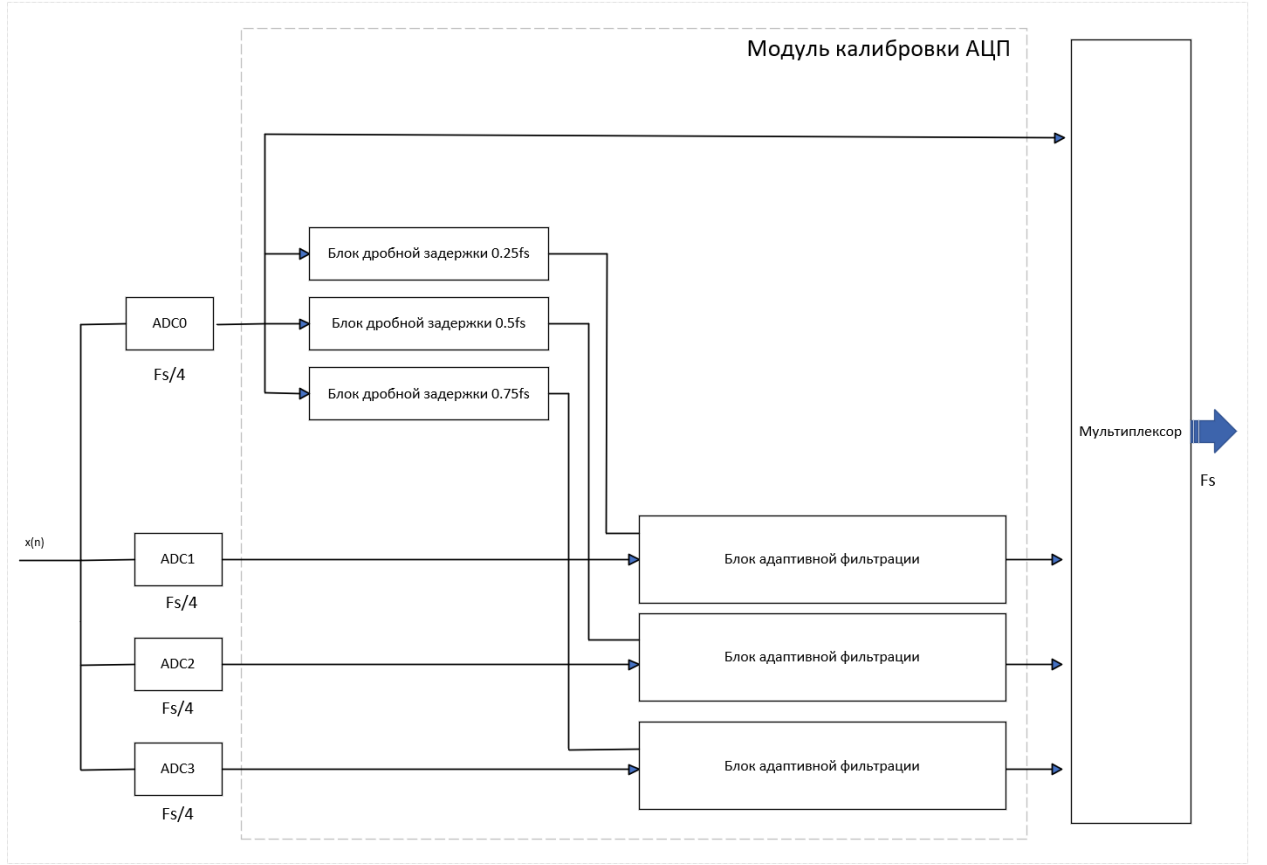


Рисунок 2. Предлагаемая блок-схема алгоритма калибровки, для $M = 4$

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

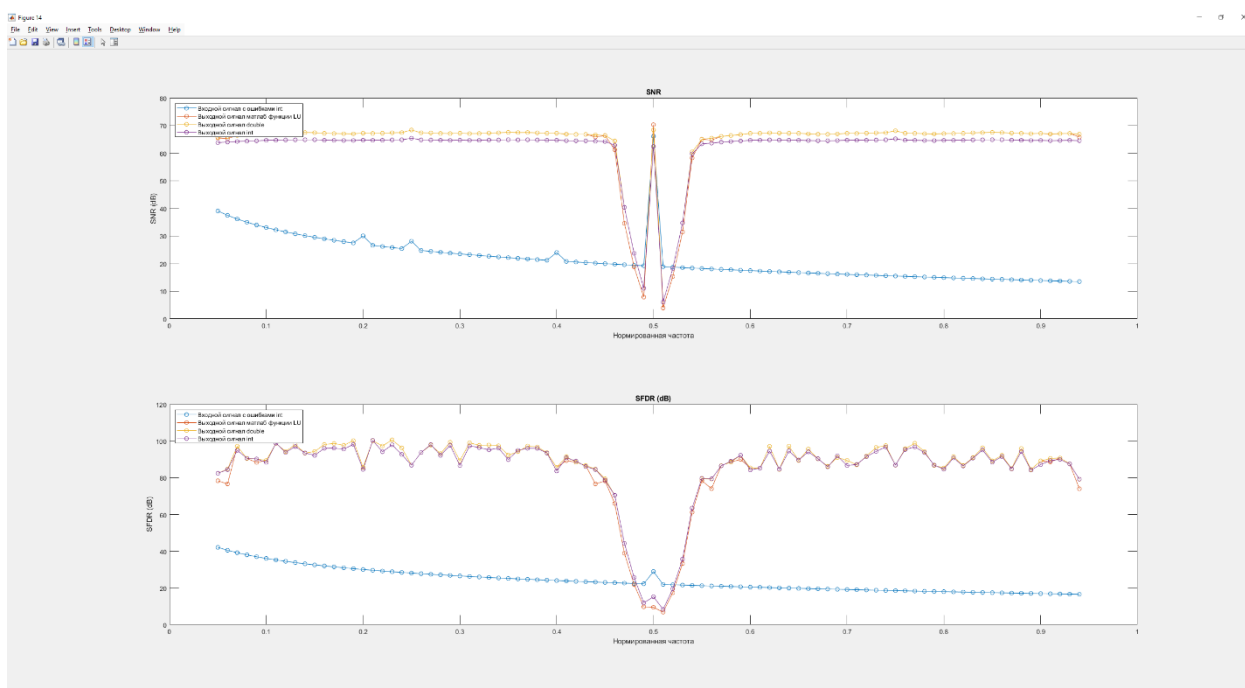


Рисунок 3. Результаты моделирования

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.

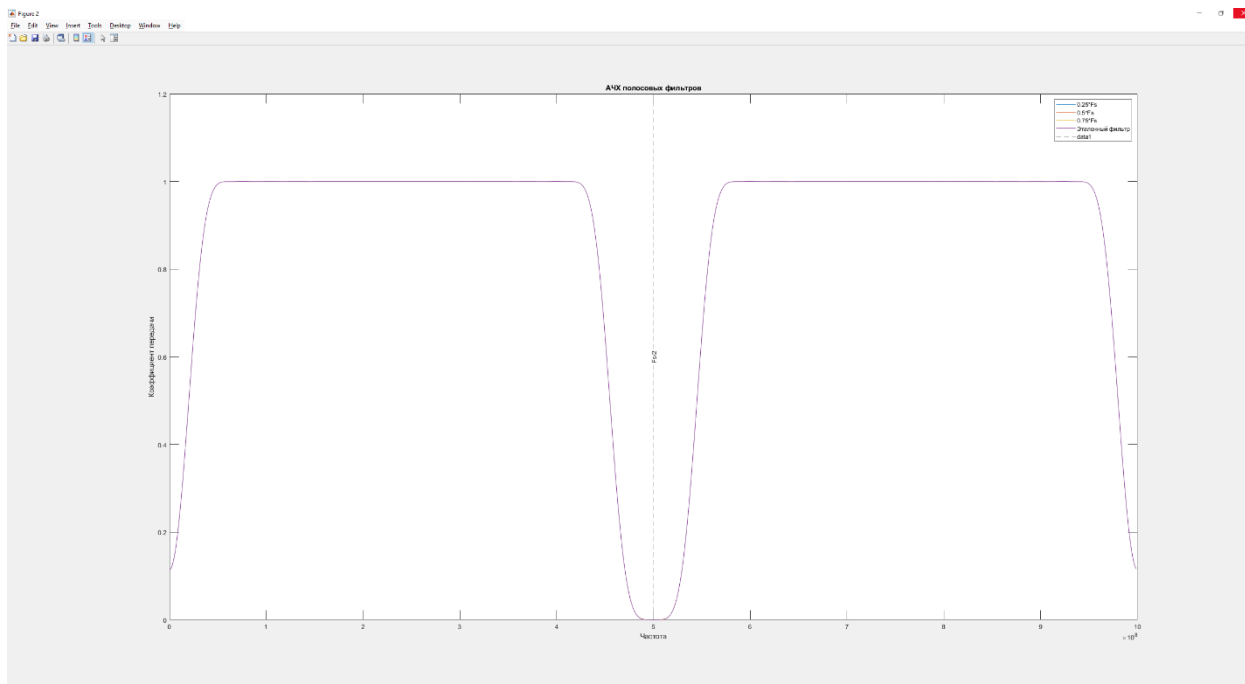


Рисунок 4. АЧХ фильтров дробной задержки

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

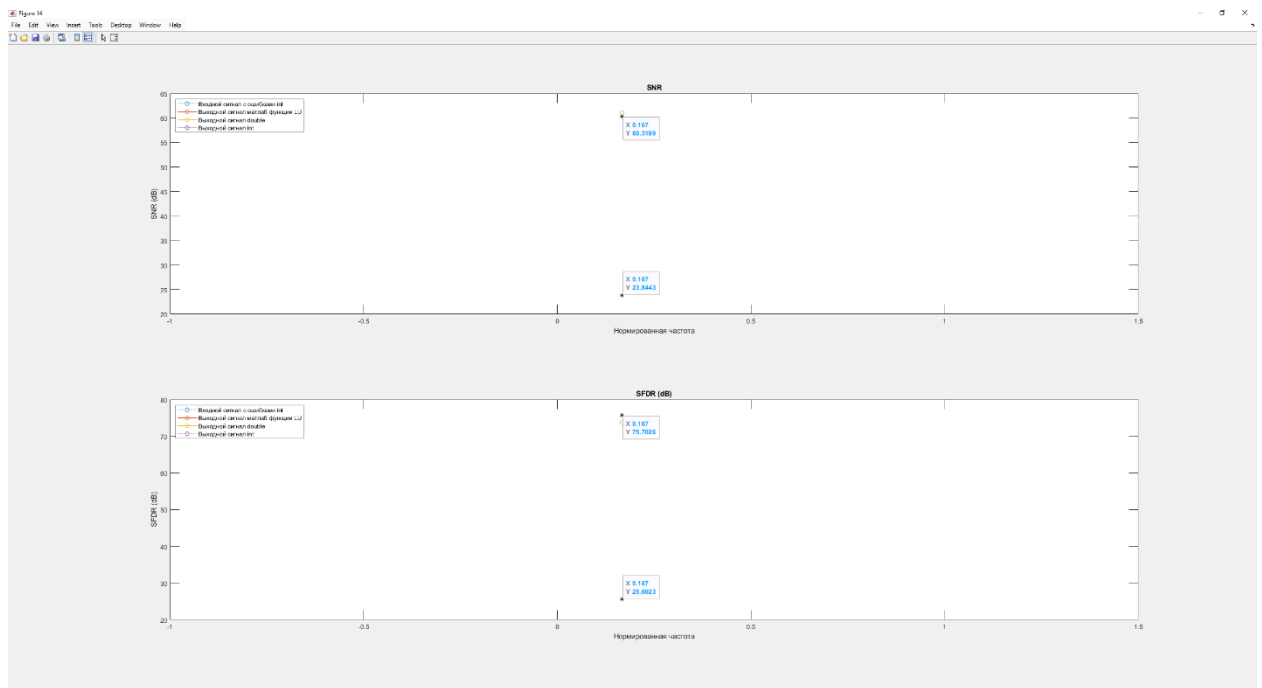


Рисунок 5. Результаты моделирования на 0.167 fs

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

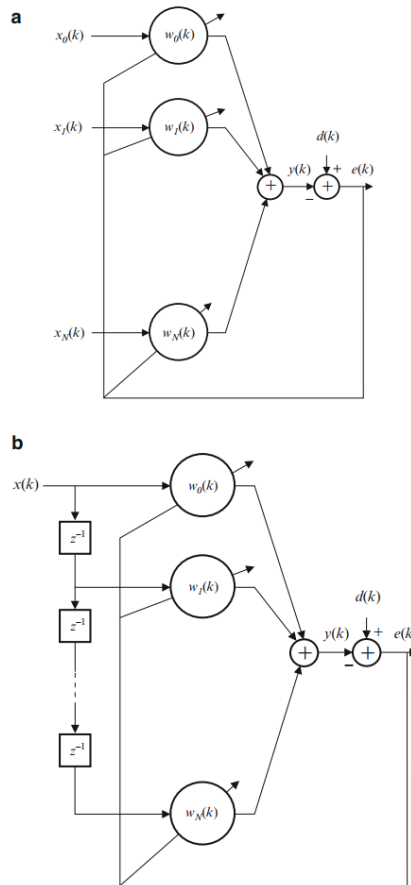
3. Вопросы

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:

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

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

Второй случай:

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

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

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.