

Адаптивная фильтрация – сложившееся научно-техническое направление в цифровой обработке сигналов

Джиган Виктор Иванович

д.т.н.

член НТОРЭС им. А.С. Попова

Senior IEEE Member

главный научный сотрудник ГУП НПЦ «ЭЛВИС»

проф. каф. МРТУС МИЭТ

Зеленоград, Москва

Что такое адаптивная обработка сигналов – это что-то новое, забытое старое, или широко используемая современная технология ?

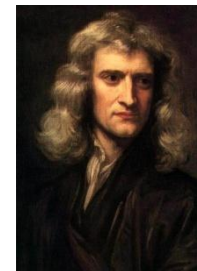
Адаптивная обработка сигналов – это направление в современной обработке сигналов, которая, в свою очередь является направлением в современных радиотехнике и связи

«Адаптация (позднелатинское adaptio – прилаживание, приспособление, от латинского adapto – приспособляю) – процесс приспособления строений и функций организмов (особей, популяций, видов) и их органов к условиям окружающей среды...»

Большая Советская Энциклопедия

Кто был первым ?

- **Йохан Карл Фредерик Гаусс (1777-1855)** в 1725 в возрасте 18 лет сформулировал критерий наименьших квадратов для обработки экспериментальных наблюдений. Этот критерий лежит в основе адаптивных фильтров, использующих рекурсивные алгоритмы по критерию наименьших квадратов
- **Исаак Ньютон (1643-1727)** предложил итерационный численный метод нахождения корня (нуля) заданной функции (также известный как метод касательных) . Этот метод лежит в основе современных градиентных адаптивных алгоритмов
- Работы по теории оптимального оценивания, лежащие в основе современной теории адаптивной обработки сигналов, относятся к 40-м – 50-м годам 20-го века:



Андрей Николаевич Колмогоров (1903 -1987)



Марк Григорьевич Крейн (1907-1989)



Норберт Винер (1894-1964)



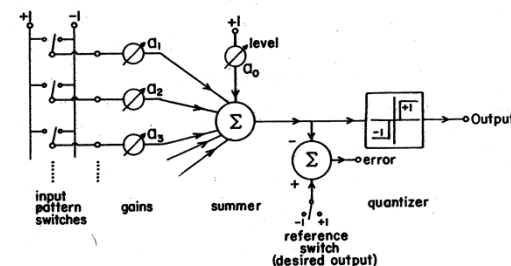
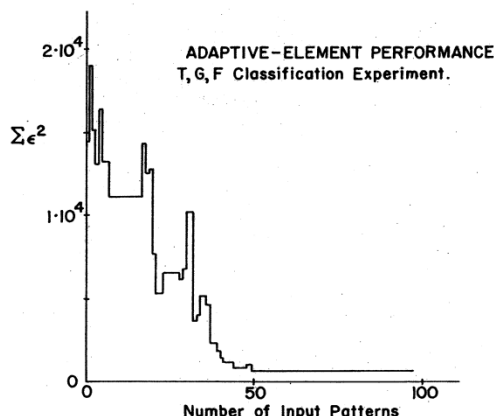
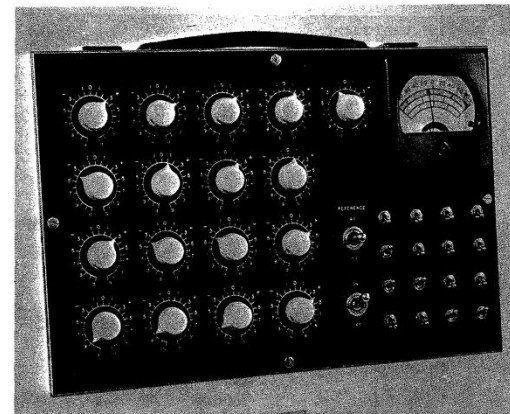
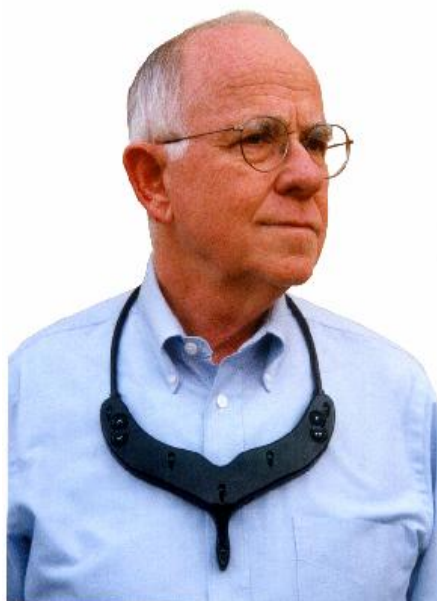
Норман Левинсон (1912-1975)

Кто сделал первый адаптивный фильтр ?

«Я был знаком с теорией винеровской фильтрации в непрерывной и дискретной формах. Для построения фильтра Винера требуется знать автокорреляционную функцию входного сигнала и взаимную корреляционную функцию между входным и требуемым сигналами. Это замечательно, когда вы делаете домашние упражнения, но что вам делать на практике, когда статистику входных сигналов никто вам не может предоставить? Все что у вас есть – это сами входные сигналы ...»

Бернард Уидроу - изобретатель LMS-алгоритма

Widrow B. Thinking about thinking: the discovery of the LMS algorithm – DSP history // IEEE Signal Processing Magazine. – 2005. – Vol. 22. – №1. – P. 100–106



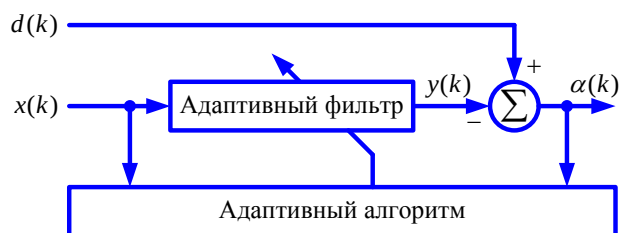
Widrow B. Hoff M.E. Adaptive switching circuits // IRE WESCON Convention Record. – 1960. – Part 4. – P. 96–104

Когда нужно использовать адаптивный фильтр?

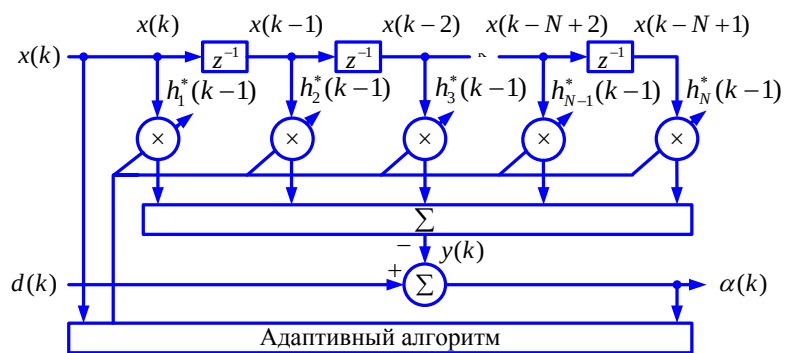
В основе большинства адаптивных устройств находится так называемый адаптивный фильтр. Фильтрация – это обработка, целью которой является извлечение интересующей информации, например, сигналов, обладающих определенными характеристиками

Свойства фильтра с фиксированными параметрами обычно определяются требуемой передаточной функцией. В свою очередь, передаточная функция определяет структуру фильтра и его вычислительную сложность. Если спецификацию к передаточной функции фильтра невозможно сформулировать заранее или когда эта спецификация может меняться в процессе работы фильтра, то вместо фильтров с фиксированными параметрами необходимо использовать фильтры с изменяемыми параметрами, например, адаптивные фильтры

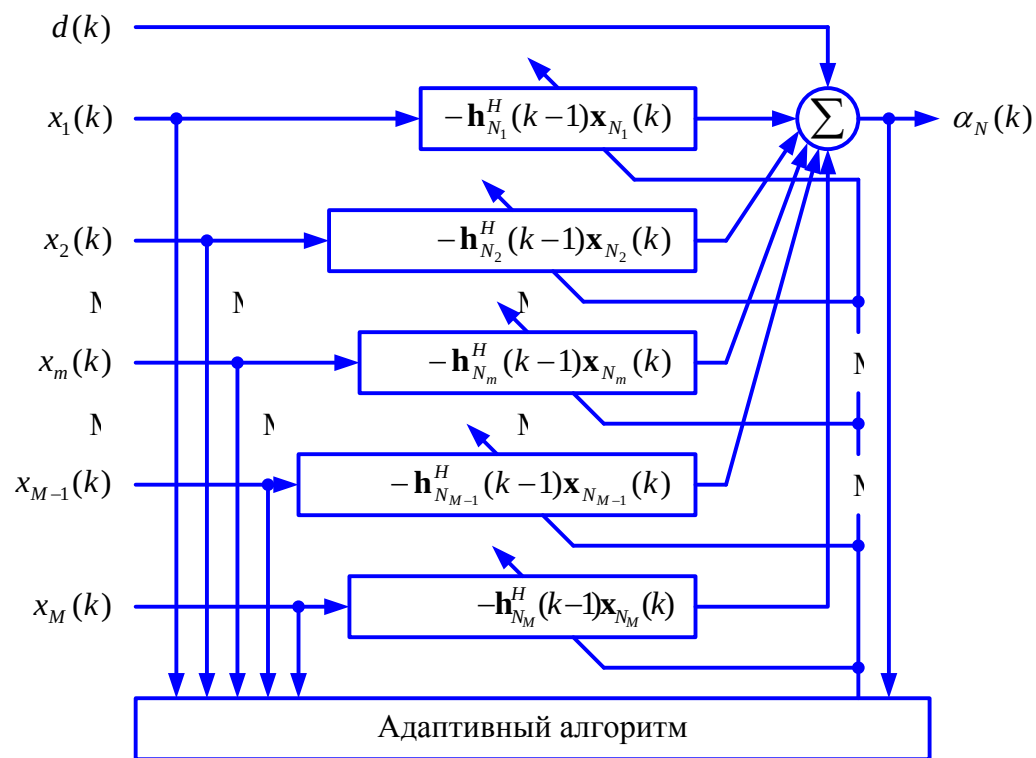
Какие бывают адаптивные фильтры (АФ) ?



Обобщенная структура АФ

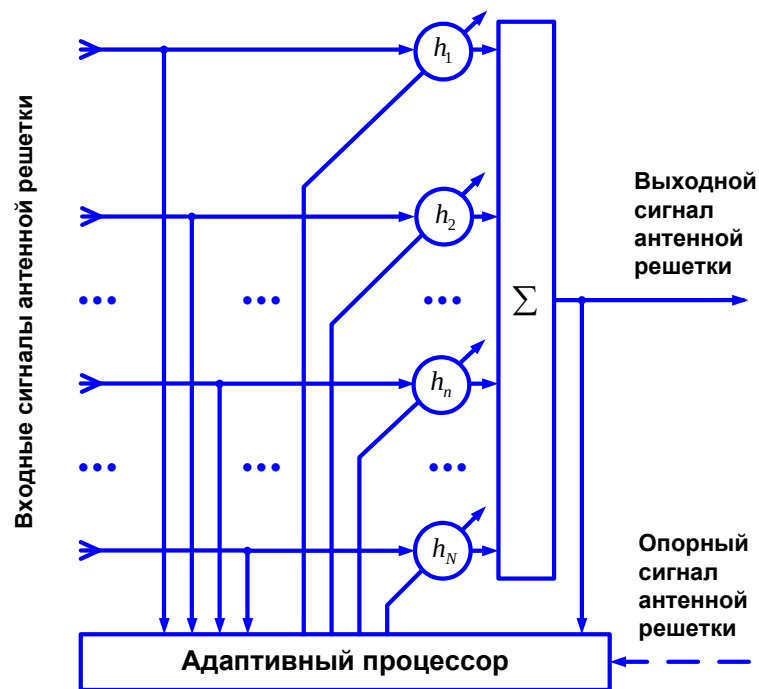


Одноканальный АФ

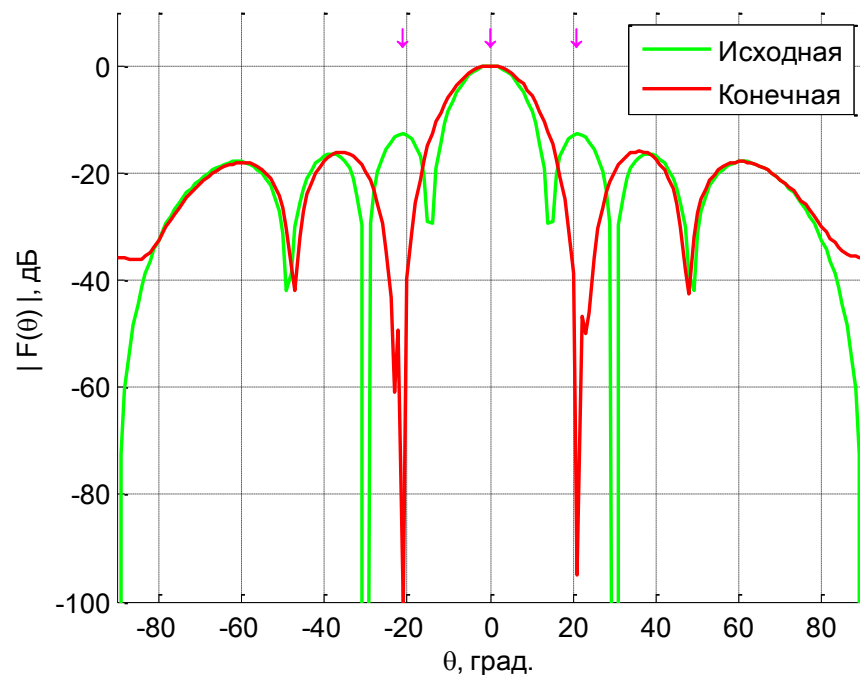


Многоканальный АФ

Где применяются адаптивные фильтры ? – Например, в адаптивных антенных решетках (ААР)

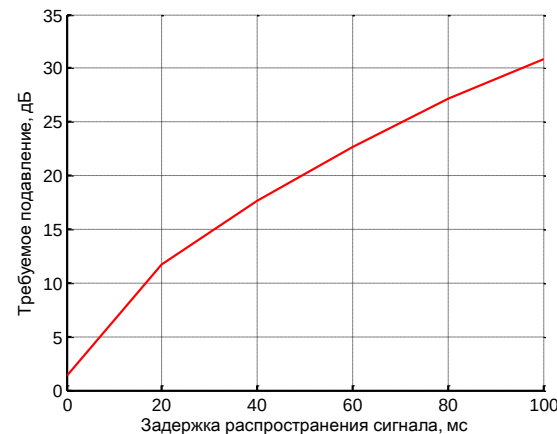
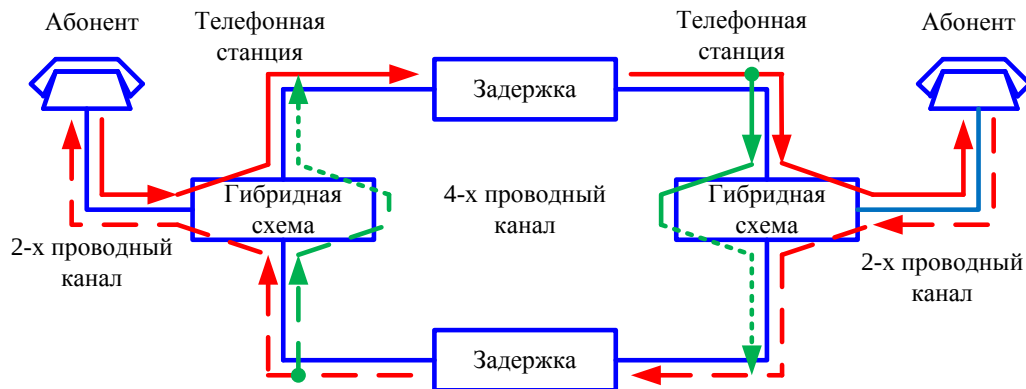


Структура ААР



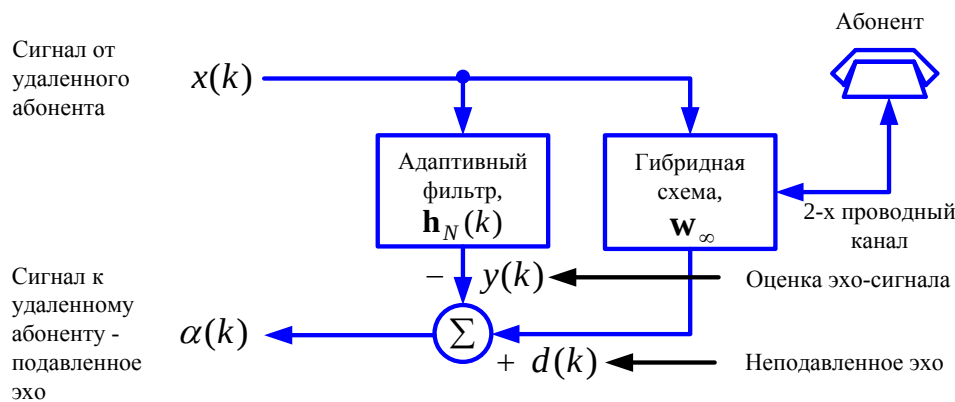
Принцип подавление помех с помощью ААР

Где применяются адаптивные фильтры ? – Например, в адаптивных эхокомпенсаторах

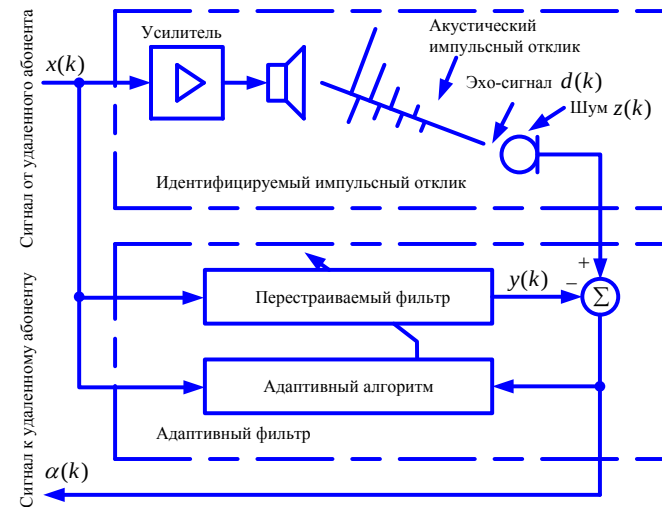


Телефонная связь

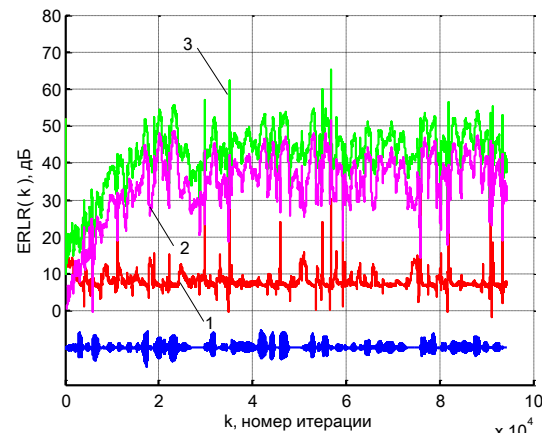
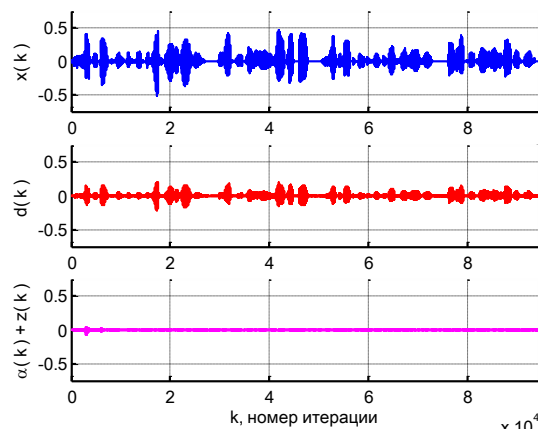
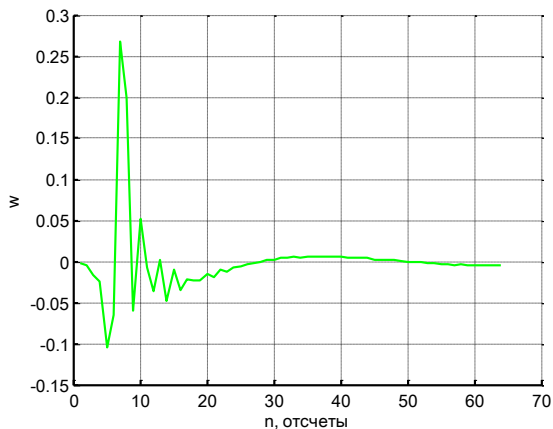
Требуемое подавление эхосигнала



Принцип эхокомпенсации



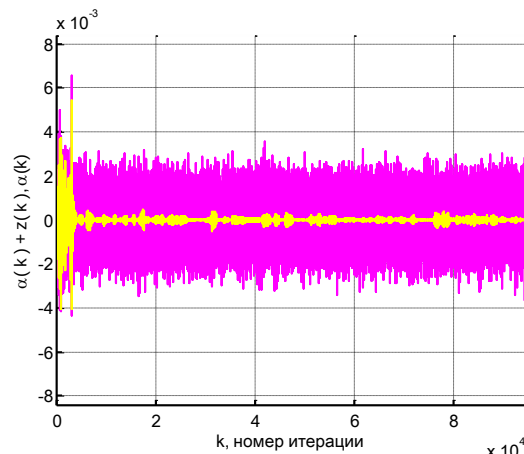
Насколько эффективна эхокомпенсация ?



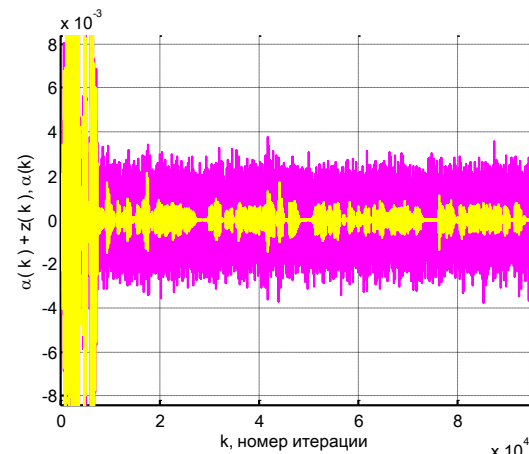
$$ERLE(k) = 10 \log_{10} \frac{E[x^2(k)]}{E[d^2(k)]}$$

$$ERLE(k) = 10 \log_{10} \frac{E[d^2(k)]}{E[\alpha^2(k)]}$$

$$ERLE(k) = 10 \log_{10} \frac{E[x^2(k)]}{E[\alpha^2(k)]}$$

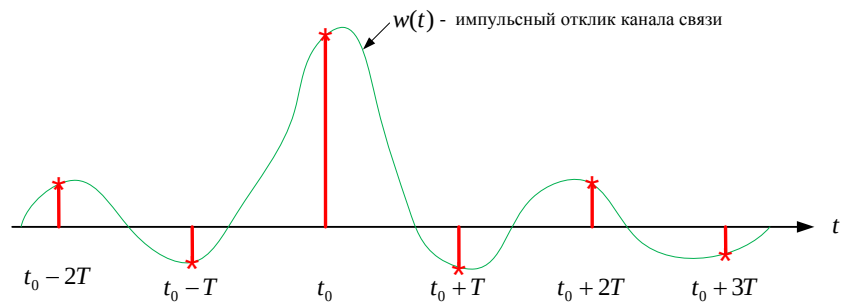
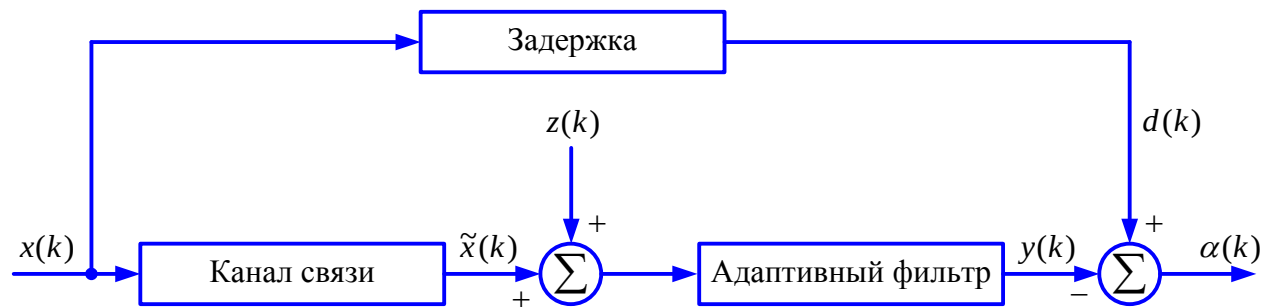


RLS-алгоритм

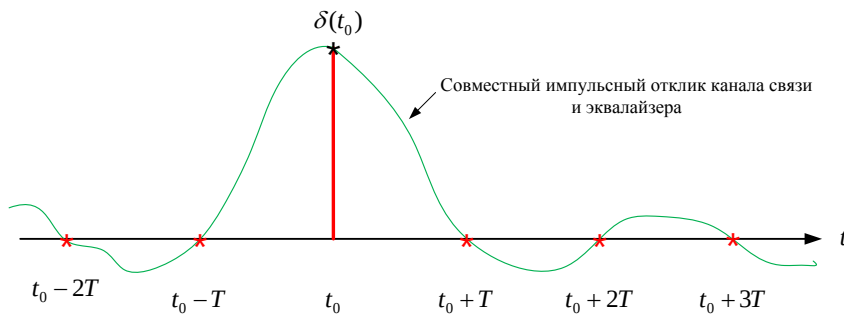


LMS-алгоритм

Где применяются адаптивные фильтры ? – Например, в адаптивных эквалайзерах каналов связи



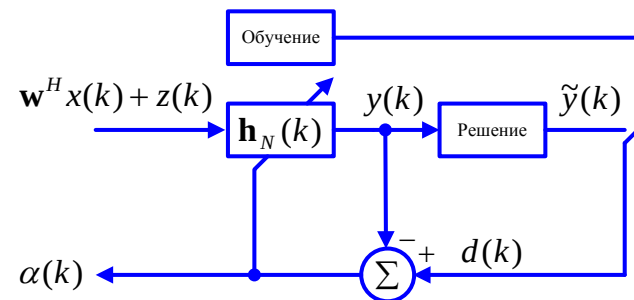
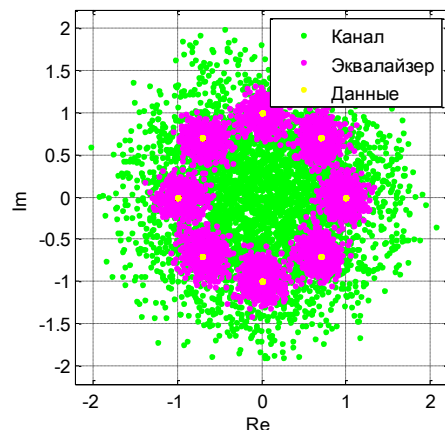
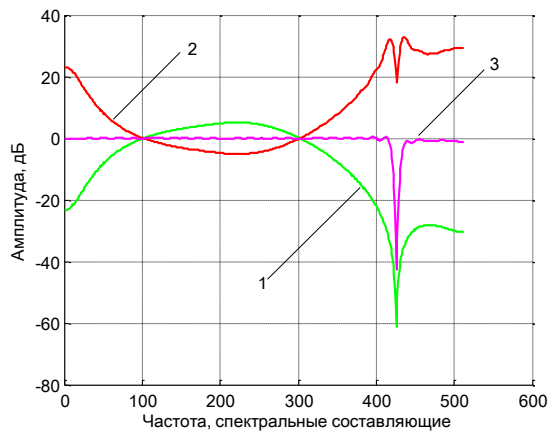
$$\tilde{x}(t_0 + kT) = w(t_0)x(t_0 + kT) + \sum_{n \neq k} w(t_0 + kT - nT)x(t_0 - nT)$$



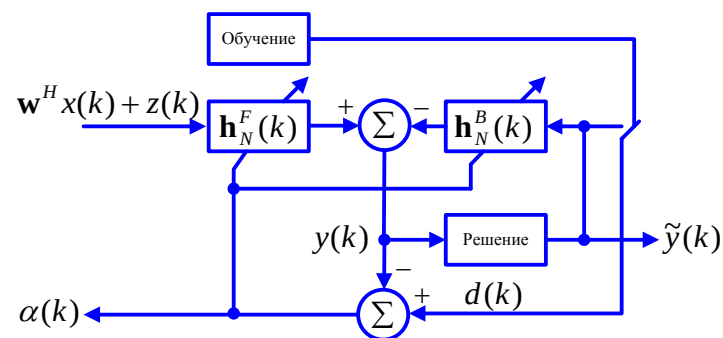
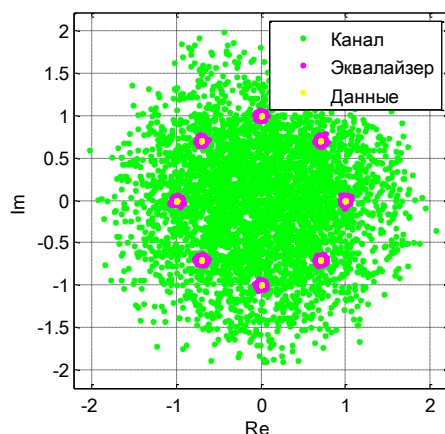
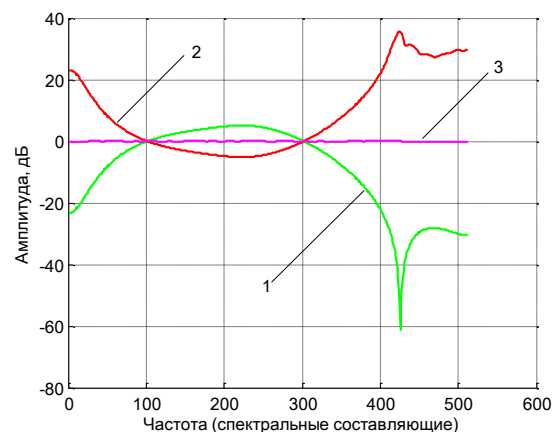
$$y(t_0 + kT) = \delta(t_0)x(t_0 + kT) + 0$$

Принцип выравнивания каналов связи

Насколько эффективно выравнивание каналов связи ?



Эквалайзер без обратной СВЯЗИ



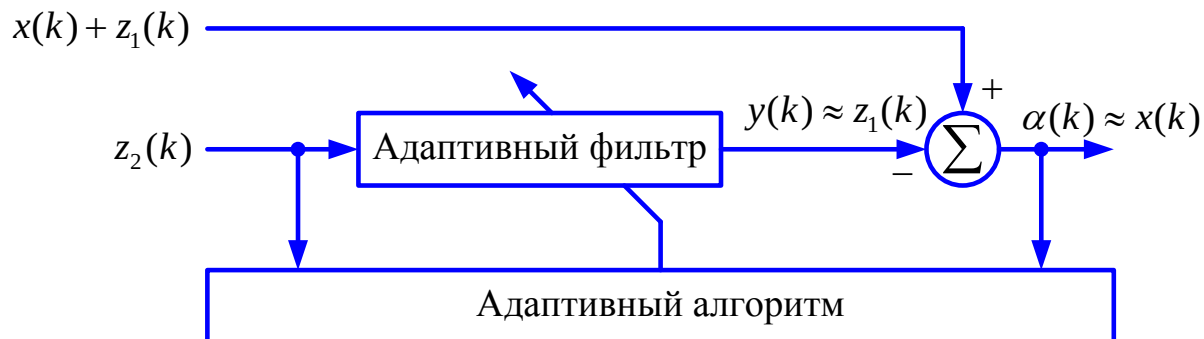
Эквалайзер с обратной СВЯЗЬЮ

АЧХ

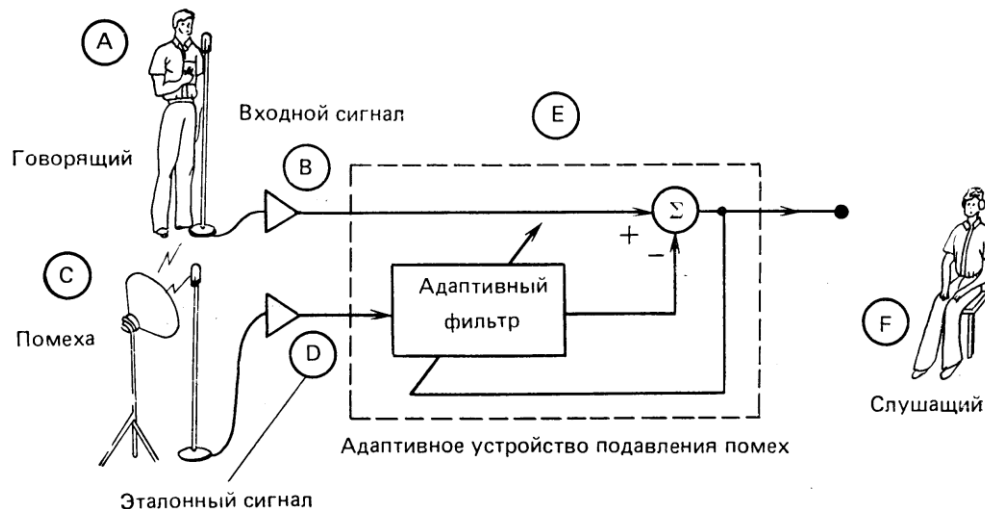
Созвездия

Где применяются адаптивные фильтры ? – Например, в адаптивном шумоподавлении

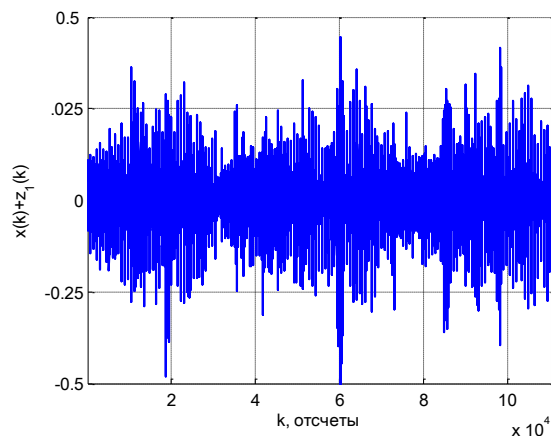
Принцип адаптивного шумоподавления



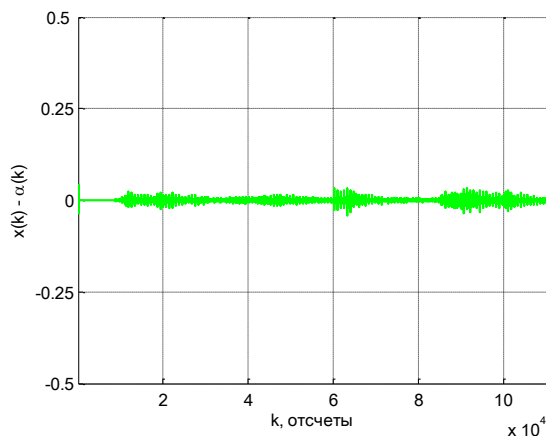
Пример адаптивного шумоподавления



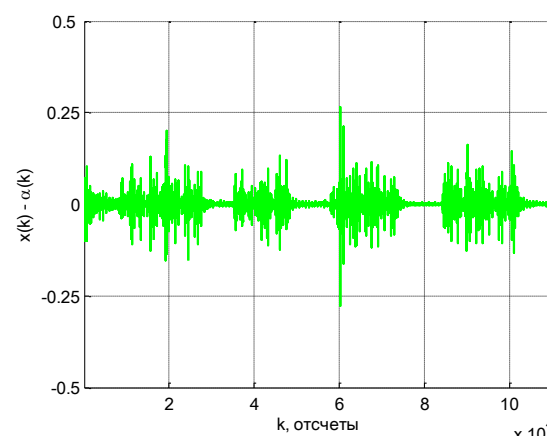
Насколько эффективно шумоподавление ?



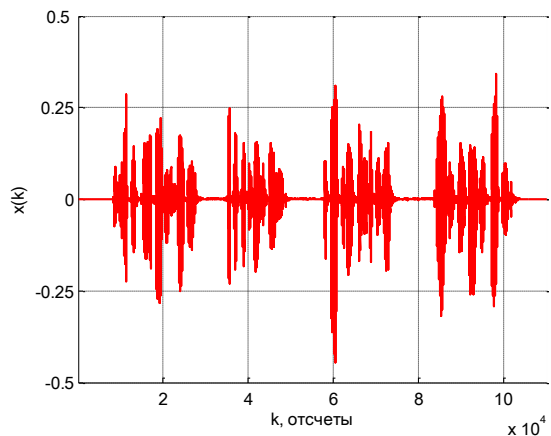
Зашумленная речь



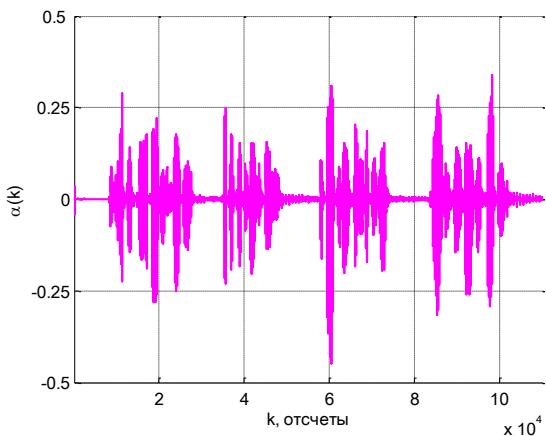
Остаточный шум, RLS



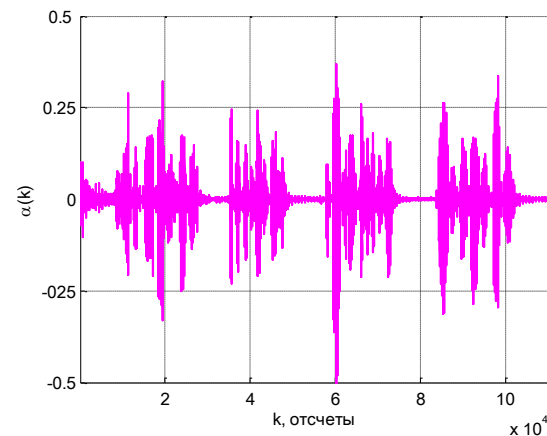
Остаточный шум, LMS



Исходная речь



RLS-алгоритм



LMS-алгоритм

АФ действительно применяются ?

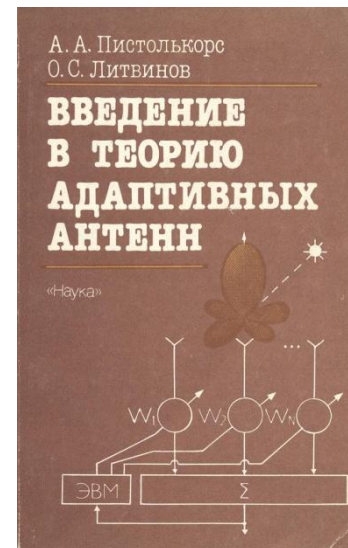
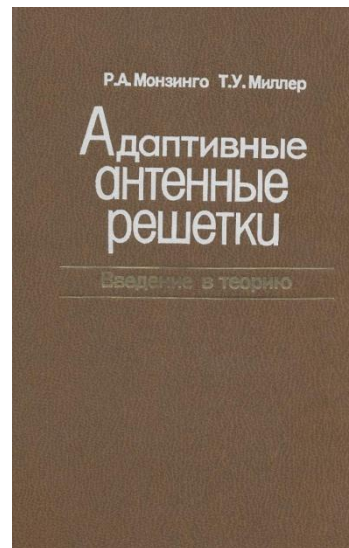
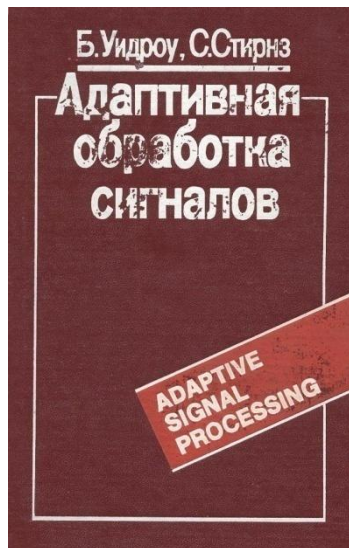


Да, применяются !

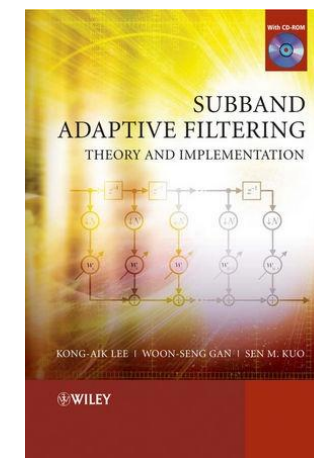
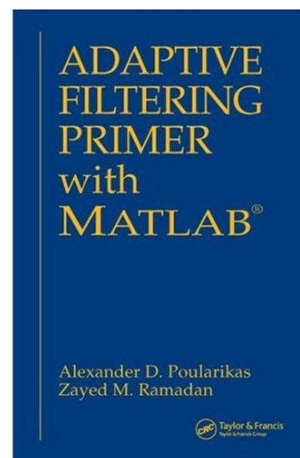
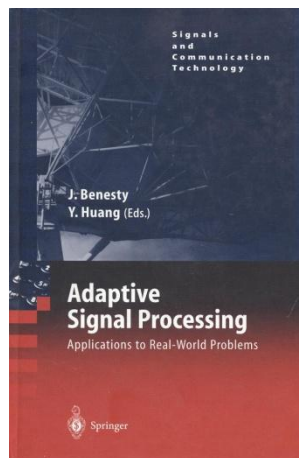
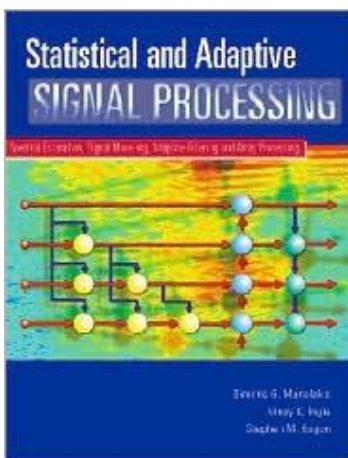
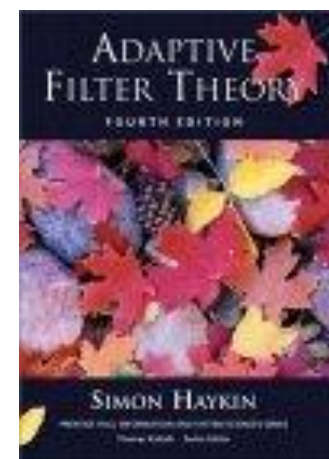
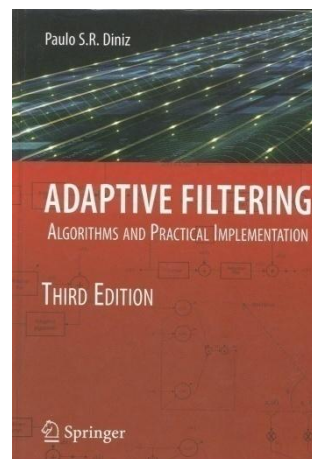
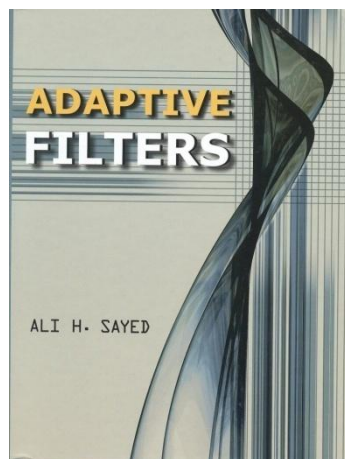
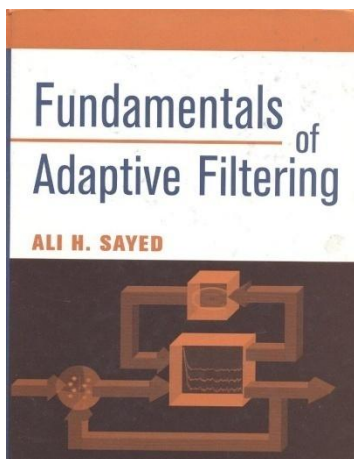


Где можно прочитать про адаптивные фильтры?

- В журнальных статьях и трудах научно-технических конференций – более 10 тысяч публикаций !
- В книгах на английском языке (несколько десятков)
- В книгах на русском языке (5 штук):



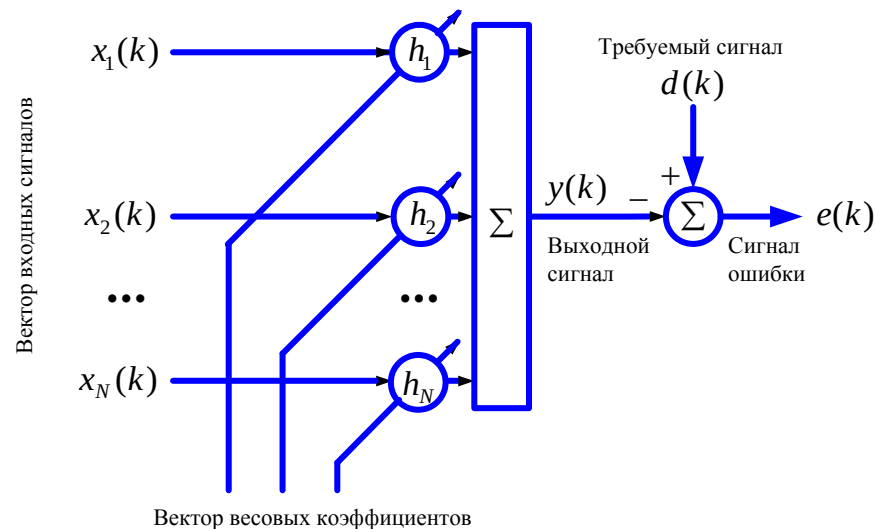
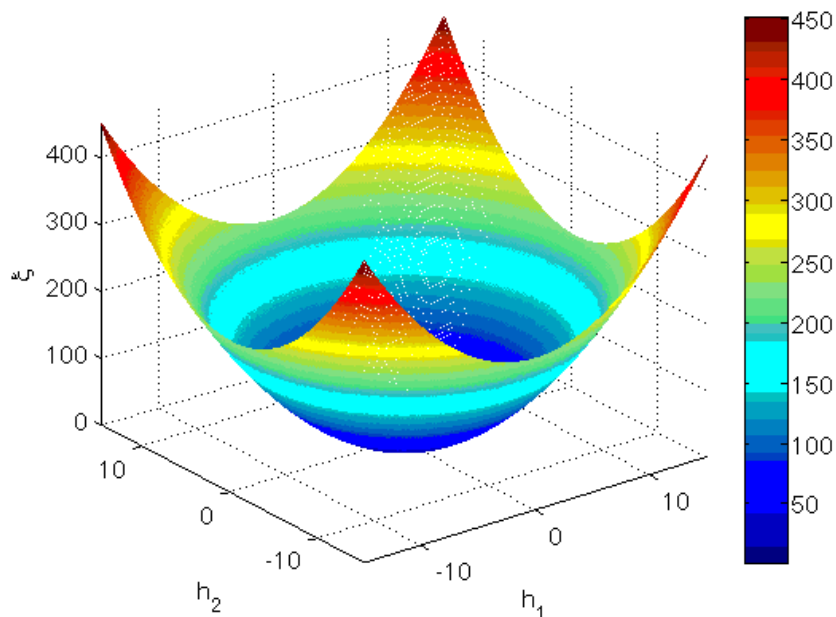
Книги на английском языке



Чему можно учить в адаптивной обработке сигналов ?

1. Введение в адаптивную фильтрацию.
2. Оператор комплексного градиента и его применение в теории адаптивной фильтрации.
3. Основы адаптивной фильтрации.
4. Поиск винеровского решения.
5. Градиентные адаптивные алгоритмы и их свойства.
6. Рекурсивная адаптивная фильтрация по критерию наименьших квадратов.
7. Рекурсивная адаптивная фильтрация по критерию наименьших квадратов на базе обратного QR-разложения.
8. Рекурсивная адаптивная фильтрация по критерию наименьших квадратов на базе прямого QR-разложения.
9. Уменьшение вычислительной сложности рекурсивных алгоритмов адаптивной фильтрации по критерию наименьших квадратов.
10. Линейно-ограниченная рекурсивная адаптивная фильтрация по критерию наименьших квадратов.
11. Быстрые рекурсивные адаптивные алгоритмы по критерию наименьших квадратов.
12. Лестничные адаптивные фильтры.
13. Многоканальные адаптивные фильтры.
14. Адаптивная фильтрация нестационарных сигналов.
15. Параллельные алгоритмы по критерию наименьших квадратов.
16. Адаптивные алгоритмы аффинных проекций.
17. Слепые алгоритмы адаптивной фильтрации.
18. Субполосные адаптивные фильтры.

Что такое винеровская фильтрация ?



Поверхность среднеквадратической

ошибки: $\xi = E[|e(k)|^2] = \sigma_d^2 - \mathbf{h}_N^H \mathbf{r}_N - \mathbf{r}_N^H \mathbf{h}_N + \mathbf{h}_N^H \mathbf{R}_N \mathbf{h}_N$

Линейный сумматор

Градиент: $\nabla_{\mathbf{h}_N^*} \xi = \left[\frac{\partial \xi}{\partial h_1^*}, \frac{\partial \xi}{\partial h_2^*}, \dots, \frac{\partial \xi}{\partial h_N^*} \right]^T$

Условие минимума: $\nabla_{\mathbf{h}_N^*} \xi = -\mathbf{r}_N + \mathbf{R}_N \mathbf{h}_N = \mathbf{0}_N$

Винеровское решение: $\mathbf{h}_{N,o} = \mathbf{R}_N^{-1} \mathbf{r}_N$

Минимальное значение: $\xi_{\min} = \sigma_d^2 - \mathbf{r}_N^H \mathbf{R}_N^{-1} \mathbf{r}_N$

Линейно-ограниченная винеровская фильтрация

Линейное ограничение: $\mathbf{C}_{NJ}^H \mathbf{h}_N = \mathbf{f}_J$

Матрица ограничений: $\mathbf{C}_{NJ}$ Вектор ограничиваемых параметров $\mathbf{f}_J$

Линейно-ограниченная целевая функция:

$$\xi_c = E[|e(k)|^2] = \sigma_d^2 - \mathbf{h}_N^H \mathbf{r}_N - \mathbf{r}_N^H \mathbf{h}_N + \mathbf{h}_N^H \mathbf{R}_N \mathbf{h}_N + \lambda_J^H [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_N] + [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{h}_N]^H \lambda_J,$$

Вектор множителей Лагранжа:

$$\lambda_J = [\lambda_1, \lambda_2, \dots, \lambda_J]^T$$

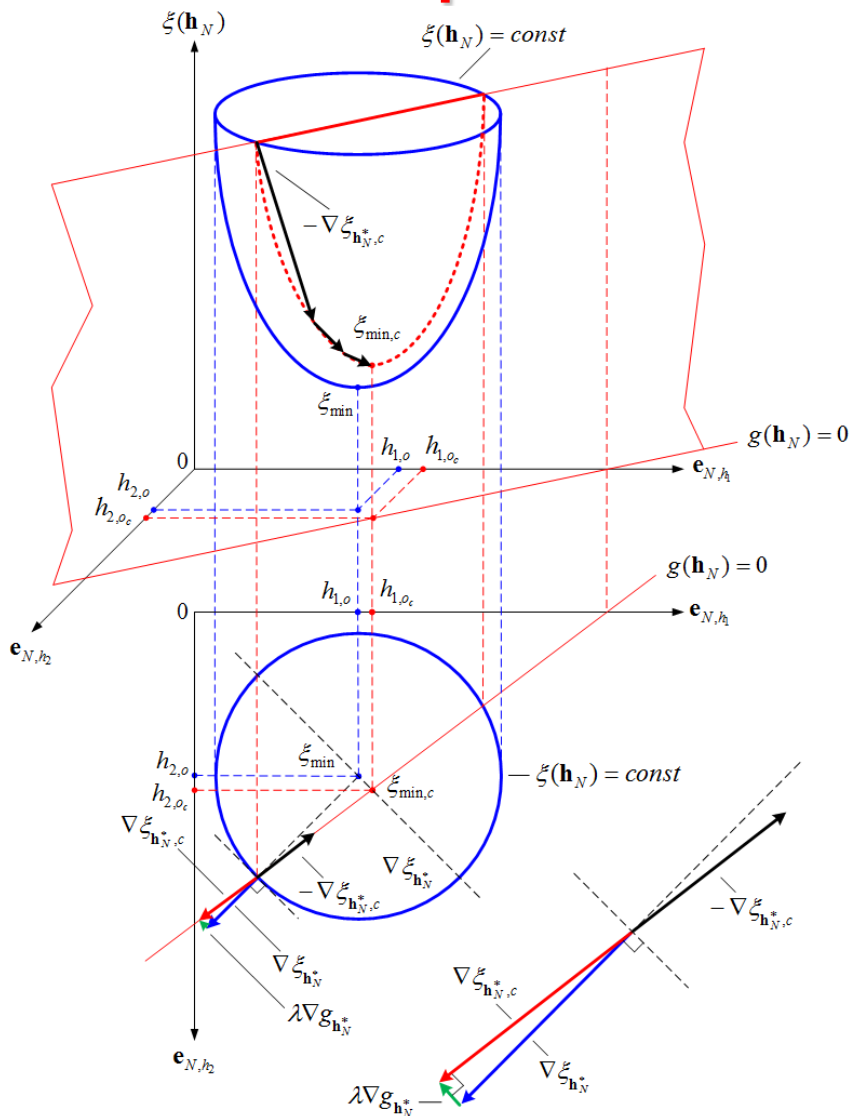
Условие минимума:

$$\nabla_{\mathbf{h}_N^*} \xi_c = \frac{\partial \xi_c}{\partial \mathbf{h}_N^*} = -\mathbf{r}_N + \mathbf{R}_N \mathbf{h}_N - \mathbf{C}_{NJ} \lambda_J = \mathbf{0}_N$$

Линейно-ограниченное винеровское решение:

$$\mathbf{h}_{N,oc} = \mathbf{R}_N^{-1} \mathbf{r}_N + \mathbf{R}_N^{-1} \mathbf{C}_{NJ} [\mathbf{C}_{NJ}^H \mathbf{R}_N^{-1} \mathbf{C}_{NJ}]^{-1} [\mathbf{f}_J - \mathbf{C}_{NJ}^H \mathbf{R}_N^{-1} \mathbf{r}_N]$$

Линейно-ограниченная винеровская фильтрация



Целевая функция: $\xi(\mathbf{h}_N)$

Линейное ограничение: $g(\mathbf{h}_N) = 0$

$$g(\mathbf{h}_N + \delta \mathbf{h}_N) = g(\mathbf{h}_N) + \delta \mathbf{h}_N^H \nabla g_{\mathbf{h}_N^*} = g(\mathbf{h}_N)$$



$$\delta \mathbf{h}_N^H \nabla g_{\mathbf{h}_N^*} = 0$$

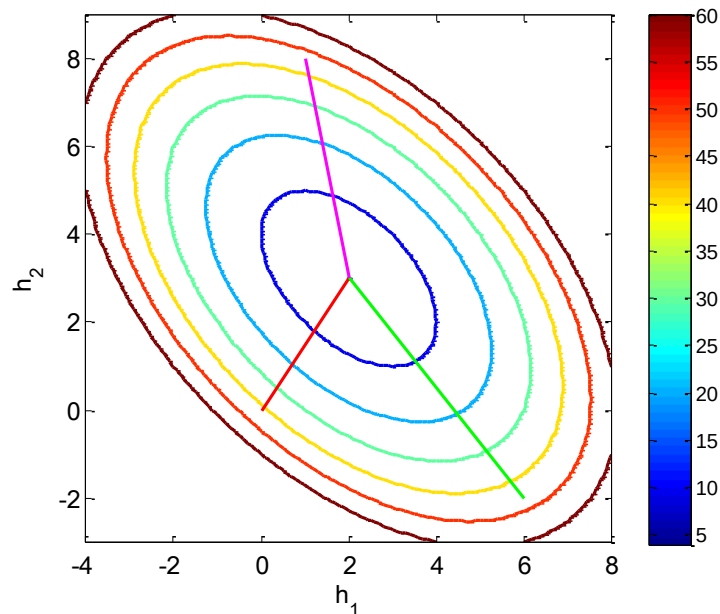
Требуемый градиент:

$$\nabla \xi_{\mathbf{h}_N^*,c} = \nabla \xi_{\mathbf{h}_N^*} + \lambda \nabla g_{\mathbf{h}_N^*}$$

Требуемая целевая функция:

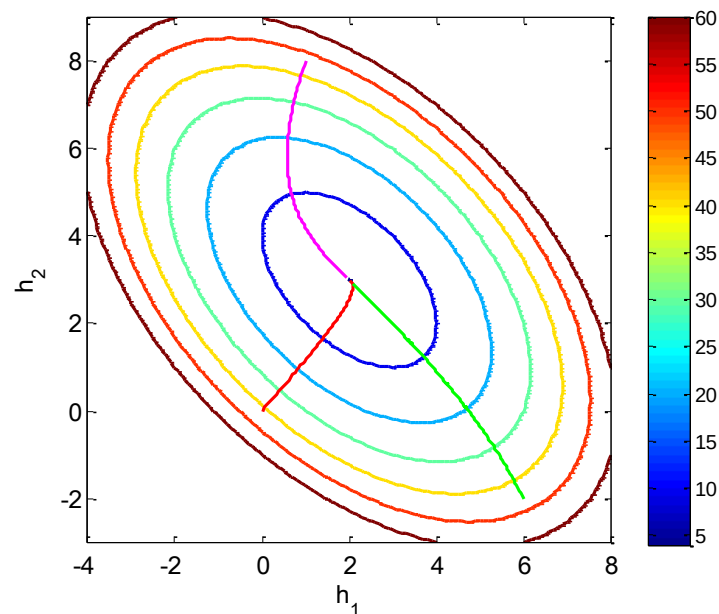
$$\xi_c(\mathbf{h}_N) = \xi(\mathbf{h}_N) + \lambda g(\mathbf{h}_N)$$

Как найти винеровское решение ?



$$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) - \mu \mathbf{R}_N^{-1} \nabla_{\mathbf{h}_N^*} \xi(k-1)$$

Алгоритм Ньютона

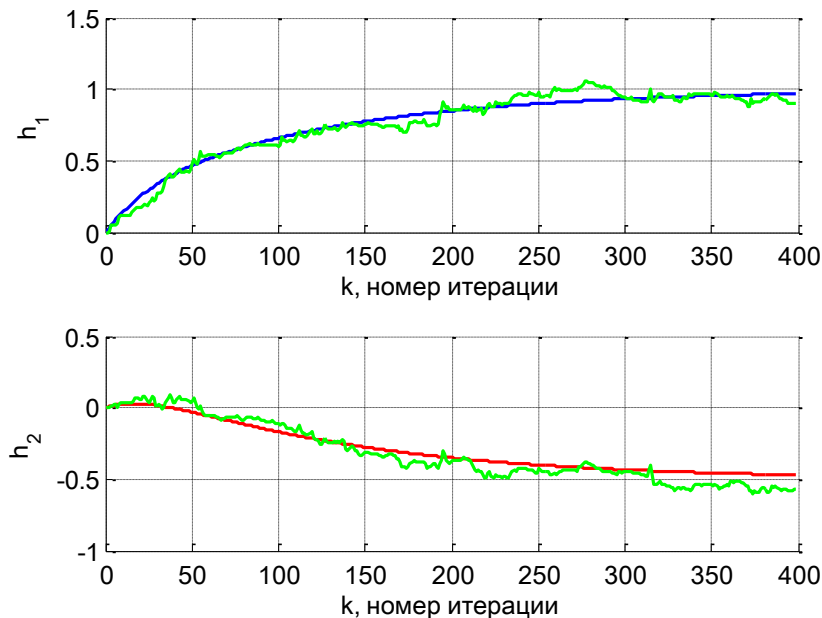
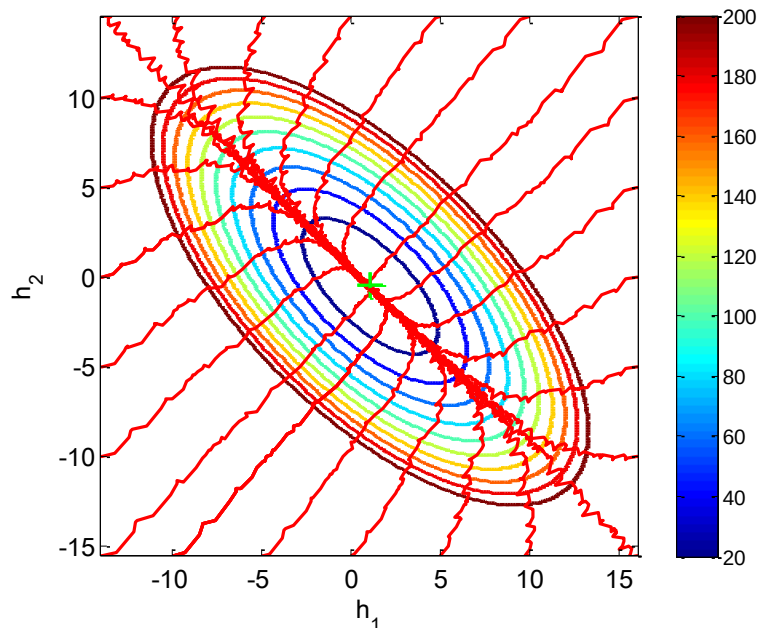


$$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \mu \left[-\nabla_{\mathbf{h}_N^*} \xi(k-1) \right]$$

Алгоритм наискорейшего спуска

Градиент: $\nabla_{\mathbf{h}_N^*} \xi = -\mathbf{r}_N + \mathbf{R}_N \mathbf{h}_N(k-1)$

Что делать, если корреляционная матрица неизвестна ?



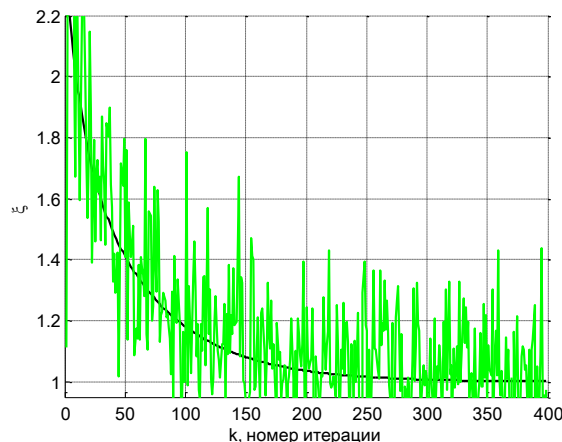
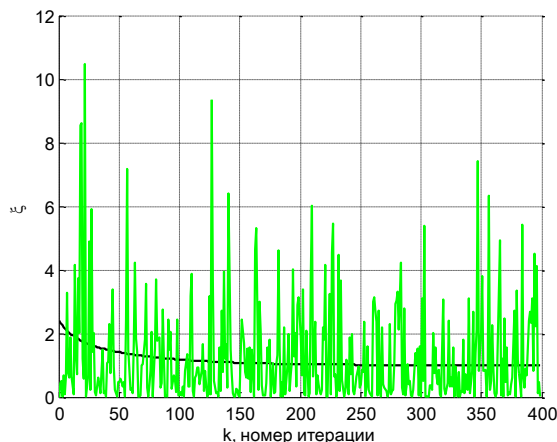
LMS-алгоритм

Градиент:

$$\hat{\nabla}_{\mathbf{h}_N^*} \xi(k-1) = -\mathbf{x}_N(k) \alpha^*(k).$$

LMS-алгоритм:
$$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \mu \left[-\hat{\nabla}_{\mathbf{h}_N^*} \xi(k-1) \right] = \mathbf{h}_N(k-1) + \mu \mathbf{x}_N(k) \alpha^*(k)$$

Действительно ли средний квадрат ?

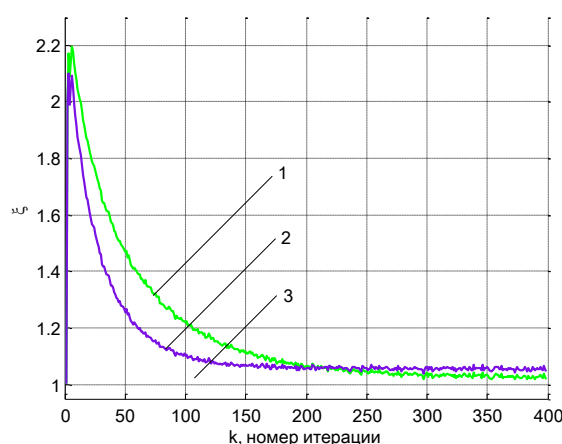
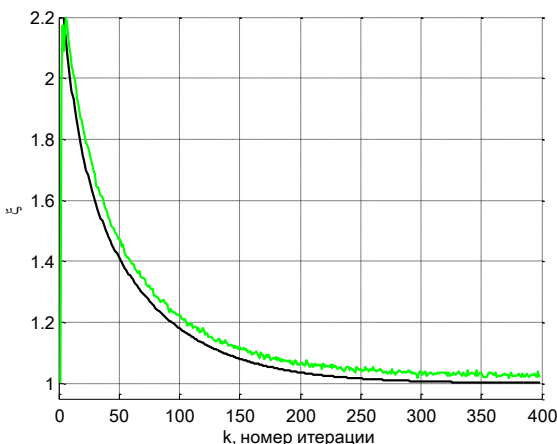


Условие сходимости:

$$0 < \mu < 2/\lambda_{\max}$$

$$0 < \mu < \frac{2}{\sum_{i=1}^N R_{nn}}$$

$$0 < \mu < \frac{2}{N\sigma_x^2}$$



Постоянные времени:

$$\frac{1}{\mu\lambda_{\max}} \leq k_{\tau_{n,h}} \leq \frac{1}{\mu\lambda_{\min}}$$

$$\frac{1}{2\mu\lambda_{\max}} \leq k_{\tau_{n,\xi}} \leq \frac{1}{2\mu\lambda_{\min}}$$

$$\xi_{exc} = \xi_{\min} \frac{\sum_{n=1}^N \mu \lambda_n}{2 - \sum_{n=1}^N \mu \lambda_n}$$

Дополнительная среднеквадратическая ошибка:

Что такое RLS-алгоритм ?

Целевая функция :

$$E_N(k) = \sum_{i=1}^k \lambda^{k-i} e_N(i) e_N^*(i) =$$

$$= E_d(k) - \mathbf{r}_N^H(k) \mathbf{h}_N(k) - \mathbf{h}_N^H(k) \mathbf{r}_N + \mathbf{h}_N^H(k) \mathbf{R}_N(k) \mathbf{h}_N(k)$$

$$E_d(k) = \sum_{i=1}^k \lambda^{k-i} d(i) d^*(i) \quad \mathbf{r}_N(k) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) d^*(i) \quad \mathbf{R}_N(k) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^H(i)$$

Условие минимума :

$$\nabla_{\mathbf{h}_N^*} [E_N(k)] = \frac{\partial E_N(k)}{\partial \mathbf{h}_N^*(k)} = -\mathbf{r}_N(k) + \mathbf{R}_N(k) \mathbf{h}_N(k) = \mathbf{0}_N$$

Оптимальное решение :

$$\mathbf{h}_N(k) = \mathbf{R}_N^{-1}(k) \mathbf{r}_N(k)$$

Лемма об обращении матриц:

$$(\mathbf{C} - \mathbf{B}^H \mathbf{A}^{-1} \mathbf{D})^{-1} = \mathbf{C}^{-1} - \mathbf{C}^{-1} \mathbf{B}^H (\mathbf{D} \mathbf{C}^{-1} \mathbf{B}^H - \mathbf{A})^{-1} \mathbf{B} \mathbf{C}^{-1}$$

$$\mathbf{C} = \lambda \mathbf{R}_N(k-1) \quad \mathbf{C}^{-1} = \lambda^{-1} \mathbf{R}_N^{-1}(k-1) \quad \mathbf{B}^H = \mathbf{x}_N^H(k) \quad \mathbf{D} = \mathbf{x}_N(k) \quad \mathbf{A}^{-1} = 1$$

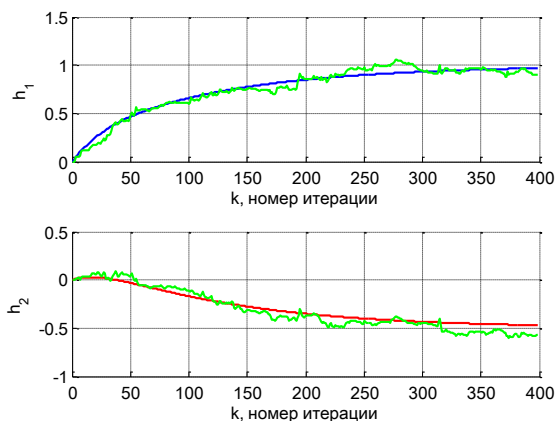
RLS-алгоритм:

$$\mathbf{g}_N(k) = \frac{\mathbf{R}_N^{-1}(k-1) \mathbf{x}_N(k)}{\lambda + \mathbf{x}_N^H(k) \mathbf{R}_N^{-1}(k-1) \mathbf{x}_N(k)}$$

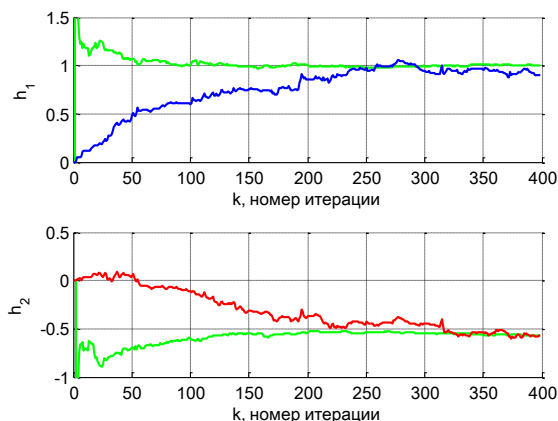
$$\mathbf{R}_N^{-1}(k) = \lambda^{-1} [\mathbf{R}_N^{-1}(k-1) - \mathbf{g}_N(k) \mathbf{x}_N^H(k) \mathbf{R}_N^{-1}(k-1)]$$

$$\alpha_N(k) = d(k) - \mathbf{h}_N^H(k-1) \mathbf{x}_N(k) \quad \mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \mathbf{g}_N(k) \alpha_N^*(k)$$

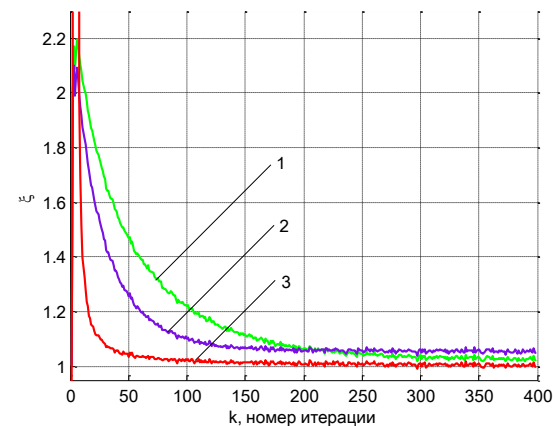
LMS или RLS ?



Весовые
коэффициенты
LMS-алгоритм



Весовые
коэффициенты
RLS-алгоритм



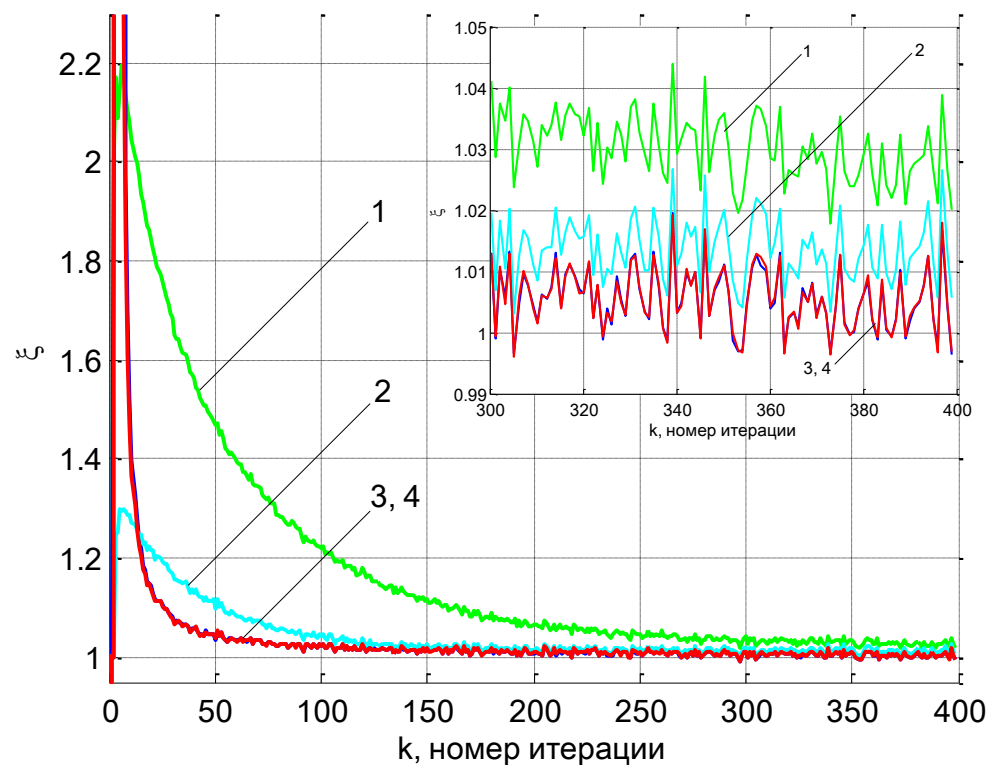
Обучающие кривые:
1, 2 – LMS, 3 - RLS

LMS-алгоритм – линейная вычислительная сложность, зависимость длительности переходного процесса и остаточной ошибки в установившемся состоянии от шага сходимости

За и против:

RLS-алгоритм – квадратичная или линейная вычислительная сложность, быстрая сходимость, отсутствие шага сходимости и дополнительной остаточной ошибки в установившемся состоянии

LMS или RLS ?



Обучающие кривые:
1, 2 – LMS, 3, 4 – RLS

За и против:

LMS-алгоритм – зависимость длительности переходного процесса и остаточной ошибки в установившемся состоянии от соотношения собственных чисел корреляционной матрицы входного сигнала адаптивного фильтра

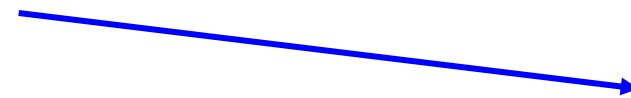
RLS-алгоритм – независимость длительности переходного процесса и остаточной ошибки в установившемся состоянии от соотношения собственных чисел корреляционной матрицы входного сигнала адаптивного фильтра

Что такое NLMS-алгоритм ?

Алгоритм Ньютона: $\mathbf{h}_N(k) = \mathbf{h}_N(k-1) - \mu \mathbf{R}_N^{-1} \nabla_{\mathbf{h}_N^*} \xi(k-1)$



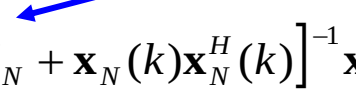
$$\hat{\mathbf{R}}_N(k) = \delta^2 \mathbf{I}_N + \mathbf{x}_N(k) \mathbf{x}_N^H(k)$$



Лемма об обращении матриц: $(\mathbf{C} - \mathbf{B}^H \mathbf{A}^{-1} \mathbf{D})^{-1} = \mathbf{C}^{-1} - \mathbf{C}^{-1} \mathbf{B}^H (\mathbf{D} \mathbf{C}^{-1} \mathbf{B}^H - \mathbf{A})^{-1} \mathbf{B} \mathbf{C}^{-1}$
 $\mathbf{C}^{-1} = \delta^{-2} \mathbf{I}_N \quad \mathbf{C} = \delta^2 \mathbf{I}_N \quad \mathbf{A}^{-1} = 1 \quad \mathbf{B}^H = \mathbf{x}_N(k) \quad \mathbf{D} = \mathbf{x}_N^H(k)$



Упрощенный алгоритм Ньютона:



$$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \hat{\mu} \left[\delta^2 \mathbf{I}_N + \mathbf{x}_N(k) \mathbf{x}_N^H(k) \right]^{-1} \mathbf{x}_N(k) \alpha^*(k).$$



NLMS-алгоритм:

$$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \frac{\hat{\mu}}{\mathbf{x}_N^H(k) \mathbf{x}_N(k) + \delta^2} \mathbf{x}_N(k) \alpha^*(k)$$

Что такое алгоритм аффинных проекций (АР) ?

Вычисление ВК:

$$\mathbf{h}_N(k) = \mathbf{h}_N(k-1) + \mu(k)\mathbf{X}_{NL}(k)\boldsymbol{\varepsilon}_L(k)$$

Вектор нормализованных ошибок:

$$\boldsymbol{\varepsilon}_L(k) = [\mathbf{X}_{NL}^H(k)\mathbf{X}_{NL}(k) + \delta\mathbf{I}_L]^{-1}\boldsymbol{\alpha}_L^*(k)$$

Вектор ошибок:

$$\boldsymbol{\alpha}_L(k) = \mathbf{d}_L(k) - [\mathbf{h}_N^H(k-1)\mathbf{X}_{NL}(k)]^T$$

Используемые векторы и матрицы:

$$\boldsymbol{\alpha}_L(k) = [\alpha(k), \alpha(k-1), \Lambda, \alpha(k-L+1)]^T$$

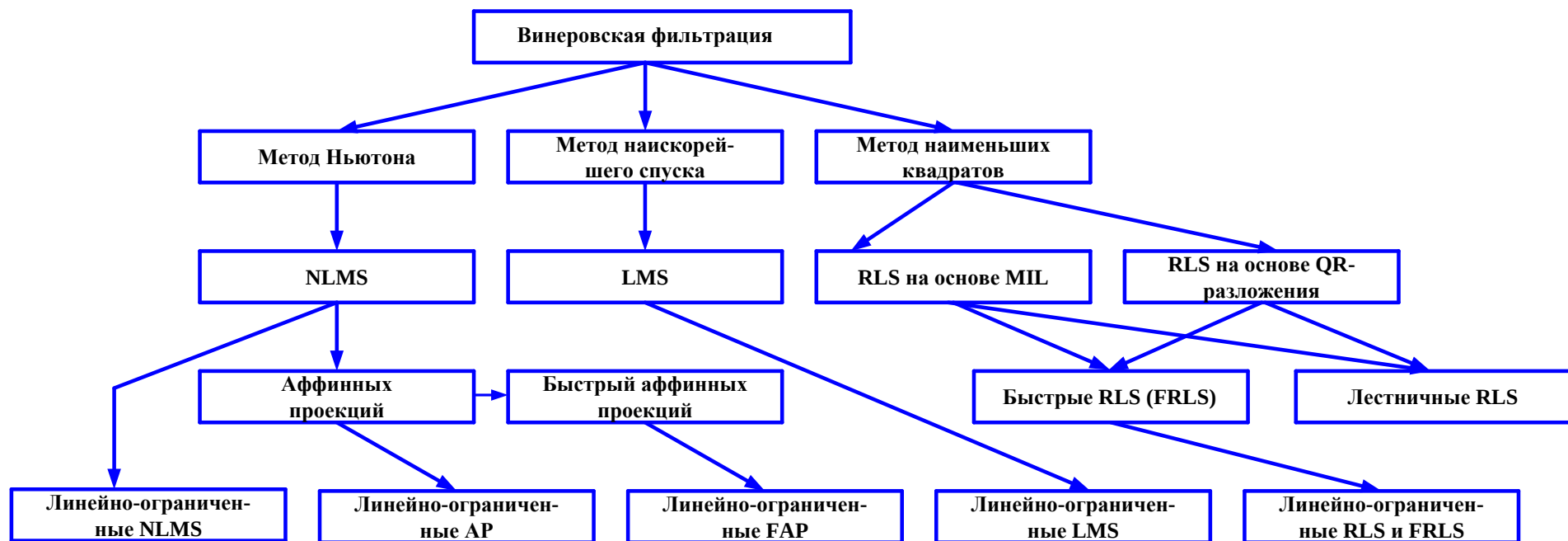
$$\boldsymbol{\varepsilon}_L(k) = [\varepsilon(k), \varepsilon(k-1), \Lambda, \varepsilon(k-L+1)]^T$$

$$\mathbf{d}_L(k) = [d(k), d(k-1), \Lambda, d(k-L+1)]^T$$

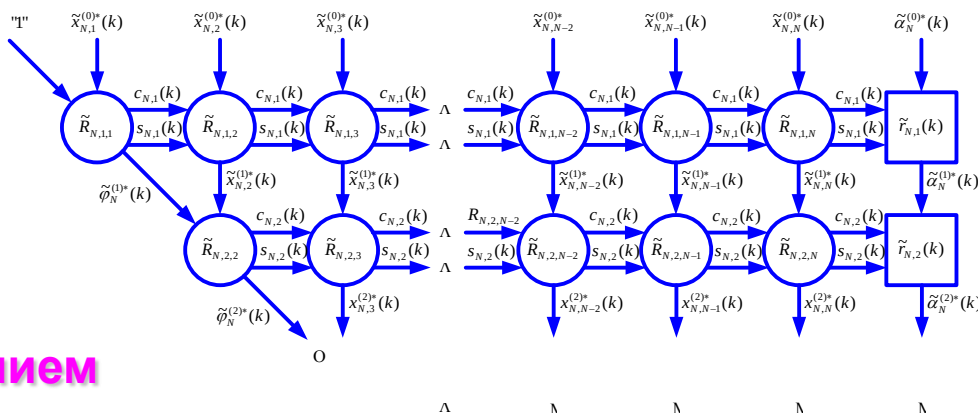
$$\mathbf{X}_{NL}(k) = [\mathbf{x}_N(k), \mathbf{x}_N(k-1), \Lambda, \mathbf{x}_N(k-L+1)]$$

Длина скользящего окна: L

Какие бывают адаптивные алгоритмы ?



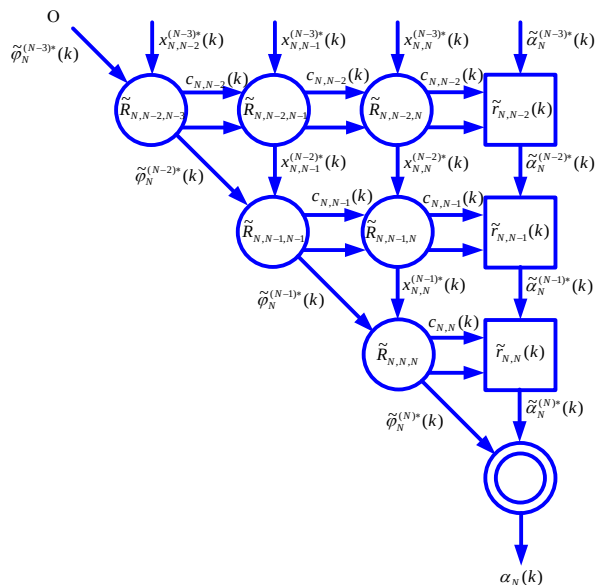
Адаптивный фильтр на основе прямого QR-разложения



С использованием вращений Гивенса

С операциями извлечения квадратного корня

Без операций извлечения квадратного корня



Адаптивный фильтр на основе обратного QR-разложения

С использованием вращений Гивенса

С операциями извлечения квадратного корня

Без операций извлечения квадратного корня



Можно ли уменьшить сложность RLS алгоритмов ?

Инвариантность
к сдвигу:

$$\mathbf{x}_{N+1}(k) = \begin{bmatrix} x(k) \\ \mathbf{x}_N(k-1) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_N(k) \\ x(k-N) \end{bmatrix}$$

Корреляционная
матрица:

$$\begin{aligned} \mathbf{R}_{N+1}(k) &= \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_{N+1}(i) \mathbf{x}_{N+1}^H(i) = \sum_{i=1}^k \lambda^{k-i} \begin{bmatrix} x(i) \\ \mathbf{x}_N(i-1) \end{bmatrix} \begin{bmatrix} x^*(i) & \mathbf{x}_N^H(i-1) \end{bmatrix} = \\ &= \begin{bmatrix} \sum_{i=1}^k \lambda^{k-i} x(i) x^*(i) & \sum_{i=1}^k \lambda^{k-i} x(i) \mathbf{x}_N^H(i-1) \\ \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i-1) x^*(i) & \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i-1) \mathbf{x}_N^H(i-1) \end{bmatrix} = \begin{bmatrix} \sigma^{f^2}(k) & \mathbf{r}_N^{fH}(k) \\ \mathbf{r}_N^f(k) & \mathbf{R}_N(k-1) \end{bmatrix}. \end{aligned}$$

Корреляционная
матрица:

$$\begin{aligned} \mathbf{R}_{N+1}(k) &= \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_{N+1}(i) \mathbf{x}_{N+1}^H(i) = \sum_{i=1}^k \lambda^{k-i} \begin{bmatrix} \mathbf{x}_N(i) \\ x(i-N) \end{bmatrix} \begin{bmatrix} \mathbf{x}_N^H(i) & x^*(i-N) \end{bmatrix} = \\ &= \begin{bmatrix} \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^H(i) & \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) x^*(i-N) \\ \sum_{i=1}^k \lambda^{k-i} x(i-N) \mathbf{x}_N^H(i) & \sum_{i=1}^k \lambda^{k-i} x(i) x^*(i) \end{bmatrix} = \\ &= \begin{bmatrix} \mathbf{R}_N(k) & \mathbf{r}_N^b(k) \\ \mathbf{r}_N^{bH}(k) & \sigma^{b^2}(k) \end{bmatrix}. \end{aligned}$$

Как уменьшить сложность RLS-алгоритмов ?

Обращение
клеточных
матриц:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^H & \mathbf{C} \end{bmatrix}^{-1} = \begin{bmatrix} (\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^H)^{-1} & -(\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^H)^{-1}\mathbf{B}\mathbf{C}^{-1} \\ -\mathbf{C}^{-1}\mathbf{B}^H(\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^H)^{-1} & \mathbf{C}^{-1}\mathbf{B}^H(\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^H)^{-1}\mathbf{B}\mathbf{C}^{-1} + \mathbf{C}^{-1} \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^H & \mathbf{C} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}(\mathbf{C} - \mathbf{B}^H\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{B}^H\mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{B}(\mathbf{C} - \mathbf{B}^H\mathbf{A}^{-1}\mathbf{B})^{-1} \\ -(\mathbf{C} - \mathbf{B}^H\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{B}^H\mathbf{A}^{-1} & (\mathbf{C} - \mathbf{B}^H\mathbf{A}^{-1}\mathbf{B})^{-1} \end{bmatrix}$$

Векторы
коэффициентов
Калмана:

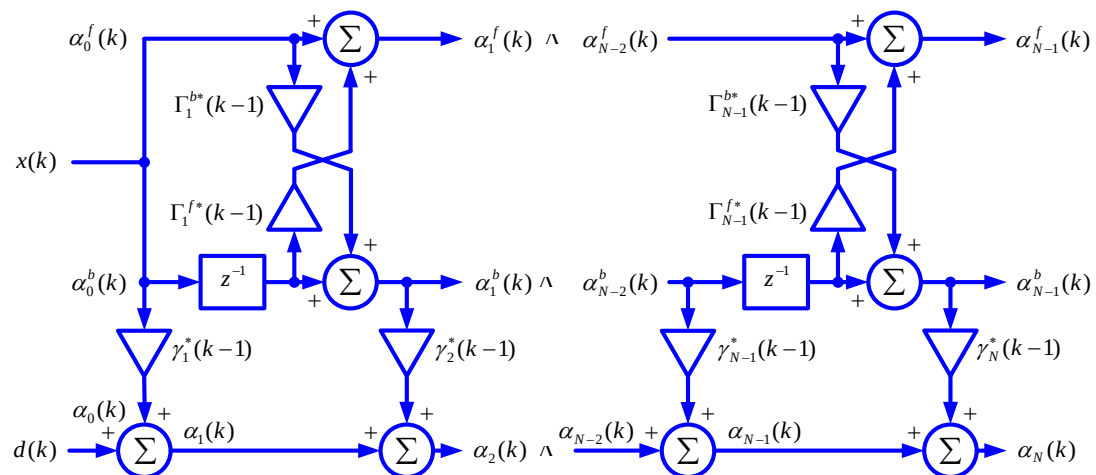
$$\begin{aligned} \mathbf{g}_{N+1}(k) &= \left\{ \begin{bmatrix} 0 & \mathbf{0}_N^T \\ \mathbf{0}_N & \mathbf{R}_N^{-1}(k-1) \end{bmatrix} + [E_{N,\min}^f(k)]^{-1} \mathbf{a}_{N+1}(k) \mathbf{a}_{N+1}^H(k) \right\} \mathbf{x}_{N+1}(k) = \\ &= \begin{bmatrix} 0 \\ \mathbf{g}_N(k-1) \end{bmatrix} + [E_{N,\min}^f(k)]^{-1} \mathbf{a}_{N+1}(k) e_N^f(k) \end{aligned}$$

$$\mathbf{g}_N(k) = \mathbf{R}_N^{-1}(k) \mathbf{x}_N(k)$$

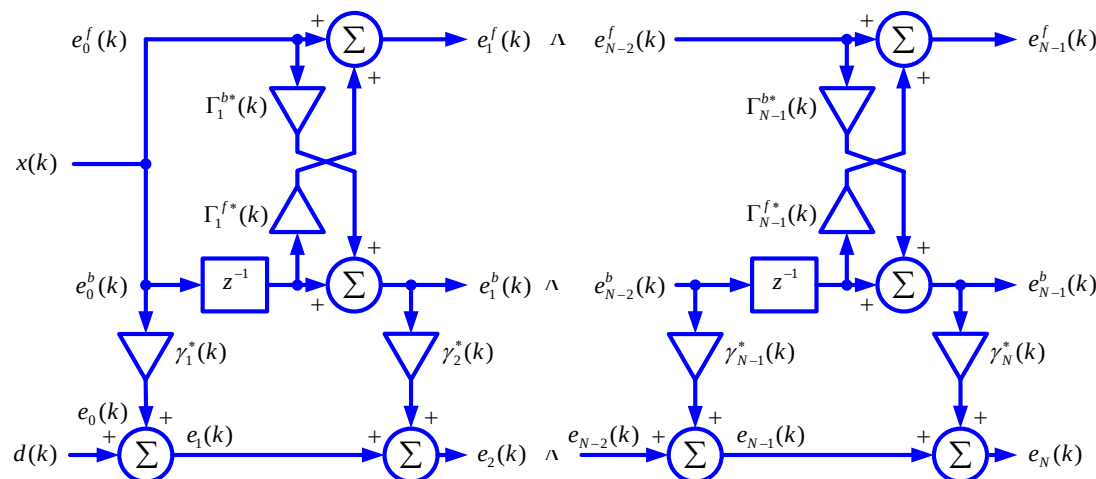
$$\begin{aligned} \mathbf{g}_{N+1}(k) &= \left\{ \begin{bmatrix} \mathbf{R}_N^{-1}(k) & \mathbf{0}_N \\ \mathbf{0}_N^T & 0 \end{bmatrix} + [E_{N,\min}^b(k)]^{-1} \mathbf{b}_{N+1}(k) \mathbf{b}_{N+1}^H(k) \right\} \mathbf{x}_{N+1}(k) = \\ &= \begin{bmatrix} \mathbf{g}_N(k) \\ 0 \end{bmatrix} + [E_{N,\min}^b(k)]^{-1} \mathbf{b}_{N+1}(k) e_N^b(k) \end{aligned}$$

Лестничные RLS-алгоритмы

Адаптивный фильтр
на основе априорных
ошибок:



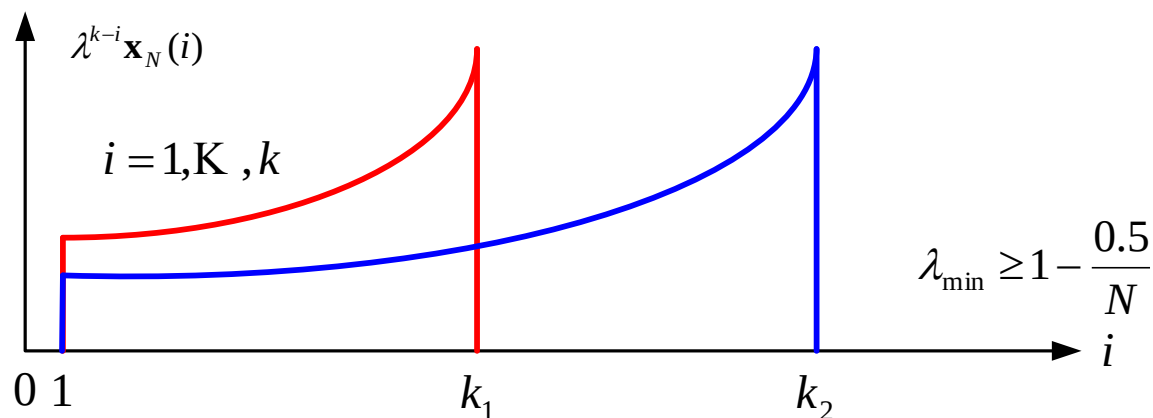
Адаптивный
фильтр на основе
апостериорных
ошибок:



Как фильтровать нестационарные сигналы ?

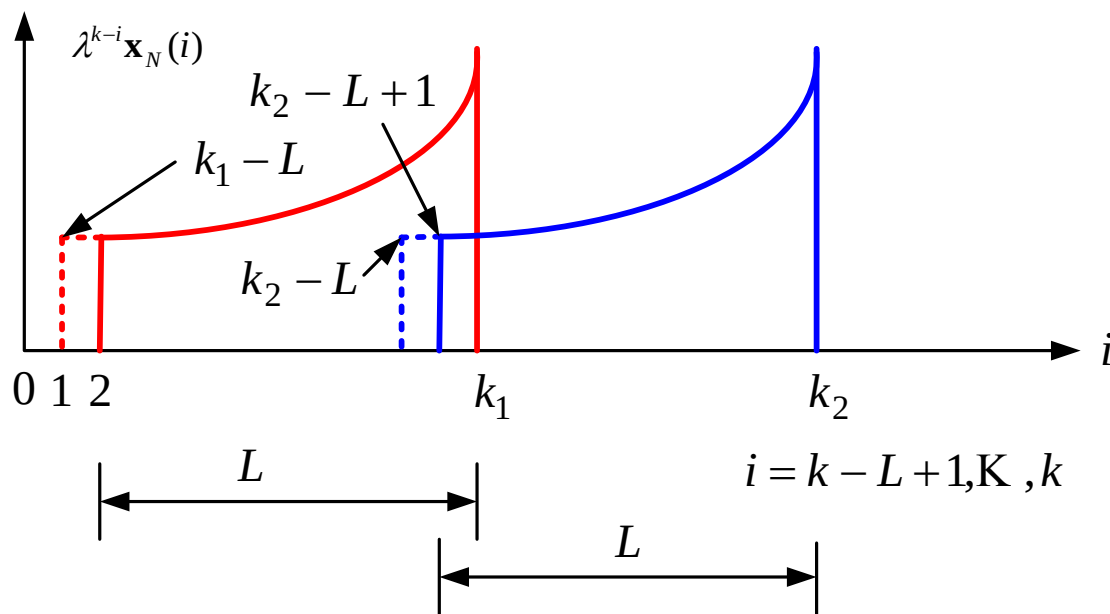
$$\begin{aligned}\mathbf{R}_N(k) &= \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^H(i) = \\ &= \lambda \mathbf{R}_N(k-1) + \mathbf{x}_N(k) \mathbf{x}_N^H(k)\end{aligned}$$

Возрастающее окно:



$$\begin{aligned}\mathbf{R}_N(k) &= \sum_{i=k-L+1}^k \lambda^{k-i} \mathbf{x}_N(i) \mathbf{x}_N^H(i) = \\ &= \lambda \mathbf{R}_N(k-1) + \mathbf{x}_N(k) \mathbf{x}_N^H(k) - \\ &\quad - \mu \mathbf{x}_N(k-L) \mathbf{x}_N^H(k-L)\end{aligned}$$

Скользящее окно:

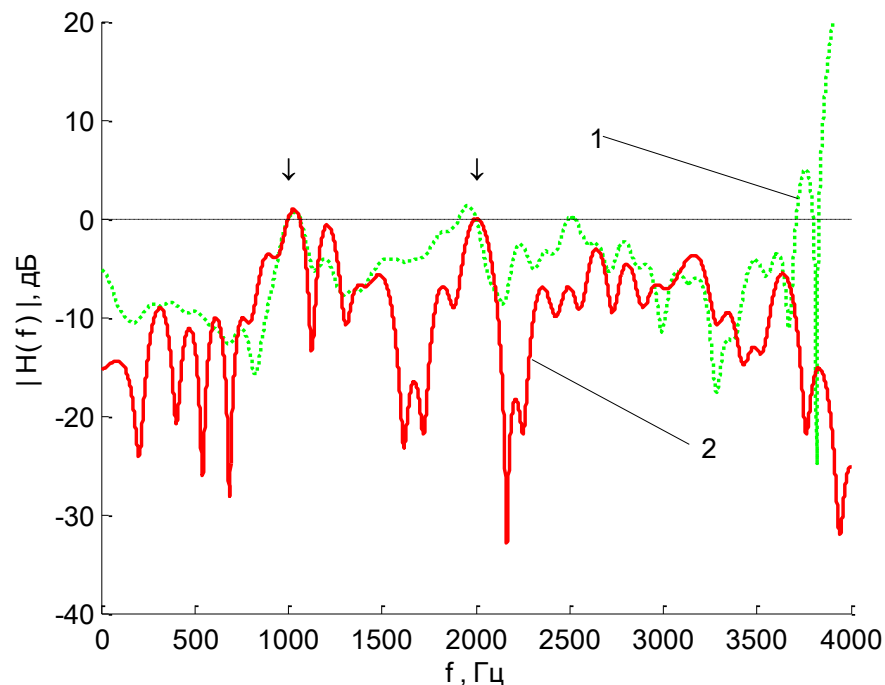


Что дает скользящее окно ?

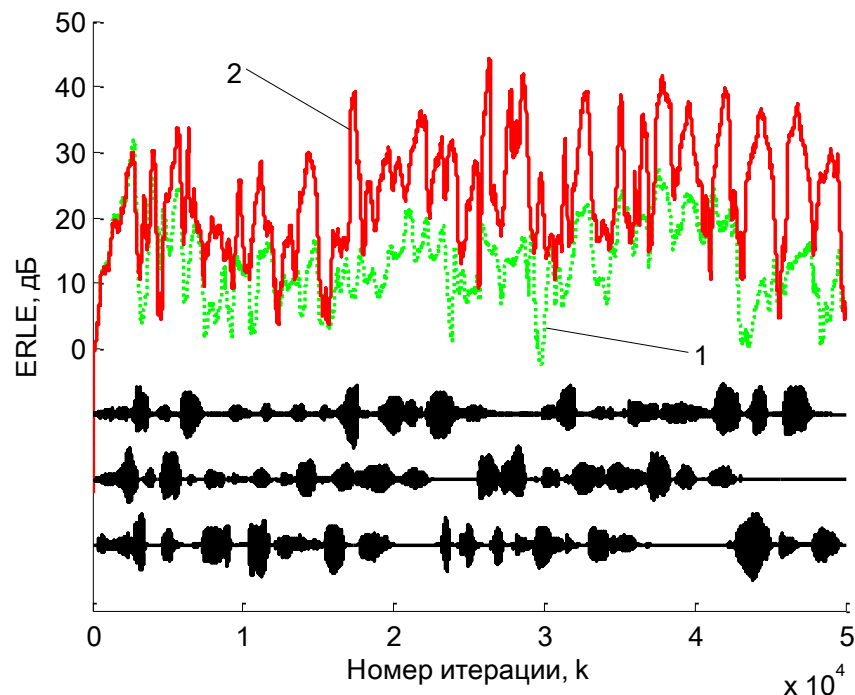
1 – RLS алгоритм без регуляризации, бесконечное окно

2 – RLS алгоритм без регуляризации, скользящее окно

АЧХ, дБ



ERLE, дБ

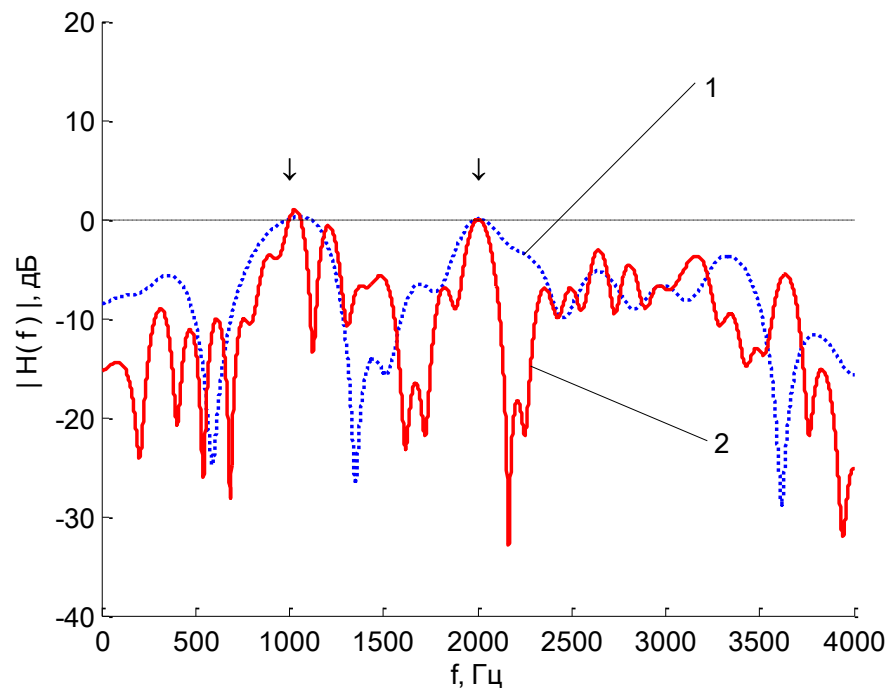


Что дает скользящее окно и регуляризация ?

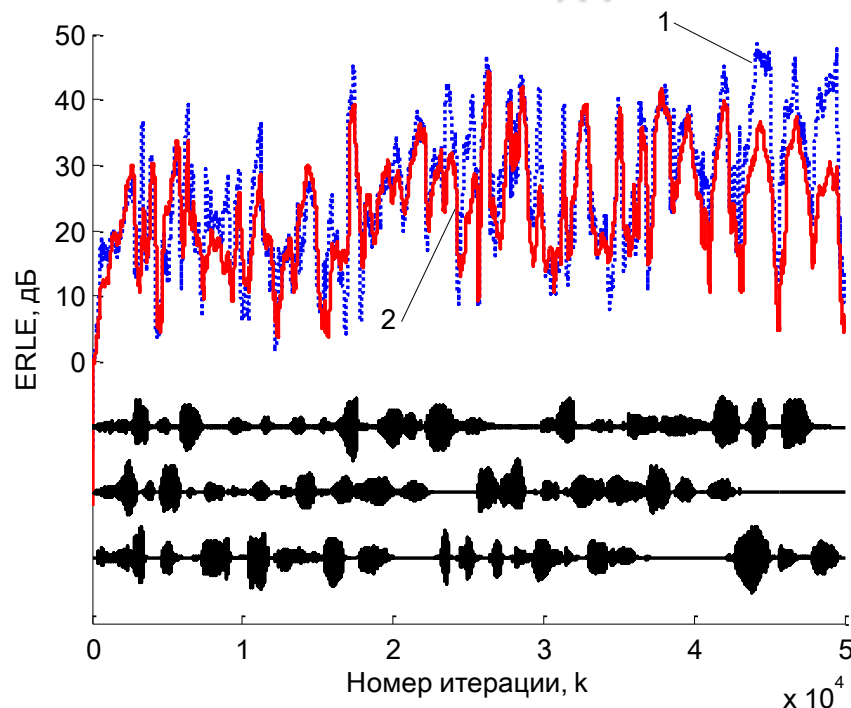
1 – регуляризованный RLS алгоритм, скользящее окно

2 – RLS алгоритм без регуляризации, скользящее окно

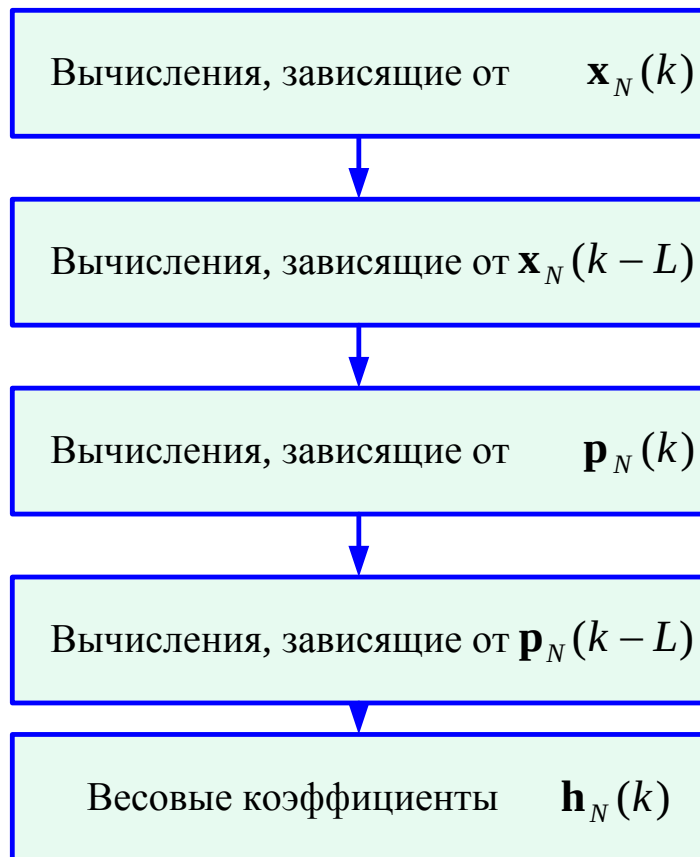
АЧХ, дБ



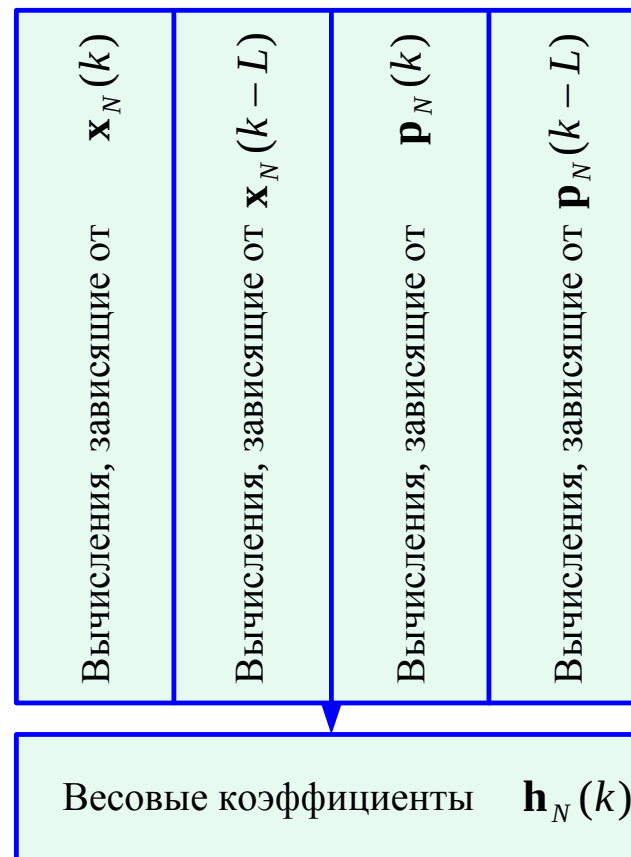
ERLE, дБ



Можно ли распараллелить вычисления ?

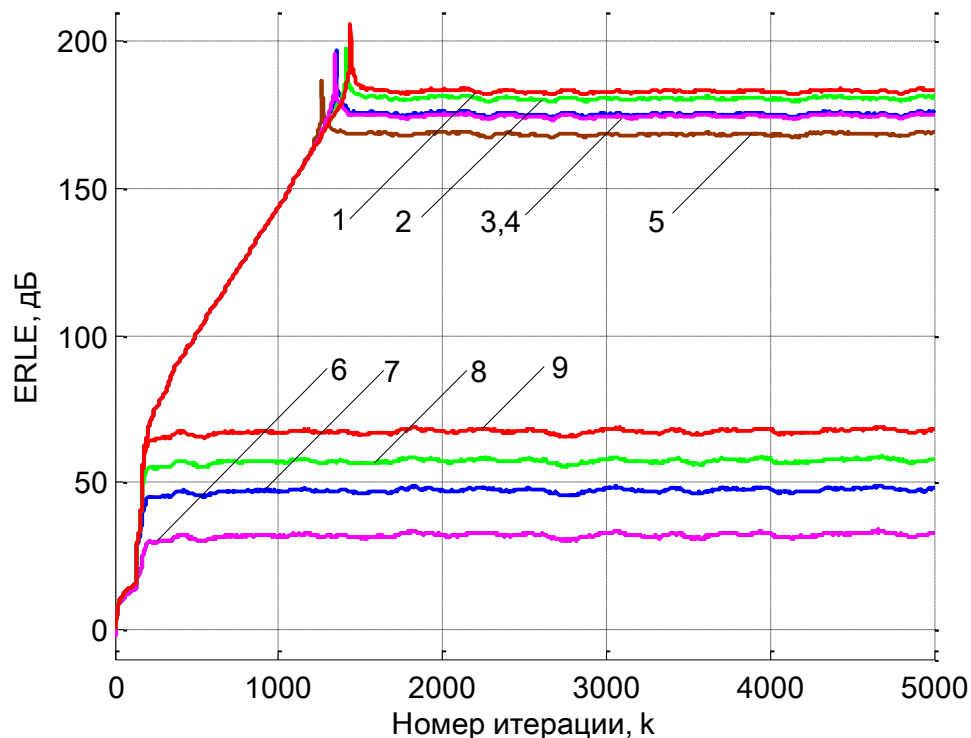


Последовательные
вычисления



Параллельные
вычисления

Одинаковые или разные RLS-алгоритмы ?



- 1 – RLS-алгоритм на основе прямого QR-разложения (с операциями извлечения квадратного корня);
- 2 – лестничный алгоритм на основе априорных и апостериорных ошибок без обратных связей;
- 3 – RLS-алгоритм;
- 4 – RLS-алгоритм на основе прямого QR-разложения (с операциями извлечения квадратного корня);
- 5 – FK-алгоритм;
- 6 – ОСШ=30 дБ;
- 7 – ОСШ=40 дБ;
- 8 – ОСШ=50 дБ;
- 9 – ОСШ=60 дБ

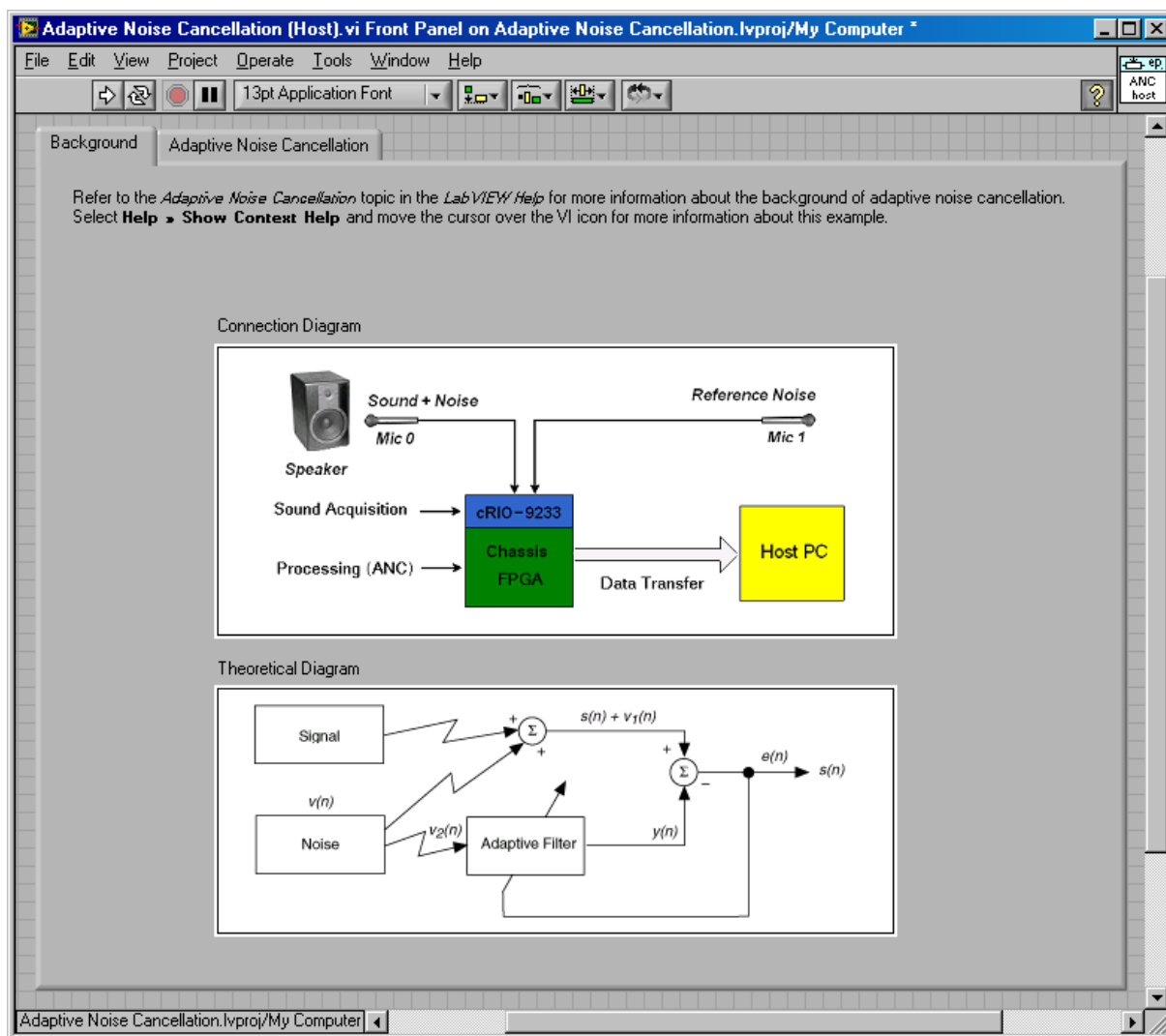
Что содержит LabVIEW Adaptive Filter Toolkit ?

1. Active Noise Control – активная компенсация шума.
2. Adaptive Channel Equalization – адаптивное выравнивание каналов связи.
3. Adaptive Inverse Control – адаптивное обратное управление.
4. Adaptive Echo Cancellation – адаптивная эхокомпенсация.
5. Adaptive Noise Cancellation – адаптивная компенсация шума.
6. Adaptive Linear Prediction – адаптивное линейное предсказание.
7. System Identification – идентификация системы.

Какие фильтры в LabVIEW Adaptive Filter Toolkit ?

1. Standard LMS
2. Normalized LMS
3. Leaky LMS
4. Normalized leaky LMS
5. Sign LMS (sign-error, sign-data, sign-sign)
6. Fast block LMS
7. Standard RLS
8. QR decomposition based RLS
9. Filtered-x LMS
10. Normalized filtered-x LMS

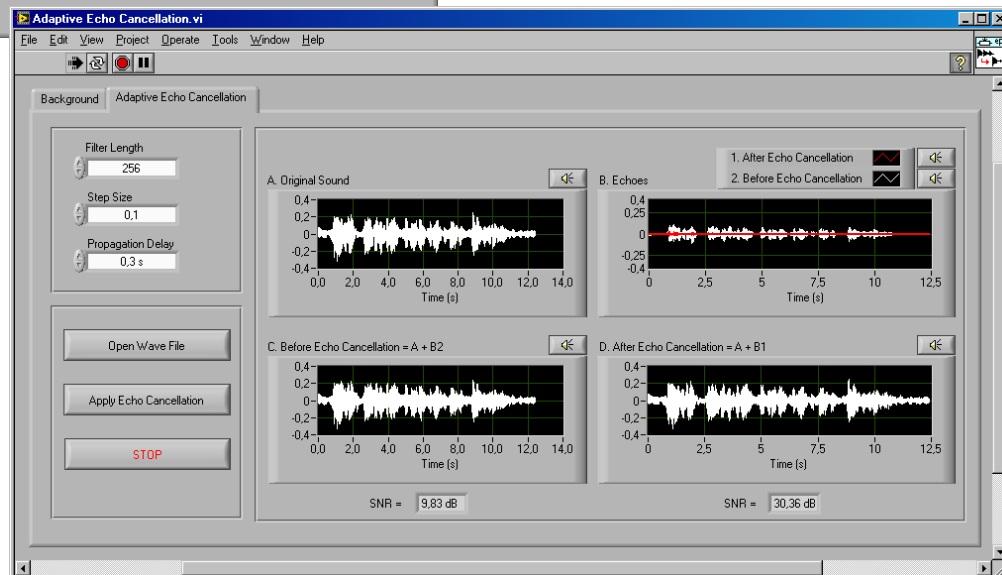
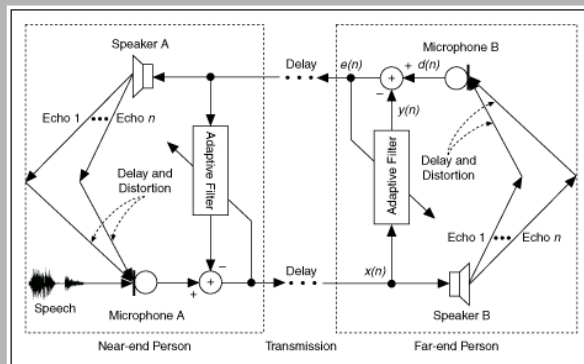
LabVIEW 2009: адаптивное подавление шума



LabVIEW 2009: эхокомпенсация

Background Adaptive Echo Cancellation

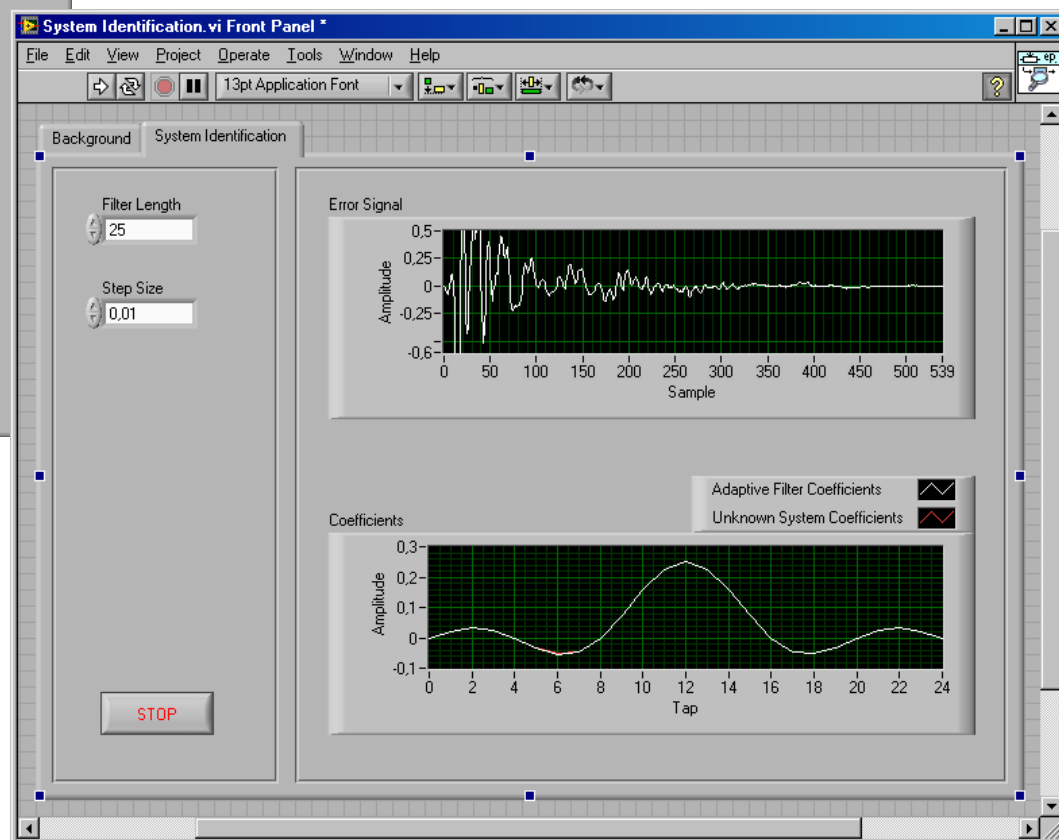
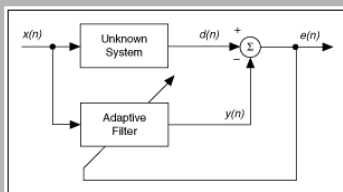
Refer to the *Adaptive Echo Cancellation* topic in the *LabVIEW Help* for more information about the background of adaptive echo cancellation. Select **Help » Show Context Help** and move the cursor over the VI icon of this example for more information.



LabVIEW 2009: идентификация системы

Background System Identification

Refer to the *System Identification* topic in the *LabVIEW Help* for more information about the background of system identification. Select **Help » Show Context Help** and move the cursor over the VI icon of this example for more information.



LabVIEW 2009: выравнивание каналов связи

Background Scatter Diagram

Refer to the *Adaptive Channel Equalization* topic in the *LabVIEW Help* for more information on channel equalization. Select **Help » Show Context Help** and move the cursor over information.

Background Scatter Diagram

Filter Length:

Forgetting Factor:

Mode:

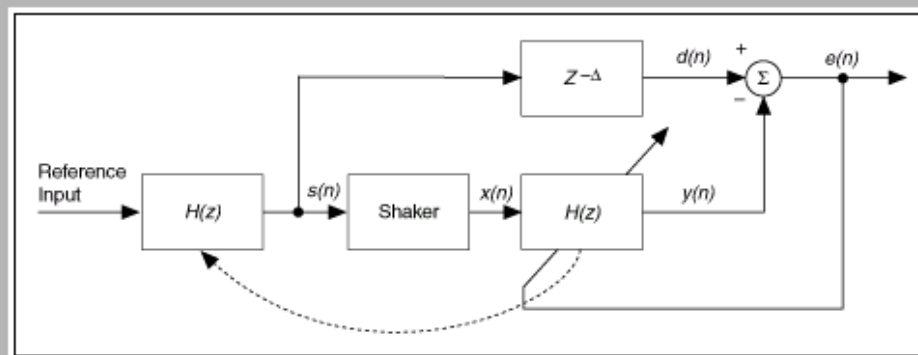
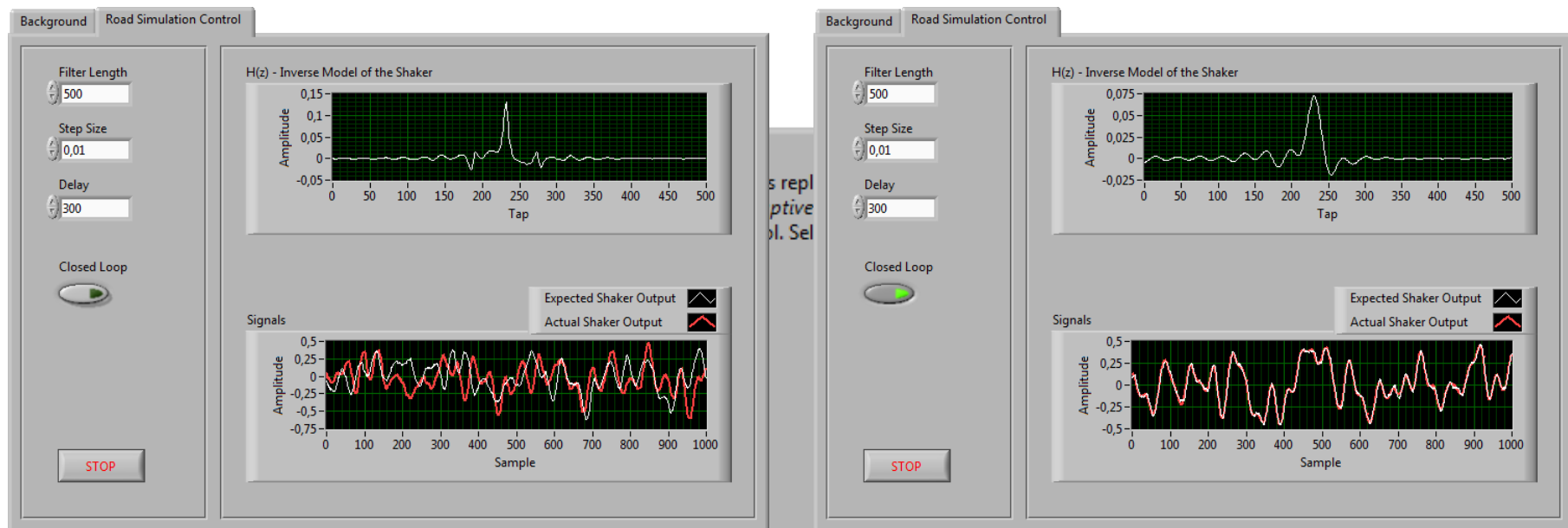
STOP

Transmitted Symbols

Received Symbols

Equalizer Output

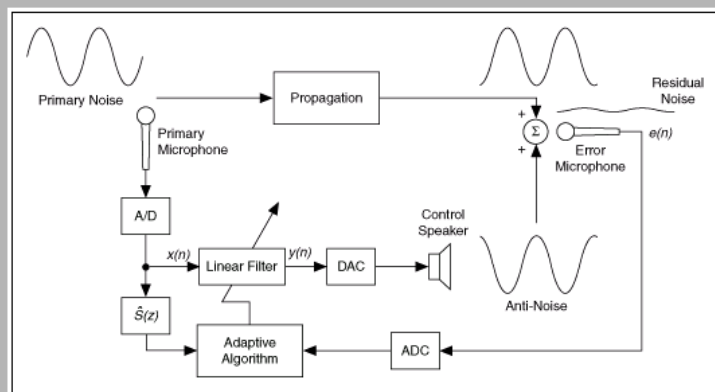
LabVIEW 2009: обратное управление



LabVIEW 2009: активная компенсация шума

Background Active Noise Control

Refer to the *Active Noise Control* topic in the *LabVIEW Help* for more information about the background of active noise control. Select **Help » Show Context Help** and move the cursor over the VI icon of this example for more information.



Background Active Noise Control

Filter Length

300

Step Size

0,0002

Primary Noise

0

frequency

120

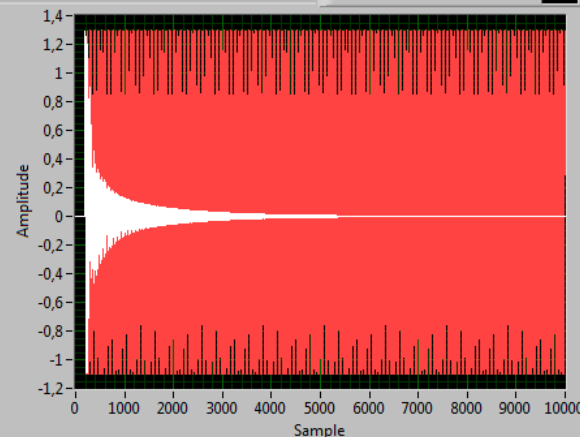
amplitude

1

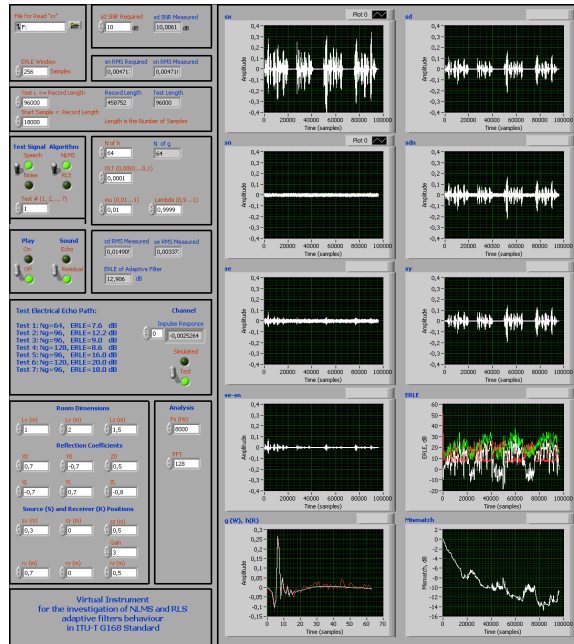
phase

55

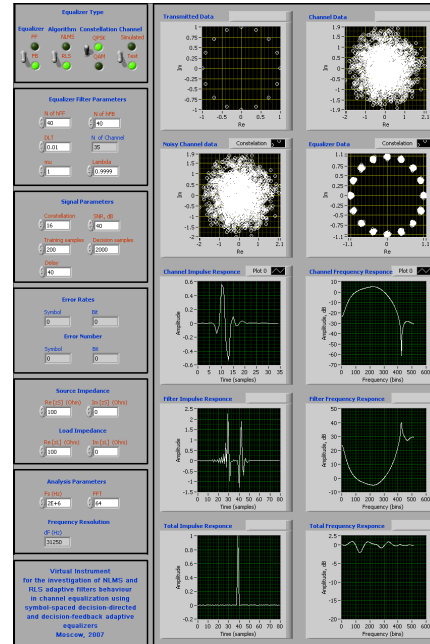
Noise



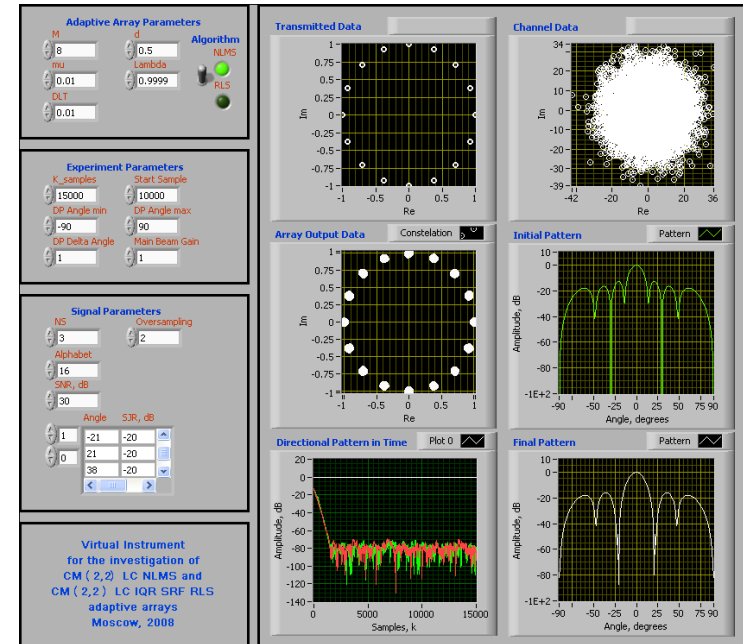
Виртуальные инструменты в LabVIEW



Эхокомпенсатор



Эквалайзер

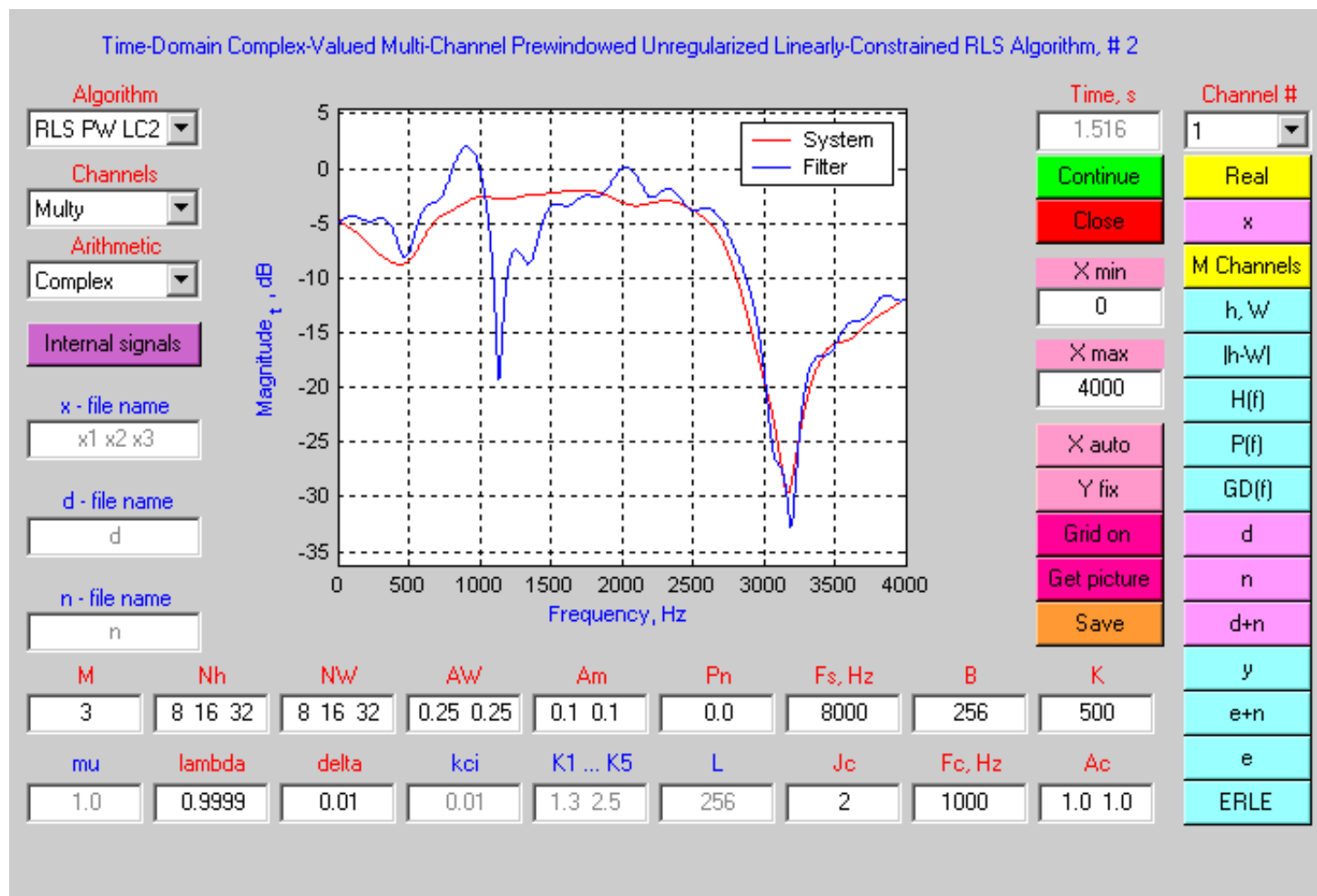


Антенная решетка

Какие фильтры в Filter Design Toolbox в MATLAB 2009?

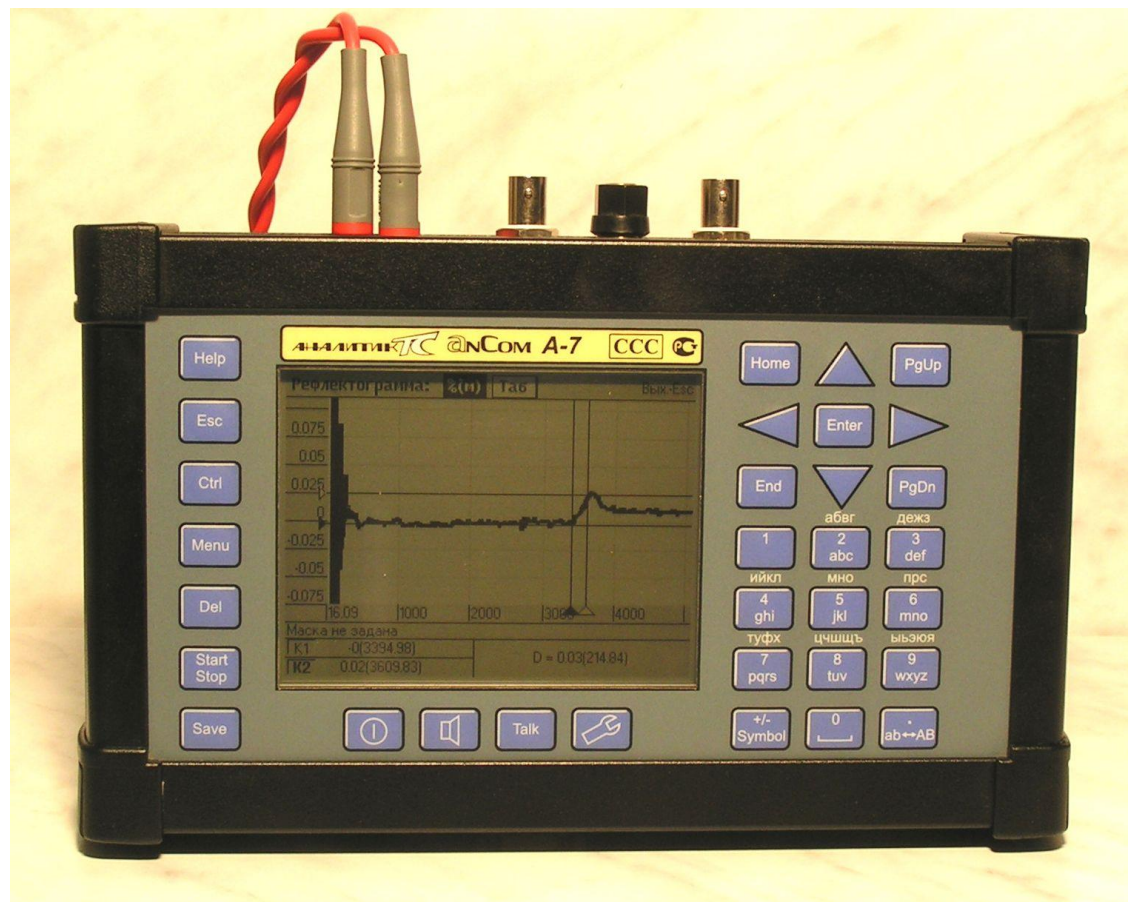
1. Least Mean Squares (LMS) - 10 шт.
2. Recursive Least Squares (RLS) - 7 шт.
3. Affine Projection (AP) - 3 шт.
4. Frequency Domain - 6 шт.
5. Lattice Filter - 3 шт.

Модели адаптивных фильтров (~ 400 шт. !)



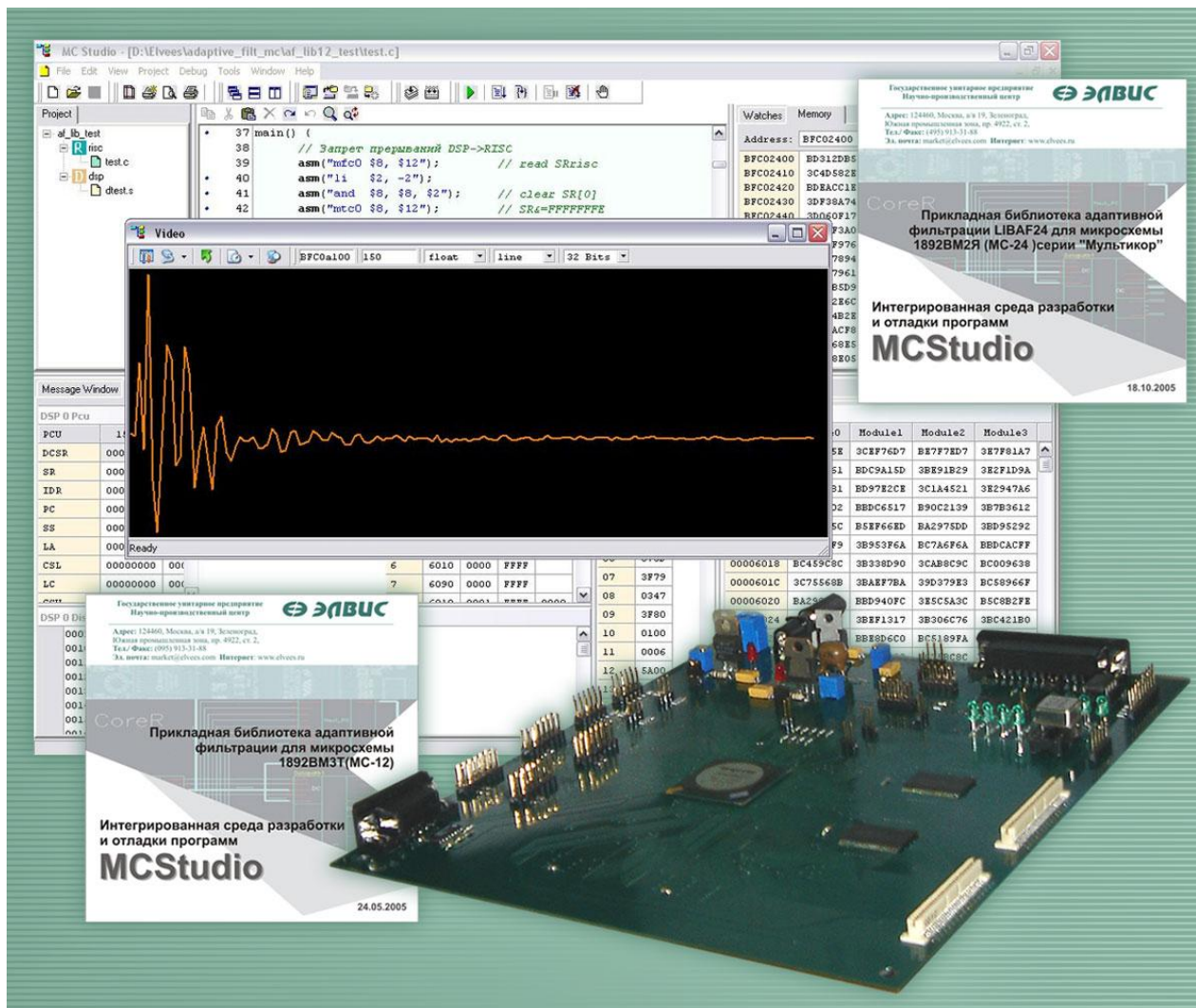
Графический интерфейс пользователя на языке MATLAB

Изделие компании «Аналитик ТелекомСистемы»



Анализатор систем передачи и кабелей связи AnCom A-7

Прикладная библиотека адаптивной фильтрации



The image displays the MCStudio integrated development environment (IDE) for the 1892BM24 microcontroller. The main window shows the source code for a test program, which includes assembly instructions for reading and clearing status registers (SR) and setting the SR4 register to FFFFFFFF. A video window in the foreground shows a waveform plot. The bottom of the screen shows a photograph of the physical microcontroller board, which is populated with various components including a microcontroller, capacitors, and connectors.

MCStudio
Интегрированная среда разработки и отладки программ
18.10.2005

Прикладная библиотека адаптивной фильтрации LIBAF24 для микросхемы 1892BM24 (MC-24) серии "Мультикор"

Государственное учреждение
Научно-производственный центр
ЭЛВИС
Адрес: 124460, Москва, в/п 19, Зеленоград,
Южная промышленная зона, стр. 4022, в/п 2,
Тел.: Факс: (495) 913-31-48
Эл. почта: market@elvis.com Интернет: www.elvis.ru

Государственное учреждение
Научно-производственный центр
ЭЛВИС
Адрес: 124460, Москва, в/п 19, Зеленоград,
Южная промышленная зона, стр. 4022, в/п 2,
Тел.: Факс: (495) 913-31-48
Эл. почта: market@elvis.com Интернет: www.elvis.ru

Прикладная библиотека адаптивной фильтрации для микросхемы 1892BM3T (MC-12)

Интегрированная среда разработки и отладки программ
MCStudio
24.05.2005



**Спасибо
за
внимание!**



Эл. почта: djigan@elvees.com Тел: +7-499-731-1961