ЛЕКЦИИ 10-11. SIMULINK-2.
Z-ПРЕОБРАЗОВАНИЕ И ДИСКРЕТНОЕ ИНТЕГРИРОВАНИЕ

DiscreteTime Integrator

Составитель:
д.т.н. Колесникова С.И.
skolesnikova@yandex.ru

На предыдущей лекции: ПРИМЕРЫ оригиналов-изображений

Общая формулы преобразования Лапласа:

$$X(s)=\frac{1}{2\pi}\int_{-\infty}^{\infty}x(t)e^{-st}dt,\quad x(t)=\frac{1}{2\pi}\int_{-\infty}^{\infty}X(j\omega)e^{-j\omega t}d\omega, \qquad (*)$$

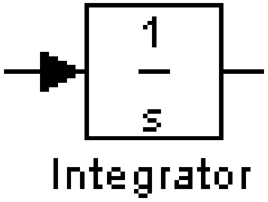
где x(t) - оригинал, X(jw) - изображение при s = jw, j - мнимая единица, w - частота.

имеются **специальные таблицы**, в которые сведены наиболее часто встречающиеся функции X(s) и их оригиналы x(t).

В частности:

Оригинал x(t)	Изображение X(s)
$\boldsymbol{a \cdot x(t)}$	$\boldsymbol{a \cdot X(s)}$
$\boldsymbol{x(t-a)}$	$X(s)e^{-as}$
$\dfrac{d^n y}{dt^n}$	$s^n X(s)$
$\int_0^t x(\tau)d\tau$	$\dfrac{X(s)}{s}$

Обозначение интегратора в Simulink.



Основы z-преобразования (дискретный аналог пр. Лапласа)

Смысл z-преобразования заключается в том, что последовательности чисел $\{x(k)\}$ ставится в соответствие функция комплексной переменной z (образ или изображение $x(k)$):

$$X(z) \doteqdot x(k) \qquad X(z) = \sum_{k=-\infty}^{\infty} x(k) z^{-k} \qquad (1)$$

Функция X(z) определена только для тех значений z, при которых ряд (1) сходится.

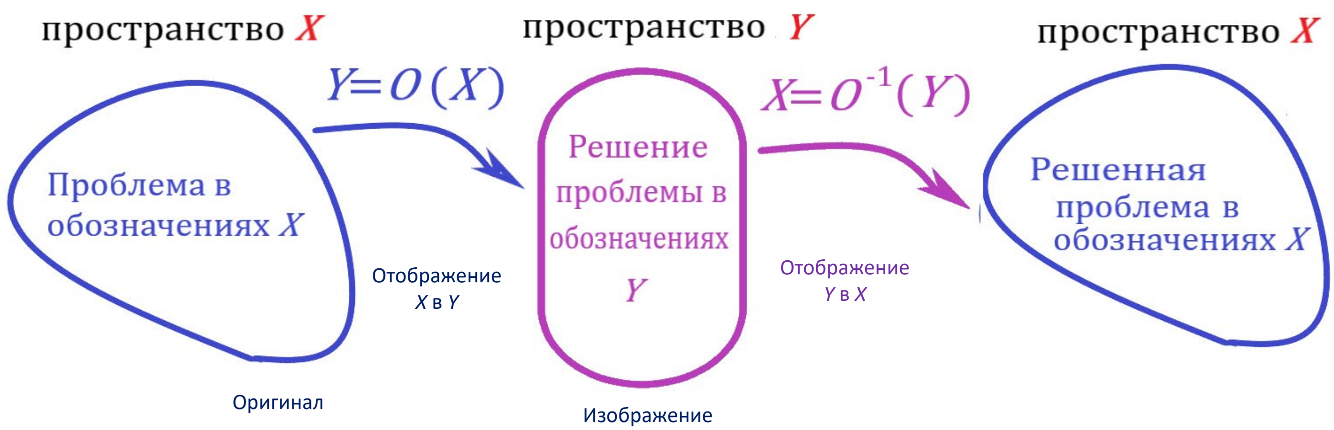
Область z, в которой z-преобразование сходится и значения X(z) конечны, называют **областью сходимости**.

<u>Z-преобразование</u> играет для дискретных сигналов и систем такую же роль, как <u>преобразование Лапласа</u> — для аналоговых (непрерывных).

z-преобразование импульсной характеристики дискретной системы является дробно-рациональной функцией переменной Z.

$$F(z) \doteqdot f(n) \quad \Leftrightarrow \quad f(n) = \frac{1}{2\pi i} \oint_C F(z) \cdot z^{n-1} dz, \; n = 0, 1, \ldots,$$

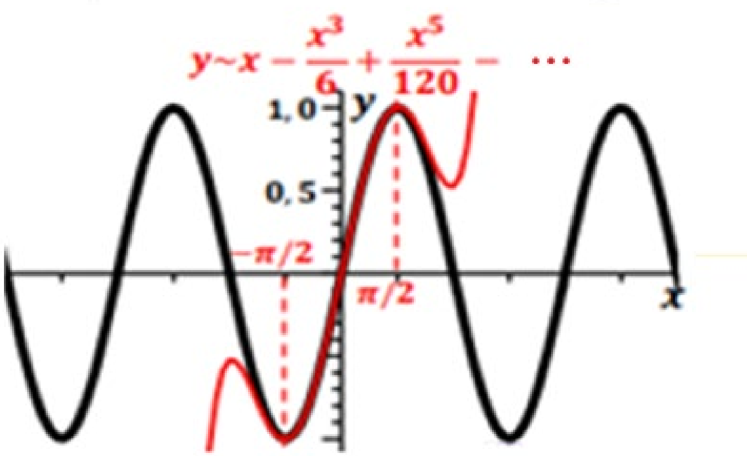
Смысл ВСЕХ преобразований



Пример 1 из матанализа. Ряд Маклорена (Тейлора) – есть представление разных «неудобных» функций в виде «удобной» суммы степенных функций.

Пример 2 из матанализа. Замена координат при взятии интеграла: площадь криволинейной фигуры в декартовой системе есть площадь прямоугольника в полярных координатах.

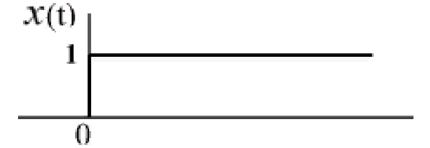
$$y(x)=\sin(x)$$



$$y(x) \sim x \quad \text{при } x \sim 0 \text{ (б.м.)}$$

ПРИМЕРЫ z-преобразования для некоторых функций

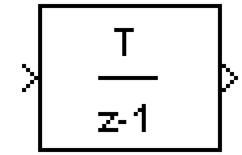
***x*(*k*)=0 при *k*<0,**
***x*(*k*)=1 при *k*≥0. Функция Хевисайда (единичный скачок)**



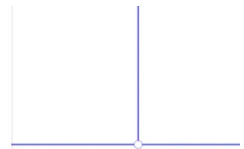
$$X(z) = z^{-0} + z^{-1} + z^{-2} + z^{-3} + \ldots = \frac{1}{1 - z^{-1}}$$

$$1 \risingdotseq \frac{z}{z-1}$$

$|z| < 1$ -- условие сходимости



Единичная импульсная функция



Единичная импульсная функция является дискретным аналогом дельта-функции и представляет собой одиночный отсчет с единичным значением:

$$x_0(k) = \begin{cases} 1, k = 0, \\ 0, k \neq 0. \end{cases}$$

Расчет его z-преобразования :

$$X_0(z) = \sum_{k=-\infty}^{\infty} x_0(k) z^{-k} = 1 \cdot z^{-0} = 1$$

Описывает **точечное воздействие**, пространственную плотность физических величин (масса, заряд, интенсивность источника тепла, сила и т. п.), **сосредоточенных или приложенных в одной точке**.

Дискретный аналог дельта-функции

- $\delta(x) = \begin{cases} +\infty, & x = 0 \\ 0, & x \neq 0 \end{cases}$
- $\int_{-\infty}^{+\infty} \delta(x)\, dx = 1.$

ПРИМЕРЫ z-преобразования для некоторых функций

- Единичный импульс $\{x_k\}=1,0,0,0\ldots$ $F(z)=1$
- Единичный скачок

$$\{x_k\}=\begin{cases}1, k\ge 0\\ 0, k<0\end{cases} \qquad F(z)=\sum_{k=0}^{\infty} z^{-k}=\frac{z}{z-1}$$

- Прямоугольный импульс

$$\{x_k\}=1,1,1,0,0\ldots F(z)=1+\frac{1}{z}+\frac{1}{z^2}=\frac{z^2+z+1}{z^2}$$

Экспоненциальная функция

$x(k)=0$ при $k<0, x(k)=a^k$ при $k\ge 0$

$$X(z)=\sum_{k=-\infty}^{\infty} x(k)z^k=\sum_{k=0}^{\infty} a^k z^k=\sum_{k=0}^{\infty}(az)^k,$$

Как и в предыдущем случае, ряд сходится при $|az|<1$, при этом:

$$X(z)=\frac{1}{1-az}, |z|<\frac{1}{a}$$

При использовании символики z^{-1}:

$$X(z)=\frac{z}{z-a}, |z|>a$$

Дискретный сигнал $x[n]$, $n\ge 0$	z-преобразование $X(z)=\sum_{n=0}^{\infty} x[n]z^{-n}$
$x[n]=\delta[n]=\begin{cases}1, & n=0\\ 0, & n\ne 0\end{cases}$	$X(z)=1$
$x[n]=u[n-N]=\begin{cases}1, & n\ge N\ge 0\\ 0, & n<N\end{cases}$	$X(z)=\dfrac{z^{-N}}{1-z^{-1}}$
$x[n]=a^n$	$X(z)=\dfrac{1}{1-az^{-1}}$
$x[n]=A$	$X(z)=\dfrac{1}{1-z^{-1}}$
$x[n]=n$	$X(z)=\dfrac{z^{-1}}{(1-z^{-1})^2}$
$x[n]=na^n$	$X(z)=\dfrac{az^{-1}}{(1-az^{-1})^2}$
$x[n]=a^n\cos n\theta$	$X(z)=\dfrac{1-a\cos\theta z^{-1}}{1-2a\cos\theta z^{-1}+a^2z^{-2}}$
$x[n]=a^n\sin n\theta$	$X(z)=\dfrac{a\sin\theta z^{-1}}{1-2a\cos\theta z^{-1}+a^2z^{-2}}$
$x[n]=\begin{cases}e^{jn2\pi fT}, & n\ge 0\\ 0, & n<0\end{cases}$	$X(z)=\dfrac{1}{1-e^{j2\pi fT}\cdot z^{-1}}$

Свойства z-преобразования дискретных функций

1. Свойство линейности

Пусть $f_i(n) \risingdotseq F_i(z)$, $i = 1, 2, \ldots, k$; $c_i = \text{const}$, $i = 1, 2, \ldots, k$.

Тогда $\sum_{i=1}^{k} c_i f_i(n) \risingdotseq \sum_{i=1}^{k} c_i F_i(z)$.

$$s(k) = a \cdot x(k) + b \cdot y(k) \Leftrightarrow S(z) = aX(z) + bY(z).$$

2. Свойство запаздывания (смещения)

Пусть $f(n) \risingdotseq F(z)$, тогда

$$f(n-1) \risingdotseq z^{-1}F(z),$$
$$f(n-2) \risingdotseq z^{-2}F(z),$$
$$\ldots$$
$$f(n-k) \risingdotseq z^{-k}F(z),\ k = 1, 2, \ldots,$$

где $f(n-k) = 0$ при $n - k < 0$.

$$y(k) = x(k-n) \Leftrightarrow Y(z) = z^{n}X(z)$$

Свойства z-преобразования дискретных функций

Свойство дифференцирования изображения

Если $f(n) \risingdotseq F(z)$, то $\quad nf(n) \risingdotseq -z\dfrac{dF(z)}{dz}$.

Удобные свойства Z-преобразования при выполнении операций над сигналами:

сдвиг сигналов,

умножение изображений,

свертка оригиналов,

теорема о предельных значениях…

Пример 5. Найдем Z-преобразование для восьми отсчетов $x = \{x[0], x[1],\ldots, x[7]\}$.

$$x[k]=\{1,2,0,-1,-2,-1,0,0\},\ k=0,1,\ldots,7$$

$$F(z)=1z^0+2z^1+0z^2-1z^3-2z^4-1z^5+0z^6+0z^7=1+2z-z^3-2z^4-z^5$$

Лабораторная работа 4. Моделирование динамических систем в MatLab Simulink.

$$\dot{x}_1(t) = -x_1(t) - 2x_2(t) + 14,$$
$$\dot{x}_2(t) = -2x_1(t) + x_2(t) + 3.$$

Дискретизация системы ДУ согласно прямому методу Эйлера

Шаг интегрирования

$$x_1(k+1) = x_1(k) + T\left(-x_1(t) - 2x_2(t) + 14\right),$$
$$x_2(k+1) = x_2(k) + T\left(-2x_1(t) + x_2(t) + 3\right),$$
$$T \sim \Delta t > 0.$$

Метод Эйлера. Разностная схема

Метод Эйлера состоит в приближенной замене производной в уравнении $y' = f(x, y)$ ее разностным аналогом. Тогда дифференциальное уравнение можно записать в виде:

$$\frac{y_{i+1} - y_i}{h} = f(x_i, y_i) \qquad \frac{y_{i+1} - y_i}{x_{i+1} - x_i} = f(x_i, y_i)$$

Эта система уравнений называется **разностной схемой Эйлера**

Расчетная формула метода Эйлера имеет следующий вид:

$$\boxed{\begin{aligned} & y_{i+1} = y_i + h \cdot f(x_i, y_i) \\ & x_{i+1} = x_i + h, \qquad (i = 0, 1, 2, \ldots, n-1) \end{aligned}}$$

Результатом численного решения задачи Коши будет таблица значений:

x	x_0	x_1	x_2	...	x_n
y	y_0	y_1	y_2	...	y_n

Связь прямого метода Эйлера и ряда Тейлора

Разложим $y(x_1)$ по формуле Тейлора в окрестности точки x_0:

$$y(x_1) = y(x_0) + y'(x_0)h + y''(x_0)\frac{h^2}{2} + \cdots$$

Взяв в разложении несколько слагаемых и отбросив остальные, получаем тот или иной метод, при этом производные точного решения можно выразить в силу исходного уравнения. Например, простейший метод на одном шаге можно записать так:

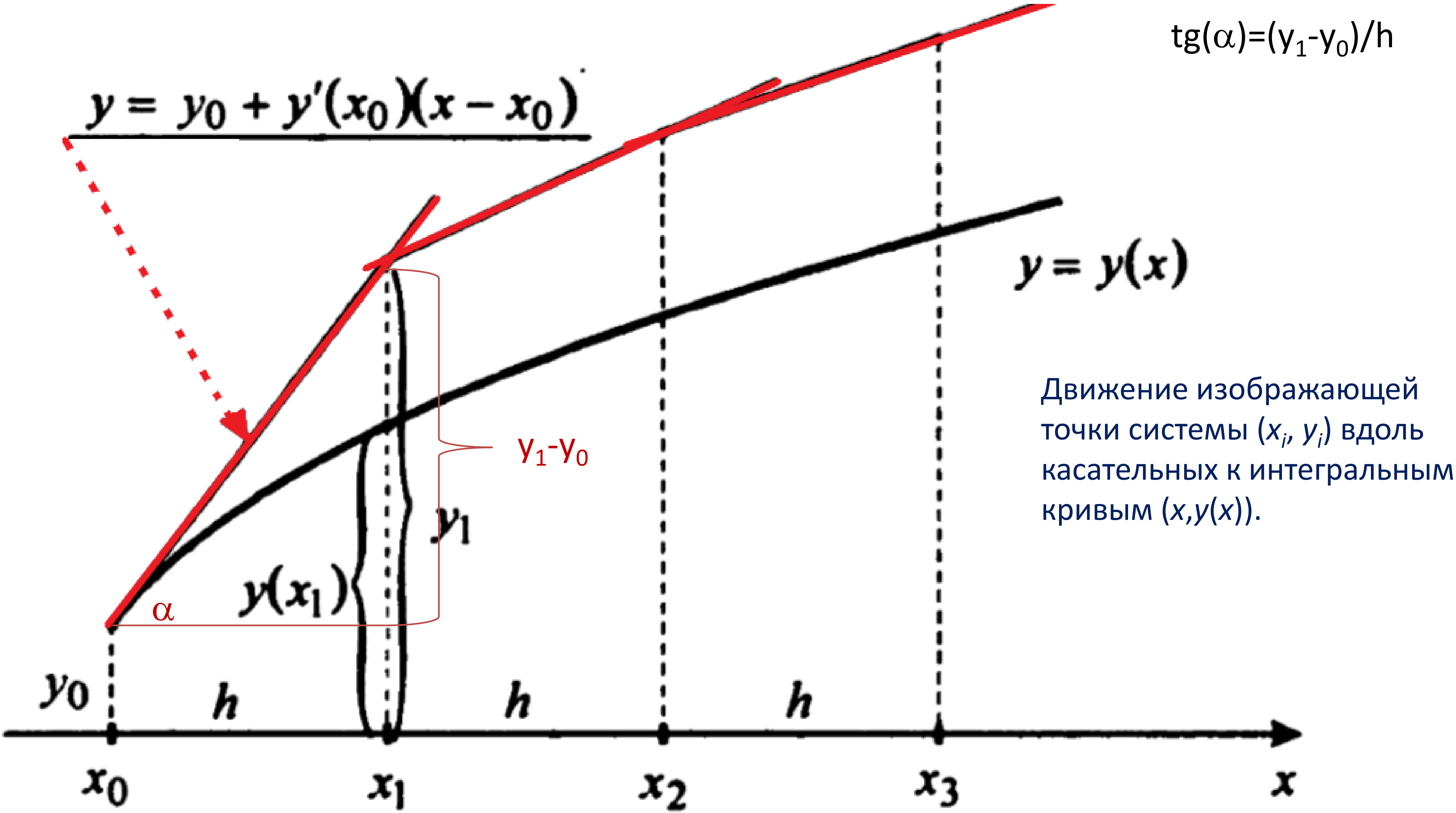
$$y_1 = y_0 + hf(x_0, y_0).$$

Аналогично можно записать алгоритм при переходе к очередному узлу:

$$y_{n+1} = y_n + hf(x_n, y_n), \; n = 0, 1, \ldots, N-1.$$

Этот метод называется методом Тейлора первого порядка, чаще его называют *методом Эйлера*. Геометрическая интерпретация метода Эйлера представляет собой движение по касательной к интегральным кривым .

Метод Эйлера. Разностная схема



Движение изображающей точки системы (x_i, y_i) вдоль касательных к интегральным кривым (x,y(x)).

Если в разностную схему подставить точное решение исходной задачи, то возникнет невязка (точное решение не удовлетворяет разностной схеме). Эта невязка называется погрешностью аппроксимации разностной схемы:

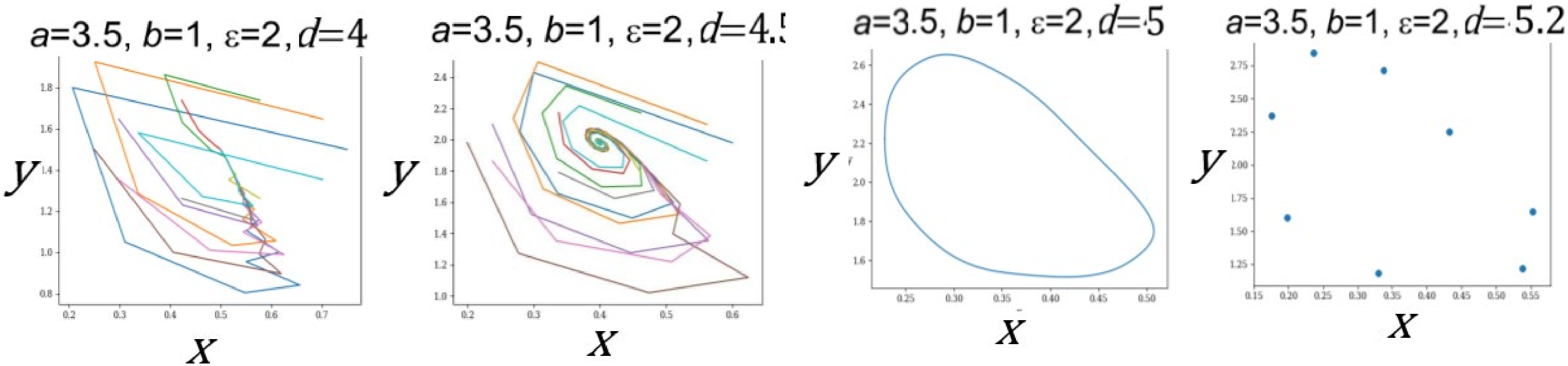
$$\frac{y_{i+1} - y_i}{h} = f(x_i, y_i)$$

$$\psi_n = y(x_{n+1}) - y(x_n) - hf(x_n, y(x_n)).$$

Пример дискретной формы системы: ХИЩНИК-ЖЕРТВА

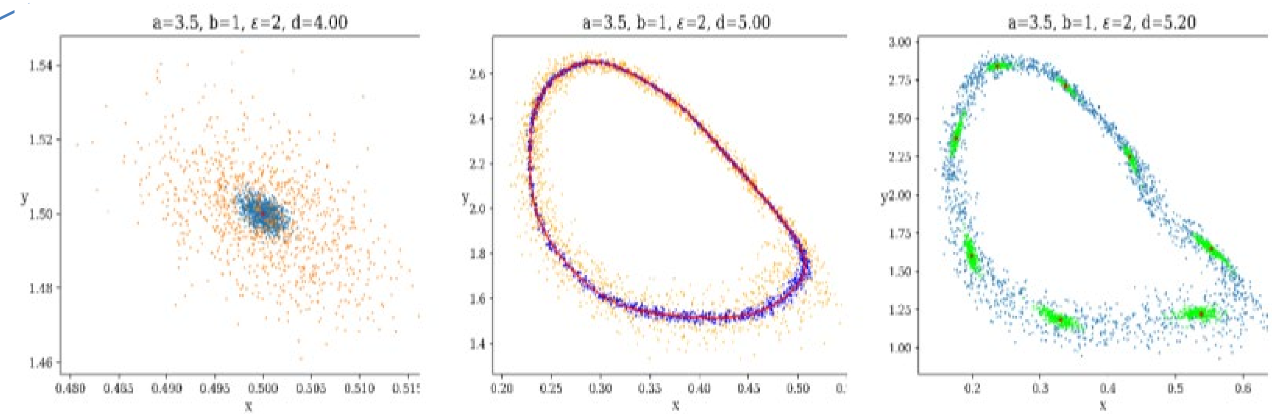
$$\begin{cases} x_{t+1} = ax_t(1 - x_t) - \dfrac{bx_t y_t}{1 + \varepsilon x_t} \\ y_{t+1} = \dfrac{dx_t y_t}{1 + \varepsilon x_t} \end{cases}$$

Здесь x_t, y_t – численность жертв и хищников, параметр a – коэффициент демографического роста,
b, d – параметры, определяющие взаимное влияние популяций друг на друга,
ε – коэффициент, степень насыщения.
Все параметры системы – положительные.

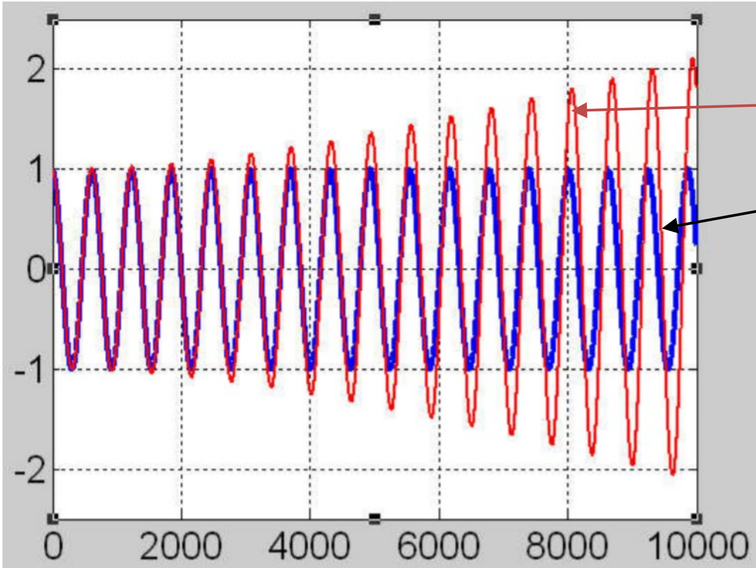


шум

$$\begin{cases} x_{t+1} = ax_t(1 - x_t) - \dfrac{bx_t y_t}{1 + \varepsilon x_t} + \delta\xi_{1,t}, \\ y_{t+1} = \dfrac{dx_t y_t}{1 + \varepsilon x_t} + \delta\xi_{2,t}. \end{cases}$$



ОПАСНОСТЬ дискретизации для ЧИСЛЕННОГО РЕШЕНИЯ СИСТЕМ ДУ

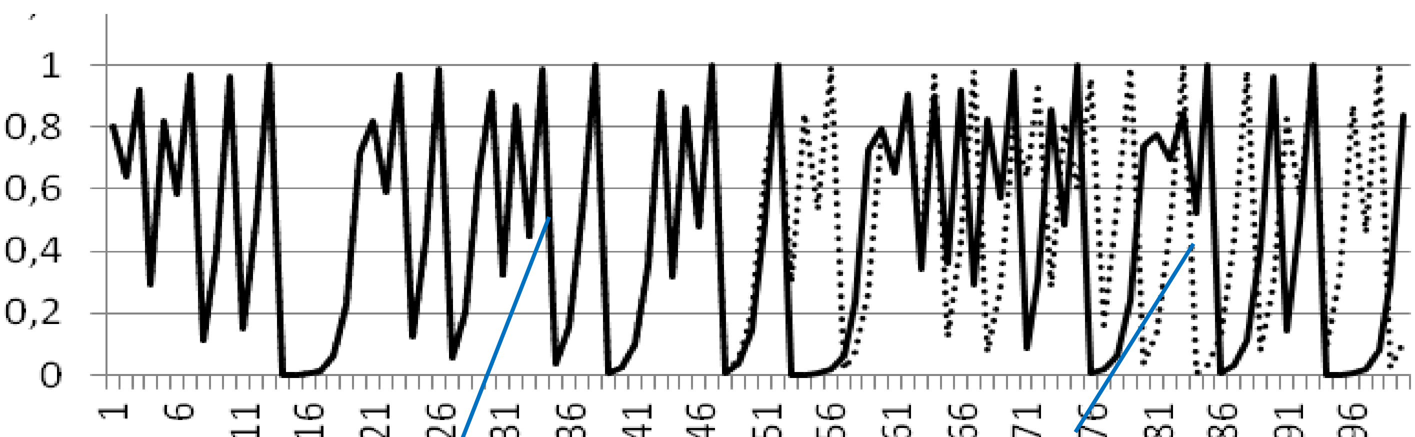


Сравнение аналитического (красная линия) и численного (фиолетовая линия) решений уравнения Дуффинга

$$\ddot{x} + x + \varepsilon x^3 = 0, x = x(t),$$

$$x(0) = x_0, \dot{x}(0) = x_{00}$$

Существенная зависимость от начальных данных – основная черта, присущая хаотической динамике.



Представление числа с плавающей запятой для хранения действительных чисел в битовой строке с некоторой конечной точностью приводит к усилению влияния ошибки округления на каждой итерации в силу нелинейности модели

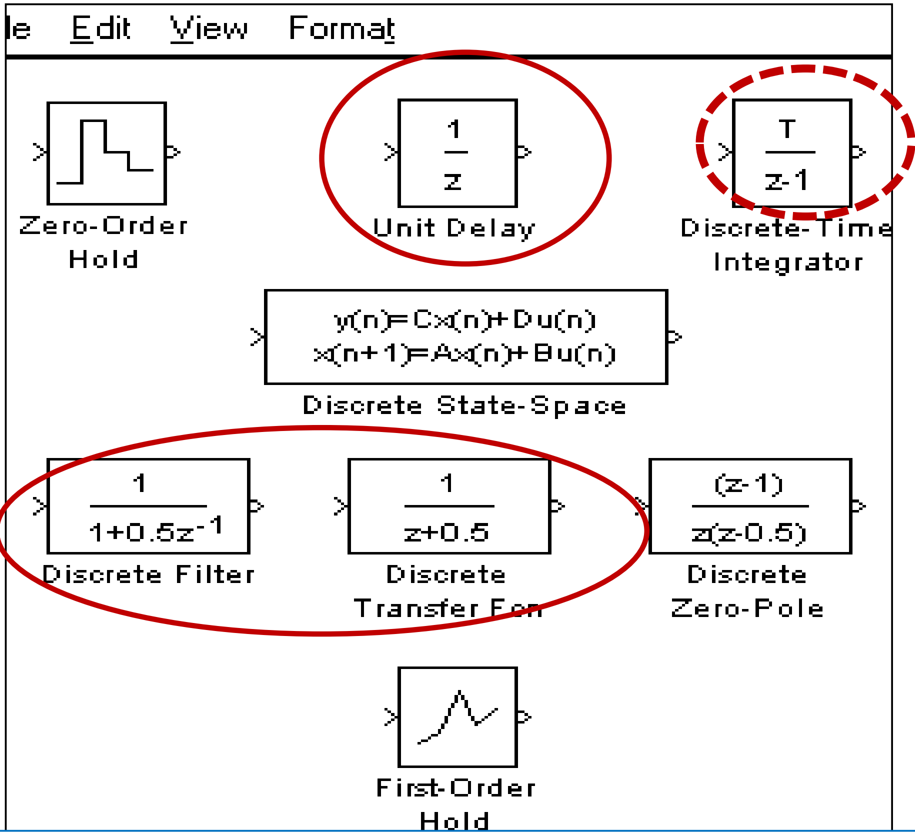
По аналитической формуле

$$x_n^{am} = \sin^2\left(2^n \arcsin\sqrt{x_0}\right).$$

По итерационной формуле

$$x_n^{it} = 4x_{n-1}^{it}(1 - x_{n-1}^{it}),\ n > 0$$

ОПИСАНИЕ БЛОКОВ. DISCRETE (дискретные блоки)



1. **Zero-Order Hold** - выполняет дискретизацию входного сигнала по времени.

2. Unit Delay - выполняет задержку входного сигнала на один шаг модельного времени.

3. Discrete-Time Integrator - операция интегрирования в дискретных системах.

4. **Discrete State-Space** - блок создает динамический объект:

$$x(n+1) = Ax(n) + Bu(n),$$
$$y(n) = Cx(n) + Du(n),$$

где x - вектор состояния, u - вектор входных воздействий, y - вектор выходных сигналов, A, B, C, D - матрицы: системы, входа, выхода и обхода, соответственно, n - номер шага моделирования.

5. **Discrete Filter** - блок дискретного фильтра задает дискретную передаточную функцию от обратного аргумента (1/z).

6. **Discrete Transfer Fcn** – передаточная функция для дискретных систем. Параметры блока:

Numerator - вектор или матрица коэффициентов числителя;

Denominator - вектор коэффициентов знаменателя;

7. **Discrete Zero-Pole** - определяет передаточную функцию с заданными полюсами и нулями.

8. **First-Order Hold** - блок задает линейное изменение выходного сигнала на каждом такте дискретизации, в соответствии с крутизной входного сигнала на предыдущем интервале дискретизации.

Блок дискретного интегратора времени Discrete Time Integrator

Блок «Дискретный интегратор» (Discrete-Time Integrator) интегрирует (накапливает) значения сигнала в дискретном времени.

Аналог блока «Интегратор» для создания полностью дискретной модели.

Принцип работы

Блок рассматривает сигнал как динамическую систему (ДС) с одним состоянием.

Динамика ДС задаётся уравнениями:

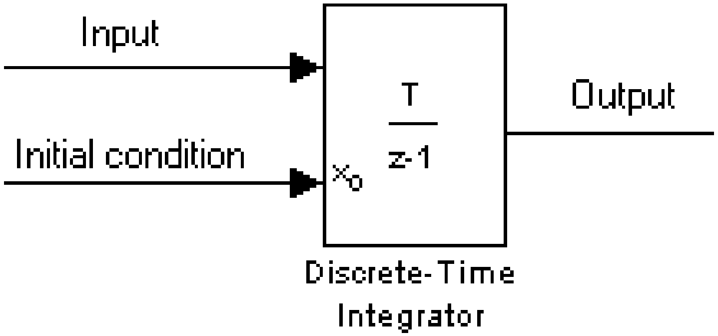
x(t) = u(t), x(t0) = x0,

y(t) = x(t), где: u — входной сигнал;

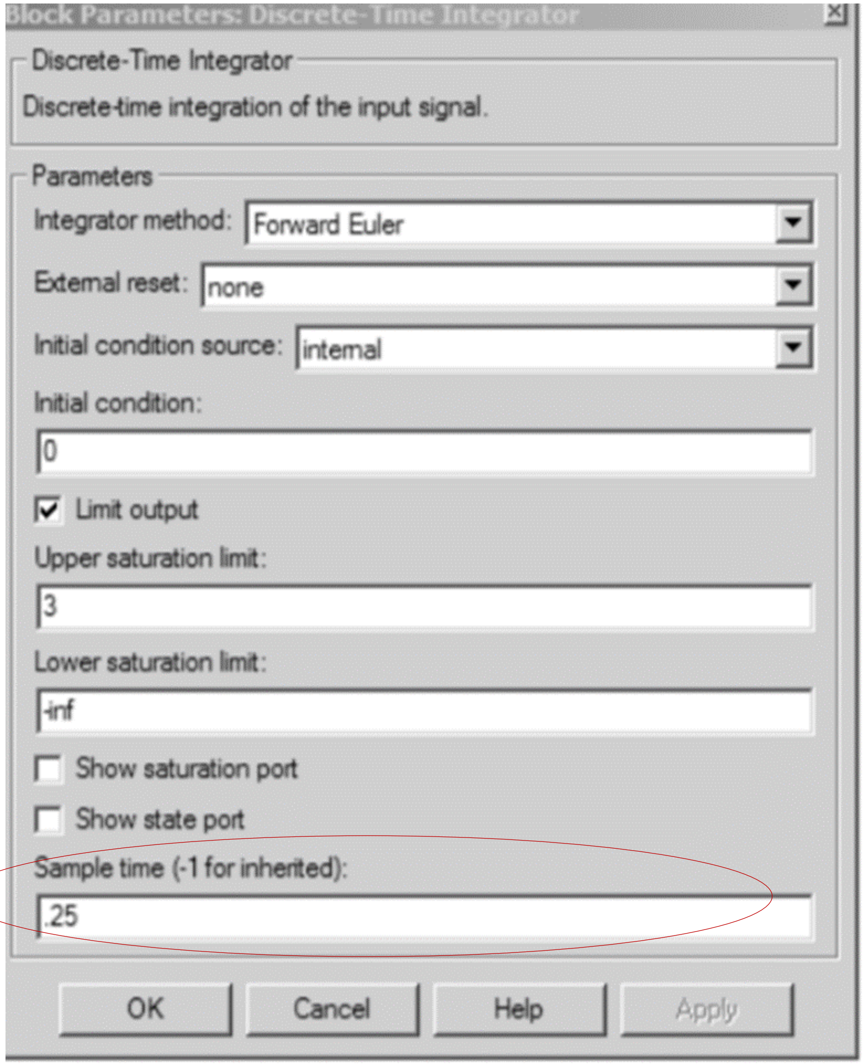
•y — выходной сигнал;

•x — состояние блока;

•x0 — начальное условие.



Блок дискретного интегратора времени Discrete Time Integrator



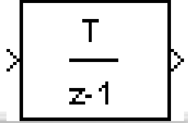
Sample time влияет на точность дискретизации, определяет шаг модельного времени и частоту отображения данных.

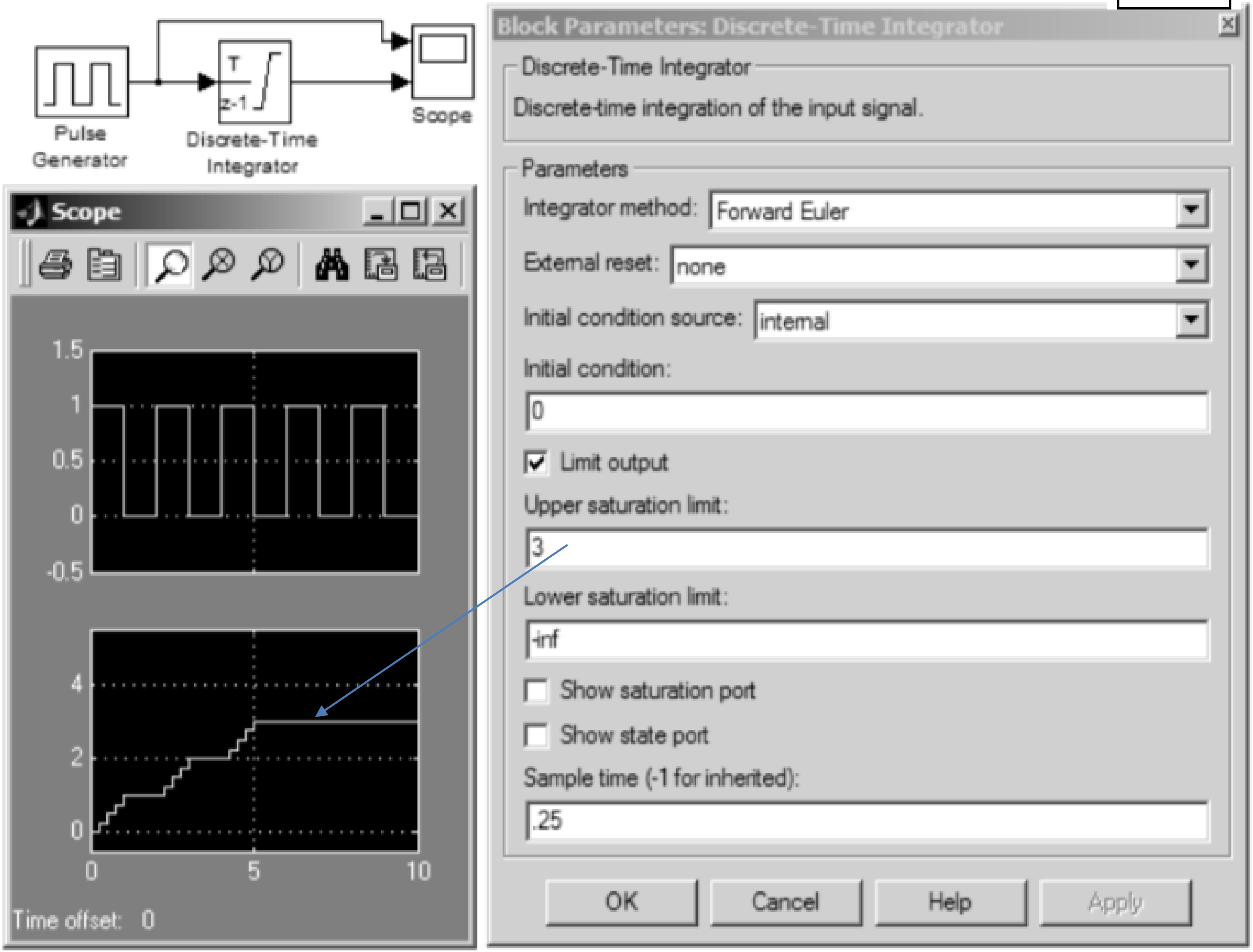
(Sample time = 0) - **моделирование непрерывных систем**, накапливается ошибка округления (на большом интервале моделирования), как правило, блок Discrete-Time Integrator «= 0» не поддерживает .

(Sample time > 0) - **моделирование дискретных систем,** шаг дискретизации по времени выходного сигнала. В этом режиме ошибка округления не накапливается, поскольку Simulink начинает отсчет номера текущего шага с нуля для каждого периода.

По умолчанию этот параметр равен Sample time =1.

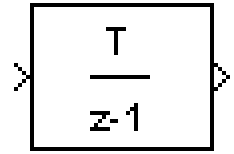
Блок дискретного интегратора времени Discrete Time Integrator



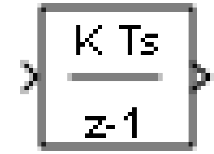


Блок дискретного интегратора времени Discrete Time Integrator

Блок интегратора дискретного времени позволяет
Определить начальные условия.
Вывести состояние блока.
Определить верхний и нижний пределы интеграла.
Сброс состояния в зависимости от дополнительного входа сброса.



1/s ~ T/(z -1)



1/s ~ Tz/(z -1)

Блок **DiscreteTime Integrator** служит для дискретного интегрирования времени.
Окно параметров:

- Integration method – метод численного интегрирования;
- External reset – сброс внешним сигналом;
- Initial condition – задание начальных значений выходного сигнала;
- Limit output – ввод ограничений на изменение выходного сигнала сверху Upper и снизу Lower;
- Show saturation port – показ порта, дающего сигнал об ограничении;
- Show state port – показ порта статуса интегратора;
- Sample time – эталонное время.

1 применяется верхний предел.
0 интеграл не ограничен.
-1 применяется нижний предел.

Методы интегрирования:

- Forward Euler – прямой метод Эйлера (прямая прямоугольная или левосторонняя аппроксимация)

$$y(k) = y(k-1) + T * u(k-1)$$

- Backward Euler – обратный метод Эйлера (обратная прямоугольная или правосторонняя аппроксимация);

$$y(k) = y(k-1) + T * u(k)$$

- Trapeziodal – метод трапеций.

1/s ~ T/2* (z +1)/(z-1)

$$x(k) = y(k-1) + T/2 * u(k-1)$$

Здесь x(k+1) приведена наилучшая оценка следующего выходного сигнала. Это не совсем состояние в том смысле, что x(k) != y(k)

Применение блока Discrete Time Integrator

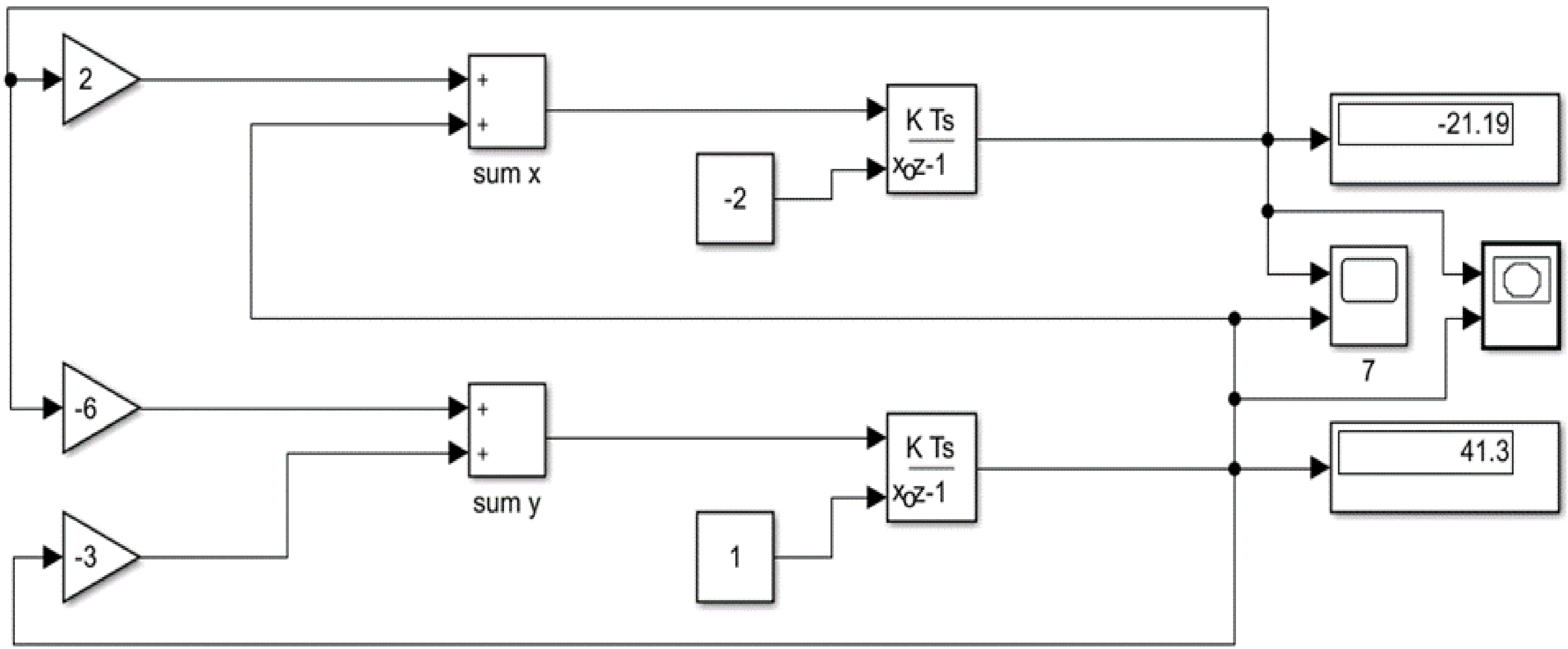
$$\begin{cases} \dfrac{dx}{dt} = 2x + y \\ \dfrac{dy}{dt} = -6x - 3y \end{cases}, \quad x(0) = -2, \quad y(0) = 1.$$

Backward Euler – обратный метод Эйлера (обратная прямоугольная или правосторонняя аппроксимация);

```
y (k) = y (k-1) + T * u(k)
```

$$x(k+1) = x(k) + T\big(2x(k) + y(k)\big),$$

$$y(k+1) = y(k) + T\big(-6x(k) + 3y(k)\big).$$

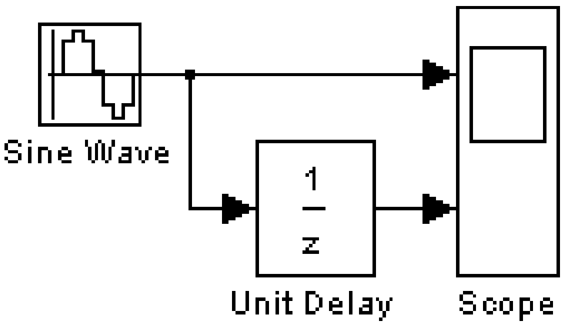


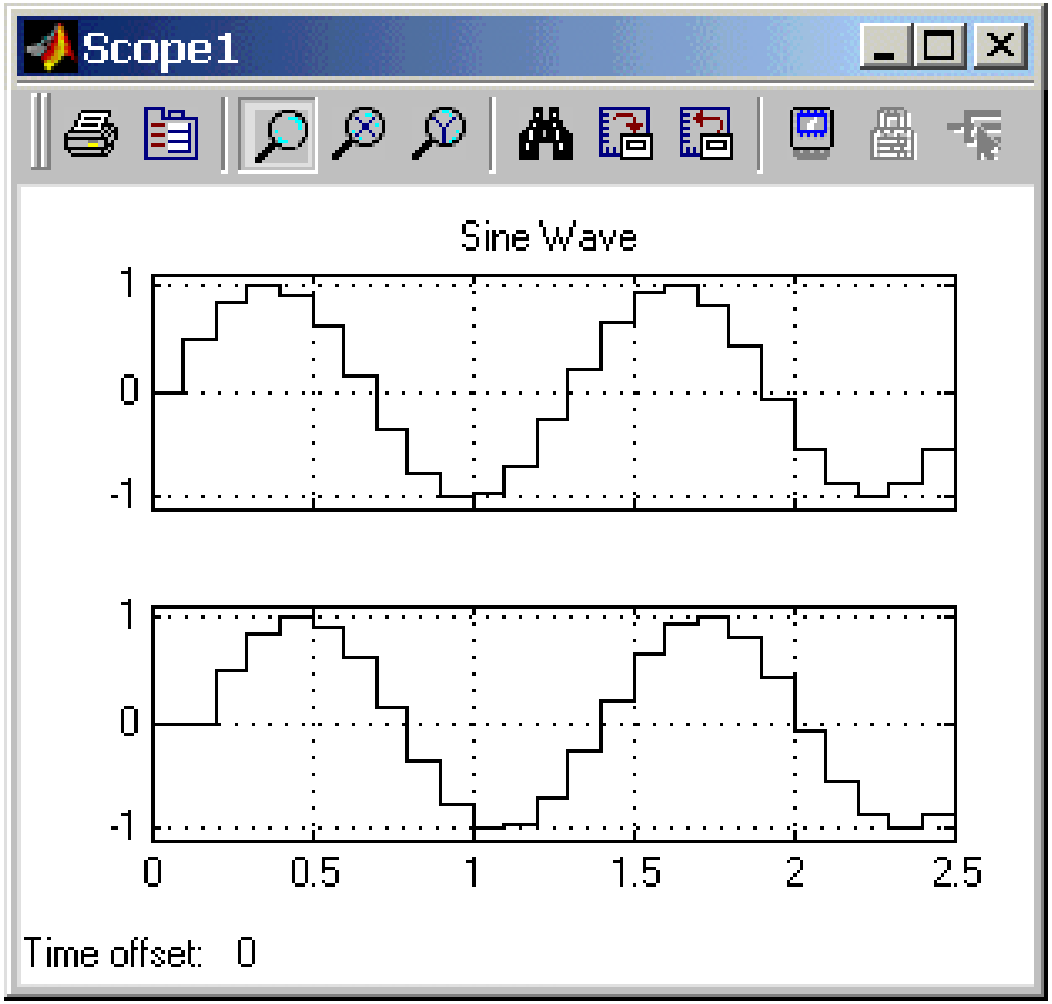
Блок единичной дискретной задержки Unit Delay (не путать с дискретным интегратором)

Задержка входного сигнала (скалярного, векторного) на один шаг модельного времени

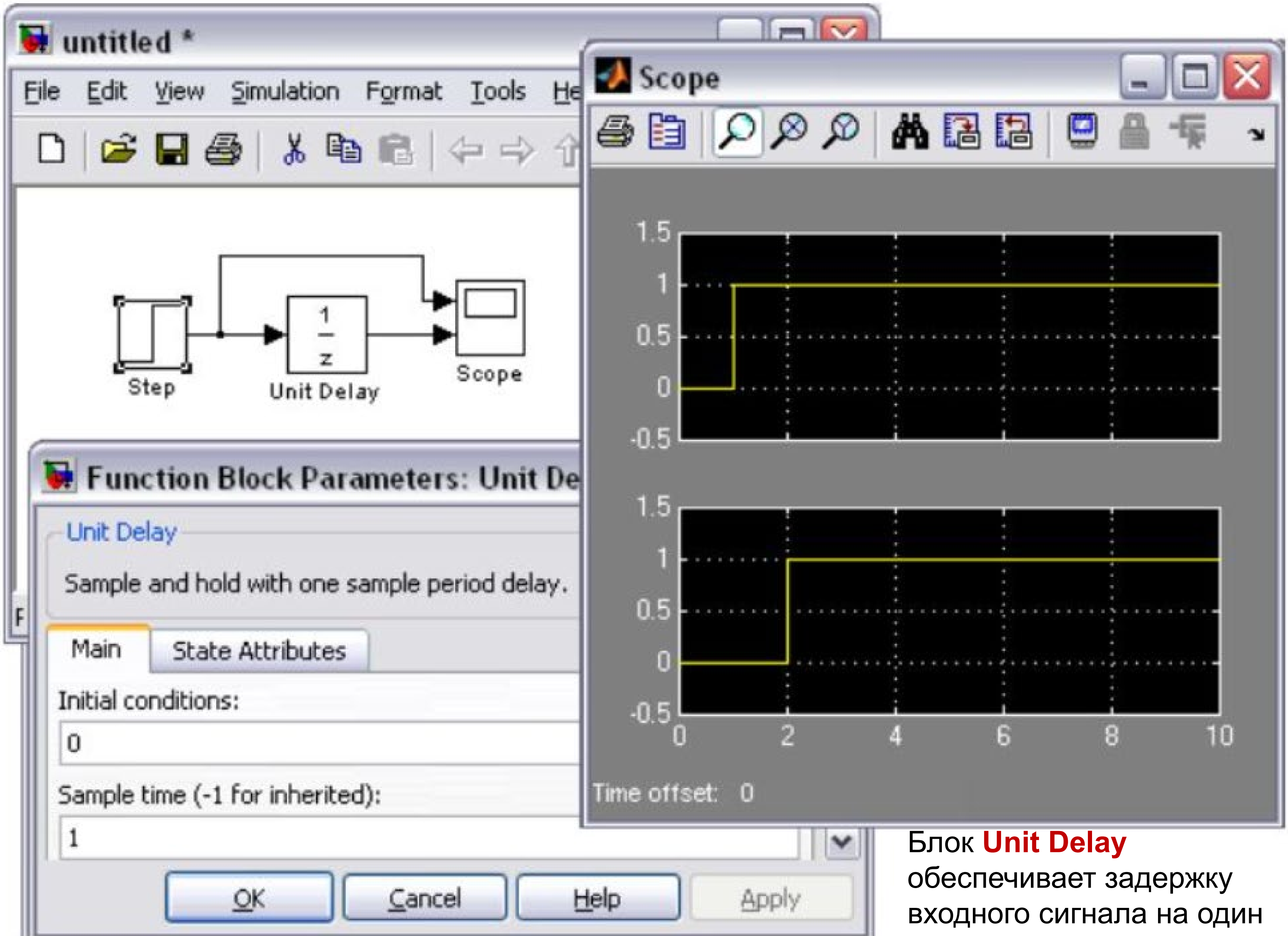
Параметры окна настроек:

- **Initial condition** – Начальное значение для выходного сигнала.
- **Sample time** – Шаг модельного времени.

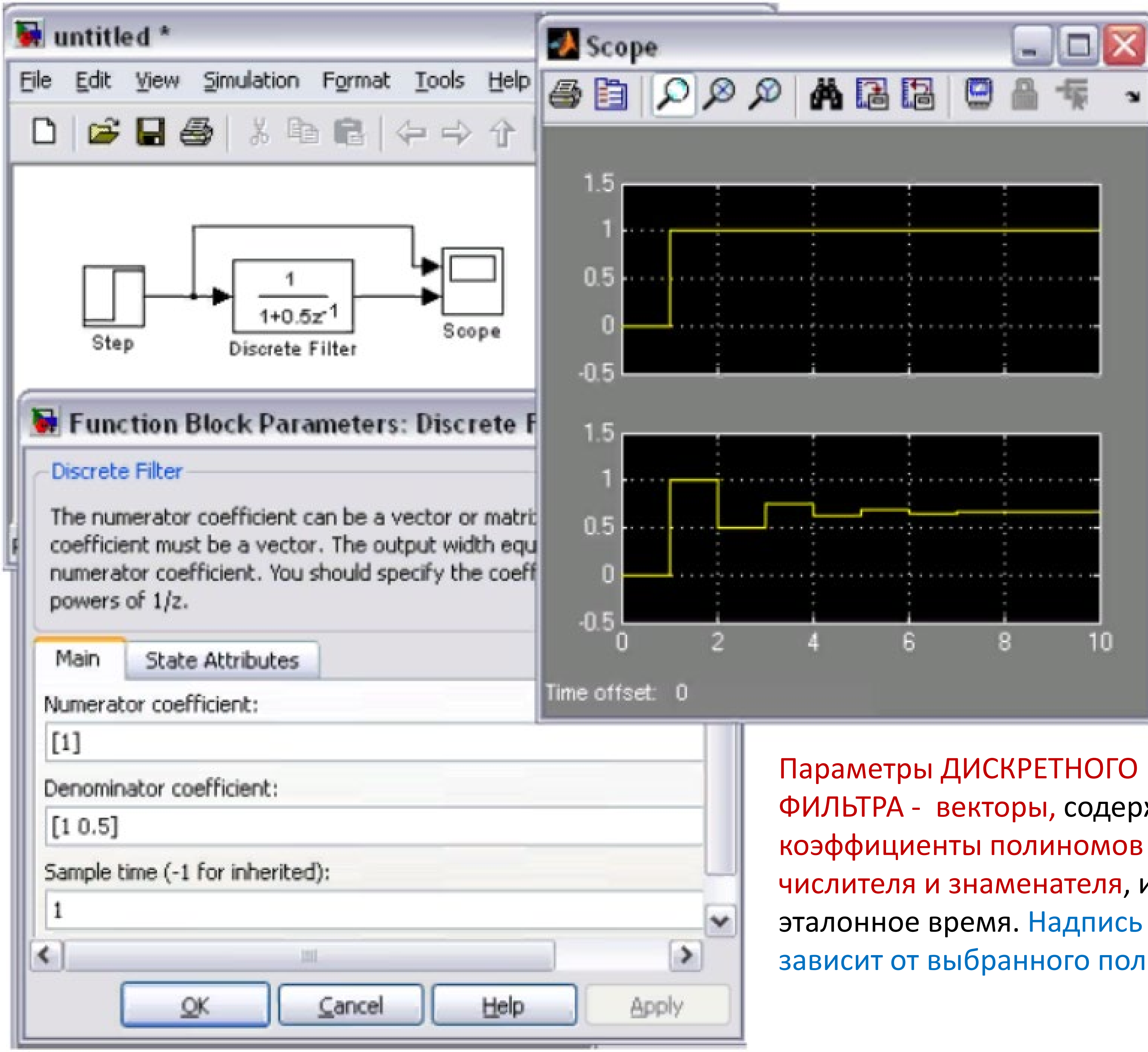




Пример с задержкой дискретного сигнала на один временной шаг= **0.1с.**



Блок **Unit Delay** обеспечивает задержку входного сигнала на один шаг эталонного времени.



Параметры ДИСКРЕТНОГО ФИЛЬТРА - векторы, содержащие коэффициенты полиномов числителя и знаменателя, и эталонное время. Надпись в блоке зависит от выбранного полинома

Блок дискретного фильтра **Discrete Filter**

задает дискретную передаточную функцию от обратного аргумента (**1/z**):

$$H(1/z) = \frac{num(1/z)}{den(1/z)} = \frac{num_0 z^0 + num_1 z^{-1} + num_2 z^{-2} + \ldots + num_m z^{-m}}{den_0 z^0 + den_1 z^{-1} + den_2 z^{-2} + \ldots + den_m z^{-n}}$$

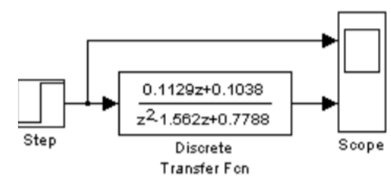
m+1 и***n+1*** – количество коэффициентов числителя и знаменателя, соответственно.***num*** – вектор или матрица коэффициентов числителя,***den*** – вектор коэффициентов знаменателя.

Параметры:

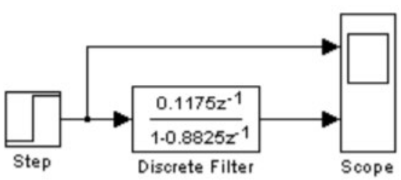
1.Numerator — Вектор или матрица коэффициентов числителя

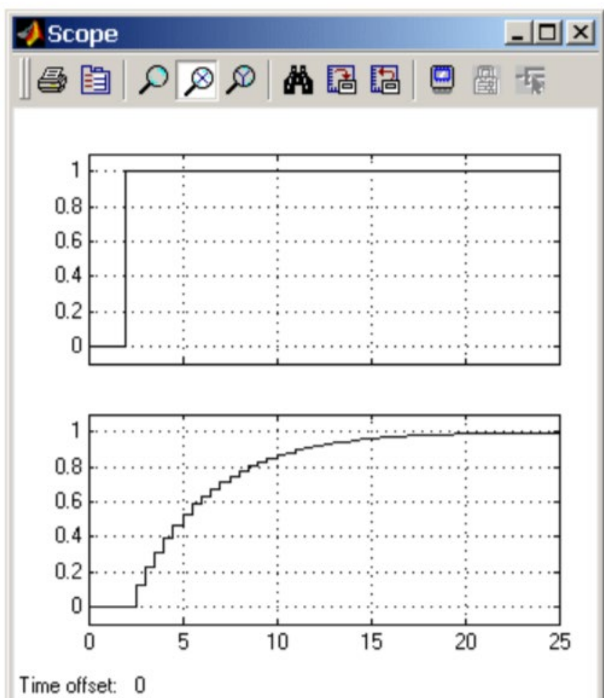
2.Denominator –Вектор коэффициентов знаменателя

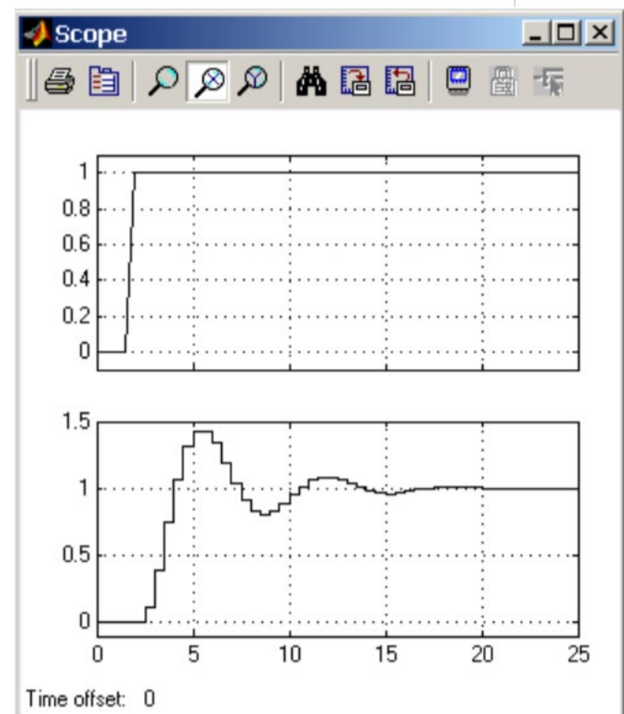
3.Sample time — Шаг дискретизации по времени.



Шаг дискретизации выбран равным **0.5**с.





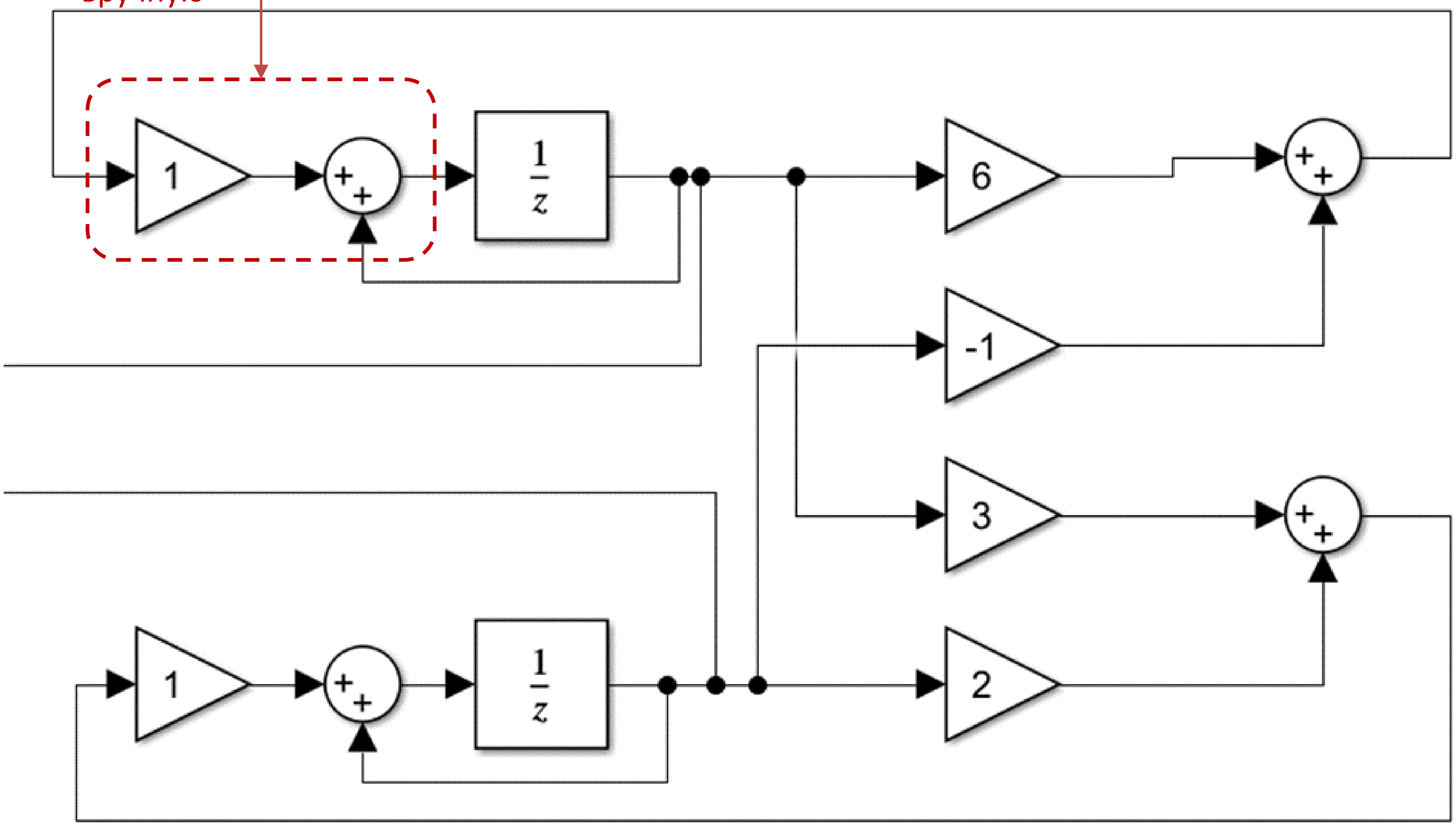


Применение блока задержки Unit Delay для дискретного интегрирования

$$\begin{cases} \dfrac{dx}{dt} = 6x - y \\ \dfrac{dy}{dt} = 3x + 2y \end{cases}, \quad x(0) = 1, \;\; y(0) = -1.$$

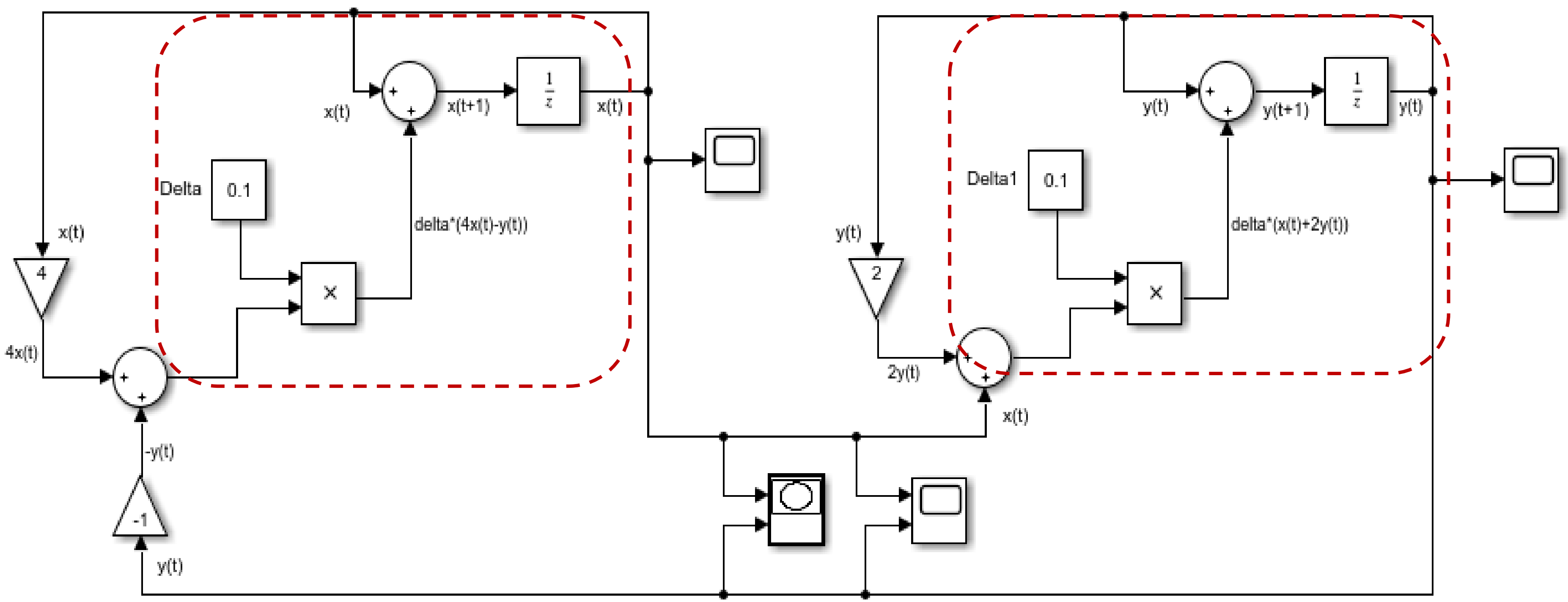
$$x(k+1) = x(k) + T\big(6x(k) - y(k)\big),$$
$$y(k+1) = y(k) + T\big(3x(k) + 2y(k)\big).$$

Организация расчета интеграла «вручную»



Искусственное использование блока задержки Unit Delay для дискретного интегрирования

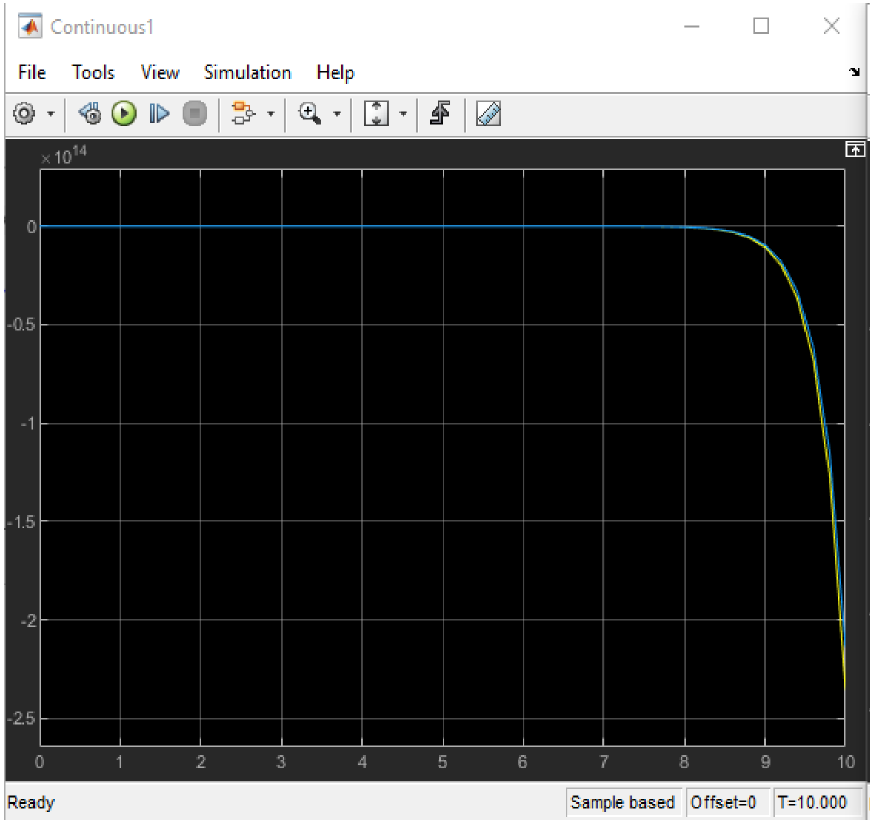
$$x(t+1) = x(t) + delta(4x(t) - y(t))$$
$$y(t+1) = y(t) + delta(x(t) + 2y(t))$$

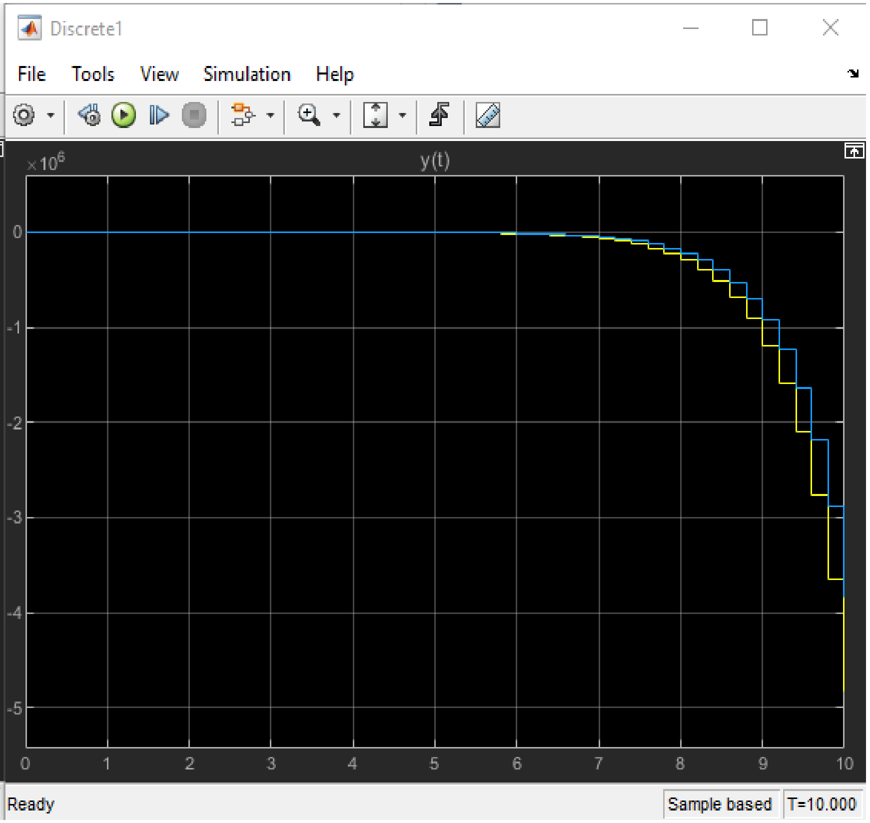


Применение блока задержки Unit Delay для дискретного интегрирования

$$\frac{dx}{dt} = 4x - y$$
$$\frac{dy}{dt} = x + 2y$$

$$x(t+1) = x(t) + delta(4x(t) - y(t))$$
$$y(t+1) = y(t) + delta(x(t) + 2y(t))$$





Применение блока задержки Unit Delay для искусственного конструирования дискретного интегратора

$$\begin{cases} \dfrac{dx}{dt} = 2x - y \\ \dfrac{dy}{dt} = x + 2y \end{cases}, \quad x(0) = -1, \quad y(0) = 0. \quad \Rightarrow \begin{cases} x(t+1) = x(t) + \Delta(2x(t) - y(t)) \\ y(t+1) = y(t) + \Delta(x(t) + 2y(t)) \end{cases}$$

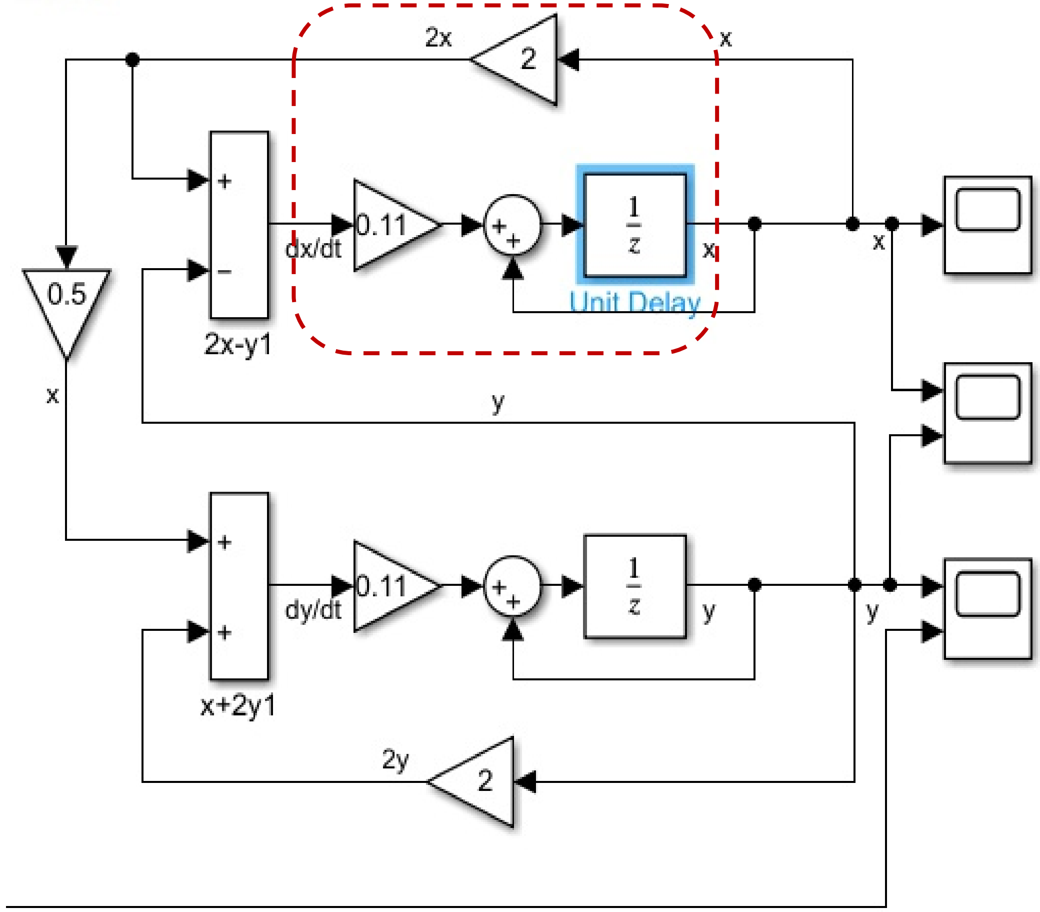
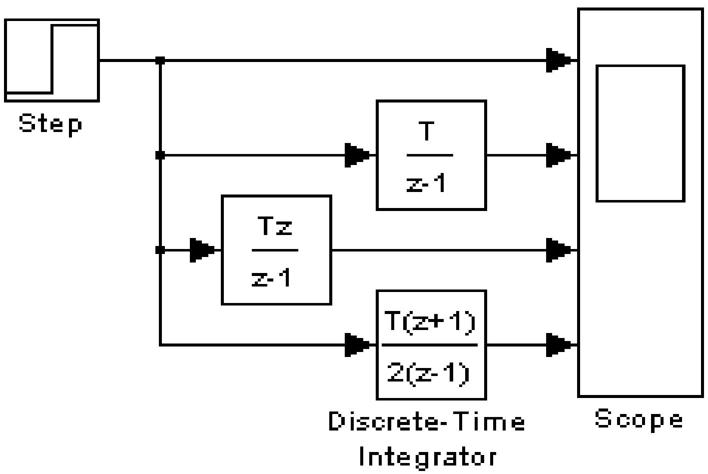


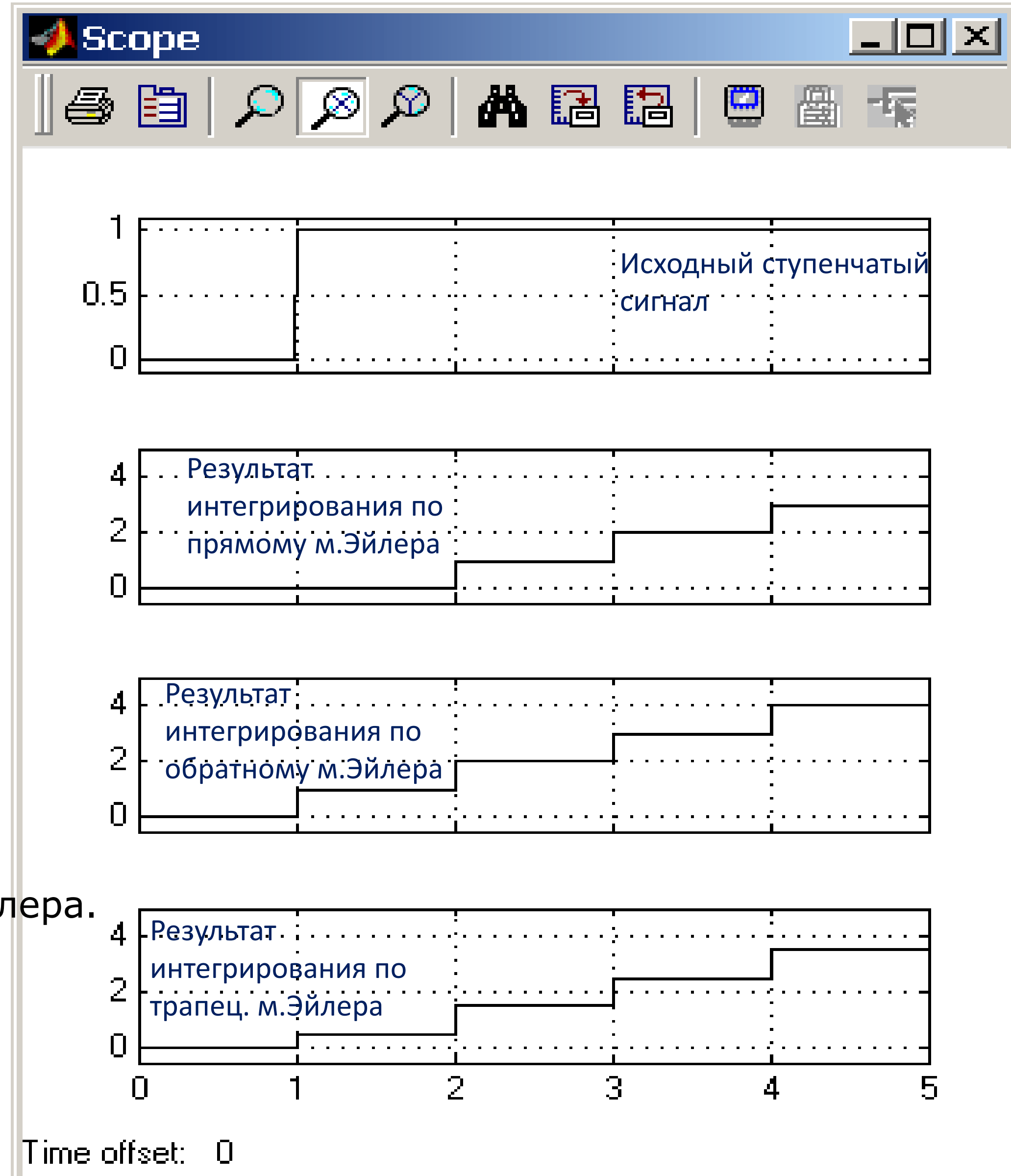
Иллюстрация трех способов численного интегрирования блока Discrete-Time Integrator

T/(z-1)

Выходной сигнал рассчитывается как

y(k) = y(k–1) + T*u(k–1),
y – выходной сигнал интегратора,
u – входной сигнал интегратора,
T – шаг дискретизации,
k – номер шага моделирования.





- **Backward Euler** – Обратный метод Эйлера.

T*z/(z–1) :
y(k) = y(k–1) + T*u(k).

- **Trapeziodal** – Метод трапеций.

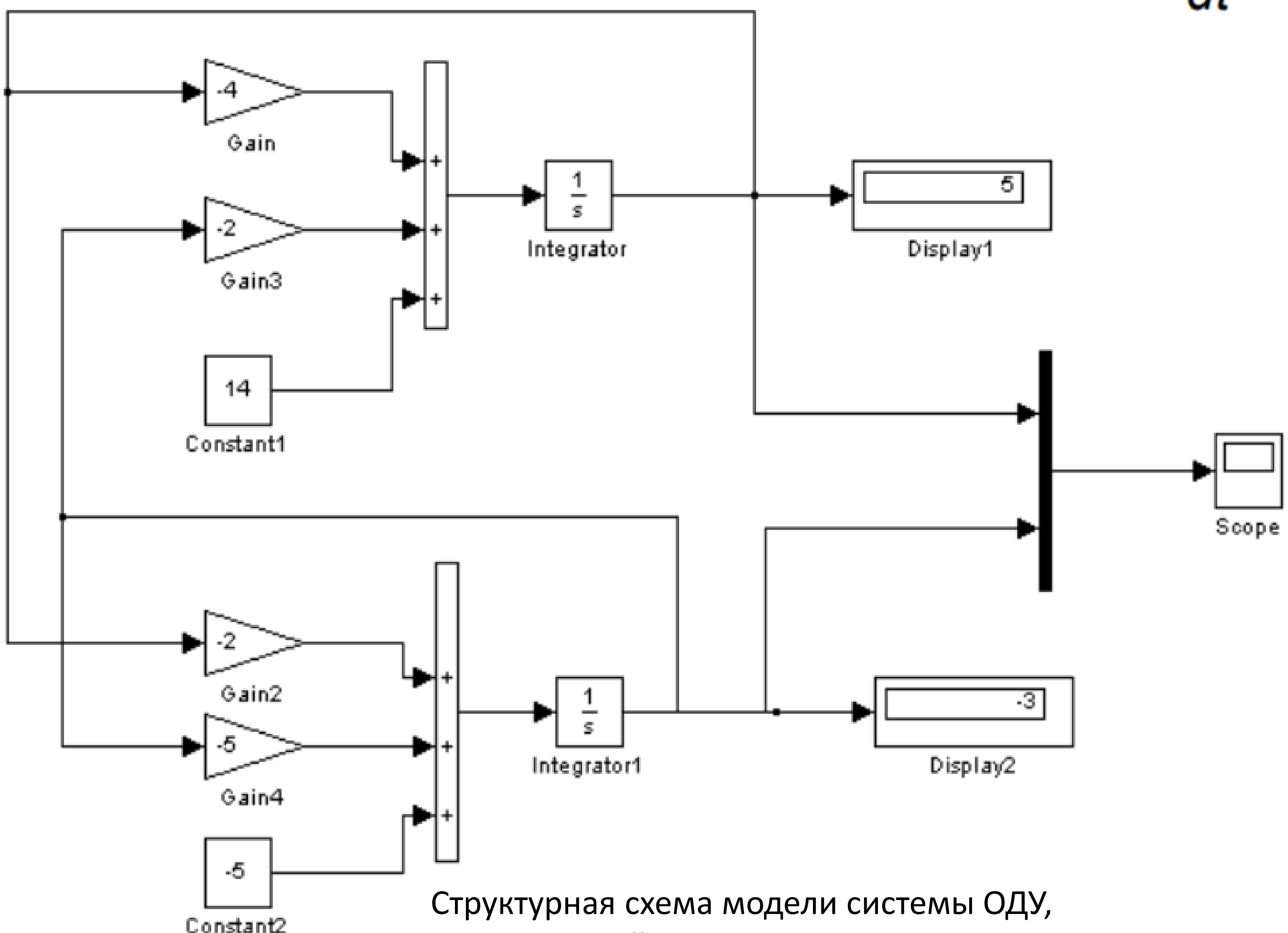
T/2*(z+1)/(z–1) :
x(k) = y(k–1) + T/2 * u(k–1).

Пример. Решение системы линейных алгебраических уравнений

$$4x_1 + 2x_2 = 14$$
$$2x_1 + 5x_2 = -5$$

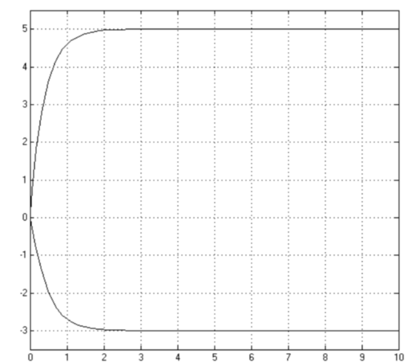
Переход к равносильной системе ОДУ

$$\frac{dx_1}{dt} = 14 - 4x_1 - 2x_2$$
$$\frac{dx_2}{dt} = -5 - 2x_1 - 5x_2$$

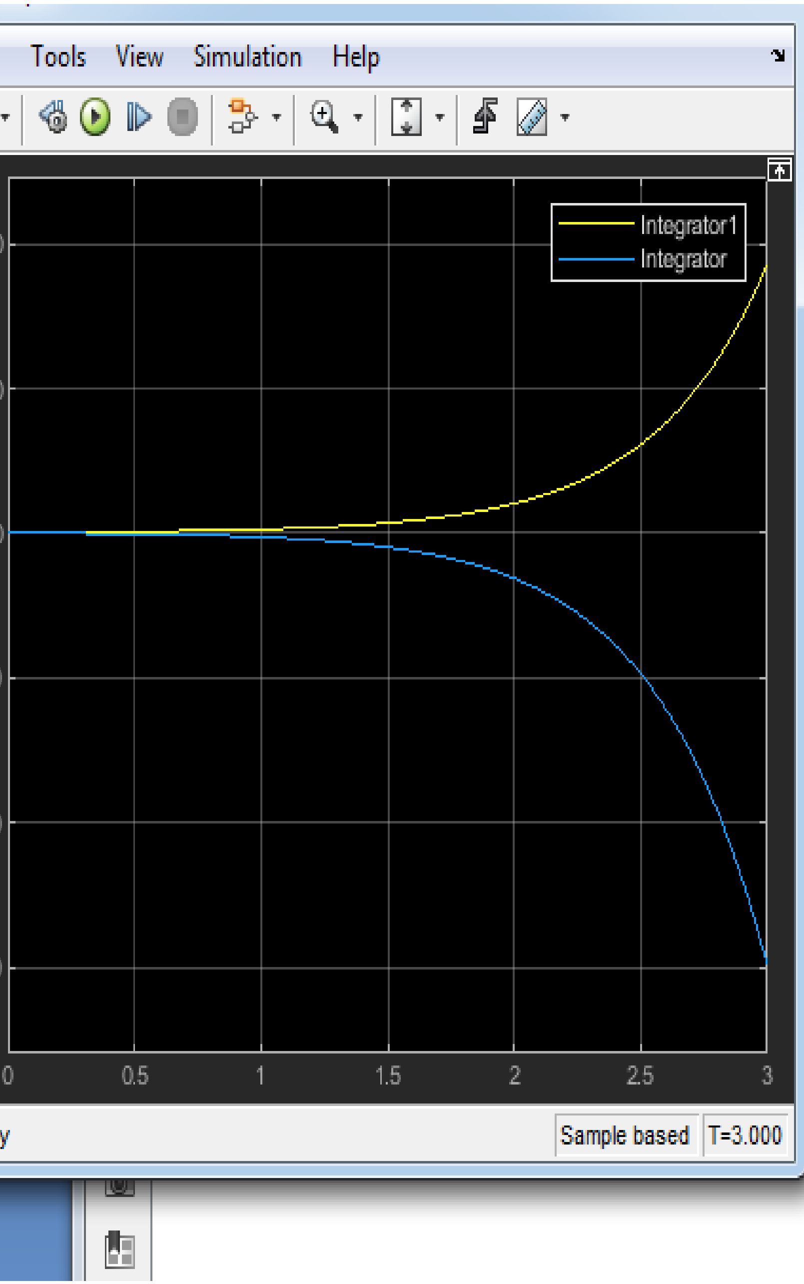


Решение после t=2

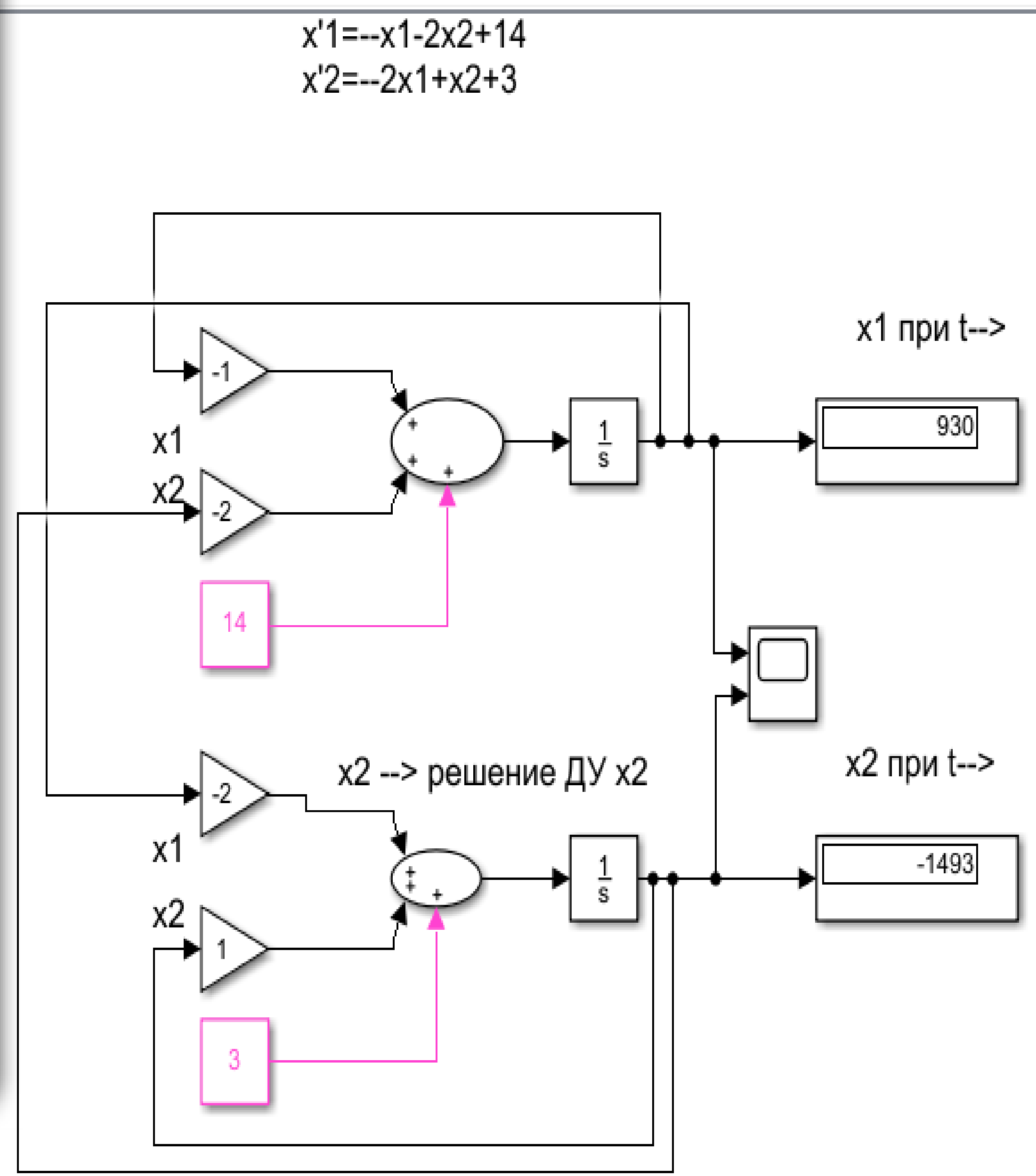
$$x_1 = 5,$$
$$x_2 = -3.$$



Структурная схема модели системы ОДУ, равносильной СЛАУ 2-го порядка

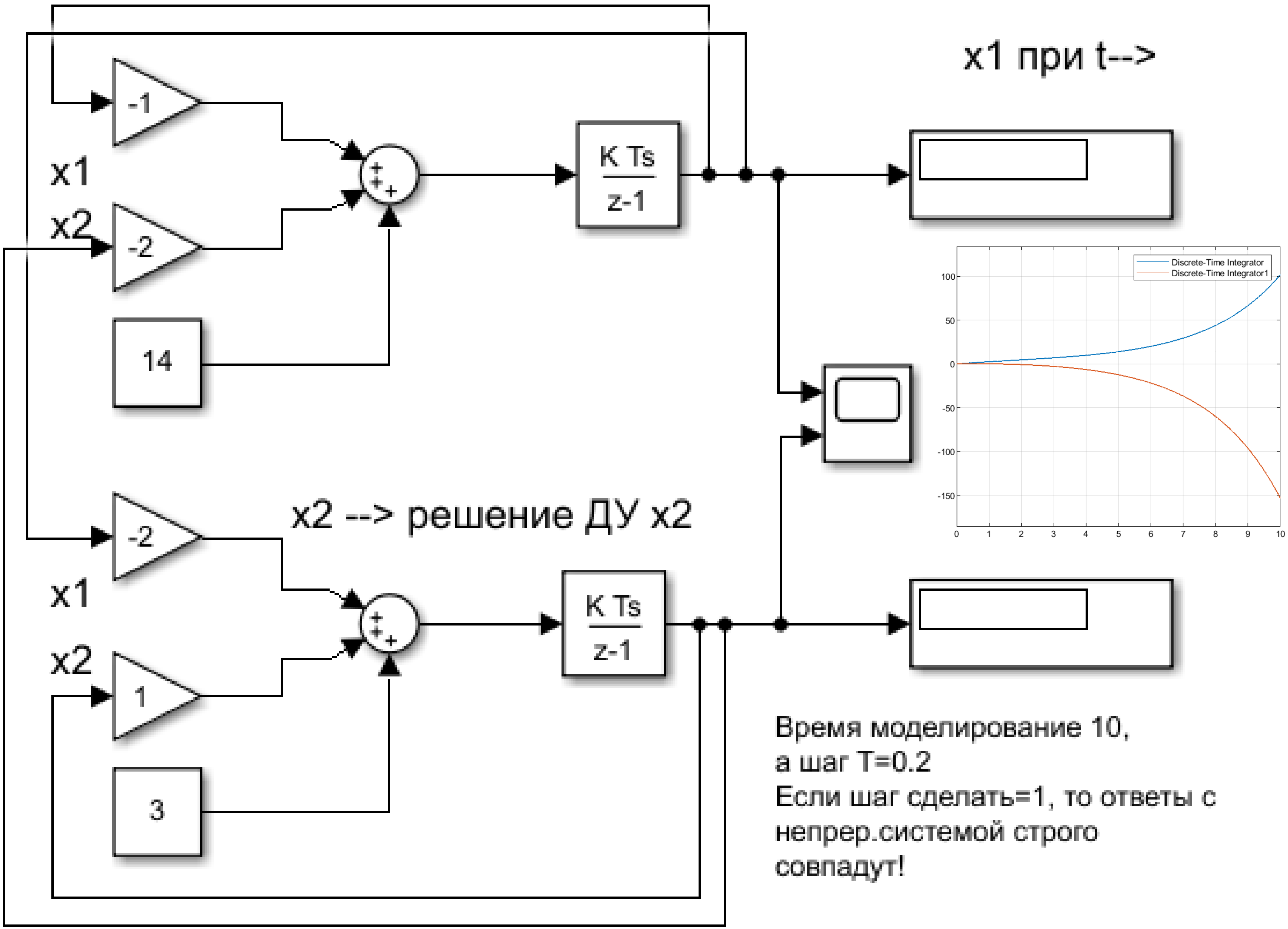


Tools View Simulation Help
Integrator1
Integrator
0 0.5 1 1.5 2 2.5 3
Sample based
T=3.000

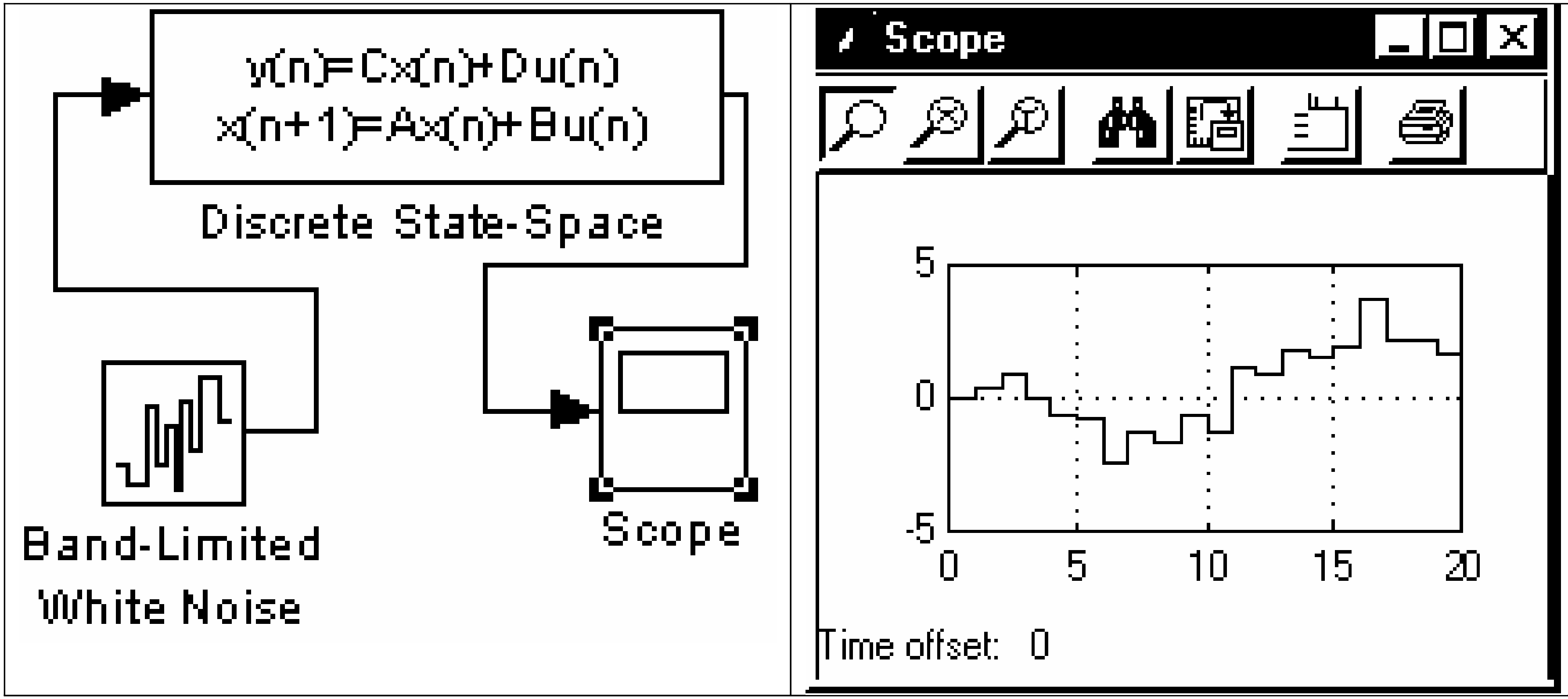


x'1=--x1-2x2+14
x'2=--2x1+x2+3
x1 при t-->
-1
x1
x2
-2
14
1/s
930
x2 --> решение ДУ x2
x2 при t-->
-2
x1
x2
1
3
1/s
-1493

x1(t+1)=x1(t)+T(--x1-2x2+14)
x2(t+1)=x2(t)+T(--2x1+x2+3)



Время моделирование 10,
а шаг T=0.2
Если шаг сделать=1, то ответы с непрер.системой строго совпадут!



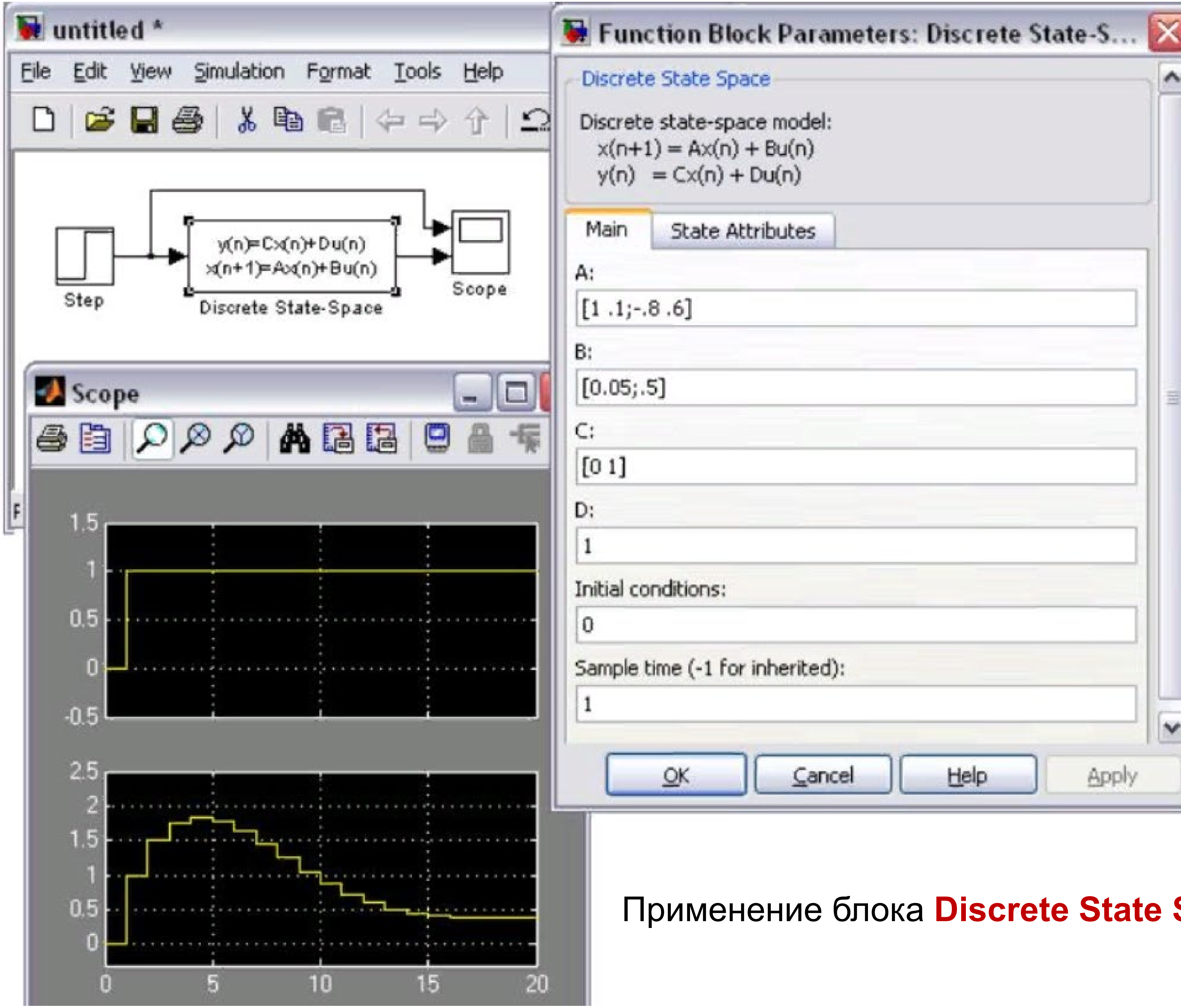
Пример использования блока **Discrete State-Space.**

Результаты моделирования для следующей дискретной системы (A=B=C=D=1)

$$x(n+1) = x(n) + u(n),$$

$$y(n) = x(n) + u(n),$$

где x и y скалярные переменные, *u* белый шум с ограниченной полосой. Начальные условия для переменной x нулевые.



Применение блока **Discrete State Space**

БАЗИСНАЯ ФУНКЦИЯ - ЭЛЕМЕНТ БАЗИСА В ФУНКЦИОНАЛЬНОМ ПРОСТРАНСТВЕ.

Причины главенствующей роли гармонических функций в радиотехнике:

- гармоническое колебание легко реализуемо на практике;
- функции, - ортогональные и определены для любого аргумента;
- гармоническое колебание сохраняет свою форму при прохождении колебания через ЛС, могут только изменяться амплитуда и фаза;
- для гармонических функций имеется удобный аппарат комплексного анализа.

Используют кроме гармонического ряда Фурье также и другие виды спектральных разложений: по функциям Уолша, Бесселя, Хаара, Лежандра, полиномам Чебышёва и др.

Спектр сигнала — коэффициенты разложения сигнала в базисе ортогональных функций. Само разложение называют спектральным разложением сигнала.

Токи и напряжения в цепи под действием сигнала описываются ДУ, соответствующими элементам цепи и способу их соединения.

Линейные цепи описываются ЛДУ, причём для линейных цепей верен принцип суперпозиции:

действие на систему сложного сигнала, который состоит из суммы простых сигналов, равно сумме действий от каждого составляющего сигнала в отдельности.

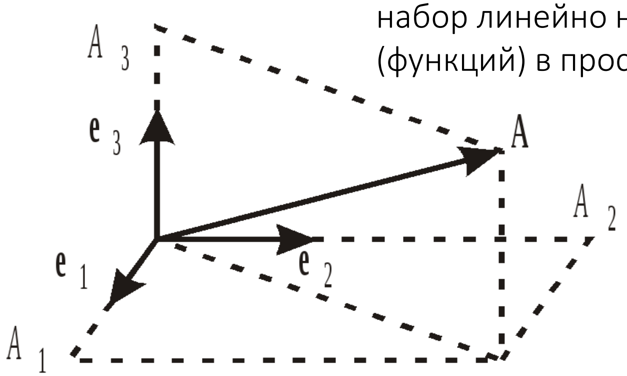
Пример. Пусть $(\varphi_n(x))$ - ортогональная система функций в $L_2[a;b]$. Выражение

$$c_0\varphi_{0(}(x)+c_1\varphi_1(x)+c_2\varphi_2(x)+\ldots+c_n\varphi_n(x)+\cdots=\sum_{n=0}^{\infty}c_n\varphi_n(x) \qquad (1)$$

называется *обобщенным рядом Фурье по ортогональной системе функций* $(\varphi_n(x))$. Если $(\varphi_n(x))$ - основная тригонометрическая система функций то ряд (1) называется *тригонометрическим рядом Фурье*.

РАЗЛОЖЕНИЕ ПРОИЗВОЛЬНОЙ ФУНКЦИИ В РАЗНЫХ БАЗИСАХ

Базис — упорядоченный (конечный или бесконечный) набор линейно независимых векторов (функций) в пространстве.



Любой вектор пространства может быть единственным образом представлен в виде линейной комбинации векторов из базисных векторов (функций).

$$x(t) = \sum_{n=-\infty}^{\infty} C_n \varphi_n(t)$$

Спектральные коэффициенты

Базисные функции

Базис $\{e_1, e_2, e_3\}$

$$x = (x_1; x_2; x_3) = x_1 \cdot e_1 + x_2 \cdot e_2 + x_3 \cdot e_3$$

Базис {2^0, 2^1, 2^2,….}

$$A_{10} = a_{n-1} \cdot 2^{n-1} + a_{n-2} \cdot 2^{n-2} + \ldots + a_1 \cdot 2^1 + a_0 \cdot 2^0$$

$$f(x) = f(0) + \frac{f'(0)}{1!}x + \frac{f''(0)}{2!}x^2 + \frac{f'''(0)}{3!}x^3 + \ldots + \frac{f^{(n)}(0)}{n!}x^n + \ldots$$

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots + \frac{(-1)^{n-1}}{(2n-1)!} \cdot x^{2n-1} + \ldots$$

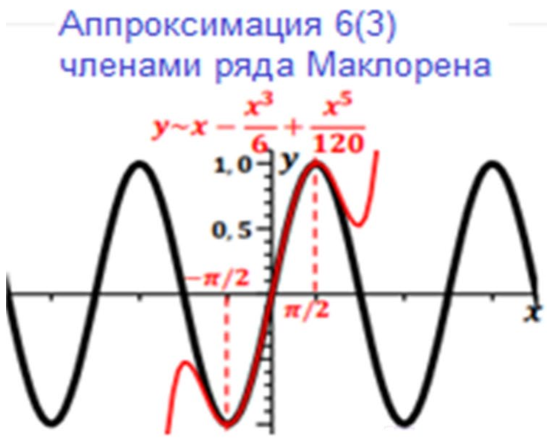
Базис {x^0, x^1, x^2,….}

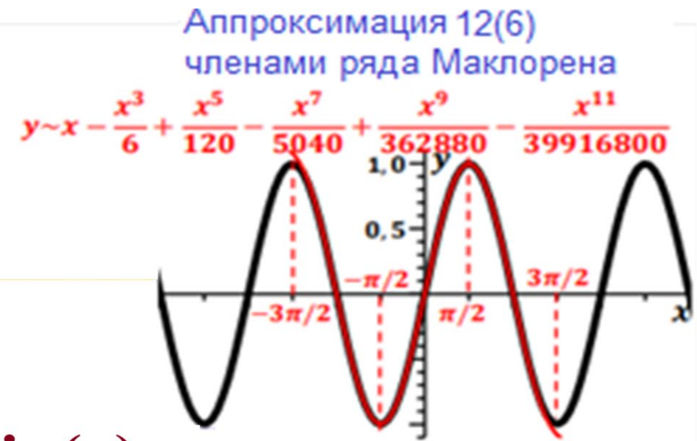
Базис {sin(0), cos(0).sin(1), cos(1),….}

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty}\left(a_k \cos k\frac{2\pi}{T}t + b_k \sin k\frac{2\pi}{T}t\right)$$

РАЗЛОЖЕНИЕ ПРОИЗВОЛЬНОЙ ФУНКЦИИ В СТЕПЕННОЙ РЯД.

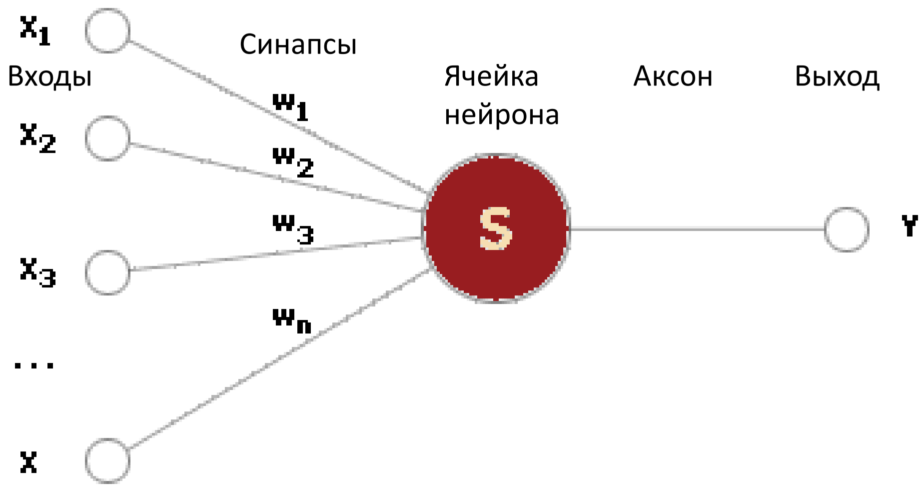
Условие: непрерывная дифференцируемость «нужное» число раз.





y=sin(x)

РАЗЛОЖЕНИЕ НЕЙРОНА ПО БАЗИСУ



Искусственный нейрон (аналогично живому) состоит из синапсов, связывающих входы нейрона с ядром; ядра нейрона осуществляют обработку входных сигналов и аксона, который связывает нейрон с нейронами следующего слоя.

Каждый синапс имеет вес, который определяет, насколько соответствующий вход нейрона влияет на его состояние. Состояние нейрона определяется по формуле

$$S = \sum_{i=1}^{n} x_i w_i,$$

n – число входов нейрона
x_i – значение i-го входа нейрона
w_i – вес i-го синапса.

ЛИТЕРАТУРА

1. Дьяконов В.П. MATLAB 5.3.1 с пакетами расширений. М.: «Нолидж», 2001.

2. Потемкин В.Г. Система инженерных и научных расчетов Матлаб 5.x. Т.1. М.: Диалог - МИФИ, 1999.

3. https://people.uncw.edu/hermanr/mat361/Simulink/ODE_Simulink.pdf .

4. Черных И.В. Simulink. Среда создания инженерных приложений. М.: Диалог-МИФИ, 2003

5. Гультяев А.К. Matlab 5.2. Имитационное моделирование в среде Windows: Практическое пособие. М.: Диалог - МИФИ, 1999.

6. Гультяев А.К. Визуальное моделирование в среде Matlab: учебный курс. СПб.: Питер, 2001.

7. Цисарь И.Ф., Крысин М.А. Matlab-Simulink. Лаборатория экономиста. М.: Анкил, 2001.

8. Документация к пакету Matlab фирмы MathWorks (pdf-files).

9. www.exponenta.ru - образовательный сайт по математическим методам и программированию с использованием пакетов прикладных программ.

10. www.matlab.ru - консультационный сайт пакета Matlab.

11. https://mmf.bsu.by/wp-content/uploads/2016/10/Goloubeva.pdf

12. https://fr.slideserve.com/wilona/introduction-to-simulink