

5. S-функции и анимация Simulink-моделей

Расширение возможностей *Simulink* при помощи блока **User-Defined functions\S-Function**. Анимация данных, полученных в *Simulink*.

5.1. Постановка задачи

При помощи блока **User-Defined functions\S-Function** создать новые блоки, расширяющие и дополняющие возможности **Simulink**. Изучить принципы построения S-функции, ее входные и выходные аргументы, **callback**-методы.

5.2. Использование S-функций в Simulink-моделях

5.2.1. Простейшая S-функция

Модель **L0501.mdl**

Одним из самых простых примеров **S-функций**, поставляемых с пакетом **MATLAB**, является функция **timestwo** (файл **MATLAB7\toolbox\simulink\blocks\timestwo.m**). Данная **S-функция** выполняет умножение входного сигнала на коэффициент 2. Пример модели с **S-функцией timestwo** приведен на рис.5.1.

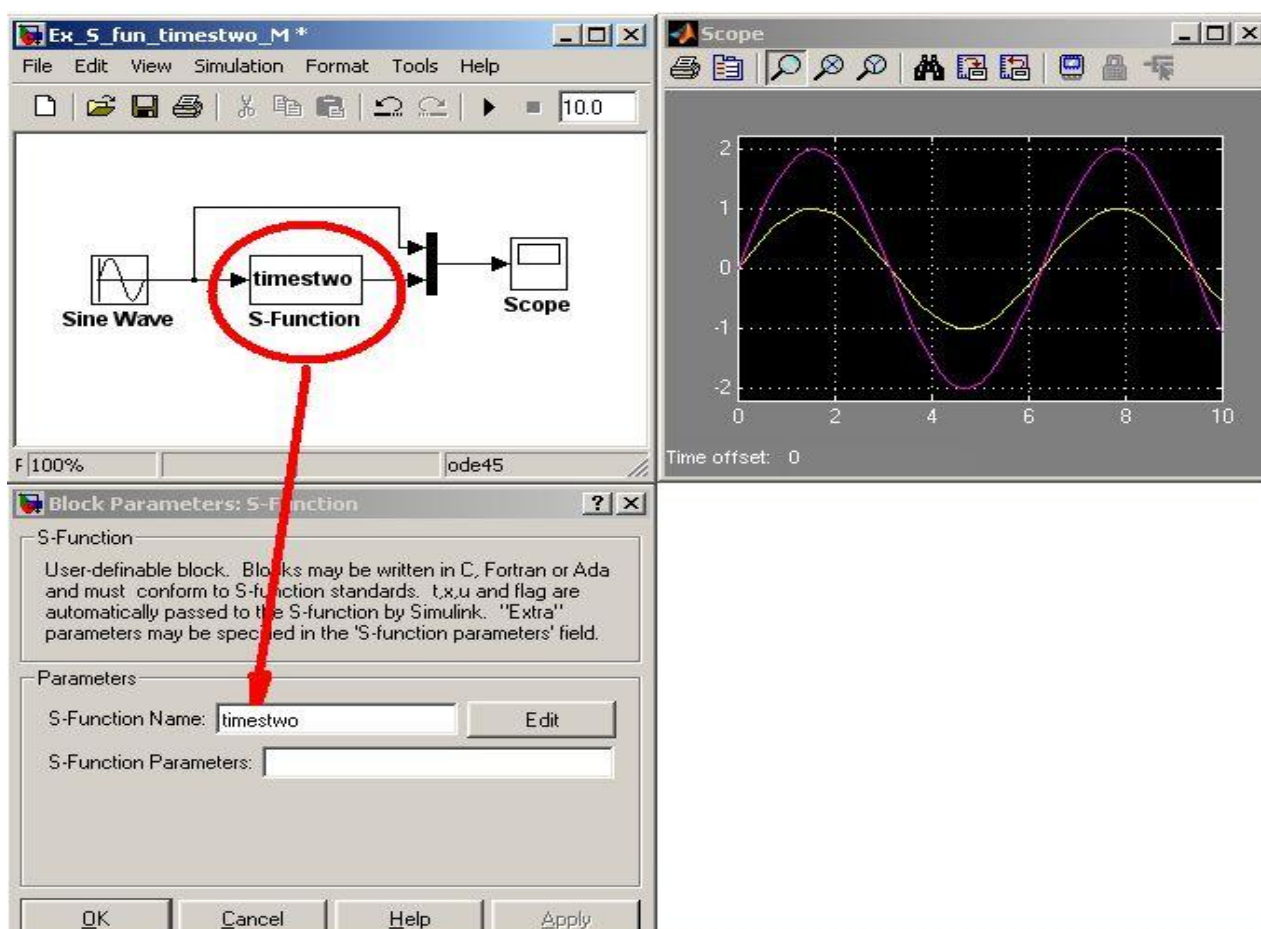


Рис. 5.1. Модель с S-функцией timestwo

Постройте **Simulink**-модель, найдите и скопируйте в тот же директорию файл **timestwo.m**, подключите его к блоку **S-Function**. Изучите структуру этого файла. Запустите **Simulink**-модель.

5.2.2. Умножение входного сигнала на коэффициент k

Модель **L0502.mdl**

Измените модель **L0501.mdl** таким образом, чтобы **S-функция** выполняла умножение входного сигнала на задаваемый в блоке **S-function** в поле **S-Function Parameters** коэффициент k .

5.2.3. Анимация «Часы»

Модель **L0503.mdl**

Создать анимацию часов, в которых движутся часовая и минутная стрелки.

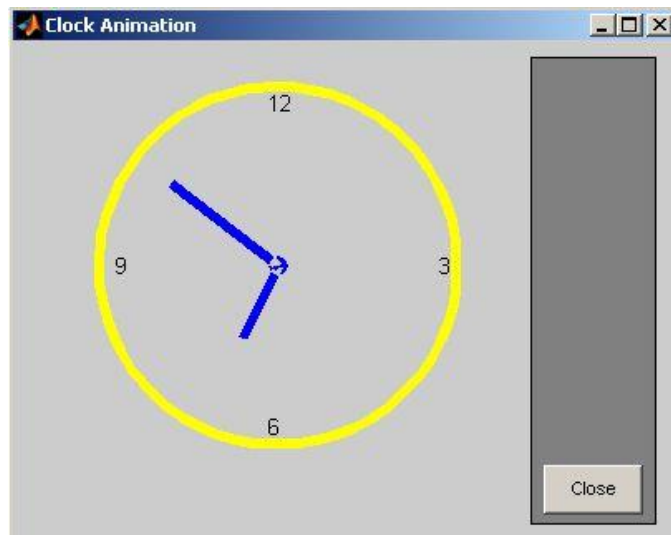


Рис. 5.2. Часы. Часовая и минутная стрелки должны двигаться

Шаг 1. Параметры конфигурации. Время моделирования **Stop time** установите равным одним суткам (в минутах). Установите следующие параметры **Solver options**: **Type** = **Variable Step**; **Solver** = **discrete (no continuous states)**; **Max step size** = **0.1**.

Шаг 2. Координаты (x, y) точки, движущейся по окружности, можно рассчитать при помощи **sin**- и **cos**-функций соответственно, как показано на Рис.5.2, где ω – частота **Frequency**.

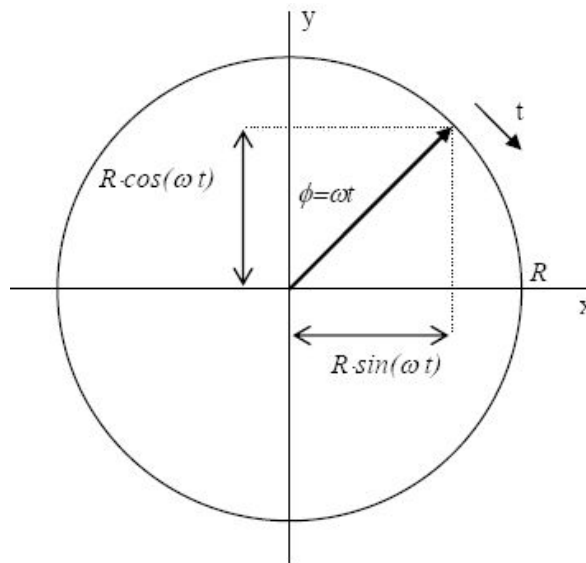


Рис. 5.3. Движение по окружности

Для этого перенесите в **Simulink**-модель четыре блока **Sources\Sine Wave**, переименуйте их как **Cos Wave Minute**, **Sin Wave Minute**, **Cos Wave Hour**, **Sin Wave Hour**. В блоках **Cos Wave** установите значение начальной фазы **Phase (rad)**= $\pi/2$. Во всех четырех блоках установите частоту **Frequency** таким образом, чтобы минутная стрелка совершала один оборот за 60 единиц модельного времени, а часовая стрелка совершала 1/12 оборота за этот же период времени.

Шаг 3. Объедините четыре сигнала в вектор при помощи мультиплексора **Signal Routing\Mux** и подайте его на вход в блок **User-Defined Functions\S-Function**. В качестве параметров блока **S-Function** введите имя анимационного **m**-файла **S-Function Name = anim** и **S-Function Parameters = 0.1**.

Шаг 4. Анимационный **m**-файл **anim.m**. Имя данного **m**-файла не должно совпадать с именем **Simulink**-модели. Поместите данный файл в тот же каталог, что и **Simulink**-модель.

%M-file 'anim.m' for clock animation:

```
function [sys,x0]=clockanim(t,x,u,flag,par_ts);
global CAnim
global handle
global min_length      % length of minute hand
global hour_length     % length of hour hand

% update model
if flag==2,                % sys = mdlUpdate(t,x,u,par_ts)
    if get(0,'Children')==1, % test, if plot exists
        set(0,'currentfigure',CAnim);
        x0=0; y0=0;
```

```

    yM = min_length*u(1);
    xM = min_length*u(2);
    yH = hour_length *u(3);
    xH = hour_length *u(4);

% Use NaNs to make the hands distinct
    x=[xH x0 NaN x0 xM];
    y=[yH y0 NaN y0 yM];
    set(handle,'XData',x,'YData',y);
    plot(x0,y0,'.','MarkerSize',30,'EraseMode','none');
    drawnow;                                % update plot
end
sys=[];

                                % sys = mdlGetTimeOfNextVarHit(t,x,u,par_ts)
elseif flag == 4                % Return next sample hit
    sys = t + par_ts;           % the constant sample interval

elseif flag==0,                % [sys,x0,str,ts]=mdlInitializeSizes;
% Initialize the figure for use with this simulation
    animinit('Clock Animation');
    CAnim = findobj('Type','figure','Name','Clock Animation');
    axis([-5 5 -5 5]);          % axis properties
    hold on
% Set up the geometry for the problem
    hour_length=1.8; min_length=3;
    x0=0; y0=0;
    xH=x0 - hour_length; yH=y0;
    xM = x0; yM = y0 + min_length;
% Use NaNs to make the hands distinct
    x=[xH x0 NaN x0 xM];
    y=[yH y0 NaN y0 yM];
% plot the clock and init setting
    handle=plot(x,y,'EraseMode','xor','LineWidth',5);
% plot center point with size 30
    plot(x0,y0,'.','MarkerSize',30,'EraseMode','none');
% write hours on the clock

```

```
text(-0.2,3.6,'12');  
text(-0.2,-3.6,'6');  
text(3.6,0,'3');  
text(-3.6,0,'9');  
% plot a circle around in yellow color  
m = 4*sin(0:0.1:2*pi+0.1);  
n = 4*cos(0:0.1:2*pi+0.1);  
plot(m,n,'y','LineWidth',5);  
%Set the relative scaling of the individual axis data values.  
set(gca,'DataAspectRatio',[1 1 1]);  
sys=[0 0 0 4 0 0]; % 4...number of inputs  
x0 = []; % No continuous states  
end;
```

Введите и внимательно изучите данный код.

Шаг 5. Запустите **Simulink**-модель и проанализируйте ее выполнение.

Шаг 6. Усовершенствуйте модель. Для этого перенесите в **Simulink**-модель два блока **Sources\Constant** и блок ручного переключателя **Signal Routing\Manual Switch**. Соедините выходы блоков **Constant** с входами блока **Manual Switch**. Установите в блоке **Constant** значение параметра **Constant Value=1**, а в блоке **Constant1** значение параметра **Constant Value=0**. Блок **Manual Switch** выполняет переключение входных сигналов по команде пользователя. Командой на переключение является двойной щелчок левой клавишей мыши на изображении блока. При этом изображение блока меняется, показывая какой входной сигнал в данный момент проходит на выход блока. Переключение блока можно выполнять как до начала моделирования, так и в процессе расчета модели. Установите изначально проход 0-го входного сигнала на выход. Подсоедините выход блока **Manual Switch** к мультиплексору **Mux**. Измените в **m**-файле **anim.m** значение числа входов **number of inputs** с 4 на 5.

Шаг 7. Измените код анимационного **m**-файла **anim.m** следующим образом. Если значение сигнала, идущего из блока **Manual Switch**, равно 0, то стрелки движутся по часовой стрелке. Если же значение сигнала, идущего из блока **Manual Switch**, переключается на 1, то стрелки начинают двигаться с текущих позиций в обратном направлении, т.е. против часовой стрелки.

5.2.4. Анимация «Маятник с ограничением движения»

Модель **L0504.mdl**

Самостоятельно создать анимацию маятника с ограничением движения.

5.2.5. Анимация «Плывущий график»

Модель **L0505.mdl**

Самостоятельно создать анимацию «плывущего» графика, воспользовавшись кодом **m**-функции, реализованной в прошлом семестре при изучении **MATLAB** (Лабораторная работа №4). **Simulink**-модель может иметь вид, как показано на Рис.5.4.

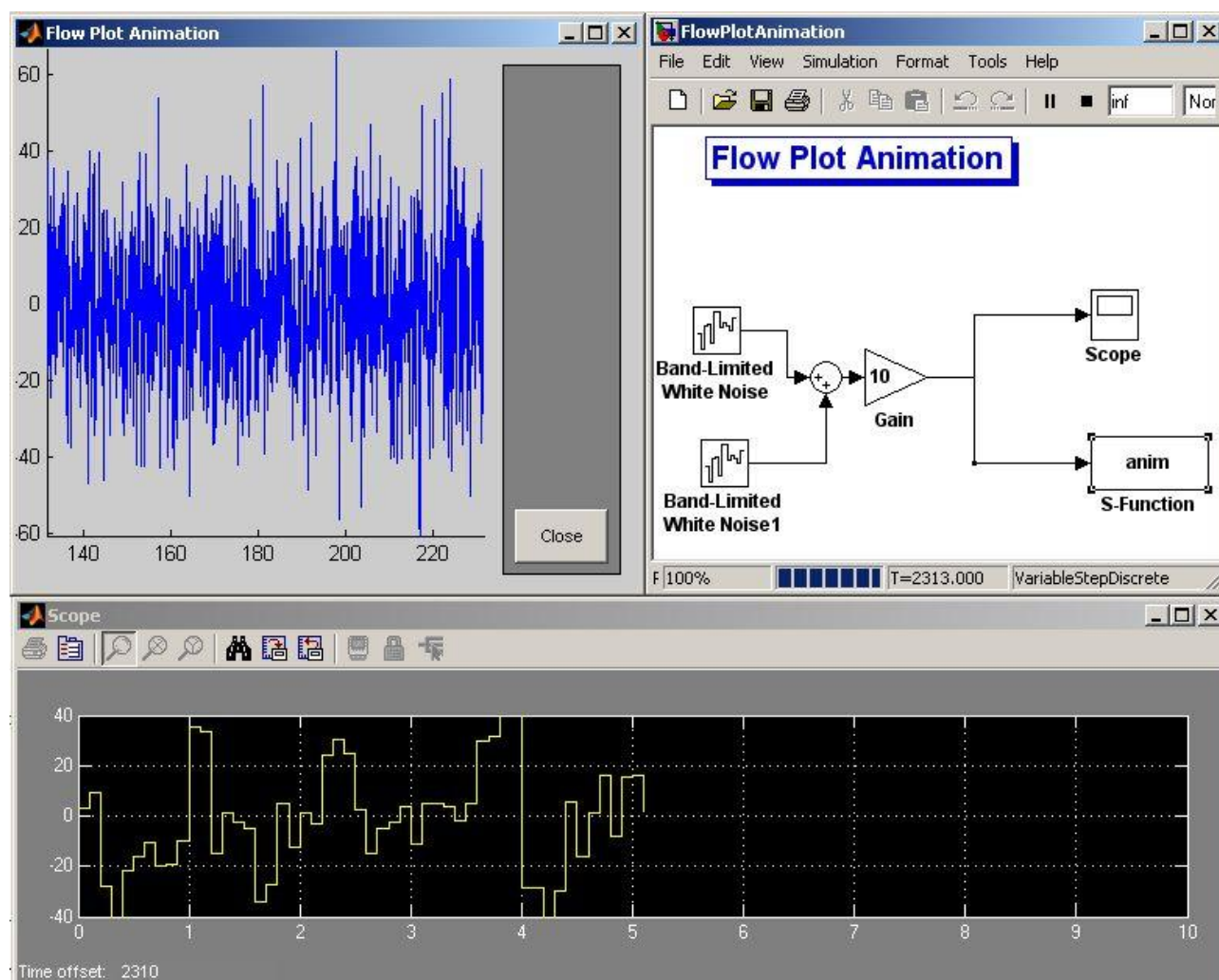


Рис. 5.4. Модель расчета «плывущего» графика

В представленной модели блоки **Sources\White Noise** являются источниками отображаемого сигнала.