

## Лекція 5

### Адаптивні цифрові фільтри

#### 1. Вступ

Адаптивний фільтр - це цифровий фільтр з характеристиками, що само налаштовуються. Такий фільтр автоматично адаптується до змін вихідних сигналів і є предметом вивчення підобласті ЦОС, відомої як адаптивна обробка сигналів.

#### 1. Коли та де варто використовувати адаптивні фільтри

У багатьох додатках часто виникає така проблема, як забруднення корисного сигналу небажаними, часто потужними сигналами або шумом. Якщо корисний сигнал і шум займають строго певні та рознесені смуги частот, для отримання корисного сигналу використовуються звичайні лінійні фільтри. Однак існують ситуації, при яких потрібно, щоб характеристики фільтра були змінними, адаптувалися до змін сигналів або змінювалися інтелектуально.

У таких випадках, наприклад, коли спектри сигналу та шуму перекриваються (рис. 1), або коли смуга шумових частот невідома чи змінна, коефіцієнти фільтра повинні змінюватися, і їх не можна задати заздалегідь.

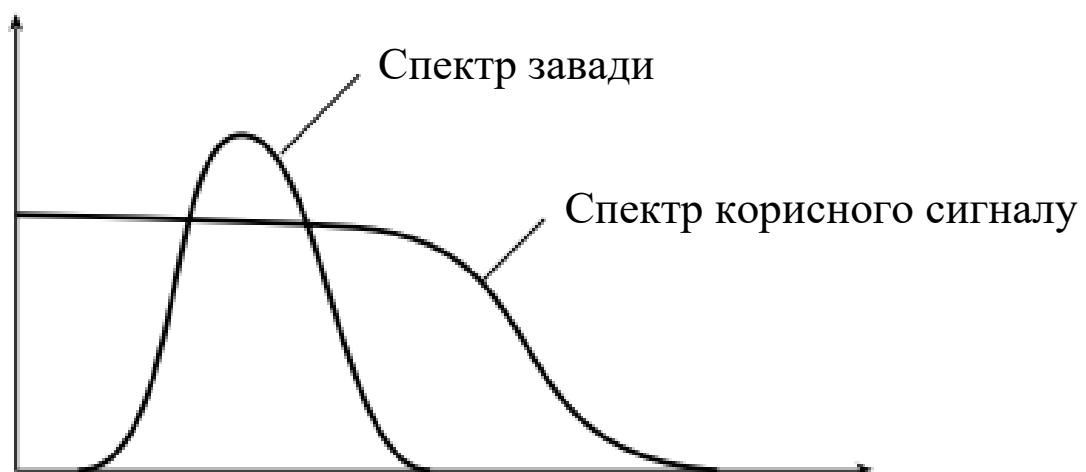


Рис. 1. Ілюстрація накладання спектрів сигналу та потужної перешкоди

Перелічимо типові програми, у яких недоречні фільтри з фіксованими коефіцієнтами.

1. Електроенцефалографія (ЕЕГ), де артефакти, або забруднення сигналу, породжене рухом або морганням ока, значно більше справжньої електричної діяльності мозку і займають ту ж смугу частот, що і сигнал, що представляє медичний інтерес. Щоб видалити артефакти, зберігши при цьому необхідні сигнали, звичайні лінійні фільтри не підходять.

2. У цифровому зв'язку є метод розширення спектра, який використовується в умовах, коли сильна активна перешкода може інтерферувати з корисним сигналом. Інтерференція часто відбувається у вузькій, але невідомій смузі частот у широкосмуговому спектрі, і єдине ефективне вирішення цієї проблеми пропонують адаптивні фільтри.

3. Цифрова передача даних телефонними каналами на високій швидкості. Спотворення сигналу, викликане поганими амплітудною та фазовою характеристиками каналу, призводить до того, що імпульси, що представляють різні цифрові коди, інтерферують між собою (міжсимвольна інтерференція), що ускладнює надійне детектування кодів у приймачі. Для компенсації внесеного каналом спотворення, яке може бути змінним або невідомим, використовується адаптивне вирівнювання.

Адаптивний фільтр відрізняється такою властивістю: його частотна характеристика автоматично регулюється або модифікується для поліпшення продуктивності фільтра відповідно до деякого критерію, що дозволяє фільтру адаптуватися до змін характеристик вхідного сигналу. Внаслідок само налаштування та властивої адаптивним фільтрам гнучкості вони використовуються в таких різних сферах, як телефонне придушення відлуння, обробка сигналів з радарів, системи навігації, вирівнювання каналів зв'язку та виділення біомедичних сигналів.

Отже, адаптивні фільтри варто використовувати, якщо:

- характеристики фільтра повинні бути змінними, адаптуватися до змінних умов;
- існує спектральне перекриття сигналу та шуму (рис. 1);
- смуга шумових сигналів невідома чи змінюється з часом.

У перерахованих випадках використання звичайних фільтрів призвело б до неприйняттого спотворення корисного сигналу. Нижче показано, що, крім зниження рівня шумів, є й інші ситуації, коли вигідно використовувати адаптивні фільтри.

## **2. Концепції адаптивної фільтрації**

### **2.1. Адаптивні фільтри як засіб придушення шуму**

Адаптивний фільтр складається з двох різних частин: цифрового фільтра з регульованими коефіцієнтами та адаптивного алгоритму, який використовується для налаштування або зміни коефіцієнтів фільтра (рис. 2). На адаптивний фільтр одночасно подаються два вхідні сигнали,  $y_k$  і  $x_k$ . Сигнал  $y_k$  забруднений і містить корисний сигнал  $s_k$  і шум  $n_k$ , причому передбачається, що ці сигнали не корелюють. Сигнал  $x_k$  — це міра забрудненого сигналу, який корелює з  $n_k$ . У цифровому фільтрі сигнал  $x_k$

обробляється щоб одержати оцінки  $\hat{n}_k$  сигналу  $n_k$ . Потім корисний сигнал оцінюється як різницю зашумленого сигналу  $y_k$  і виходу цифрового фільтра  $\hat{n}_k$ :

$$\hat{s}_k = y_k - \hat{n}_k = s_k + n_k - \hat{n}_k \quad (1)$$

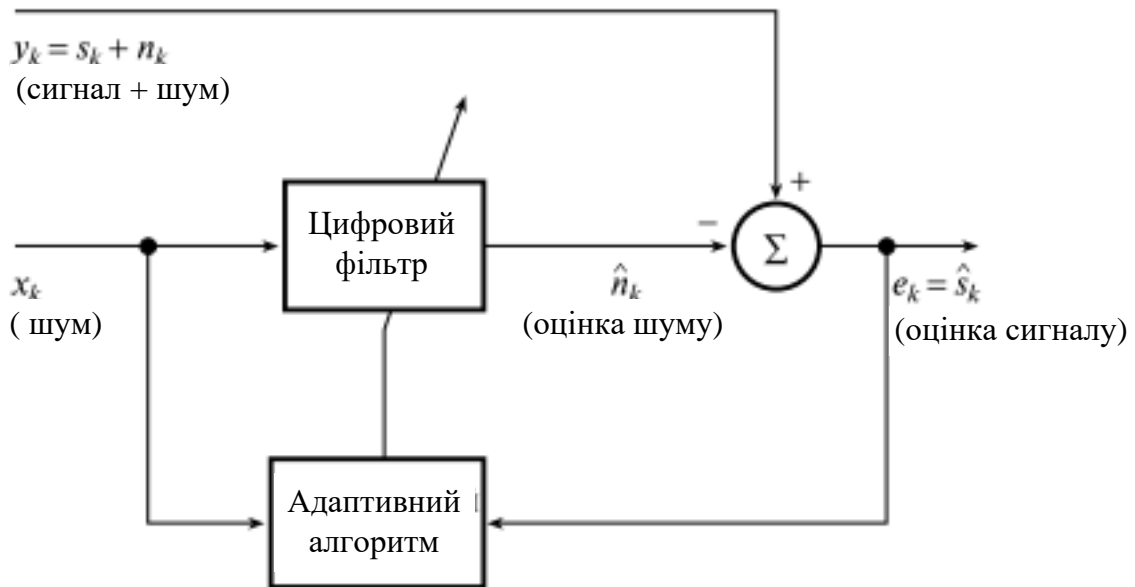


Рис. 2. Блок-схема адаптивного фільтра-шумоподавлювача

Основна мета шумоподавлення полягає в отриманні оптимальної оцінки шуму в зашумлених сигналах, а отже оптимальної оцінки бажаного сигналу. Для цього  $\hat{s}_k$  подається в ланцюг зворотного зв'язку, і на основі цього сигналу і відповідного адаптивного алгоритму регулюються коефіцієнти цифрового фільтра для мінімізації шуму в сигналі  $\hat{s}_k$ . Вихідний сигнал  $\hat{s}_k$  використовується подвійно: 1) як оцінка бажаного сигналу і 2) як сигнал помилки, який застосовується для регулювання коефіцієнтів фільтра.

## 2.2. Інші конфігурації адаптивного фільтра

У наведених вище міркуваннях передбачається, що фільтр використовується для адаптивного шумоподавлення. Важливо пам'ятати, що адаптивні фільтри можуть використовуватися і в інших цілях, наприклад, для лінійного передбачення, адаптивного виділення сигналу та адаптивного управління. Взагалі завжди варто мати на увазі, що значення сигналів  $x_k$ ,  $y_k$  і  $e_k$  або способу їх отримання залежить від програми. Деякі приклади конфігурацій адаптивного фільтра наведено на рис. 3.

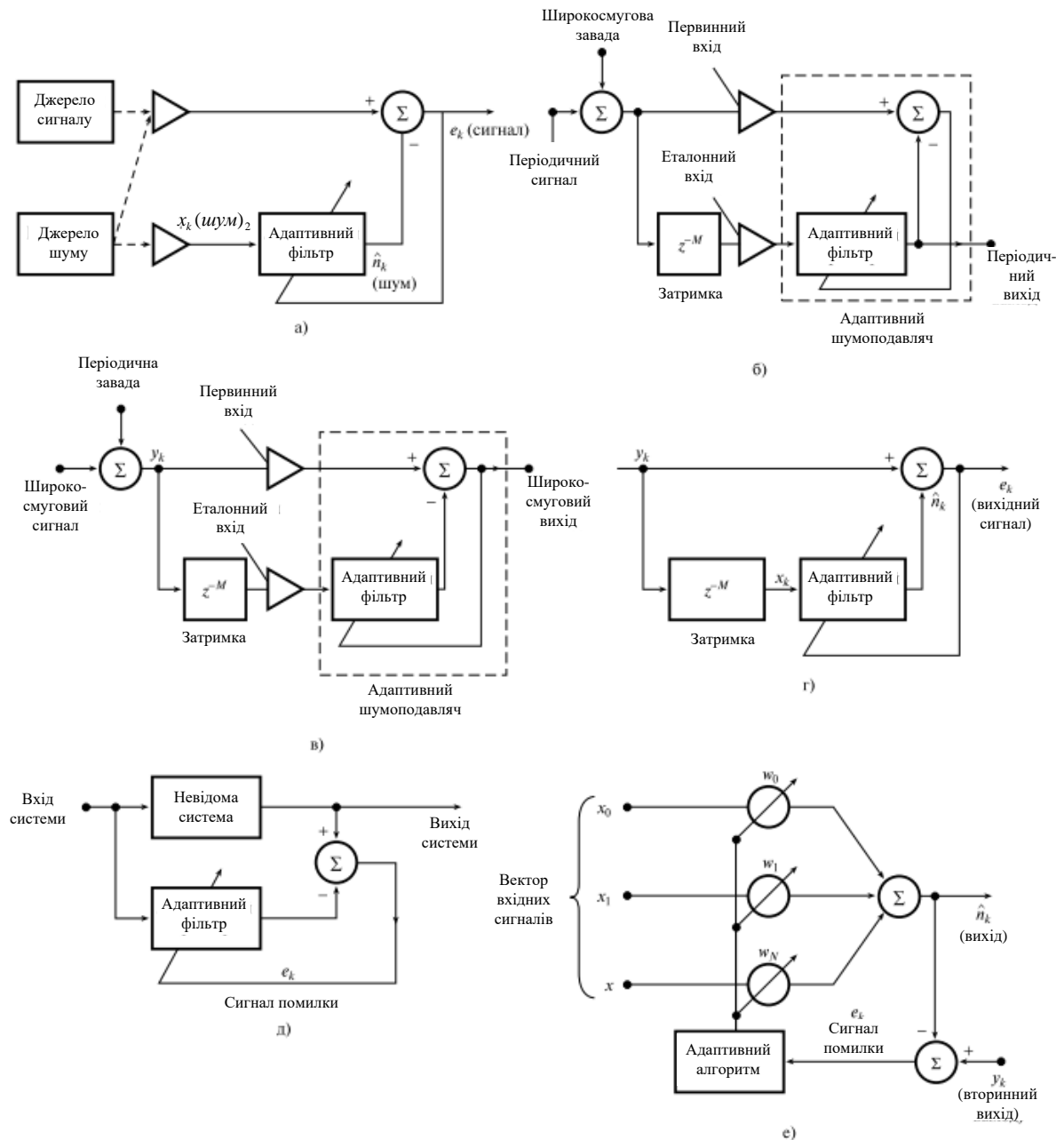


Рис. 3. Деякі конфігурації адаптивного фільтра: а) адаптивний шумоподавлювач; б) адаптивний фільтр, що само налаштовується; в) придушення періодичної завади без зовнішнього еталонного джерела; г) адаптивний селективний фільтр; д) моделювання системи; е) підбір лінійної комбінації

### 2.3. Основні компоненти адаптивного фільтра

У більшості адаптивних систем цифровий фільтр показаний на рис. 2, реалізований з використанням трансверсальної структури або схеми з кінцевою імпульсною характеристикою (КІХ) (рис. 4). Іноді використовуються й інші форми, наприклад, схема з нескінченною імпульсною характеристикою (БІХ) або ґратчаста структура, але КІХ-структура застосовується найбільш широко через свою простоту і гарантовану стійкість. Вихід  $N$ -точкового фільтра, зображеного на рис. 4 виражається наступним чином:

$$\hat{n}_k = \sum_{i=0}^{N-1} w_k(i) x_{k-i} \quad (2)$$

де  $w_k(i)$ ,  $i = 0, 1, \dots$  – регульовані коефіцієнти фільтра (або вагові коефіцієнти), а  $x_k(i)$  та  $\hat{n}_k$  — вхід та вихід фільтра відповідно. На рис. 4 зображено систему з одним входом і одним виходом. У системі з кількома входами та одним виходом  $x_k$  може включати одночасні входи  $N$  різних джерел сигналу.

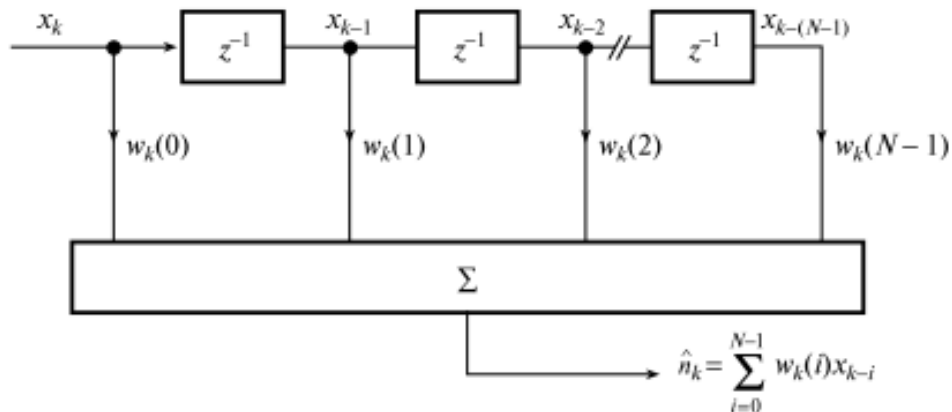


Рис. 4. Структура фільтра з кінцевою імпульсною характеристикою

## 2.4. Адаптивні алгоритми

Для налаштування коефіцієнтів цифрового фільтра (на рис. 2) використовуються адаптивні алгоритми, в яких сигнал помилки  $e_k$  мінімізується згідно з деяким критерієм, наприклад, за схемою найменших квадратів. Найбільшого поширення набули алгоритми, у яких використовуються схема найменших квадратів, рекурсивна схема найменших квадратів і фільтр Калмана. З погляду обчислювальної складності та вимог до пам'яті найбільш ефективна схема найменших квадратів. Більше того, для неї несуттєвою є проблема чисельної нестійкості, властива двом іншим алгоритмам. З цих причин у багатьох ситуаціях спочатку пробують схему найменших квадратів. Втім, рекурсивна схема найменших квадратів має найкращу збіжність.

### Приклад 1

Оцінка бажаного сигналу на виході адаптивного шумоподавлювача виражається наступним чином:

$$\hat{s}_k = y_k - \hat{n}_k = s_k + n_k - \hat{n}_k$$

Покажіть, що при мінімізації загальної потужності на виході шумоподавлювача максимізується вихідне відношення сигнал-шум.

*Рішення*

Зашумлений сигнал записується як

$$y_k = s_k + n_k \quad (3)$$

а оцінка корисного сигнал дорівнює

$$\hat{s}_k = y_k - \hat{n}_k = s_k + n_k - \hat{n}_k \quad (4)$$

Зводячи рівняння (4) у квадрат, отримуємо

$$\hat{s}_k^2 = s_k^2 + (n_k - \hat{n}_k)^2 + 2s_k(n_k - \hat{n}_k). \quad (5)$$

Обчислюючи математичне очікування виразів праворуч і ліворуч від знака “=”, отримуємо

$$E[\hat{s}_k^2] = E[s_k^2] + E[(n_k - \hat{n}_k)^2] + 2E[s_k(n_k - \hat{n}_k)]. \quad (6)$$

Оскільки корисний сигнал  $s_k$  не корелює з  $n_k$  або з  $\hat{n}_k$ , останній доданок у формулі (6) дорівнює нулю, так що отримуємо

$$E[\hat{s}_k^2] = E[s_k^2] + E[(n_k - \hat{n}_k)^2], \quad (7)$$

де  $E[s_k^2]$  представляє загальну потужність сигналу,  $E[\hat{s}_k^2]$  – оцінку потужності сигналу (а також загальну вихідну потужність), а  $E[(n_k - \hat{n}_k)^2]$  – залишкову потужність шуму, який все ще може утримуватися в  $s_k$ . З рівняння (7) очевидно, що, якщо оцінка  $\hat{n}_k$  це точна копія  $n_k$ , вихідна потужність це потужність тільки корисного сигналу. При налаштуванні адаптивного фільтра до оптимального положення мінімізується потужність залишкового шуму, а отже, загальна вихідна потужність. На потужність корисного сигналу налаштування не впливає, оскільки  $s_k$  не корелює з  $n_k$ . Отже,

$$\min E[\hat{s}_k^2] = E[s_k^2] + \min E[(n_k - \hat{n}_k)^2], \quad (8)$$

З рівняння (8) очевидно, що сумарний ефект мінімізації загальної вихідної потужності полягає у максимізації вихідного відношення сигнал-шум. Якщо фільтр налаштований так, що  $\hat{n}_k = n_k$ , то  $s_k = s_k$ . У цьому випадку вихід адаптивного шумоподавлювача вільний від шуму. Якщо сигнал  $y_k$  не містить шуму, тобто. коли  $n_k = 0$ , адаптивний фільтр само вимикається (принаймні теоретично) шляхом присвоєння всім ваговим коефіцієнтам нульових значень.

### 3. Основи теорії фільтрів Вінера

Багато адаптивних алгоритмів можна розглядати як апроксимації дискретного фільтра Вінера (рис. 5). На фільтр одночасно подаються два сигнали  $x_k$  і  $y_k$ . Зазвичай  $y_k$  складається з компонента, який корелює з  $x_k$ , і компонента, який з  $x_k$  не корелює. Вінерівський фільтр видає оптимальну

оцінку тієї частини  $Y_k$ , яка корелює з  $x_k$ , потім ця оцінка віднімається з  $Y_k$  і виходить  $e_k$ .

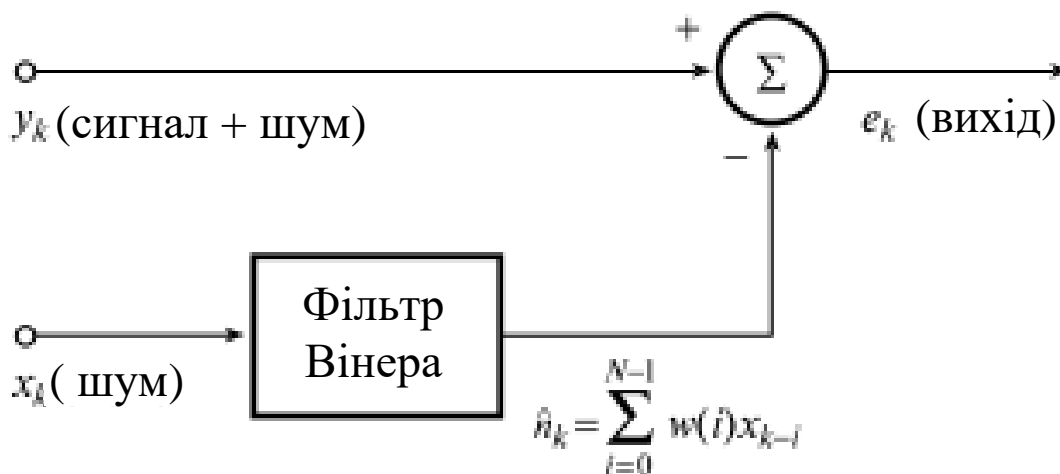


Рис. 5. Фільтр Вінера (стандартна форма)

Припустимо, що використовується КІХ-фільтр з  $N$  коефіцієнтами (або ваговими коефіцієнтами - дуже популярний термін у літературі). Помилка  $e_k$  між виходом фільтра Вінера і вихідним сигналом  $Y_k$  записується як

$$e_k = y_k - \hat{n}_k = y_k - W^T X_k = y_k - \sum_{i=0}^{N-1} w(i)x_{k-i} \quad (9)$$

де  $X_k$  і  $W^T$  — вектор вхідного сигналу та вектор вагових коефіцієнтів відповідно, що мають вигляд

$$X_k = \begin{bmatrix} x_k \\ x_{k-1} \\ \vdots \\ x_{k-(N-1)} \end{bmatrix}, \quad W = \begin{bmatrix} w(0) \\ w(1) \\ \vdots \\ w(N-1) \end{bmatrix}. \quad (10)$$

Квадрат помилки дорівнює

$$e_k^2 = y_k^2 - 2y_k X_k^T W + W^T X_k X_k^T W. \quad (11)$$

Щоб отримати середньоквадратичну помилку  $J$ , потрібно знайти математичне очікування обох частин рівняння (11), припускаючи, що вектор входу  $X_k$  і сигнал  $Y_k$  спільно стаціонарні:

$$\begin{aligned} J &= E[e_k^2] = E[y_k^2] - 2E[y_k X_k^T W] + E[W^T X_k X_k^T W] = \\ &= \sigma^2 + 2P^T W + W^T R W, \end{aligned} \quad (12)$$

де  $E[\cdot]$  означає математичне очікування,  $\sigma^2 = E[y_k^2]$  – дисперсія  $y_k$ ,  $P = E[y_k X_k]$  –  $N$ -компонентний вектор взаємної кореляції, а  $R = E[X_k X_k^T]$  – автокореляційна матриця  $N \times N$ . Графік залежності середньоквадратичної помилки від коефіцієнтів  $W$  – це чашоподібна поверхня з однозначно визначеною нижньою точкою (див. рис. 6). Ця структура називається поверхнею продуктивності (performance surface), і вона не негативна. Градієнт поверхні продуктивності записується так:

$$\nabla = \frac{dJ}{dW} = -2P + 2RW. \quad (13)$$

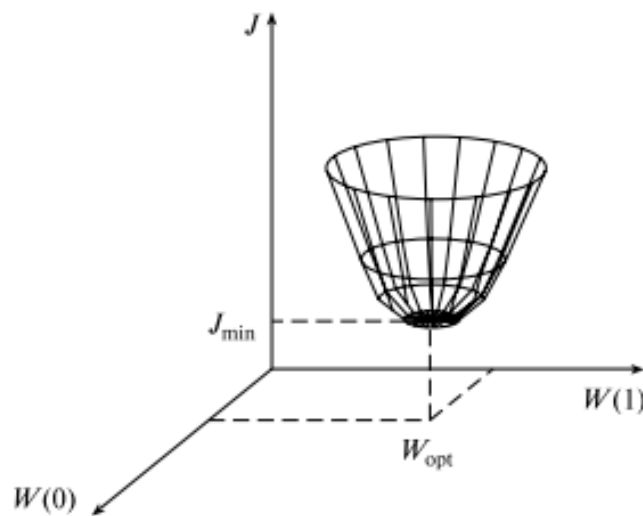


Рис. 6. Поверхня продуктивності

Кожен набір коефіцієнтів  $w(i)$  ( $i = 0, 1, \dots, N - 1$ ) відповідає точці на поверхні. У точці мінімуму градієнт дорівнює нулю і вектор вагових коефіцієнтів фільтра досягає оптимального значення  $W_{opt}$  (див. приклад 2):

$$W_{opt} = R^{-1}P. \quad (14)$$

Формула (14) називається рівнянням Вінера-Хопфа. Завдання адаптивної фільтрації полягає в налаштуванні вагових коефіцієнтів фільтра  $w(0)$ ,  $w(1)$ , ... за допомогою відповідного алгоритму до досягнення оптимальної точки на поверхні продуктивності.

Практичне використання вінерівських фільтрів обмежене через такі недоліки:

- вимагає застосування автокореляційної матриці  $R$  та вектора взаємної кореляції  $P$ , причому обидві величини апіорі невідомі;
- фільтри включають звернення матриць, що є трудомістким процесом;
- якщо сигнали нестационарні, то  $R$  і  $P$  будуть змінюватися з часом, так що  $W_{opt}$  доведеться обчислювати багаторазово.



У додатках реального часу  $W_{opt}$  потрібно перераховувати в міру надходження вибірок. Використання адаптивних алгоритмів дозволяє не звертати при цьому матриці та не обчислювати явно  $R$  та  $P$ .

## Приклад 2

Починаючи з виразу для середньоквадратичної помилки (12), виведіть рівняння Вінера-Хопфа.

*Рішення*

Середньоквадратична помилка записується так:

$$J = \sigma^2 + 2P^T W + W R W. \quad (15)$$

Градiєнт  $\nabla$  помилки  $J$  виходить диференціюванням  $J$  по вектору вагових коефіцієнтів  $W$  і прирівнюванням результату до нуля (див. [4]):

$$\nabla = \frac{dJ}{dW} = \frac{d\sigma^2}{dW} + \frac{d(P^T W)}{dW} + \frac{d(W^T R W)}{dW}. \quad (16)$$

У нашому випадку

$$\begin{aligned} \frac{d\sigma^2}{dW} &= 0, \\ \frac{d(2P^T W)}{dW} &= -2P, \\ \frac{d(W^T R W)}{dW} &= 2RW. \end{aligned}$$

Використовуючи ці результати та поклавши  $\nabla = 0$  у рівнянні (16), отримуємо

$$\nabla = \frac{dJ}{dW} = -2P + 2RW. \quad (17)$$

Вектор оптимальних коефіцієнтів записується у вигляді

$$W_{opt} = R^{-1} P. \quad (18)$$

## 4. Стандартний адаптивний алгоритм найменших квадратів

Одним із найбільш вдалих адаптивних алгоритмів є алгоритм найменших квадратів, розроблений групою Уїдроу. Натомість обчислення  $W_{opt}$  за один крок за формулою (18) у схемі найменших квадратів коефіцієнти налаштовуються при послідовній обробці вибірок з мінімізацією середньоквадратичної помилки. Результат рівносильний спуску поверхнею, зображеної на рис. 6 до її дна.

Схема найменших квадратів заснована на алгоритмі якнайшвидшого спуску,

в якому вектор вагових коефіцієнтів наступним чином оновлюється при обробці вибірок:

$$\mathbf{W}_{k+1} = \mathbf{W} - \mu \nabla_k \quad (19)$$

де  $\mathbf{W}_k$  і  $\nabla_k$  – вектори вагових коефіцієнтів і дійсного градієнта відповідно в  $k$ -й момент вибірки; параметр  $\mu$  визначає стійкість та швидкість збіжності.

Алгоритм найшвидшого спуску у формулі (19), як і раніше, вимагає знання  $\mathbf{R}$  і  $\mathbf{P}$ , оскільки ці величини необхідні для отримання (див. формулу (18)). Алгоритм найменших квадратів – це практичний метод оцінки вагових коефіцієнтів фільтра  $\mathbf{W}_k$  в реальному часі без обігу матриць (див. формулу (18)) або прямого обчислення автокореляційної матриці та вектора взаємної кореляції. Алгоритм найменших квадратів Уїдроу-Хопфа послідовного оновлення вагових коефіцієнтів записується так:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + 2\mu e_k \mathbf{X}_k, \quad (20)$$

де

$$e_k = y_k - \mathbf{W}^T \mathbf{X}_k.$$

Очевидно, що алгоритм найменших квадратів не вимагає апріорних знань статистики сигналу (тобто кореляції  $\mathbf{R}$  і  $\mathbf{P}$ ), а використовує їх миттєві оцінки (див. приклад 3). Вагові коефіцієнти, отримані за допомогою алгоритму найменших квадратів, – це лише оцінки, які поступово покращуються у міру того, як налаштовуються коефіцієнти та фільтр вивчає характеристики сигналів. Зрештою вагові коефіцієнти сходяться до якихось значень. Умова збіжності:

$$0 < \mu < 1/\lambda_{\max}, \quad (21)$$

де  $\lambda_{\max}$  – максимальне значення коваріаційної матриці даних. На практиці  $\mathbf{W}_k$  ніколи не досягає теоретичного максимуму (рішення Вінера), а флюктує близько цього значення (рис. 7).

#### 4.1. Реалізація стандартного алгоритму найменших квадратів

Нижче наводиться схема обчислень, прийнята алгоритмі найменших квадратів.

1. Спочатку кожному ваговому коефіцієнту  $w_k(i)$ ,  $i = 0, 1, \dots, N - 1$  надається довільне фіксоване значення, наприклад, 0.

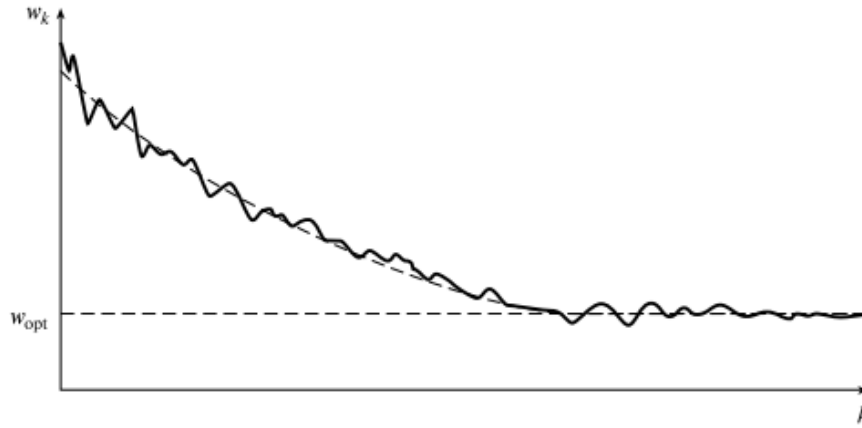


Рис. 7. Зміна вагових коефіцієнтів фільтра

У кожний наступний момент вибірки  $k = 1, 2, \dots$  виконуються етапи 2-4.

2. Обчислити вихід фільтра

$$\hat{n}_k = \sum_{i=0}^{N-1} w_k(i) x_{k-i}$$

3. Обчислити оцінку помилки

$$e_k = y_k - \hat{n}_k.$$

4. Оновити набір вагових коефіцієнтів

$$w_{k+1}(i) = w_k(i) + 2\mu e_k x_{k-i}.$$

Простота алгоритму найменших квадратів і легкість реалізації, очевидні з наведених вище викладок, пояснюють, чому даний метод використовується в системах реального часу, коли це можливо. Описаний алгоритм вимагає приблизно  $2N + 1$  операцій множення і  $2N + 1$  операцій додавання для кожного нового набору вхідних і вихідних вибірок. Основні операції множення-накопичення, що фігурують в алгоритмі, можна виконувати на більшості процесорів сигналів, що наводить на думку використовувати пряму реалізацію алгоритму.

Функціональна схема алгоритму найменших квадратів показано на рис. 8. На рис. 9 та 10 відповідно зображені псевдокоди програми та апаратної реалізації.



Рис. 8. Функціональна схема адаптивної фільтрації за схемою найменших квадратів

```

Вхід:  $x_k(i)$  вектор останніх вхідних вибірок
 $y_k$  поточна зашумлена вибірка сигналу
 $w_k(i)$  вектор коефіцієнтів фільтра
Вихід:  $e_k$  поточна бажана вихідна вибірка (або помилка)
 $w_k(i)$  вектор оновлених коефіцієнтів фільтра
/*розрахунок поточної оцінки помилки*/
 $e_k = y_k$ 
for i=1 to N do
 $e_k = e_k - x_k(i) * w_k(i)$ 
end
/*оновлення коефіцієнтів фільтра*/
 $g_k = 2u * e_k$ 
for i=1 to N do
 $w_k(i) = w_k(i) + x_k(i) * g_k$ 
end return

```

Рис. 10.9. Код реалізації адаптивної фільтрації за схемою найменших квадратів

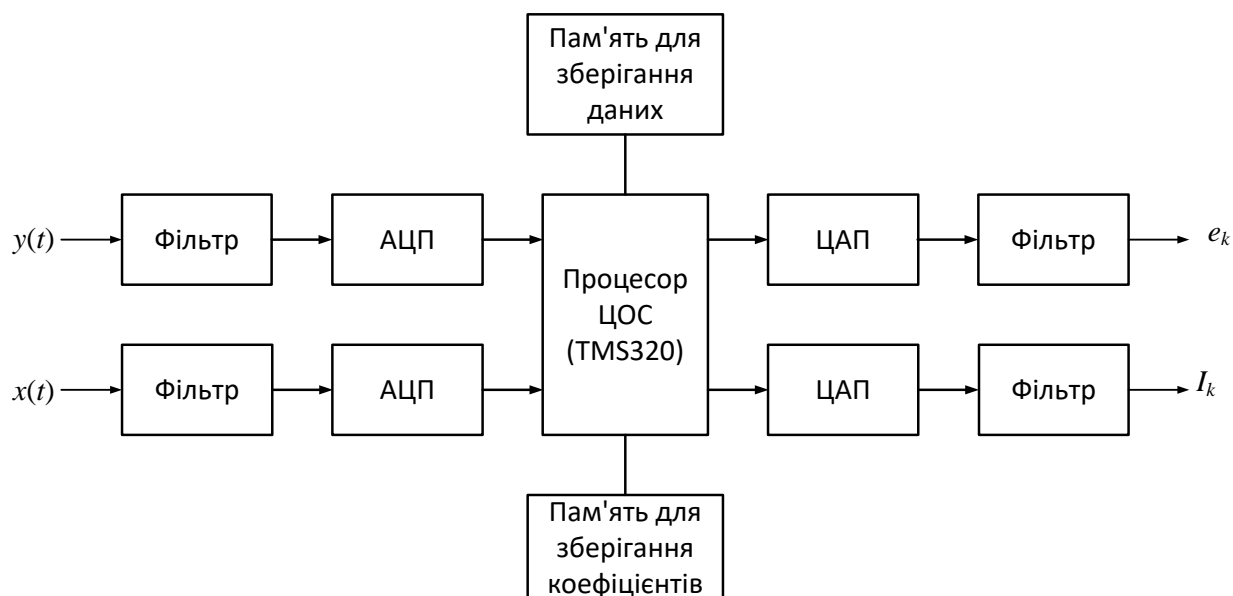


Рис. 10. Апаратна реалізація адаптивної фільтрації реального часу за схемою найменших квадратів

### Приклад 3

Почавши з алгоритму найшвидшого спуску

$$\mathbf{W}_{k+1} = \mathbf{W} - \mu \nabla_k$$

де  $\mathbf{W}_k$  – вектор вагових коефіцієнтів фільтра в момент  $k$ -ої вибірки, параметр  $\mu$  визначає стійкість і швидкість збіжності,  $\nabla_k$  - дійсний градієнт на поверхні продуктивності, виведіть алгоритм найменших квадратів Уїдрой-Хопфа для адаптивного шумопридушення, вказавши зроблені припущення.

*Рішення*

Алгоритм найшвидшого спуску формулюється так:

$$\mathbf{W}_{k+1} = \mathbf{W} - \mu \nabla_k \quad (22)$$

Вектор-градієнт  $\nabla$ , взаємна кореляція первинного та вторинного входів  $\mathbf{P}$  та автокореляція первинного входу  $\mathbf{R}$  пов'язані співвідношенням

$$\nabla = -2\mathbf{P} + 2\mathbf{R}\mathbf{W} \quad (23)$$

В алгоритмі найменших квадратів при обчисленні  $\nabla$  використовуються миттєві значення. Отже,

$$\begin{aligned} \nabla_k &= -2\mathbf{P}_k + 2\mathbf{R}_k\mathbf{W}_k = -2\mathbf{X}_k y_k + 2\mathbf{X}_k\mathbf{X}_k^T \mathbf{W}_k = \\ &= -2\mathbf{X}_k (y_k - \mathbf{X}_k^T \mathbf{W}_k) = -2 e_k \mathbf{X}_k \end{aligned} \quad (24)$$

де

$$e_k = y_k - \mathbf{X}_k^T \mathbf{W}_k.$$

Підставляючи рівняння (24) у вираз для алгоритму найшвидшого спуску, одержуємо стандартний алгоритм найменших квадратів Уїдрой-Хопфа:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + 2\mu e_k \mathbf{X}_k \quad (25,a)$$

де

$$e_k = y_k - \mathbf{W}_k^T \mathbf{X}_k. \quad (25,b)$$

## 4.2. Практичні обмеження стандартного алгоритму найменших квадратів

Застосування стандартного алгоритму найменших квадратів пов'язане з деякими проблемами, що призводить до зниження продуктивності системи. Розглянемо ці проблеми.

### 4.2.1. Вплив нестационарності

У стаціонарному середовищі поверхня продуктивності фільтра має постійну форму і орієнтацію, а характеристики адаптивного фільтра просто сходяться до оптимального значення або коливаються біля нього. Якщо статистики сигналу змінюються після сходження вагових коефіцієнтів до оптимального значення, характеристика фільтра повинна також перейти в новий набір оптимальних значень шляхом повторного налаштування вагових коефіцієнтів. При цьому

вважається, що зміна статистик сигналу відбувається досить повільно, щоб коефіцієнти встигали збігатися до оптимального значення між змінами. Проте в нестационарному середовищі точка мінімуму безперервно рухається, крім того, можуть змінюватися орієнтація та кривизна траєкторії її руху (див. рис. 11). Отже, у такому випадку алгоритм повинен не тільки знайти точку мінімуму поверхні, а й відстежити її мінливе становище, що значно знижує продуктивність схеми. Змінна називається нестационарною, якщо її статистики (такі як середнє, дисперсія, автокореляція) змінюються з часом. Причиною таких змін можуть бути, наприклад, випадкові перешкоди малої тривалості (рис. 12) або неправильні дані, причому такі зміни часто призводять до розладу вагових коефіцієнтів фільтра (у цьому випадку особливо яскраво виявляються недоліки лінійних фільтрів).

Зараз розроблено кілька схем, у яких ця проблема зникає, але це ускладнює стандартний алгоритм найменших квадратів. Однією з таких схем є впорядкований у часі адаптивний фільтр.

#### 4.2.2. Вплив компонентів сигналу на перешкоди у вхідному каналі

Продуктивність алгоритму залежить від виміряного сигналу перешкоди  $x_k(i)$ , який сильно корелює з реальною перешкодою, але слабо корелює (теоретично зовсім не корелює) з корисним сигналом. У більшості випадків ця умова не виконується. У деяких програмах зашумлений вхідний сигнал може містити небажану перешкоду, і слабкі компоненти сигналу.

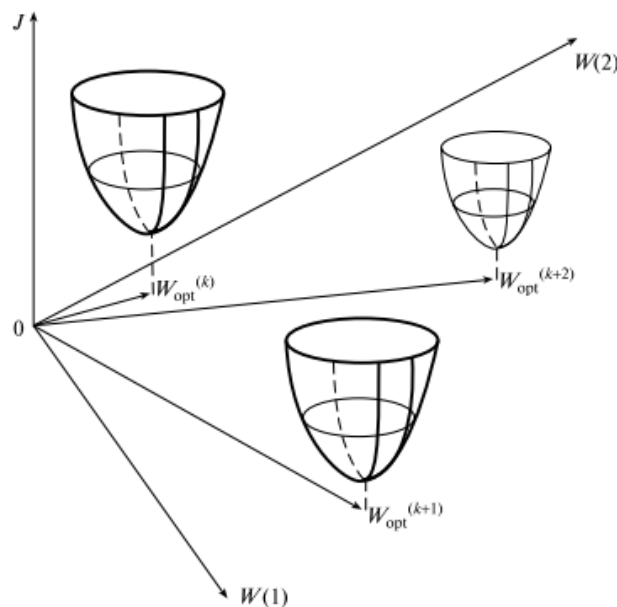


Рис. 11. Нестационарна поверхня продуктивності

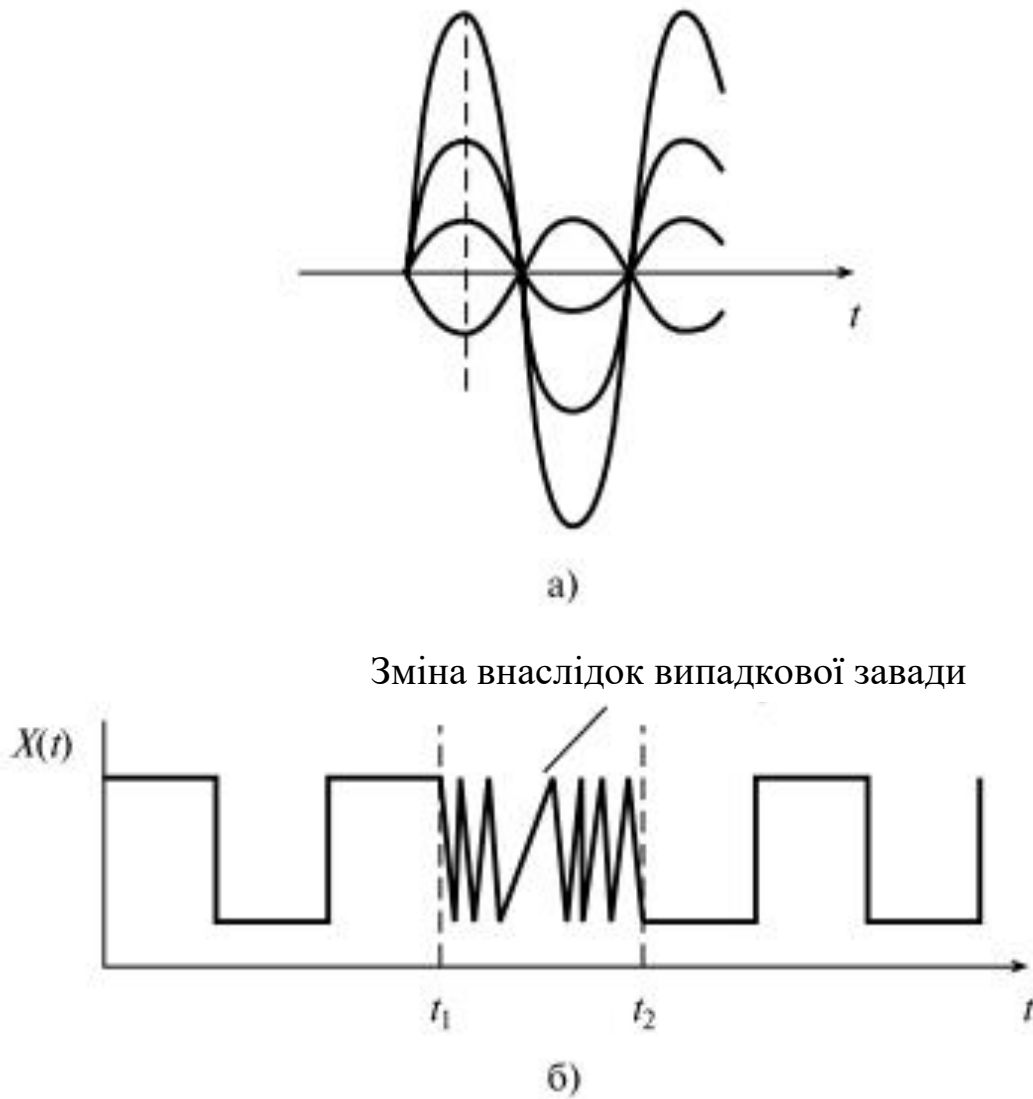


Рис. 12. Ілюстрація нестационарних процесів: а) модульований сигнал; б) спорадична інтерференція

В результаті разом із шумом пригнічуються деякі компоненти корисного сигналу. Подібна ситуація ілюструється на рис. 13. У роботі [11] показано, що в цих випадках процес адаптивного шумопридушення як і раніше, веде до значного поліпшення відношення корисний сигнал-шум, але тільки за рахунок невеликого спотворення сигналу. Втім, якщо  $x_k$  містить тільки сигнали і зовсім не містить шумових компонентів, корисний сигнал в  $y_k$  може бути повністю знищений.



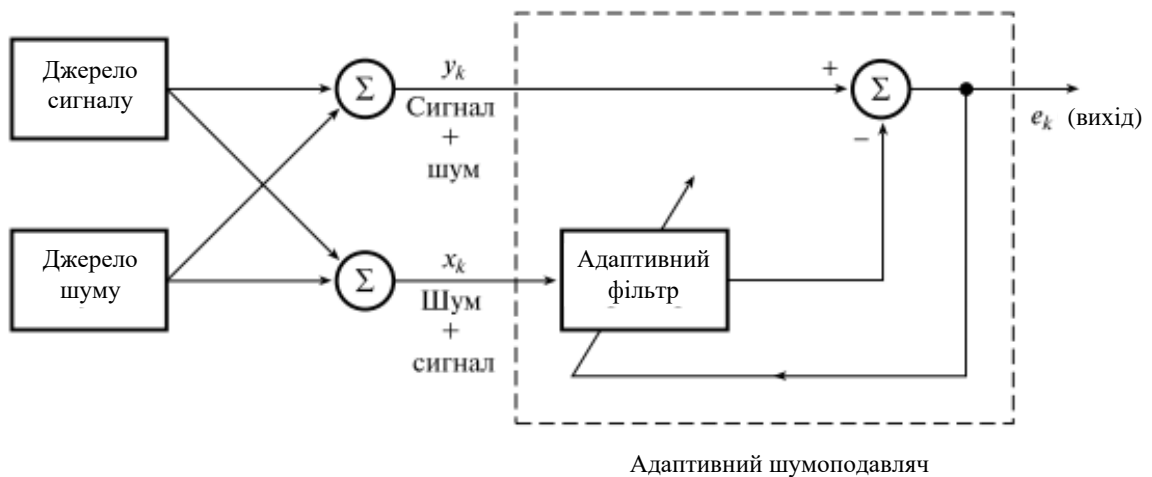


Рис. 13. Адаптивне шумозаглушення за наявності компонентів сигналу в каналі перешкоди та каналі корисного сигналу

### 4.2.3. Вимоги до довжини комп'ютерного подання

Адаптивний КІХ-фільтр, заснований на методі найменших квадратів, характеризується такими рівняннями:

$$\text{цифровий фільтр} \quad \hat{n}_k = \sum_{i=0}^{N-1} w_k(i) x_{k-i} \quad (26,а)$$

$$\text{адаптивний алгоритм} \quad \mathbf{W}_{k+1} = \mathbf{W}_k + 2\mu e_k \mathbf{X}_k \quad (26,б)$$

де

$$e_k = y_k - \mathbf{W}_k^T \mathbf{X}_k.$$

Коли адаптивні фільтри реалізуються практично, вагові коефіцієнти фільтра  $w_k$  і входні змінні  $x_k$  і  $y_k$  неминуче представляються кінцевим числом бітів. Таким чином, необхідні чисельні операції виконуються з використанням арифметики кінцевої точності. Рекурсивна природа алгоритму найменших квадратів означає, що довжина слова зростатиме без обмежень і деякі біти доведеться відкидати перед збереженням кожного оновленого вагового коефіцієнта.

Отже,  $y_k$ ,  $e_k$  і  $w_k(i)$  можуть істотно відрізнятиметься від своїх істинних значень. Використання вагових коефіцієнтів фільтра та результатів арифметичних операцій з обмеженою точністю може вводити помилки в адаптивні фільтри, вплив яких проявляється так: 1) можлива несхожість адаптивного фільтра до оптимального рішення, що зробить його малоефективним (наприклад, якщо фільтр використовується як шумоподавляч, можливий певний залишковий рівень завад); 2) вихід фільтра може містити шум, що призведе до випадкових коливань виходу; 3) можливе передчасне завершення алгоритму. Отже, щоб утримати ці помилки в межах допустимого,

потрібно використовувати достатню кількість бітів. У більшості адаптивних систем, описаних у літературі, цифрові сигнали  $x_{k-i}$  і  $y_k$  представляються як числа з фіксованою комою, мають довжину від 8 до 16 біт, а квантовані коефіцієнти мають довжину від 16 до 24 біт. Розмір помножувачів змінюється в межах від  $8 \times 8$  біт до  $24 \times 16$  біт, а розмір накопичувачів - від 16 до 40 біт. Схоже, що для фільтрів невеликих порядків (до 100 коефіцієнтів) достатньо зберігати коефіцієнти не більше ніж з 16-бітовою точністю та використовувати помножувач  $16 \times 16$  біт з накопичувачем довжиною 32 біт.

#### 4.2.4. Відхід коефіцієнтів

За наявності входів певних типів (наприклад, вузькосмугових сигналів) коефіцієнти фільтра можуть поступово зміщуватися від оптимального значення і повільно зростати, з часом перевищуючи дозволена довжину слова. Дана проблема характерна для алгоритмів найменших квадратів і в перспективі призводить до погіршення продуктивності. Щоб перешкоджати відходу коефіцієнтів, практично вводиться коефіцієнт втрат, який трохи зміщує коефіцієнти до нуля. Два приклади подібних схем наведено нижче.

$$w_{k+1}(i) = \delta w_k(i) + 2\mu e_k x_{k-i} \quad 0 < \delta < 1. \quad (27,a)$$

$$w_{k+1}(i) = w_k(i) + 2\mu e_k x_{k-i} \pm \delta \quad 0 < \delta < 1. \quad (27,b)$$

Невеликий коефіцієнт втрат  $\delta$  гарантує, що відхід коефіцієнтів присутній, але вводить в помилку  $e_k$  псевдовипадковий шум.

Як згадувалося раніше, ефективність стандартного алгоритму найменших квадратів можна збільшити, дещо його модифікувавши. Назвемо деякі покращені версії схеми найменших квадратів.

1. Комплексний алгоритм найменших квадратів, що дозволяє обробляти комплексні дані.
2. Блоковий алгоритм найменших квадратів, що має значну обчислювальну перевагу та в деяких випадках швидшу збіжність.
3. Алгоритми найменших квадратів з упорядкуванням у часі, що дозволяють вирішувати деякі нестационарні завдання.

### 4.3. Інші алгоритми з урахуванням схеми найменших квадратів |

#### 4.3.1. Комплексний алгоритм найменших квадратів

Оновлення вагових коефіцієнтів фільтра за допомогою комплексного алгоритму найменших квадратів викладено згідно з [12]:

$$\tilde{W}_{k+1} = \tilde{W}_k + 2\mu \tilde{e}_k \tilde{X}_{k-i}, \quad (28)$$

де символом "Тільда" позначена комплексна змінна. Комплексний алгоритм

найменших квадратів ідеально втілюється з урахуванням процесорів PDSP16XXX (Mitel), які можуть виконувати арифметичні операції безпосередньо з комплексними даними, що вигідно відрізняє їхню відмінність від звичайних процесорів.

#### 4.3.2. Швидкі алгоритми найменших квадратів

Значно заощадити на обчисленнях, особливо при великій кількості коефіцієнтів фільтра, дозволяють кілька блокових алгоритмів найменших квадратів. Для зменшення обсягу обчислень дані таких алгоритмах обробляються блоками, а не окремими вибірками. Вивчивши уявлення у частотній області реалізації блочного алгоритму найменших квадратів, можна назвати, наскільки вигідно під час виконання згортки використовувати швидке перетворення Фур'є (ШПФ). Відповідний ефективний фільтр, представлений у частотній області, зображений на рис. 14

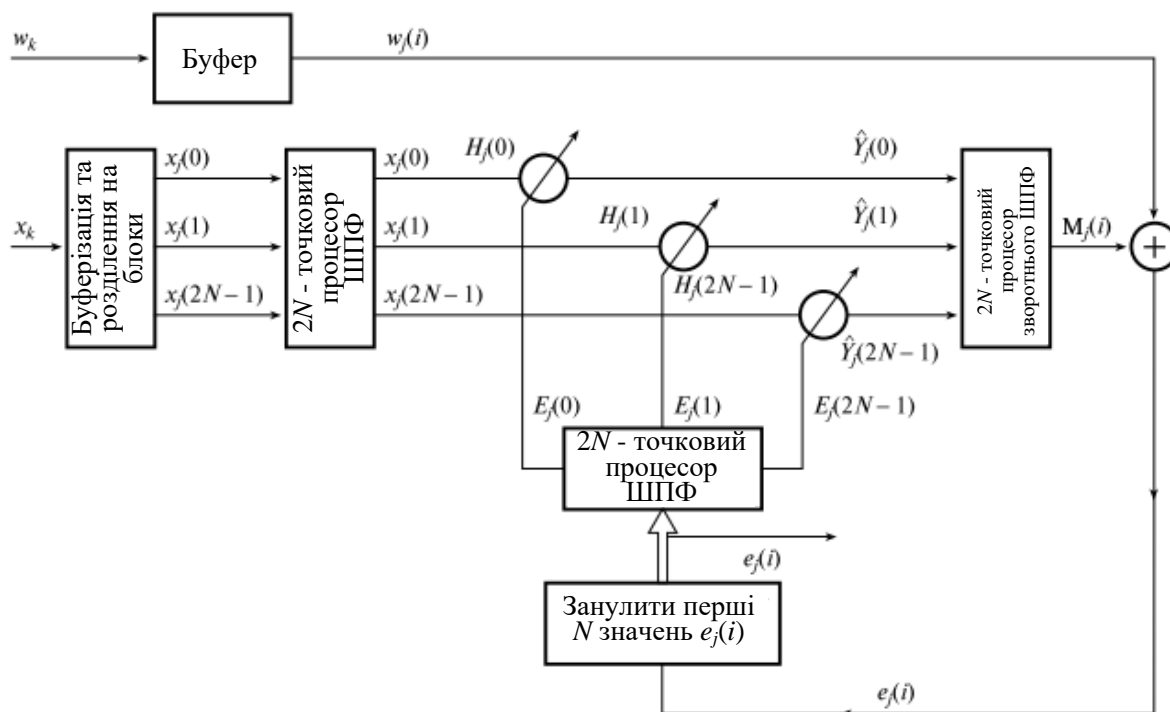


Рис. 14. Спрощена блок-схема фільтру, побудованого на основі схеми найменших квадратів

#### 10.5. Рекурсивний алгоритм найменших квадратів

Рекурсивний алгоритм найменших квадратів ґрунтується на добре відомій схемі найменших квадратів (рис. 15).

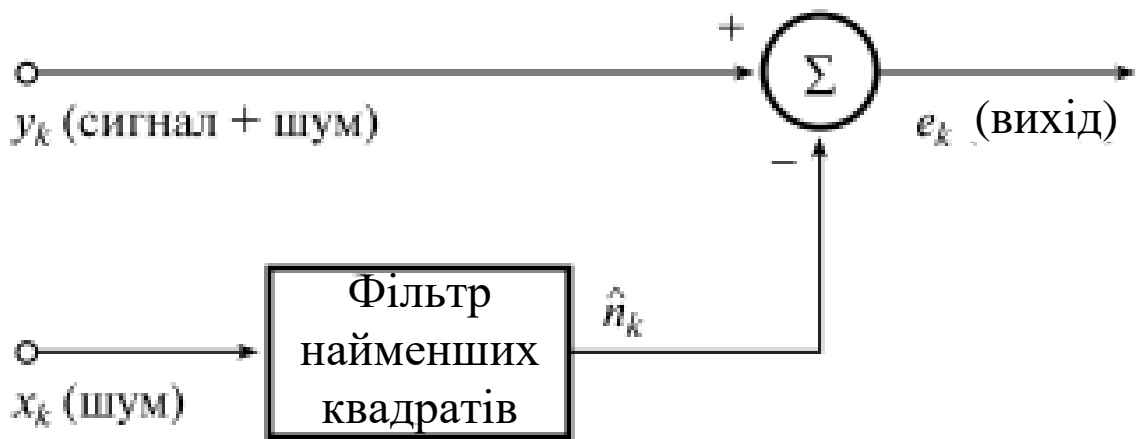


Рис. 15. Ілюстрація основної ідеї методу найменших квадратів

У відповідь на набір вхідних сигналів  $x_k(i)$ ,  $i = 1, 2, \dots, n$  виходить вихідний сигнал  $y_k$ , який вимірюється в дискретні моменти часу  $k$ . Вхідні та вихідний сигнали пов'язані наступною регресійною моделлю:

$$y_k = \sum_{i=0}^{N-1} w(i)x_k(i) + e_k \quad (29)$$

де  $e_k$  представляє помилки виміру або інші ефекти, які не можна врахувати, а  $w(i)$  представляє частку  $i$ -го входу в первинному сигналі  $y_k$ . Завдання найменших квадратів формулюється як отримання за даними  $x_k(i)$  і  $y_k$  оцінок величин з  $w(0)$  по  $w(n-1)$ .

Оптимальні оцінки (за схемою найменших квадратів) вагових коефіцієнтів фільтра  $w(i)$  визначаються виразом

$$\mathbf{W}_m = [\mathbf{X}_m^T \mathbf{X}_m]^{-1} \mathbf{X}_m^T \mathbf{Y}_m \quad (30)$$

де  $\mathbf{Y}_m$ ,  $\mathbf{W}_m$  і  $\mathbf{X}_m$  записуються як

$$\mathbf{Y}_m = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{m-1} \end{bmatrix} \quad \mathbf{X}_m = \begin{bmatrix} \mathbf{x}^T(0) \\ \mathbf{x}^T(1) \\ \mathbf{x}^T(2) \\ \vdots \\ \mathbf{x}^T(m-1) \end{bmatrix} \quad \mathbf{W}_m = \begin{bmatrix} w(0) \\ w(1) \\ w(2) \\ \vdots \\ w(n-1) \end{bmatrix}$$

$$\mathbf{x}^T(k) = [x_k(0) \quad x_k(1) \quad \dots \quad x_k(n-1)], \quad k = 0, 1, \dots, m-1.$$

Індекс  $m$  показує, що кожна з наведених вище матриць обчислюється для всіх інформаційних точок, а через  $T$  позначений транспонований вектор. Рівняння (30) визначає оптимальну оцінку  $\mathbf{W}_m$  за схемою найменших квадратів,

яку можна отримати за допомогою будь-якого зручного методу обігу матриць. Нарешті, вихід фільтра записується як

$$\hat{n}_k = \sum_{i=0}^{N-1} \hat{w}(i)x_{k-i}, k = 1, 2, \dots, m. \quad (31)$$

### 5.1. Рекурсивний алгоритм найменших квадратів

Обчислення  $W_m$  за формулою (30) вимагає трудомісткого обчислення зворотної матриці. Очевидно, що описаний метод найменших квадратів не підходить для фільтрації в реальному часі. Насправді при отриманні безперервних даних, коли потрібно поліпшити оцінку  $W_m$  з допомогою нових даних, більш доцільні рекурсивні методи. При рекурсивному методі найменших квадратів оцінки  $W_m$  можна оновлювати для кожного нового отриманого набору даних без прямого повторного трудомісткого звернення матриці.

Відповідний рекурсивний алгоритм виходить, якщо враховувати дані з експоненціально загасаючими ваговими коефіцієнтами, щоб поступово усунути вплив старих даних на  $W_m$  і дозволити відстежувати характеристики сигналу, що повільно міняються. Отже,

$$W_k = W_{k-1} + G_k e_k, \quad (32,a)$$

$$P_k = \frac{1}{\gamma} [P_{k-1} - G_k x^T(k) P_{k-1}], \quad (32,b)$$

де

$$G_k = \frac{P_{k-1} x(k)}{\alpha_k},$$

$$e_k = y_k - x^T(k) W_{k-1},$$

$$\alpha_k = \gamma + x^T(k) P_{k-1} x(k).$$

Власне, введення  $P_k$  дозволяє рекурсивно обчислювати зворотну матрицю  $[X_k^T X_k]^{-1}$ .

Аргумент  $k$  використовується, щоб підкреслити той факт, що величини обчислюються в кожен момент отримання вибірки;  $\gamma$  називається коефіцієнтом забування. Якщо покласти  $\gamma = 1$ , наведена схема зведеться до методу найменших квадратів. Значення  $\gamma$  зазвичай вибирається між 0,98 і 1. При менших значеннях найбільш свіжим даним присвоюються занадто великі вагові коефіцієнти, що призводить до сильної флуктуації оцінок. Число попередніх вибірок, що вносять значний внесок у  $W$  у кожний момент вибірки, називається асимптотичною довжиною вибірки (asymptotic sample length - ASL) і виражається такою формулою:

$$\sum_{k=0}^{\infty} \gamma^k = \frac{1}{1-\gamma} \quad (33)$$

Даний вираз ефективно визначає пам'ять фільтра, реалізованого з використанням рекурсивної схеми найменших квадратів. При  $\gamma = 1$  фільтр має нескінченну пам'ять.

## 5.2. Обмеження рекурсивного алгоритму найменших квадратів

Число арифметичних операцій з вибірками у рекурсивній схемі найменших квадратів дорівнює розмірності  $\mathbf{W}_k$  та  $\mathbf{P}_k$  у рівнянні (32). Втім, існує дві основні проблеми, які можуть зустрітися за безпосередньої реалізації рекурсивного алгоритму найменших квадратів. Перша, іменована “роздуванням” (“blow-up”), виникає, якщо сигнал  $x_k(i)$  дорівнює нулю протягом значного часу, внаслідок чого матриця  $\mathbf{P}_k$  зростатиме експоненційно в результаті ділення на  $\gamma$  ( $\gamma < 1$ ) в кожен момент отримання вибірки:

$$\lim_{k \rightarrow \infty} \mathbf{P}_k = \lim_{k \rightarrow \infty} \left( \frac{\mathbf{P}_{k-1}}{\gamma_{k-1}} \right) \quad (34)$$

Друга проблема рекурсивного алгоритму найменших квадратів - це його чутливість до помилок комп'ютерного округлення, через що матриця  $\mathbf{P}$  стає негативно визначеною, а згодом - нестійкою. Для успішної оцінки  $\mathbf{W}$  потрібно, щоб матриця  $\mathbf{P}$  була не негативно визначеною, що у методі найменших квадратів еквівалентно вимозі оборотності матриці  $\mathbf{X}^T \mathbf{X}$ . Однак внаслідок того, що в рівнянні (32,б) фігурує різниця величин, позитивну визначеність  $\mathbf{P}$  гарантувати не можна. Ця проблема може ще більше зіпсувати життя в багатопараметричних моделях, особливо якщо змінні лінійно залежні і алгоритм реалізований у невеликій системі з кінцевою довжиною слова. Після того, як в алгоритмі буде виконано багато ітерацій, два члени в дужках у рівнянні (32,б) можуть стати майже рівними. Віднімання таких членів у системі з кінцевою довжиною слова може призвести до помилок і того, що матриця  $\mathbf{P}_k$  стане негативно визначеною.

Для вирішення проблеми чисельної нестійкості можна відповідним чином факторизувати матрицю  $\mathbf{P}$  щоб уникнути віднімання в рівнянні (32,б). Подібні алгоритми факторизації краще сходяться, а їх точність можна порівняти з точністю рекурсивного алгоритму найменших квадратів з подвійною точністю. Два таких алгоритми – це алгоритм квадратного кореня та алгоритм UD-факторизації. З точки зору пам'яті та обчислювальної складності UD-алгоритм ефективніше, а отже, краще. Фактично UD-алгоритм - це формулювання алгоритму квадратного кореня без використання квадратного кореня, отже, властивості обох алгоритмів однакові.

## 5.3. Алгоритми факторизації

### 5.3.1. Алгоритм квадратного кореня

У методі квадратного кореня матриця  $P_k$  факторизується так:  $X_k^T X_k$

$$P_k = S_k S_k^T \quad (35)$$

де  $S_k$ , верхня трикутна матриця, і  $S_k^T$ , транспонована матриця  $S_k$ , рівні квадратному кореню з  $P_k$ . Отже, якщо в ході ітерацій оновлювати не  $P_k$ , а  $S_k$ , позитивна визначеність  $P_k$  гарантована, оскільки добуток двох квадратних коренів завжди позитивний. Величина  $S_k$  оновлюється таким чином:

$$S_k = S_{k-1} \frac{1}{\gamma^{1/2}} H_{k-1} \quad (35)$$

де  $H_{k-1}$  - верхня трикутна матриця.

### 5.3.2. Алгоритм UD-факторизації

При UD-факторизації  $P_k$  факторизується так:

$$P_k = U_k D_k U_k^T \quad (35)$$

де  $U_k$  – одинична верхня трикутна матриця,  $U_k^T$  – транспонована матриця  $U_k$ , а  $D_k$  - діагональна матриця. Отже, замість оновлення  $P_k$ , як у рекурсивному алгоритмі найменших квадратів, оновлюються її фактори  $U$  і  $D$ .