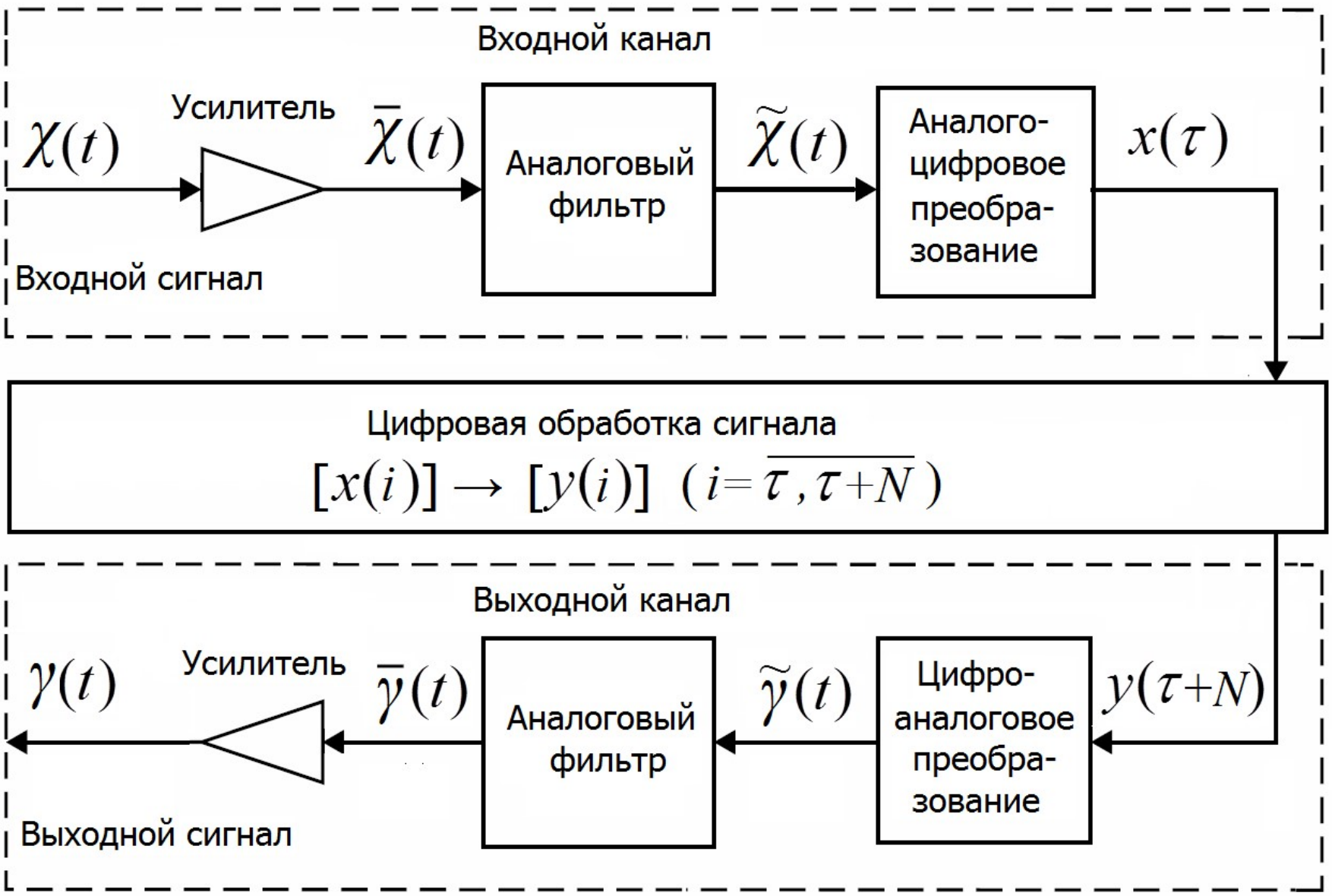
Микропроцессорные устройства обработки сигналов

Лекция L16
«Стандартная библиотека»

http://vykhovanets.ru/course67/

Обработка сигналов



Методы обработки сигналов

$$x(0),\ x(1),\ ...,\ x(N-1) \Rightarrow y(0),\ y(1),\ ...,\ y(M-1)$$

- **Передискретизация** – изменении частоты дискретизации
- **Преобразование** – функциональное преобразование сигнала
- **Свертка и корреляция** – обработка периодических сигналов
- **Фильтрация** – обработка непериодических сигналов
- **Спектральная обработка** – обработка сигнала в спектральной области

Стандартные библиотеки

- CSL - Chip Support Library (библиотека поддержки) [P01] – функции синхронизации (csl_ppl.h), питания (csl_pwr.h), канала прямого доступа к памяти (csl_dma.h), прерывания (csl_irq.h), аналого-цифрового преобразователя (csl_adc.h), входав-выходов общего назначения (csl_gpio.h), интерфейса I2C (csl_i2c.h), …

- DSPLIB (dsplib.h, 55xdspx.lib) [P02, D09] – функции свертки, корреляции, КИХ- и БИХ-фильтрации, адаптивной фильтрации, быстрого преобразования Фурье, математические функции, матричные функции, вспомогательные функции.

- IMGLIB (imglib.h, 55ximage.lib) – функции компрессии и декомпрессии, анализа изображений, фильтрации изображений, преобразования форматов.

Библиотека обработки сигналов

- Быстрое преобразование Фурье (FFT)
- Фильтрация нерекурсивная (FIR)
- Фильтрация рекурсивная (IIR)
- Фильтрация адаптивная (DLMS)
- Свертка (convolution)
- Корреляция (CORR)
- Тригонометрические функции (SINE, …)
- Векторные функции (SQRT, SUB, DIV, …)
- Матричные функции (MUL, TRANS, …)
- Вспомогательные функции (RAND, …)

Передискретизация

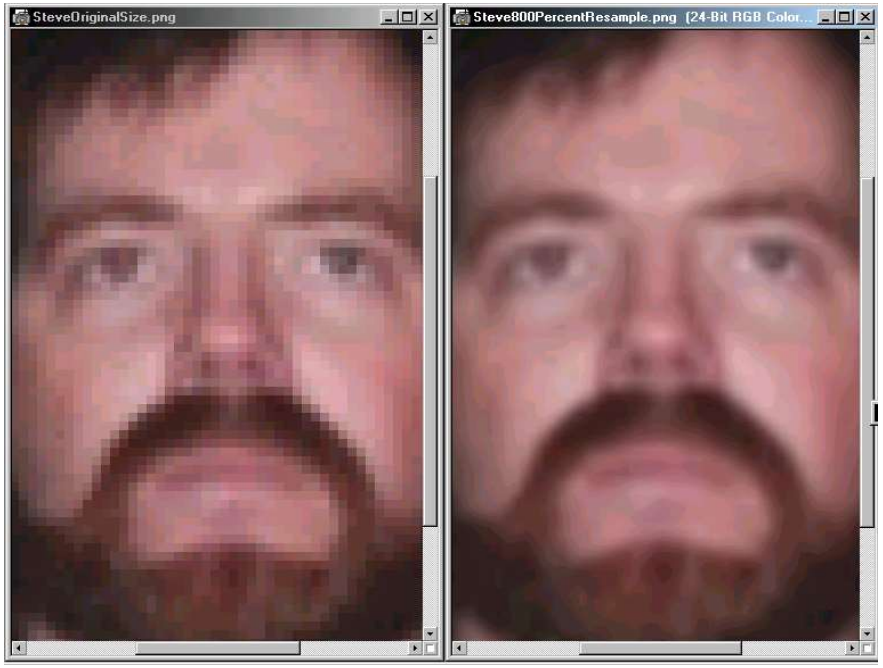
$$x(0),\ x(1),\ \ldots,\ x(N-1) \Rightarrow y(0),\ y(1),\ \ldots,\ y(M-1)$$

- **Интерполяция** – увеличение частоты дискретизации,

$$N < M.$$

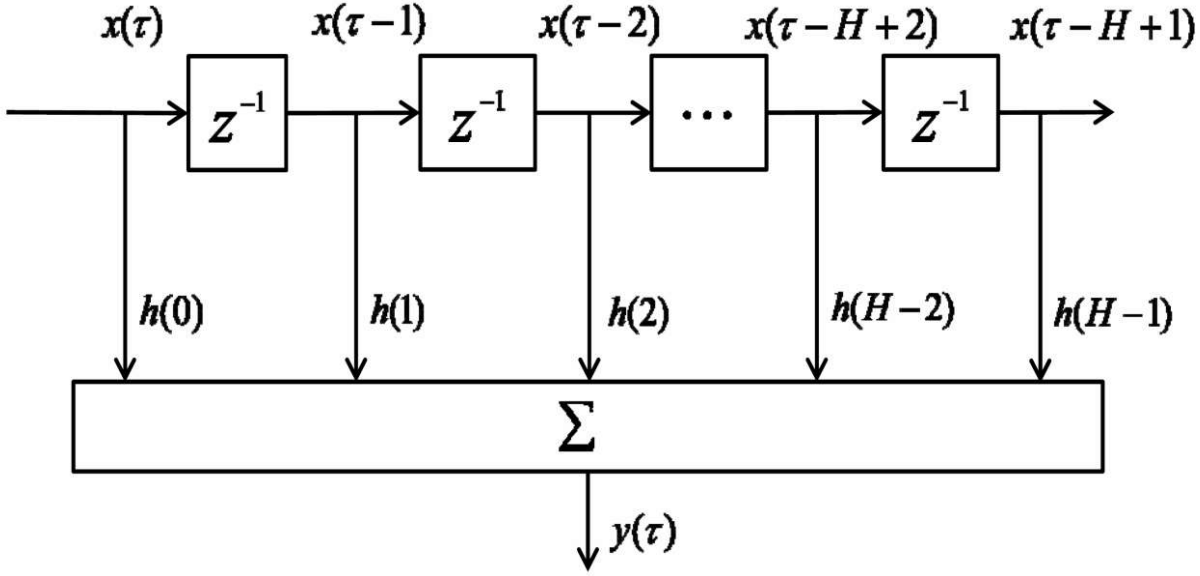
- **Децимация** – уменьшение частоты дискретизации,

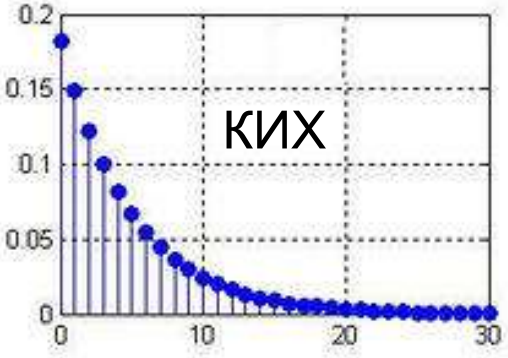
$$N > M.$$

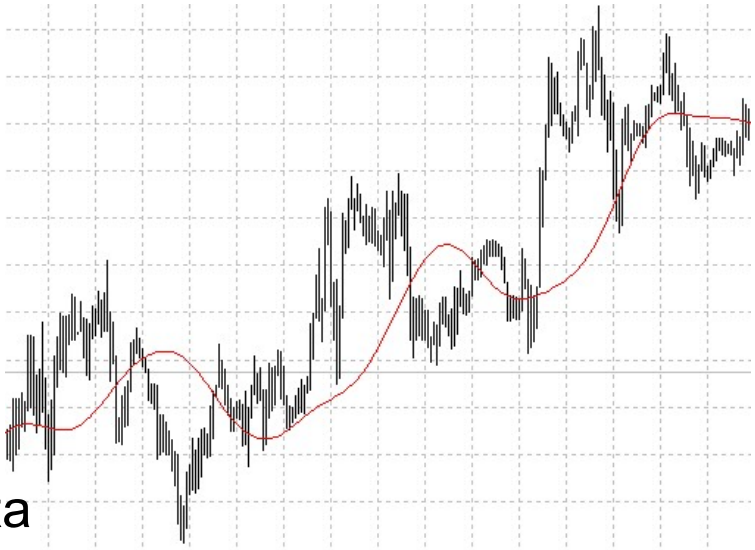


Нерекурсивный фильтр

$$y(\tau) = \sum_{i=0}^{H-1} h(i) \times x(\tau - i) \ (\tau = \overline{0, N-1})$$



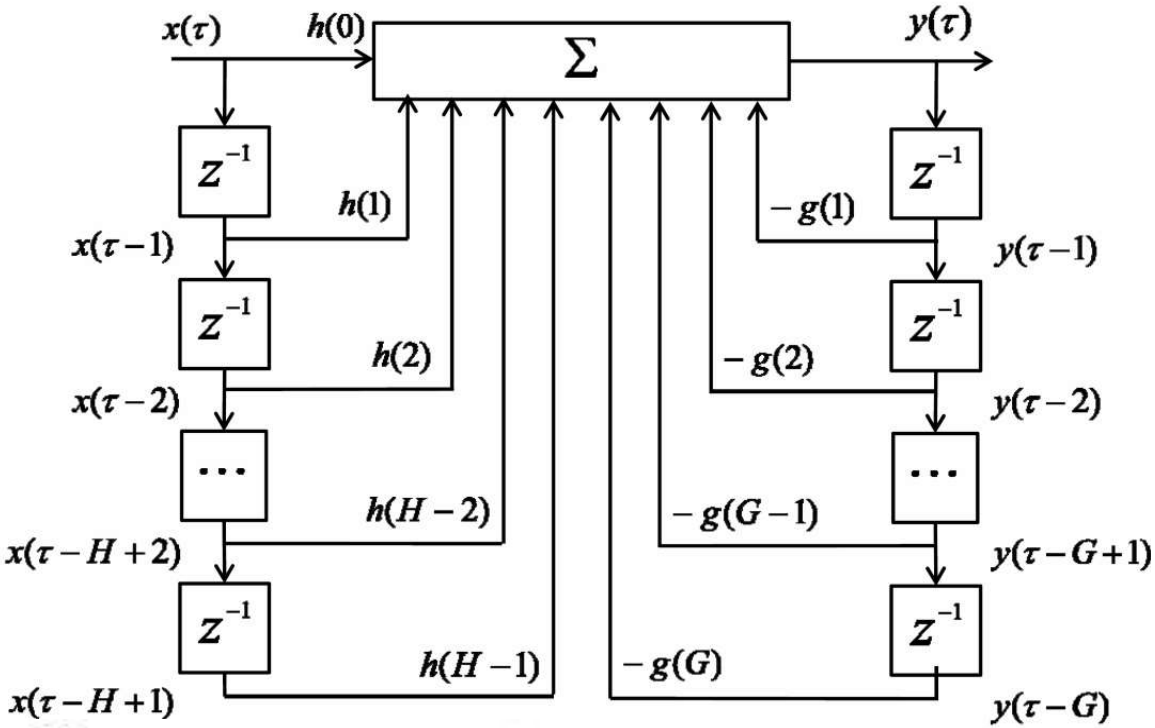


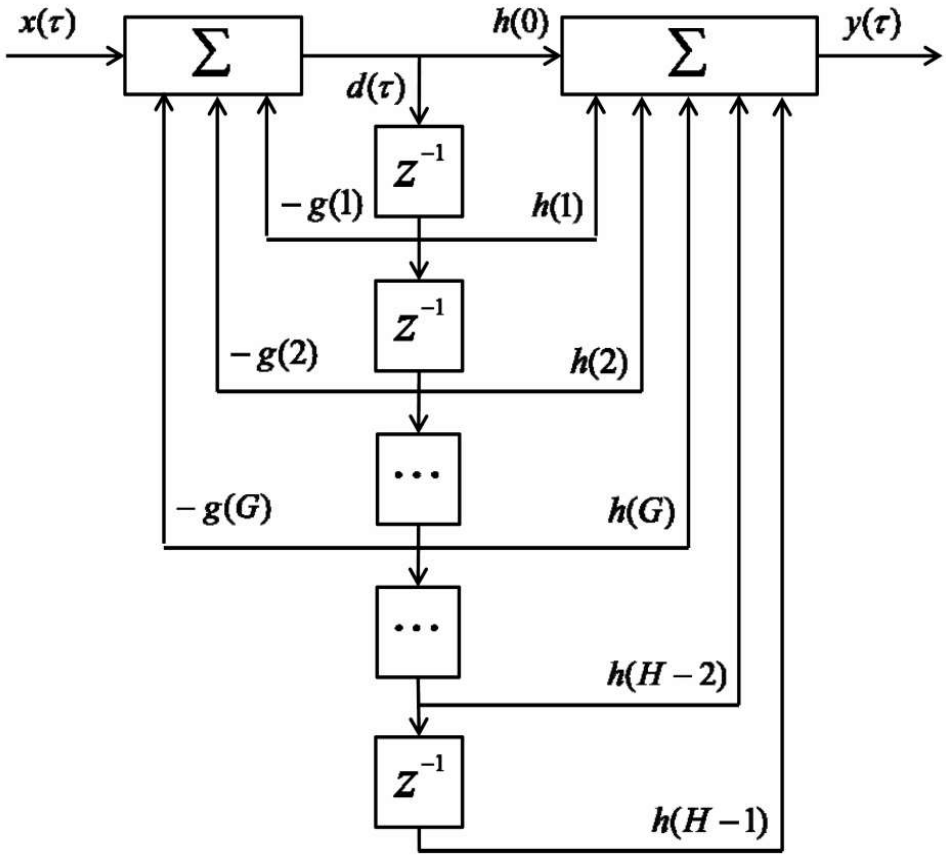


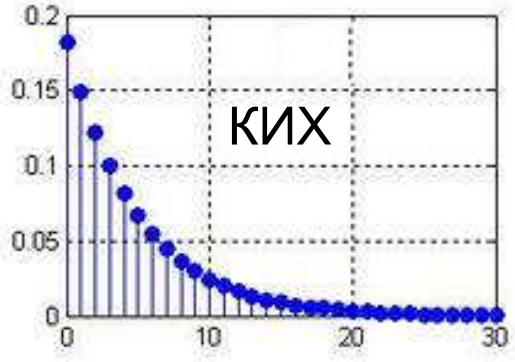
КИХ (**FIR**) – конечная импульсная характеристика

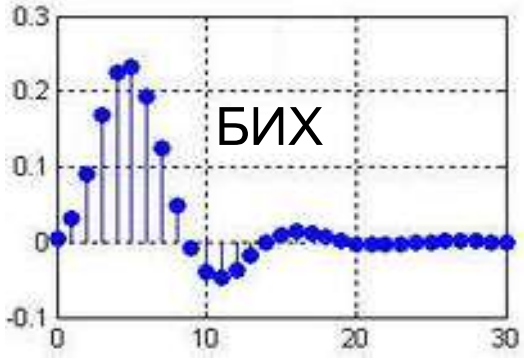
Рекурсивный фильтр

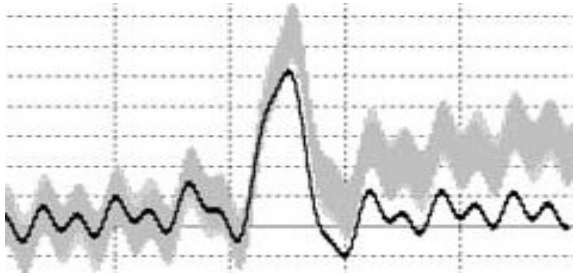
$$y(\tau) = \sum_{i=0}^{H-1} h(i) \times x(\tau - i) - \sum_{i=1}^{G} g(j) \times y(\tau - j) \, (\tau = \overline{0, N-1})$$





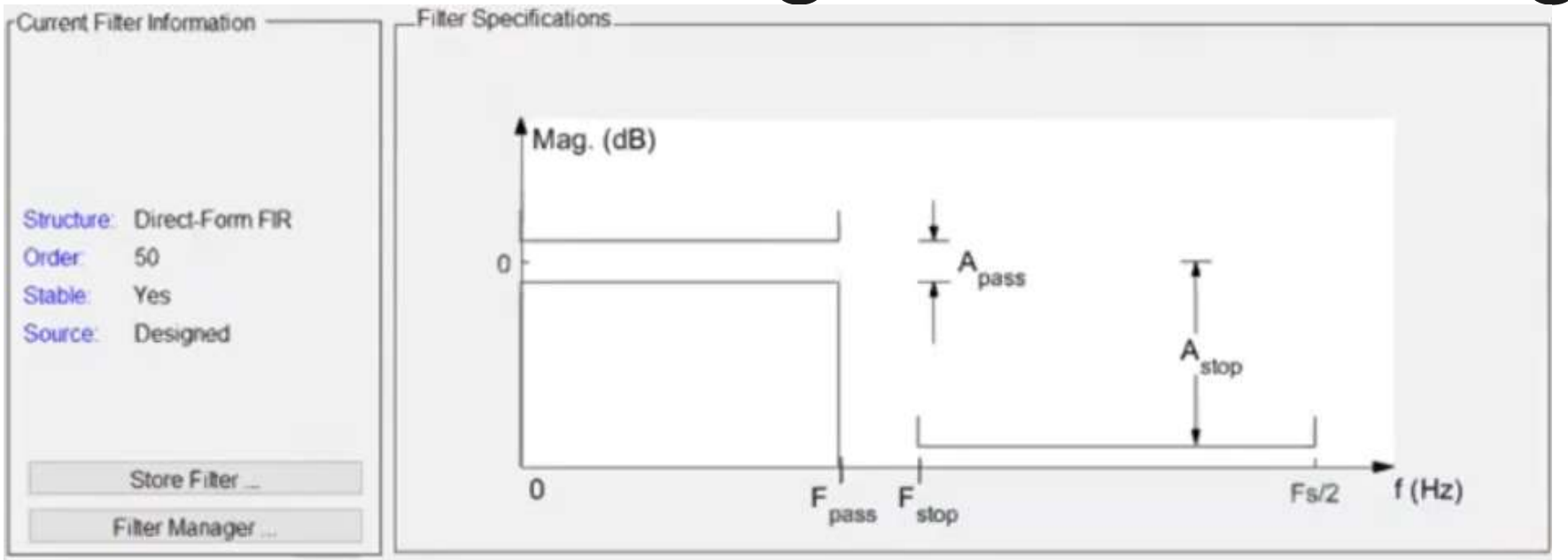


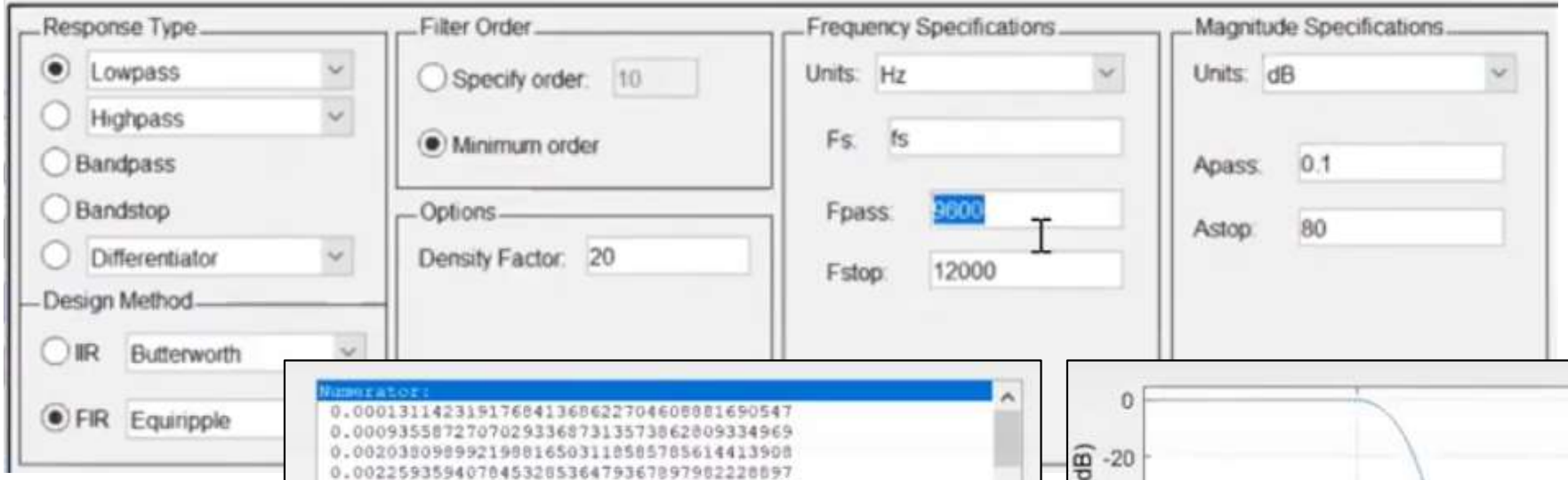


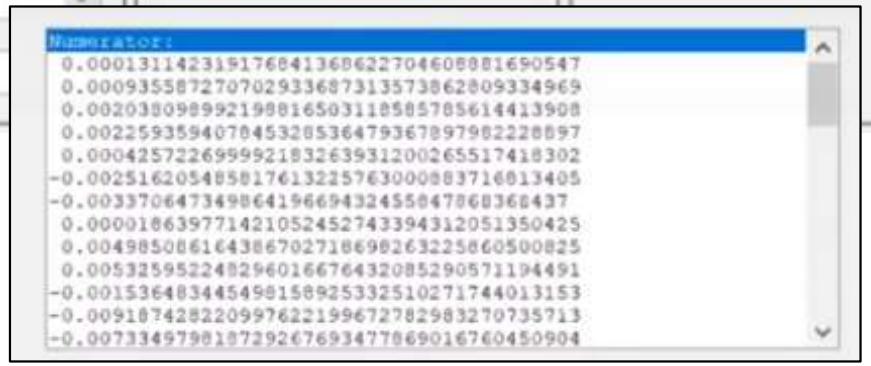


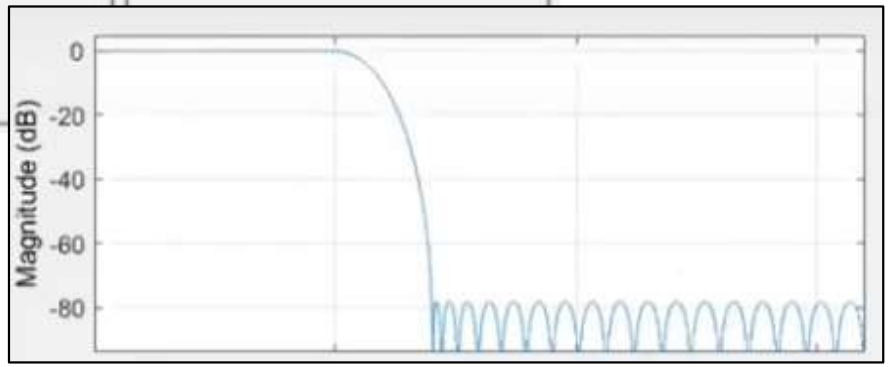
БИХ (**IIR**) – бесконечная импульсная характеристика

MatLab Signal Processing

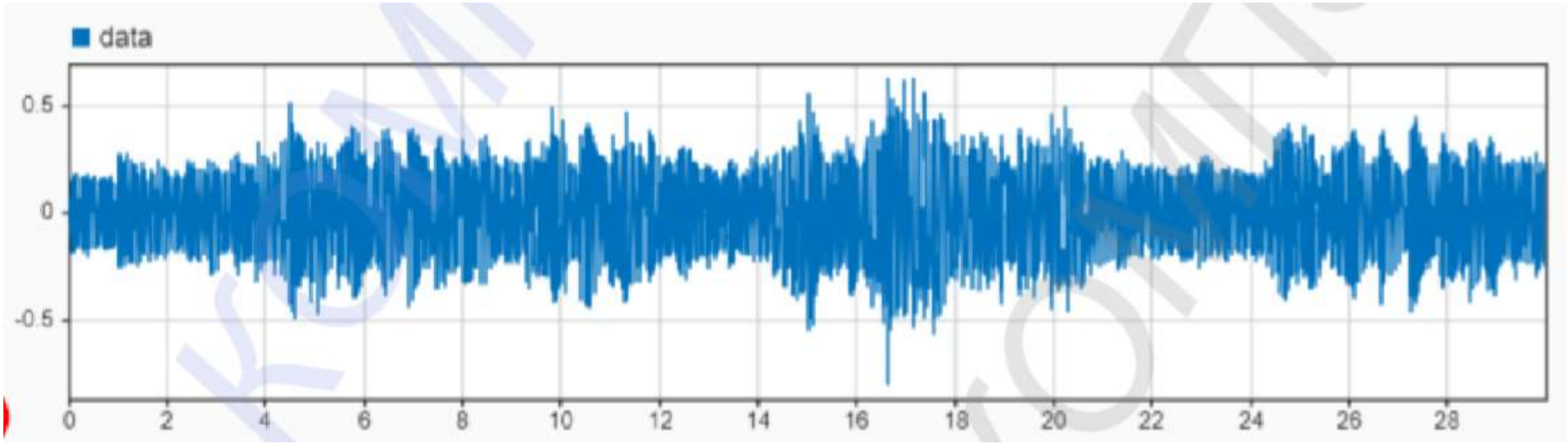


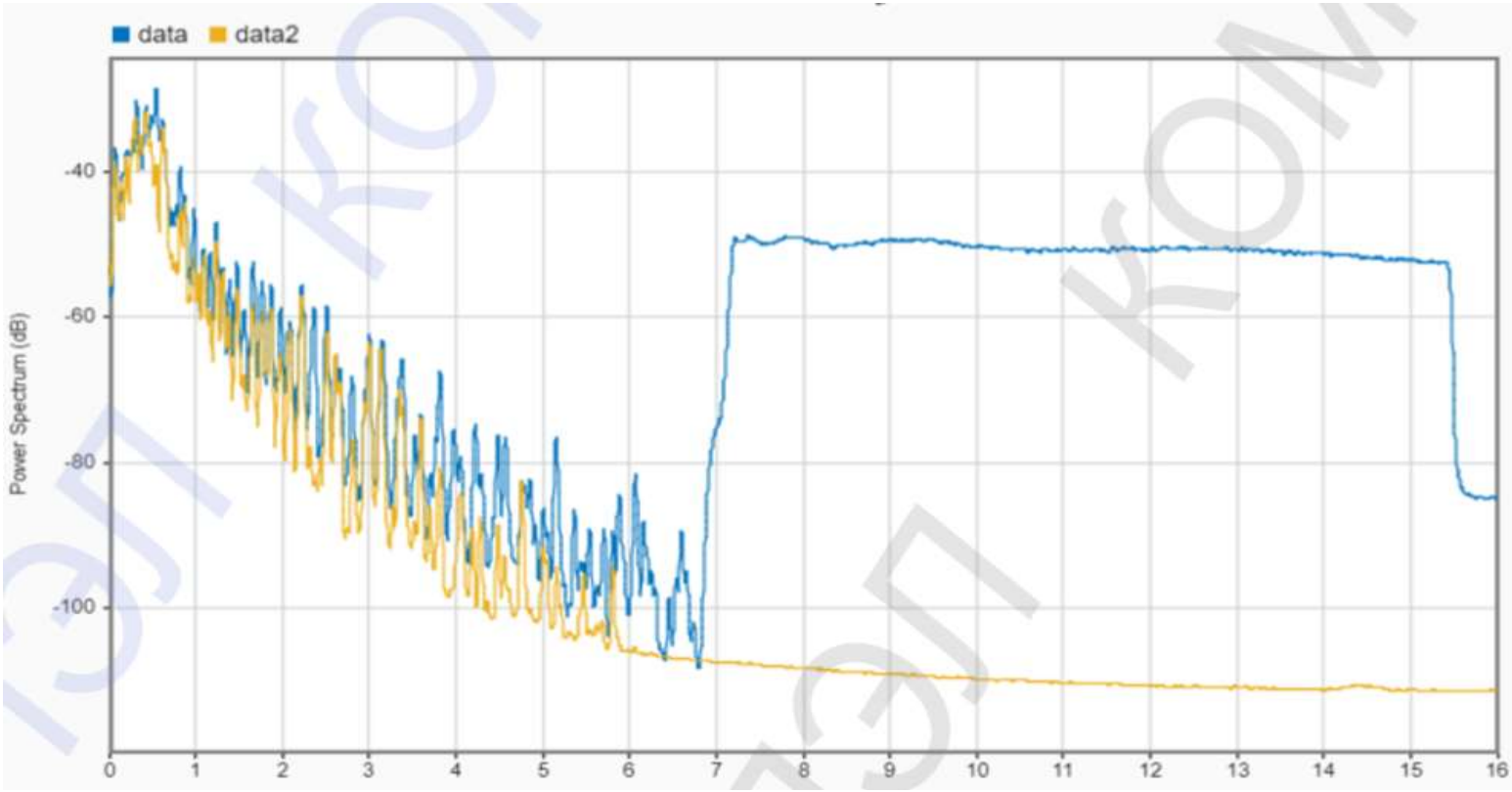






Фильтрация


data
0.5
0
-0.5
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28


data
data2
Power Spectrum (dB)
-40
-60
-80
-100
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

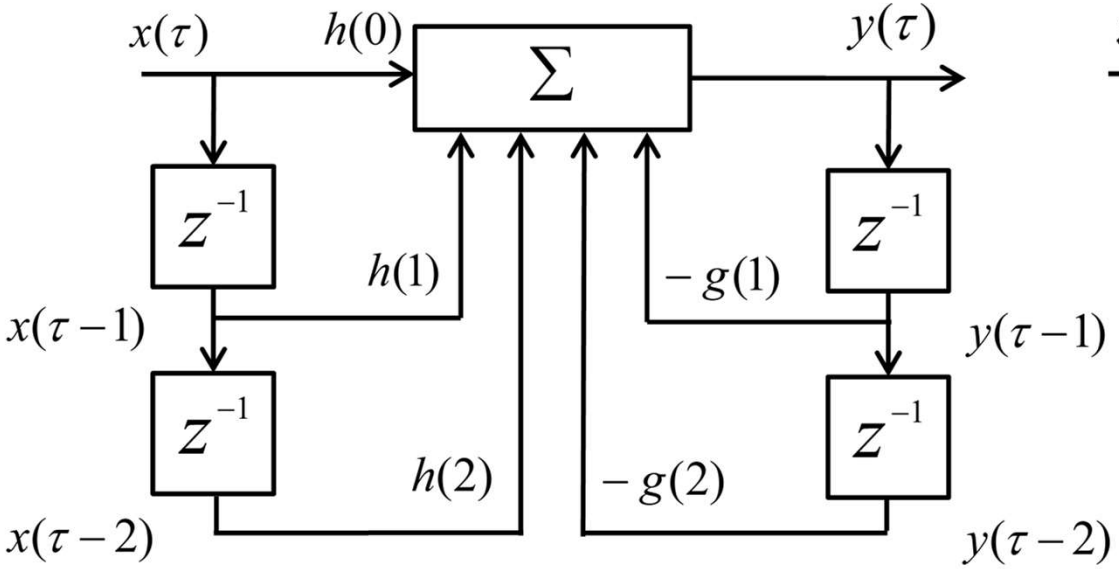
Сравнение фильтров

БИХ-фильтры	КИХ-фильтры
Более эффективны	Менее эффективны
Есть аналоговый эквивалент	Нет аналогового эквивалента
Могут быть нестабильными	Всегда стабильные
Нелинейная фазовая характеристика	Линейная фазовая характеристика
Больше «звон» при наличии ложных сигналов	Меньше «звон» при наличии ложных сигналов
Доступны средства САПР	Доступны средства САПР
Децимация не влияет на эффективность	Децимация увеличивает эффективность

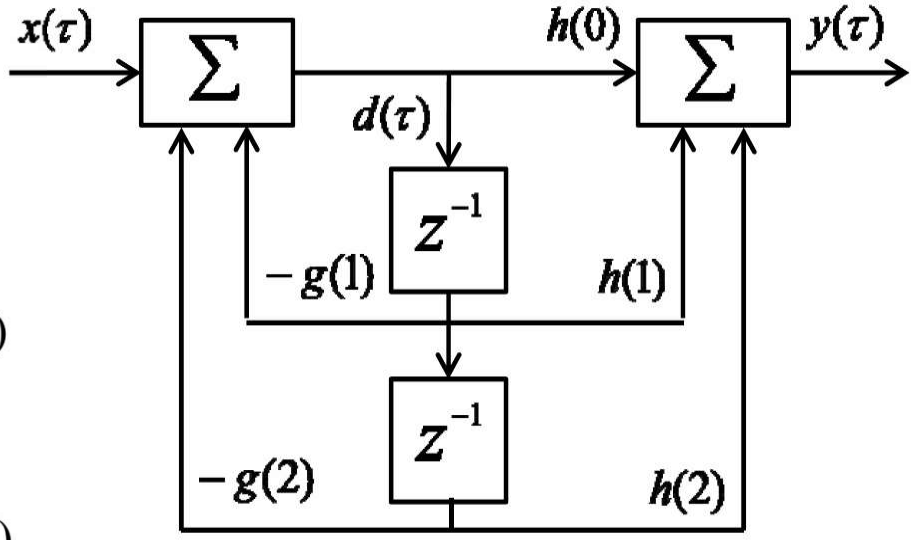
Биквадратный фильтр

$$K(z)=\frac{Y(z)}{X(z)}=\frac{\sum_{i=0}^{H-1}h(i)z^{-i}}{1+\sum_{j=1}^{G-1}g(j)z^{-j}}=\frac{H(z)}{G(z)}=\frac{H_1(z)}{G_1(z)}\frac{H_2(z)}{G_2(z)}\cdots\frac{H_n(z)}{G_n(z)}$$

$$y(\tau)=h(0)\times x(\tau)+h(1)\times x(\tau-1)+h(2)\times x(\tau-2)-$$
$$-g(1)\times y(\tau-1)-g(2)\times y(\tau-2)$$



Форма 1

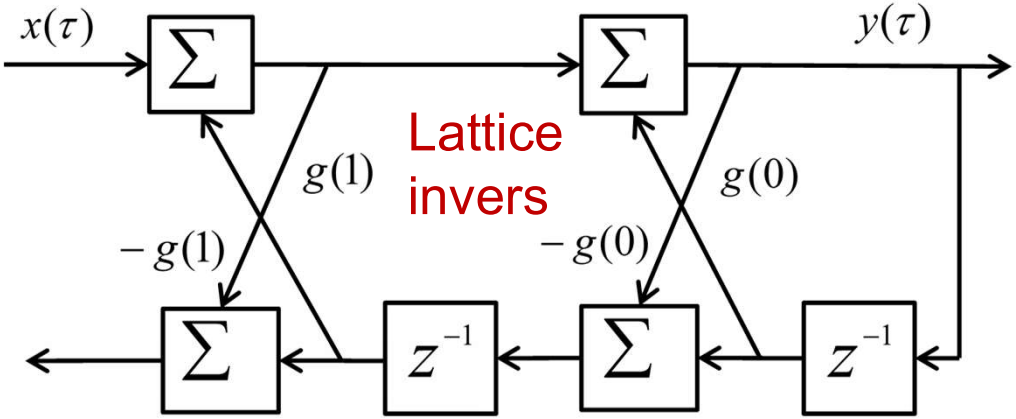


Форма 2

Решетчатый фильтр

БИХ $K(z) = \dfrac{1}{G(z)}$

КИХ $K(z) = H(z)$

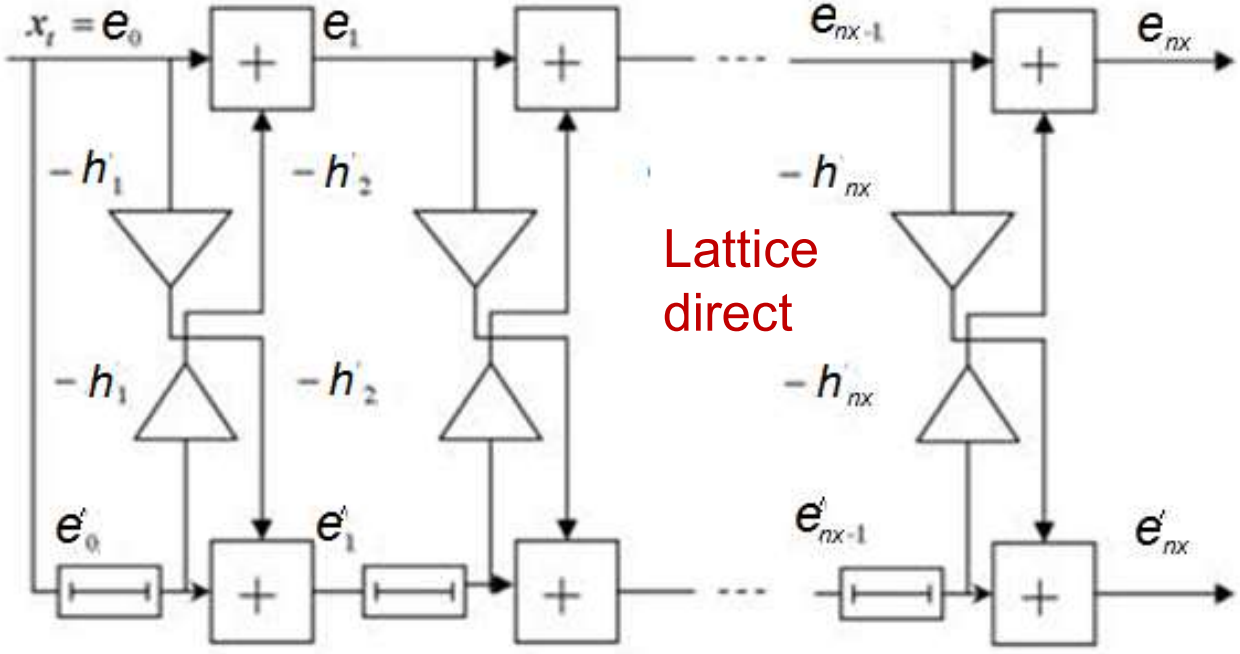


$$e_0[n] = e'_0[n] = x[n],$$

$$e_i[n] = e_{i-1}[n] - h_i e'_{i-1}[n-1], \quad i = 1, 2, \ldots, n_X$$

$$e'_i[n] = h_i e_{i-1}[n] + e'_{i-1}[n-1], \quad i = 1, 2, \ldots, n_X$$

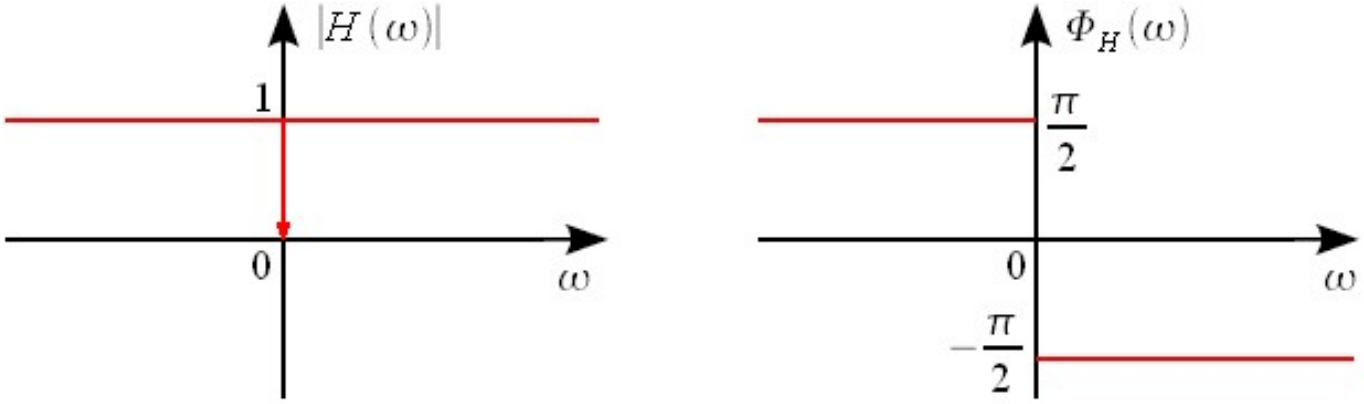
$$y[n] = e_{n_X}[n]$$



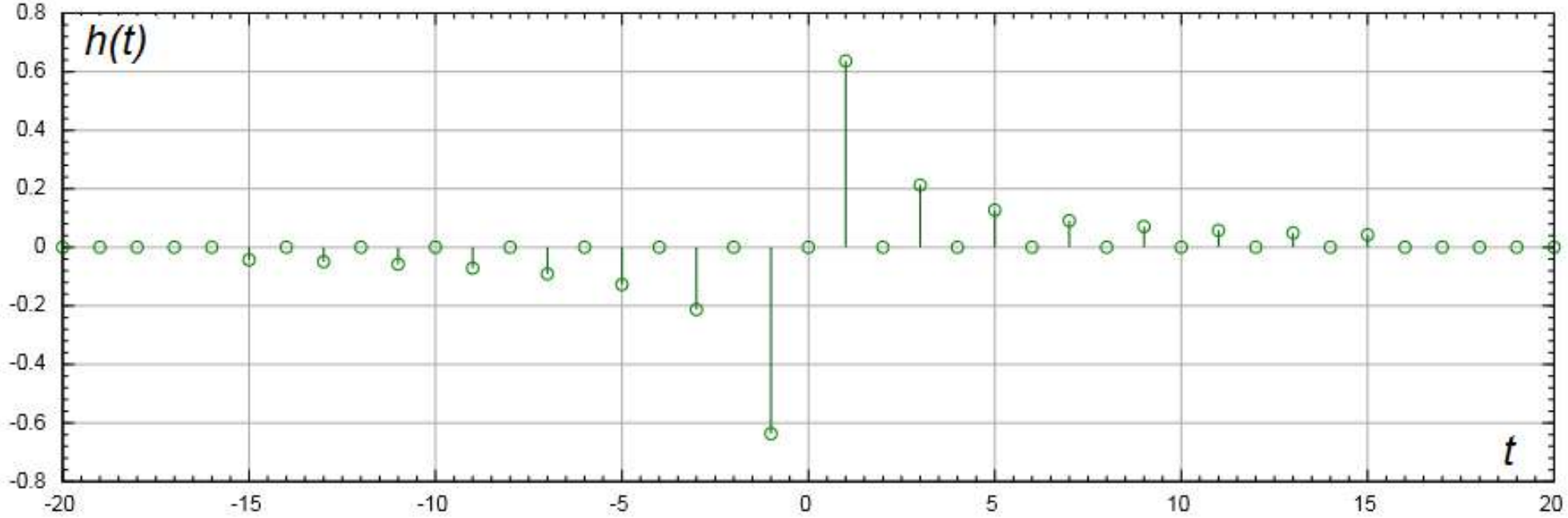
Преобразование Гильберта

$$\chi(t) = \frac{1}{\pi}\int_{-\infty}^{+\infty}\frac{x(\tau)}{t-\tau}d\tau, \qquad x(t) = \frac{1}{\pi}\int_{-\infty}^{+\infty}\frac{\chi(\tau)}{\tau-t}d\tau, \qquad \int_{-\infty}^{+\infty}\chi(t)x(t)dt = 0.$$

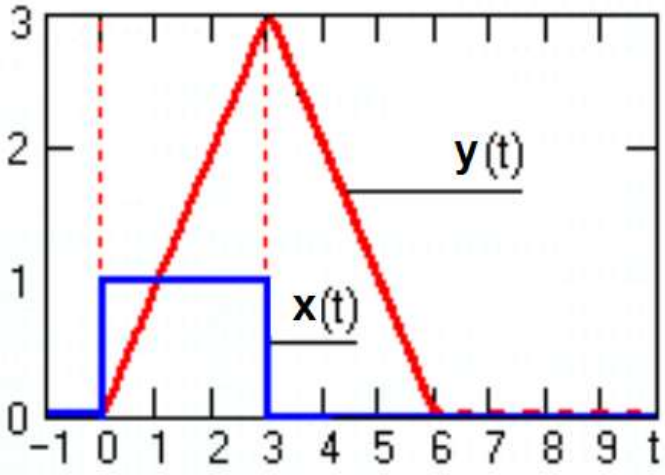
$$h(t) = \frac{1}{\pi t}$$

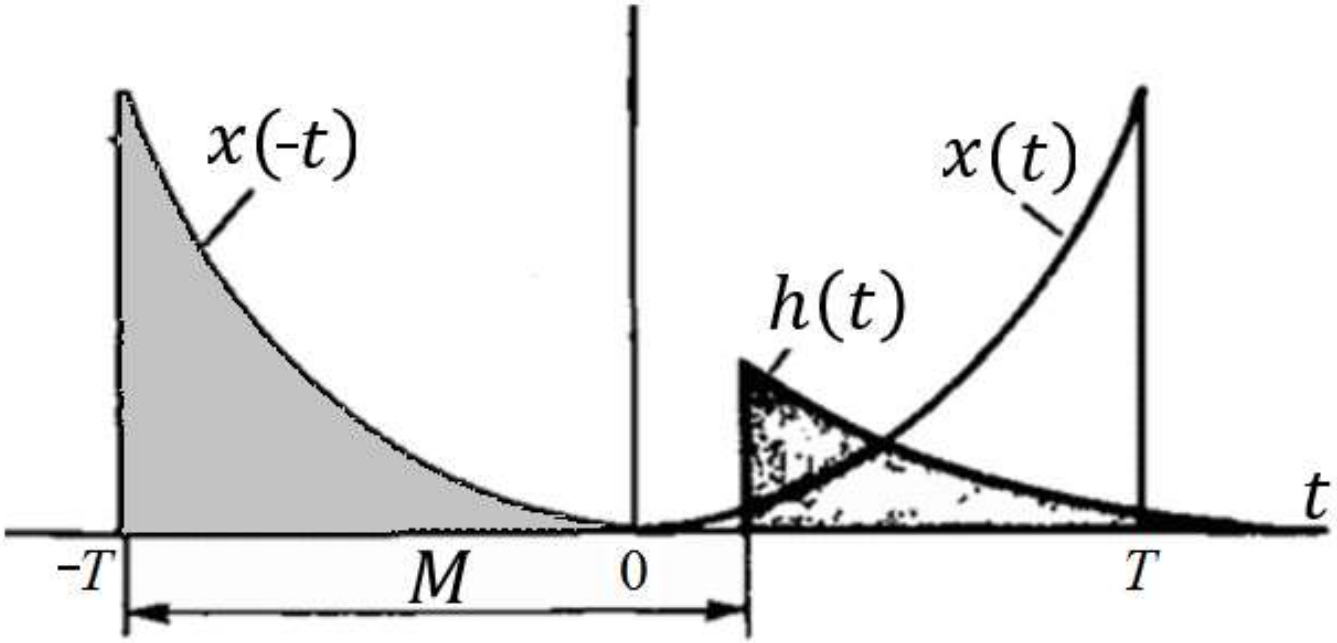


идеальный широкополосный фазовращатель



Согласованный фильтр





$$y(i) = \sum_{j=0}^{N-1} h(j)x(i-j) \qquad y(M) \to max$$

$$\left|\sum_{j=0}^{N-1} h(j)x(M-j)\right| \le \sqrt{\sum_{j=0}^{N-1} h^2(j) \sum_{j=0}^{N-1} x^2(M-j)}$$

$$x(M-j) = kh(j) \quad x(t) = kh(M-t) \qquad M \ge T$$

Фильтрация нерекурсивная 1

- FIR direct form
  ushort fir (DATA *x, DATA *h, DATA *r, DATA *dbuffer,
  ushort nx, ushort nh)
- FIR direct form (DUAL−MAC)
  ushort fir2 (DATA *x, DATA *h, DATA *r, DATA *dbuffer,
  ushort nx, ushort nh)
- Symmetric FIR direct form
  ushort firs (DATA *x, DATA *h, DATA *r, DATA *dbuffer,
  ushort nx, ushort nh2)
- Complex FIR direct form
  ushort cfir (DATA *x, DATA *h, DATA *r, DATA *dbuffer,
  ushort nx, ushort nh)

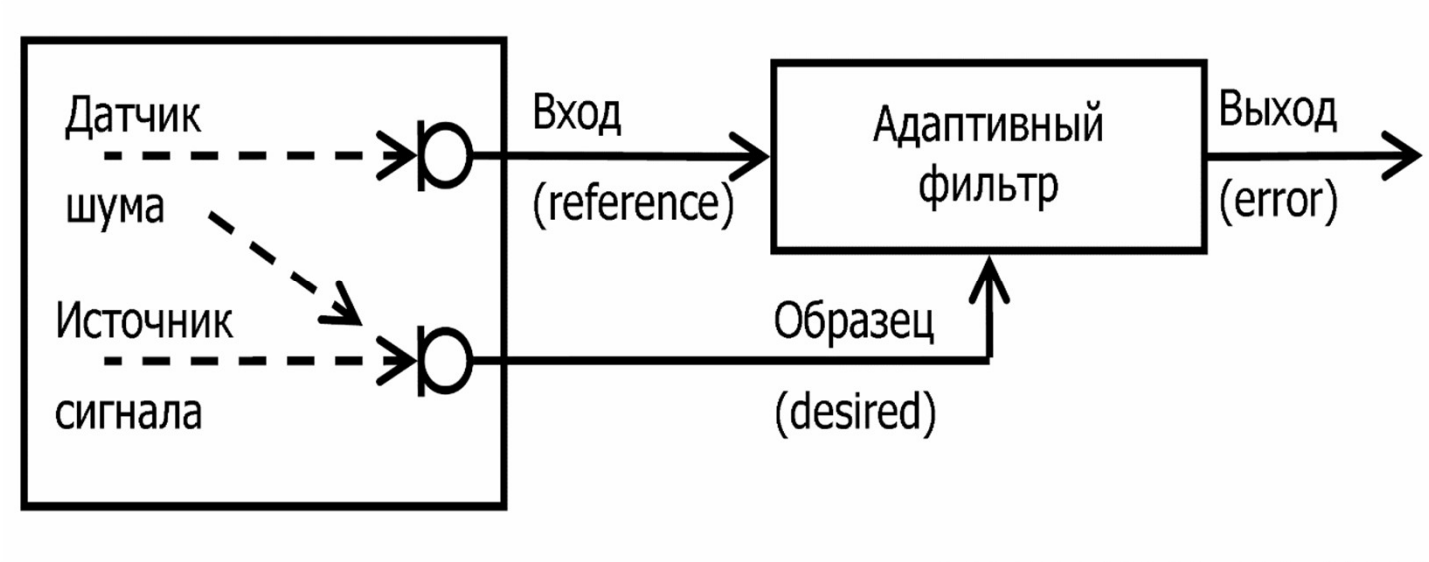
Фильтрация нерекурсивная 2

- Decimating FIR filter
  ushort firdec (DATA *x, DATA *h, DATA *r, DATA *dbuffer, ushort nh, ushort nx, ushort D)
- Interpolating FIR filter
  ushort firinterp (DATA *x, DATA *h, DATA *r, DATA *dbuffer, ushort nh, ushort nx, ushort I)
- FIR Hilbert Transformer (преобразование Гильберта)
  ushort hilb16 (DATA *x, DATA *h, DATA *r, DATA *dbuffer, ushort nx, ushort nh)
- Lattice forward FIR filter (решетчатый фильтр)
  ushort firlat (DATA *x, DATA *g, DATA *r, DATA *pbuffer, int nx, int nh)

Фильтрация рекурсивная

- IIR cascade direct 4-biquad form 2
  ushort iircas4 (DATA *x, DATA *h, DATA *r, DATA *dbuffer, ushort nbiq, ushort nx)
- IIR cascade direct 5-biquad form 2
  ushort iircas5 (DATA *x, DATA *h, DATA *r, DATA *dbuffer, ushort nbiq, ushort nx)
- IIR cascade direct 5-biquad form 1
  ushort iircas51 (DATA *x, DATA *h, DATA *r, DATA *dbuffer, ushort nbiq, ushort nx)
- Double-precision IIR filter
  ushort iir32 (DATA *x, LDATA *h, DATA *r, LDATA *dbuffer, ushort nbiq, ushort nr)
- Lattice inverse IIR filter
  ushort iirlat (DATA *x, DATA *h, DATA *r, DATA *pbuffer, int nx, int nh)

Адаптивный фильтр

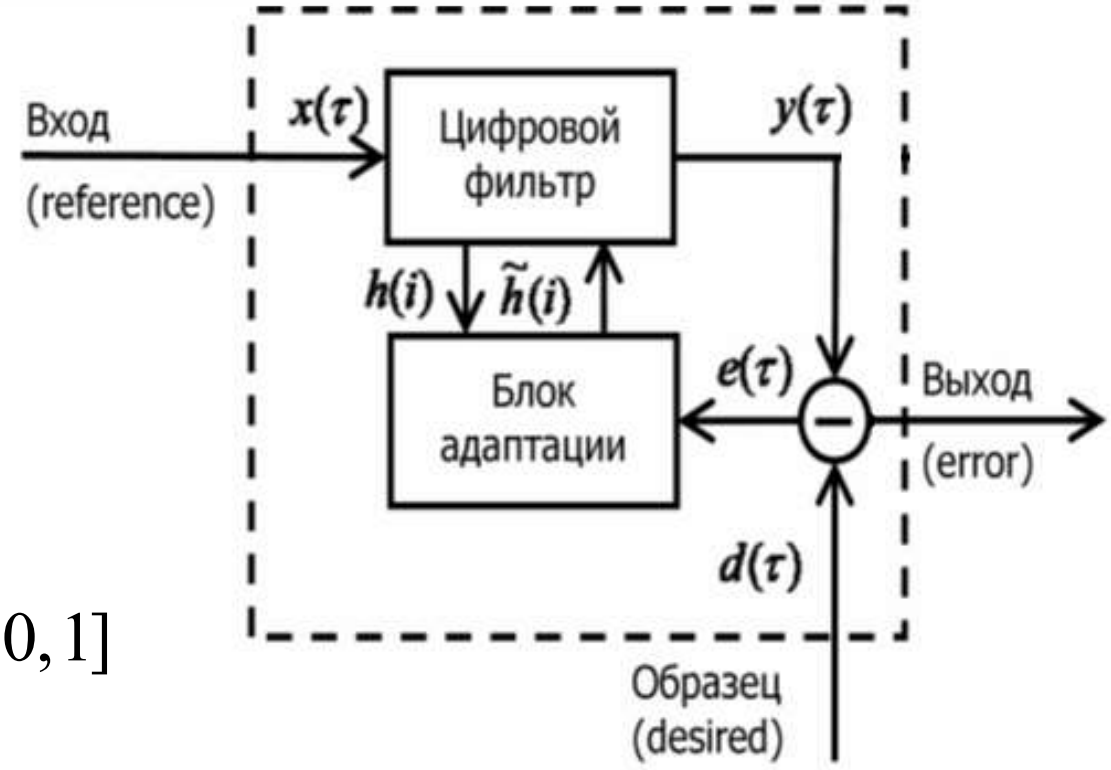


$$y(\tau) = \sum_{i=0}^{H-1} h(i) \times x(\tau - i)$$

$$e(\tau) = d(\tau) - y(\tau)$$

$$\widetilde{h}(i) = h(i) + \mu \times e(\tau) \times x(\tau - i)$$

$$i = \overline{0, H-1} \qquad \tau = \overline{0, N-1} \qquad \mu \in (0, 1]$$



Адаптивная фильтрация

- LMS FIR (delayed version)
  ushort dlms (DATA *x, DATA *h, DATA *r, DATA *des,
  DATA *dbuffer, DATA step, ushort nh, ushort nx)
- Adaptive delayed LMS filter (fast implemented)
  ushort dlmsfast (DATA *x, DATA *h, DATA *r, DATA *des,
  DATA *dbuffer, DATA step, ushort nh, ushort nx)

LMS – Least Mean Square (минимальная среднеквадратическая ошибка)

Свертка

$$y(\tau) = \sum_{i=0}^{H-1} h(i) \times x(\tau - i) \quad (\tau = \overline{0, N-1})$$

Циклическая свертка:

x[*N*+*H*-1] – входной вектор 1;

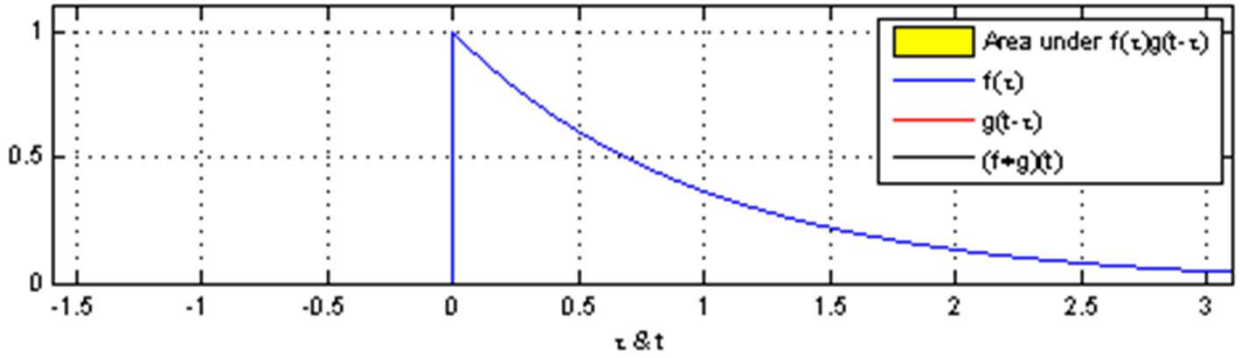
h[*H*] – входной вектор 2;

y[*N*] – выходной вектор.

Вычисляется без использования буфера задержки:

h[*i*] = 0 при *i* < 0 и *i* >= *H;*

x[*j*] = 0 при *j* < 0 и *j* >= *N* + *H* - 1.



Примечание. Линейная свертка реализуется как КИХ-фильтр.

Корреляция

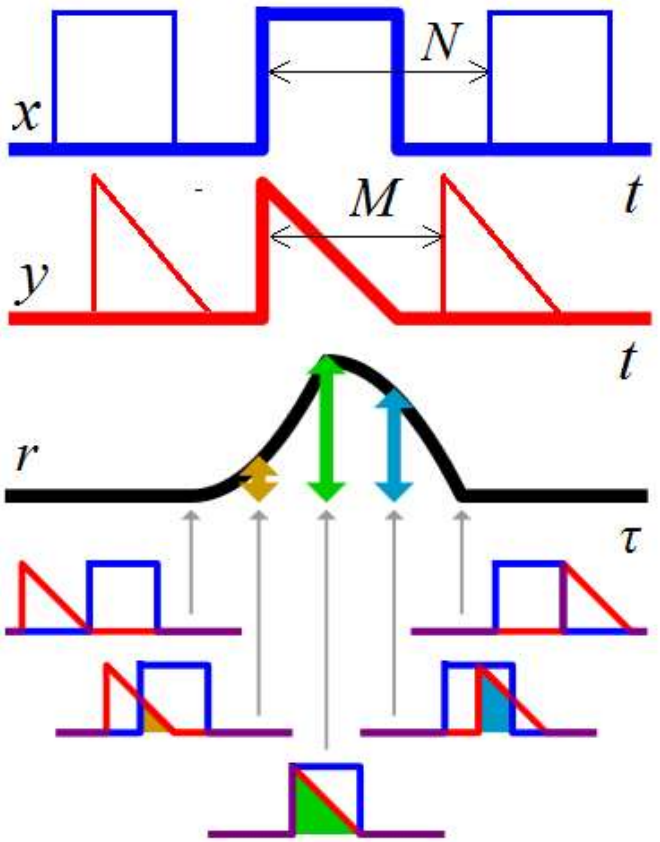
$$r(\tau) = \frac{1}{N} \sum_{k=0}^{K-\tau-1} x(\tau + k) \times y(k)$$

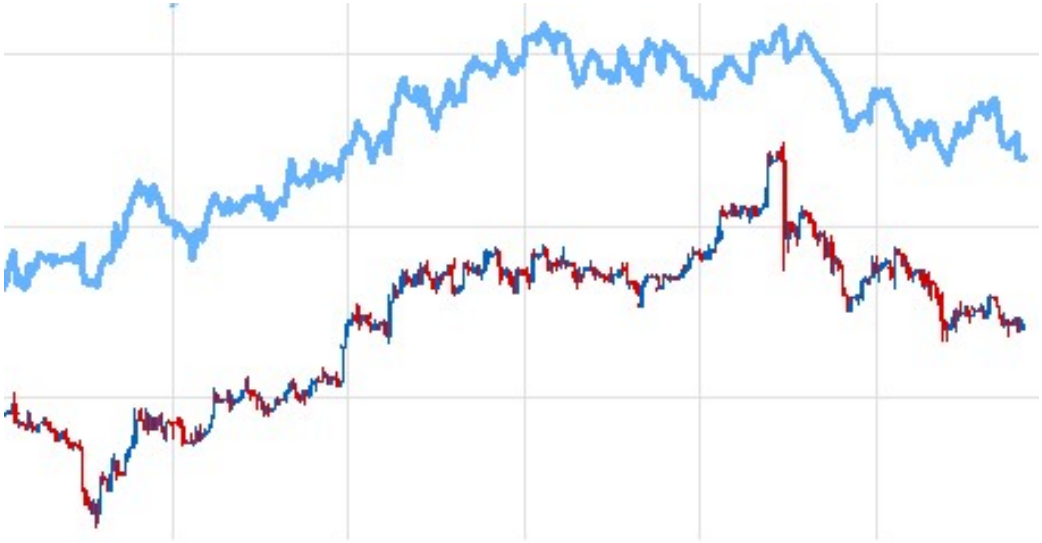
$$(\tau = \overline{0, N-1})$$

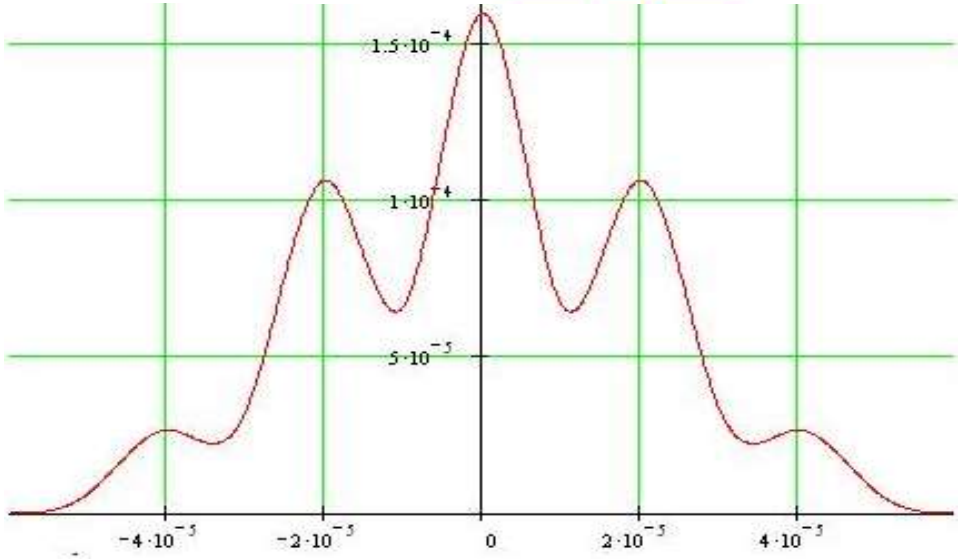
$$K = N + M - 1$$

$$x(i \pm N) = x(i) \quad (i = \overline{0, N-1})$$

$$y(j \pm M) = y(j) (j = \overline{0, M-1})$$







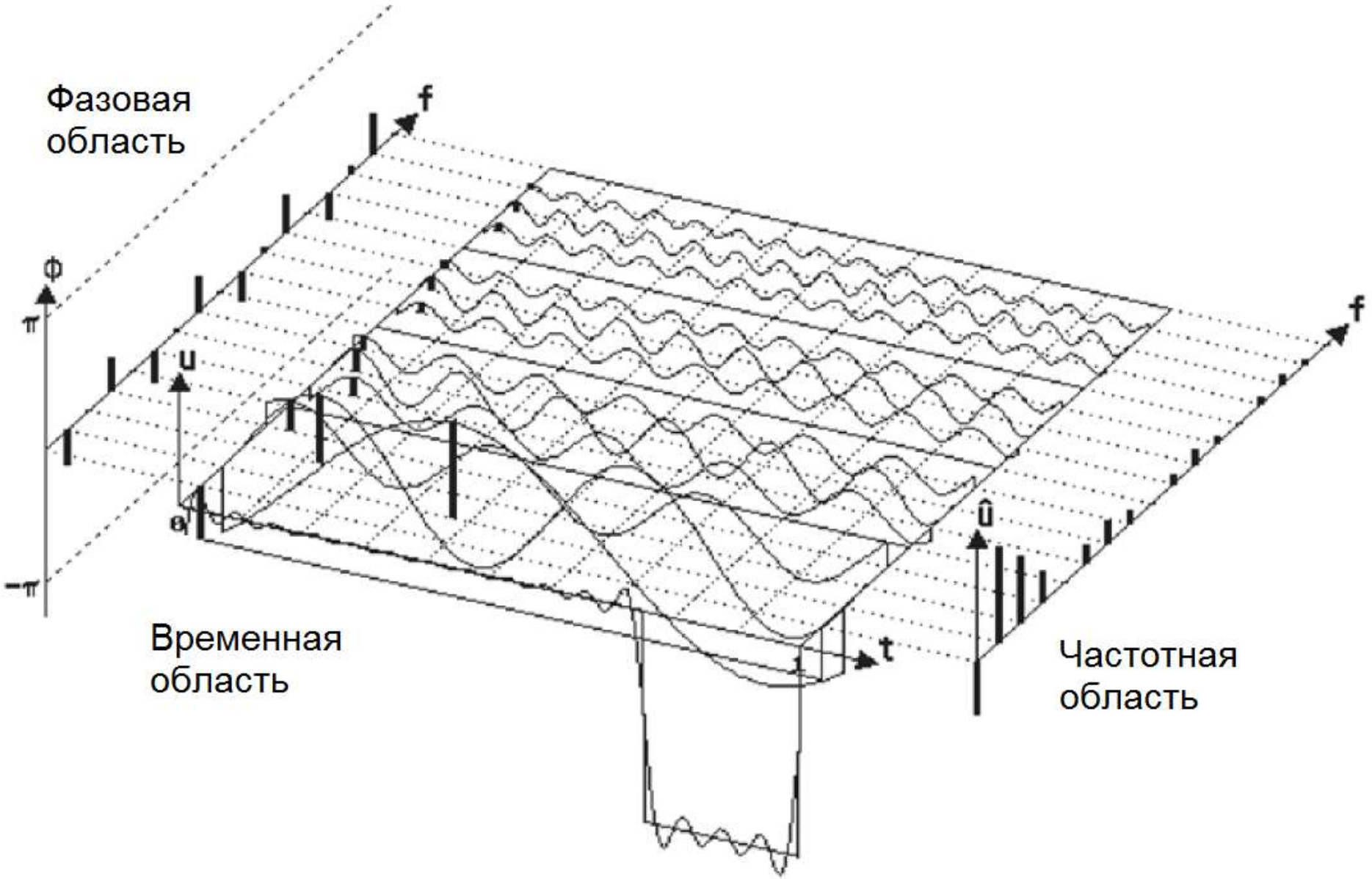
Свертка и корреляция

- Convolution
  ushort convol (DATA *x, DATA *h, DATA *r, ushort nr, ushort nh)
- Convolution (DUAL−MAC)
  ushort convol1 (DATA *x, DATA *h, DATA *r, ushort nr, ushort nh)
- Convolution (DUAL−MAC)
  ushort convol2 (DATA *x, DATA *h, DATA *r, ushort nr, ushort nh)
- Autocorrelation
  ushort acorr (DATA *x, DATA *r, ushort nx, ushort nr, type)
- Correlation
  ushort corr (DATA *x, DATA *y, DATA *r, ushort nx, ushort ny, type)

- Примечание. Параметр type задает нормировку.

Спектральная обработка

$$y[k] = \frac{1}{(scale\ factor)} * \sum_{i=0}^{nx-1} x[i] * \left(\cos\left(\frac{-2*\pi*i*k}{nx}\right) + j\sin\left(\frac{-2*\pi*i*k}{nx}\right)\right)$$



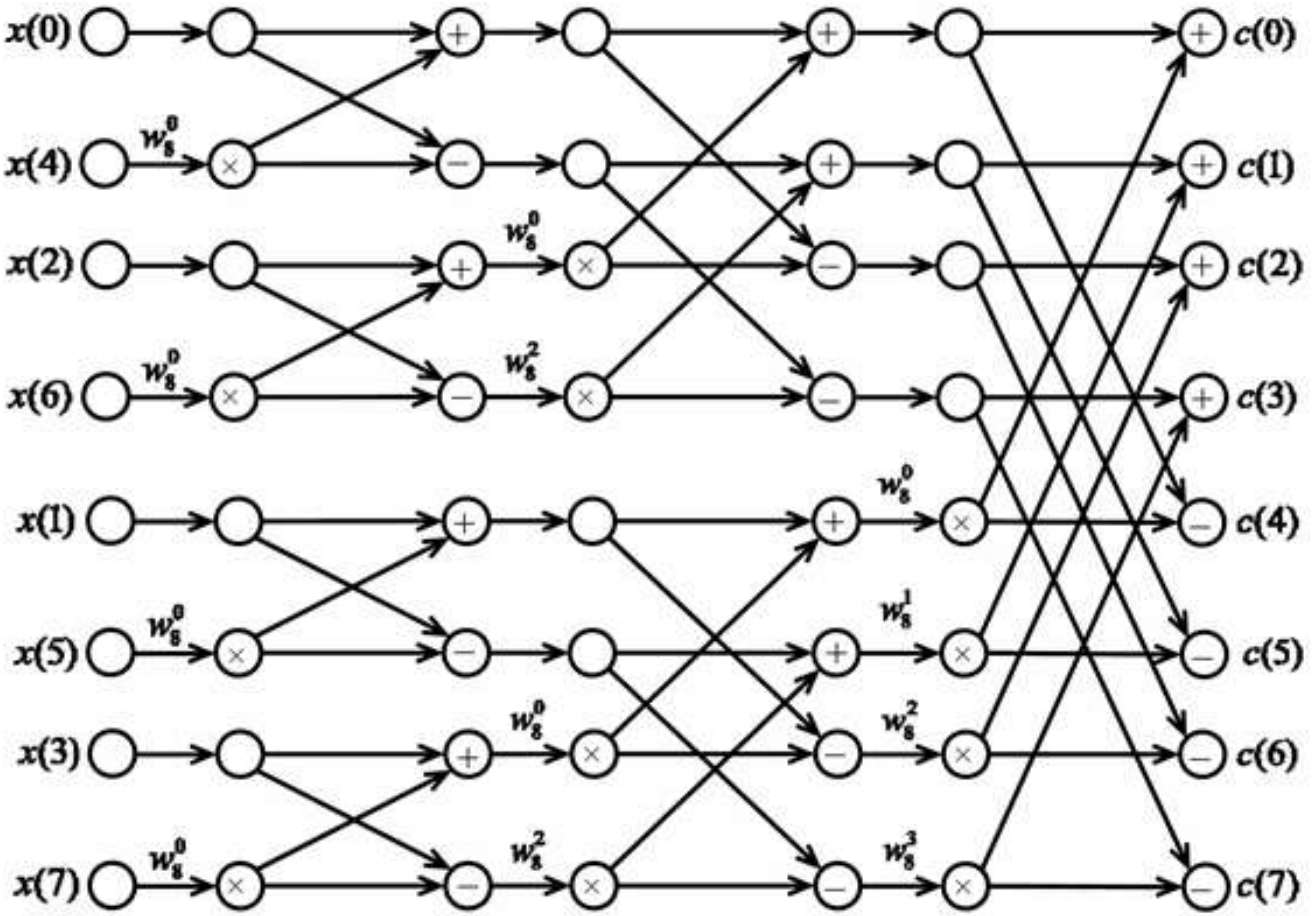
Преобразование Фурье

- 16-bit complex forward FFT
  void cfft (DATA *x, ushort nx, type t)
- 32-bit forward complex FFT
  void cfft32 (LDATA *x, ushort nx, type t)
- 16-bit complex inverse FFT
  void cifft (DATA *x, ushort nx, type t)
- 32-bit inverse complex FFT
  void cifft32 (LDATA *x, ushort nx, type t)
- 16-bit real forward FFT
  void rfft (DATA *x, ushort nx, type t)
- 16-bit real inverse FFT
  void rifft (DATA *x, ushort nx, type t)
- 32-bit real forward FFT
  void rfft32 (LDATA *x, ushort nx, type t)
- 32-bit real inverse FFT
  void rifft32 (LDATA *x, ushort nx, type t)

type:
SCALE = nx,
NOSCALE = 1.

Бит-реверсивная перестановка

$$c(i) = \sum_{\tau=0}^{N-1} x(\tau) \times \omega_N^i(\tau), \quad \omega_N^i = \exp(-2\pi j i \tau / N), \quad w_N^i = \exp(-2\pi j i / N).$$



$c(*) - x(*)$
000 – 000 (0)
001 – 100 (4)
010 – 010 (2)
011 – 110 (6)
100 – 001 (1)
101 – 101 (5)
110 – 011 (3)
111 – 111 (7)

*(ARx -TxB)

*(ARx+TxB)

AR0= 01100000
T0 = 00000100

01100000 (+0)
+ 00000100
= 01100100 (+4)
+ 00000100
= 01100010 (+2)
+ 00000100
= 01100110 (+6)
+ 00000100
= 01100001 (+1)
+ 00000100
= 01100101 (+5)
+ 00000100
= 01100011 (+3)
+ 00000100
= 01100111 (+7)

Примечание. Реализована путем сложения (вычитания) с обратным распространением переноса (заема).

Вспомогательные функции

- 16-bit Complex bit-reverse function
  void cbrev (DATA *x, DATA *r, ushort n)
- 32-bit complex bit reverse
  void cbrev32 (LDATA *a, LDATA *r, ushort)
- Floating-point to Q.15 conversion
  ushort fltoq15 (float *x, DATA *r, ushort nx)
- Q.15 to floating-point conversion
  ushort q15tofl (DATA *x, float *r, ushort nx)
- Random number generation
  ushort rand16 (DATA *r, ushort nr)
- Random number generation initialization
  void rand16init(void)

Тригонометрические функции

- Four quadrant inverse tangent of a vector
  ushort atan2_16 (DATA *q, DATA *i, DATA *r, ushort nx)
- Arctan of a vector
  ushort atan16 (DATA *x, DATA *r, ushort nx)
- Sine of a vector
  ushort sine (DATA *x, DATA *r, ushort nx)

Векторные функции 1

- Optimized vector addition
  ushort add (DATA *x, DATA *y, DATA *r, ushort nx, ushort scale)
- Exponent of all values in a vector
  short bexp (DATA *x, ushort nx)
- Exponent of a vector
  ushort expn (DATA *x, DATA *r, ushort nx)
- Log base 2 of a vector
  ushort log_2 (DATA *x, LDATA *r, ushort nx)
- Log base 10 of a vector
  ushort log_10 (DATA *x, LDATA *r, ushort nx)
- Natural log of a vector
  ushort logn (DATA *x, LDATA *r, ushort nx)
- 32-bit by 16-bit long division
  void ldiv16 (LDATA *x, DATA *y, DATA *r, DATA *rexp, ushort nx)

Векторные функции 2

- Index for maximum magnitude in a vector
  short maxidx (DATA *x, ushort ng, ushort size)
- Index of the maximum element of a vector ≤ 34
  short maxidx34 (DATA *x, ushort nx)
- Maximum magnitude in a vector
  short maxval (DATA *x, ushort nx)
- Index and value of the maximum element of a vector
  void maxvec (DATA *x, ushort nx, DATA *val, DATA *idx)
- Index for minimum magnitude in a vector
  short minidx (DATA *x, ushort nx)
- Minimum element in a vector
  short minval (DATA *x, ushort nx)
- Index and value of the minimum element of a vector
  void minvec (DATA *x, ushort nx, DATA *val, DATA *idx)
- 32-bit vector multiply
  ushort mul32 (LDATA *x, LDATA *y, LDATA *r, ushort nx)

Векторные функции 3

- 16-bit vector negate
  short neg (DATA *x, DATA *r, ushort nx)
- 32-bit vector negate
  short neg32 (LDATA *x, LDATA *r, ushort nx)
- Sum of squares of a vector (power)
  short power (DATA *x, LDATA *r, ushort nx)
- Vector reciprocal (обратные значения)
  void recip16 (DATA *x, DATA *r, DATA *rexp, ushort nx)
- Square root of a vector (квадратные корни)
  ushort sqrt_16 (DATA *x, DATA *r, short nx)
- Vector subtraction (поэлементная разность)
  short sub (DATA *x, DATA *y, DATA *r, ushort nx, ushort scale)

Матричные функции

- Matrix multiply
  ushort mmul (DATA *x1, short row1, short col1, DATA *x2, short row2, short col2, DATA *r)
- Matrix transponse
  ushort mtrans (DATA *x, short row, short col, DATA *r)