

АДАПТИВНЫЙ ФИЛЬТР В ЧАСТОТНОЙ ОБЛАСТИ

ФОРМИРОВАНИЕ СИГНАЛА

- Модуляция - QPSK (3 отсчета на символ)

```
clc; close all; clear;
M = 4; % Порядок модуляции
D = 16; % Согласующая задержка фильтра (при D = 0 не работает)
size = 2048; % Число символов в пакете
data = randi([0, M - 1], size + D, 1); % Генерирует данные на множестве {0, 1, 2, 3}
```

- Формирование узкополосной помехи

```
g = 0.1; % Энергия
SNR = 28; % Отношение сигнал шум дБ
narrowband_noise = g * pskmod(randi([0, M - 1], size + D, 1), M); % Узкополосная помеха
tx = pskmod(data, M, pi / 4); % Маппинг согласно QPSK
rx = awgn(tx + narrowband_noise, SNR, 'measured'); % сигнал + узкополосная помеха + шум
rx = rx(1 + D: D + size); % Искусственная задержка на D (Без нее фильтр не работает)
tx = tx(1 : size); % Срезаем первые size элементов, чтобы длины tx и rx совпадали
```

Созвездие

```
figure;
scatter(real(rx), imag(rx))
title('Созвездие сигнал + наведенная помеха + шум')
xlabel('Re')
ylabel('Im')
grid on;
```



ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Тип: Frequency domain fast least mean squares.

Основные параметры:

1. α - шаг сходимости алгоритма
2. γ - фактор забывания (параметр веса взвешенного экспоненциального среднего)
3. $\widehat{W}(k)$ - вектор весовых коэффициентов адаптивного фильтра (частотная область)
4. $U(k)$ - диагональная матрица входного сигнала в частотной области
5. $e(k)$ - сигнал ошибки во временной области
6. $E(k)$ - сигнал ошибки в частотной области
7. $P_i(k)$ - экспоненциально взвешенная сумма мощностей входного сигнала
8. $D(k)$ - коэффициент нормирования, выравнивает скорость сходимости каждого отсчета спектра адаптивного фильтра

Структурная схема

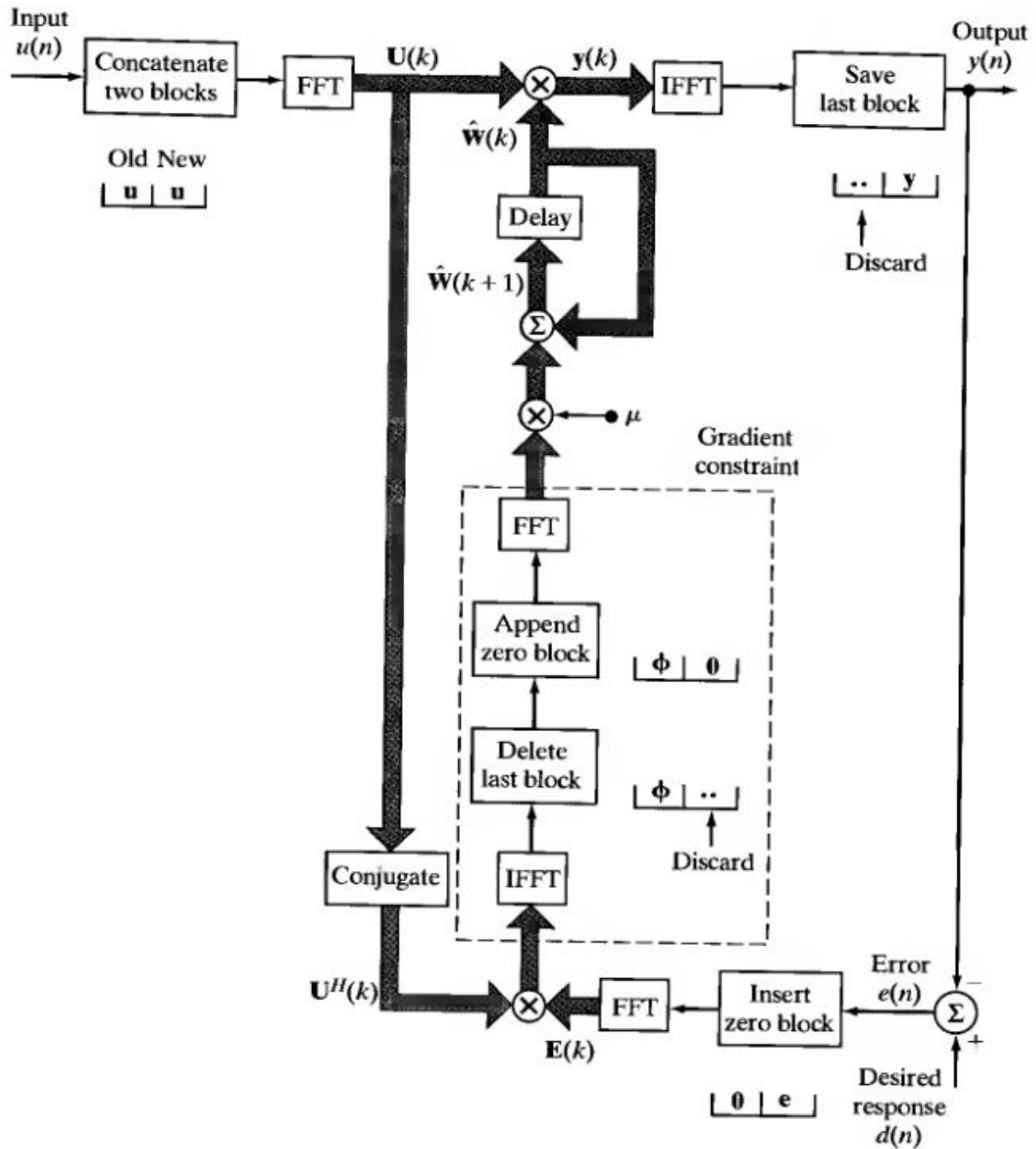


FIGURE 7.3 Overlap-save FDAF. This FDAF is based on the overlap-save sectioning procedure for implementing linear convolutions and linear correlations. (Taken from *IEEE SP Magazine* with permission of the IEEE)

Алгоритм

Инициализация:

$$\hat{W}(0) = 0_{(2M-1) \times 1}$$

$$P_i(0) = \delta_i, \forall i = \overline{0, 2M-1}$$

Этап фильтрации:

- $U(k) = \text{diag}[\text{FFT}[u(kM - M), \dots, u(kM - 1), u(kM), \dots, u(kM + M - 1)]]^T$
- $y^T(k)$ срез последних M элементов из вектора $\text{IFFT}[U(k)\widehat{W}(k)]$:

Формирование сигнала ошибки

- $e(k) = d(k) - y(k)$
- $E(k) = \text{FFT}\left(\begin{bmatrix} 0_{M \times 1} \\ e(k) \end{bmatrix}\right)$

Оценка мощности сигнала:

- $P_i(k) = \gamma P_i(k - 1) + (1 - \gamma)|U_i(k)|^2$
- $D(k) = \text{diag}\{P_0^{-1}, P_1^{-1}, \dots, P_{2M-1}^{-1}\}$

Обновление весов фильтра:

- $\phi(k)$ - срез из первых M элементов вектора $\text{IFFT}[D(k)U^H(k)E(k)]$
- $\widehat{W}(k + 1) = \widehat{W}(k) + \alpha \text{FFT}\left(\begin{bmatrix} \phi(k) \\ 0_{M \times 1} \end{bmatrix}\right)$

Реализация:

```
% Инициализация
alpha = 0.29601;
gamma = 0.64693;
M = 32 * 2 ^ 0
```

```
M = 32
```

```
N = 2 * M; % Длина БПФ
W = zeros(N, 1); % Веса АФ
P = ones(N, 1); % Инициализация экспоненциально взвешенной суммы мощностей входа

% Алгоритм
for k = 1 : length(rx) / M - 1

    u = rx(k * M - M + 1 : k * M + M); % Скользящее окно входа
    d = tx(k * M + 1 : k * M + M); % Скользящее окно опоры

    % Этап фильтрации:
    U = diag(fft(u));
    y = ifft(U * W);
    y = y(M + 1 : end);

    % Формирование сигнала ошибки
```

```

e = d - y;
E = fft([complex(zeros(M, 1), zeros(M, 1)); e]);

% Оценка мощности сигнала:
P = gamma * P + (1 - gamma) * U * U';
D = P ^ -1;

%Обновление весов фильтра:
phi = ifft(D * U' * E);
W = W + alpha * fft([phi(1 : M); complex(zeros(M, 1), zeros(M, 1))]);

% Извлечение векторов в массив для построения
out(k * M - M + 1 : k * M) = y;
error(k * M - M + 1 : k * M) = e;

end
out = out(1000 : end); % Срез, чтобы избавиться от переходного процесса

```

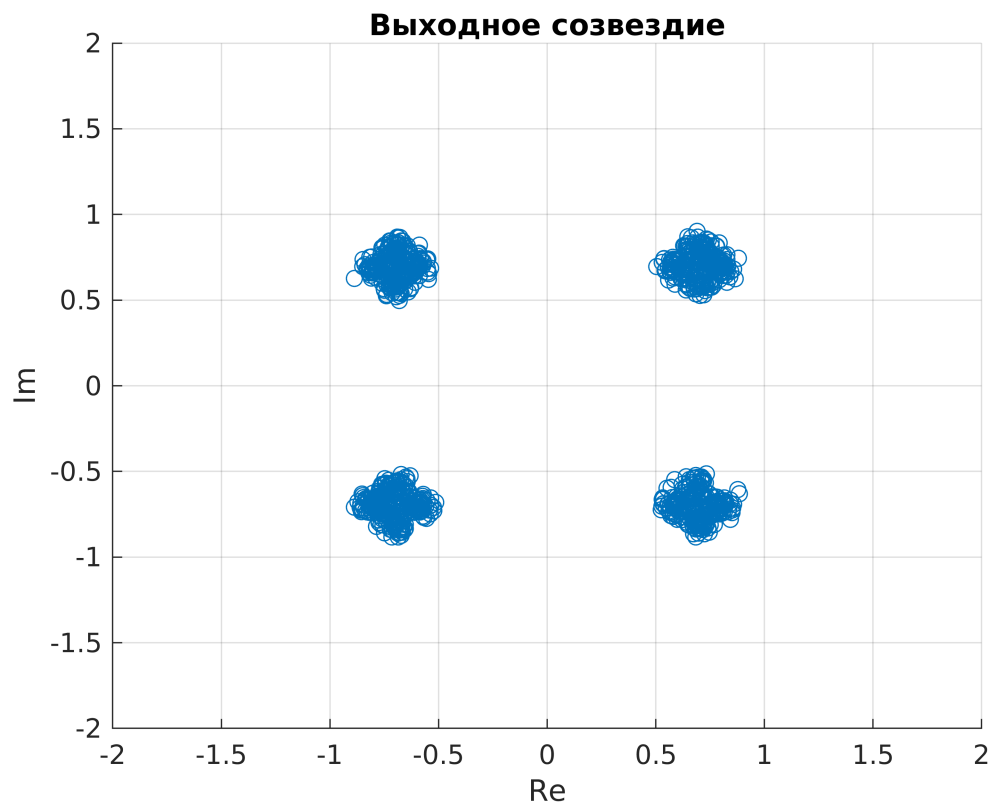
ОТОБРАЖЕНИЕ РЕЗУЛЬТАТОВ

Созвездие выхода

```

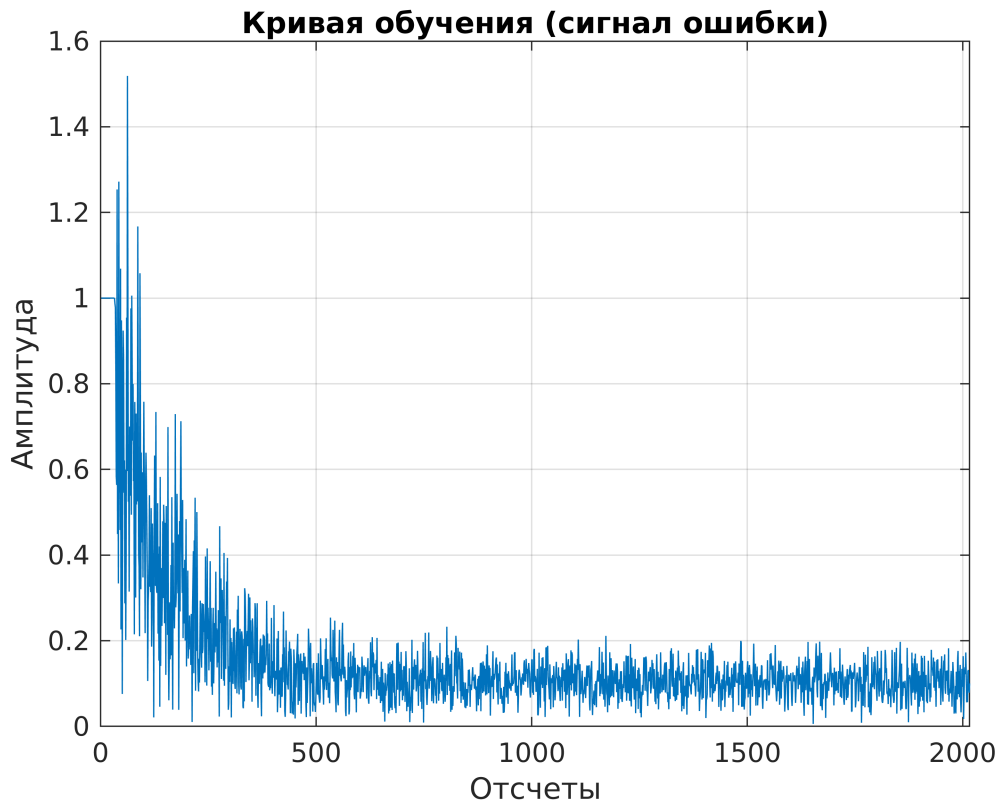
figure;
scatter(real(out), imag(out))
title('Выходное созвездие')
xlabel('Re')
ylabel('Im')
xlim([-2, 2])
ylim([-2, 2])
grid on;

```



Сигнал ошибки

```
figure;
plot(abs(error))
title('Кривая обучения (сигнал ошибки)')
xlabel('Отсчеты')
ylabel('Амплитуда')
xlim([0 length(error)])
grid on
```



Отмечу, что на фоне узкополосной помехи (которая принуждает сигнал вести себя так, будто он в нестационарном канале) смещена вверх на 0,1. Так может быть. Так как, Винер решал задачу идентификации стационарной(неподвижной по параметрам во времени) системы.

Ссылки:

- 1) <https://www.mathworks.com/help/dsp/ref/dsp.frequencydomainadaptivefilter-system-object.html> - Встроенная функция (сверял по ней)
- 2) Simon Haykin - ADAPTIVE FILTER THEORY
- 3) Прокис - цифровая связь
- 4) Грант, Коуэн - адаптивные фильтры