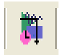
# 2. Моделирование динамических систем в `Simulink`. Непрерывные модели.

## 2.1. Запуск Simulink

Запустите *MatLab*. Установите свой рабочий каталог в качестве текущего каталога *MatLab*.

Запустите Simulink одним из способов:

- В командном окне ввести команду **`>> Simulink`**
- На панели инструментов нажать кнопку  (*Simulink*)
- В глобальном меню **`Start\Simulink`** выбрать желаемую утилиту
- Из главного меню **`File\Open...`** открыть готовую модель **`"*.mdl"`**.

Появится окно **`"Simulink Library Browser"`**. Нажмите в его панели инструментов кнопку **`Create a new model`** или выберите пункт меню **`File\new ►Model`** или комбинацию клавиш **`Ctrl+N`** или из меню **`File\Open...`** откройте готовую модель **`"*.mdl"`**. Появится окно модели с заголовком. С помощью команды **`Save as`** модель можно сохранить в виде файла с расширением **`mdl`**.

## 2.2. Демонстрационные примеры

Откройте, внимательно изучите модель, запустите симуляцию (процесс выполнения модели), проанализируйте результат симуляции для следующих демонстрационных примеров из справочника *Simulink* (команда **`Demos`**, раздел справки **`Help\Demos\Simulink`**):

- **`General Applications\Tracking a Bouncing Ball`** – прыгающий мячик

## 2.3. Движение тела в поле тяжести с учетом сопротивления воздуха

**Цель** задания – построить **S**-модель и исследовать движение тела, брошенного под углом к горизонту, с учетом сопротивления воздуха. Также необходимо ответить на вопрос, при каком угле бросания дальность полета будет достигать максимального значения, если учитывать сопротивление воздуха.

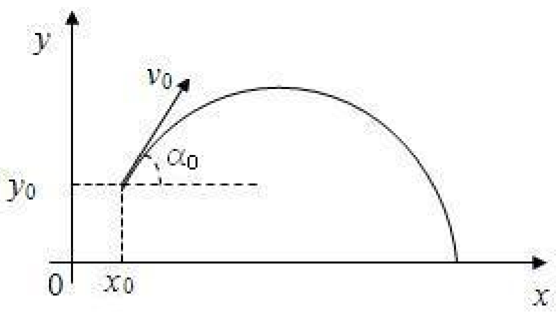



**Рис. 1.** Траектория полета тела, брошенного под углом к горизонту

### 2.3.1. Аналитическое исследование

Пусть тело массой $\boldsymbol{m}$ брошено под углом $\boldsymbol{\alpha}_0$ к горизонту с начальной скоростью $\boldsymbol{V}_0$. Требуется вывести уравнение движения тела, с учетом сопротивления воздуха, и построить соответствующую **`S`**-модель. **`S`**-модель должна вычислять положение тела в любой момент времени.

Исходные данные:

$\boldsymbol{m}$ – масса тела;

$\boldsymbol{V}_0$ – начальная скорость;

$\boldsymbol{R}_0$ ($\boldsymbol{x}_0$, $\boldsymbol{y}_0$) – начальные координаты;

$\boldsymbol{\alpha}_0$ – угол броска тела.

Полагаем:

- тело считаем материальной точкой массой $\boldsymbol{m}$, положение которой совпадает с центром масс тела;
- движение тела происходит под действием силы тяжести с постоянным ускорением свободного падения в плоскости, перпендикулярной поверхности земли, и описывается законами классической механики Ньютона.

Введём прямоугольную систему координат, как показано на рис.1. В начальный момент времени тело массой $\boldsymbol{m}$ находится в точке $\boldsymbol{R}_0$ ($\boldsymbol{x}_0$, $\boldsymbol{y}_0$). Вектор ускорения свободного падения $\vec{g}$ направлен вертикально вниз и имеет координаты (**0**, **-*g***). $\vec{v}_0$ – вектор начальной скорости. Разложим этот вектор по базису: $\vec{v}_0 = v_{0x}\vec{i} + v_{0y}\vec{j}$ . Здесь $v_{0x} = v_0 \cos\alpha_0, \cdots\cdots v_{0y} = v_0 \sin\alpha_0$ , $\boldsymbol{\alpha}_0$ - угол бросания.

Запишем второй закон Ньютона: $\vec{a} = \dfrac{\vec{F}}{m}$.

Ускорение в каждый момент времени есть (мгновенная) скорость изменения скорости, то есть производная от скорости по времени: $\vec{a} = \dfrac{d\vec{v}}{dt}$.

Следовательно, 2-й закон Ньютона можно переписать в следующем виде:

$$m\frac{d\vec{v}}{dt} = \vec{F},$$

где $\vec{F}$ – это равнодействующая всех сил, действующая на тело. Так как на тело действуют сила тяжести $m\vec{g}$ и сила сопротивления воздуха $\vec{F}_c$, то имеем:

$$m\frac{d\vec{v}}{dt} = m\vec{g} + \vec{F}_c. \tag{1}$$

Рассмотрим два случая:

1) Сила сопротивления воздуха равна 0: $\vec{F}_c = 0$.
2) Сила сопротивления воздуха противоположно направлена с вектором скорости, и её величина пропорциональна скорости: $\vec{F}_c = -k\vec{v}, \cdots k > 0$.

Составим ***математическую модель*** системы.

**Модель без учета сопротивления воздуха**

Рассмотрим **<u>первый случай</u>**, когда отсутствует сопротивление воздуха. Тогда из (1) имеем $m\dfrac{d\vec{v}}{dt} = m\vec{g}$, или

$$\frac{d\vec{v}}{dt} = \vec{g}\,. \tag{2}$$

Из (2) следует, что

$$\vec{v} = \vec{v}_0 + \vec{g}\,t\,. \tag{3}$$

Т.е. в отсутствии сопротивления воздуха скорость неограниченно увеличивается (равноускоренное движение).

Так как

$$\vec{v} = \frac{d\vec{R}}{dt}, \tag{4}$$

где $\vec{R}$ - радиус-вектор, то из (3) и с учетом (4) имеем: $\frac{d\vec{R}}{dt} = \vec{v}_0 + \vec{g}\,t$.

Отсюда получаем формулу закона движения тела при равноускоренном движении:

$$\vec{R} = \vec{R}_0 + \vec{v}_0\,t + \frac{\vec{g}\,t^2}{2}\,. \tag{5}$$

Запишем равенство (2) в скалярном виде:

$$\frac{dv_x}{dt} = 0, \cdots \frac{dv_y}{dt} = -g\,. \tag{6}$$

Согласно второму закону Ньютона и с учетом (4) и (6) дифференциальные уравнения движения в проекциях на оси $x$ и $y$ имеют вид

$$m\,\frac{d^2x}{dt^2} = 0, \quad m\,\frac{d^2y}{dt^2} = -m\,g, \quad v_x = \frac{dx}{dt}, \quad v_y = \frac{dy}{dt} \tag{7}$$

при следующих начальных условиях:

$$x(0) = x_0,\; y(0) = y_0,\; v_x(0) = v_0 \cos \alpha_0,\; v_y(0) = v_0 \sin \alpha_0. \tag{8}$$

Математическая постановка соответствует задаче Коши для системы обыкновенных дифференциальных уравнений с заданными начальными условиями.

Найдем зависимости $x(t)$, $y(t)$, $v_x(t)$, $v_y(t)$. Из (7) запишем систему ОДУ первого порядка:

$$\frac{dv_x}{dt} = 0, \quad \frac{dv_y}{dt} = -g, \quad v_x = \frac{dx}{dt}, \quad v_y = \frac{dy}{dt}. \tag{9}$$

После интегрирования системы (9) и с учетом начальных условий (8) получаем аналитическое решение

$$\begin{gathered} x(t) = x_0 + v_0 \cos\alpha_0 \cdot t, \quad y(t) = y_0 + v_0 \sin\alpha_0 \cdot t - \frac{g\,t^2}{2}, \\ v_x(t) = v_0 \cos\alpha_0, \quad v_y(t) = v_0 \sin\alpha_0 - g\,t, \end{gathered} \tag{10}$$

из которого следует, что полет тела, брошенного под углом к горизонту, при отсутствии сопротивления воздуха происходит по параболической траектории.

**Модель с учетом сопротивления воздуха**

Теперь рассмотрим **второй случай**, когда сила сопротивления воздуха противоположно направлена с вектором скорости, и ее величина пропорциональна скорости: $\vec{F}_c = -k\vec{v}, \cdots k > 0$.

В этом случае второй закон Ньютона имеет вид $m\dfrac{d\vec{v}}{dt} = m\vec{g} - k\,\vec{v}$, отсюда

$$\frac{d\vec{v}}{dt} = \vec{g} - \frac{k}{m}\vec{v}\,.$$

Запишем это равенство в скалярном виде:

$$\frac{dv_x}{dt} = -\frac{k}{m}v_x, \cdots \frac{dv_y}{dt} = -g - \frac{k}{m}v_y\,. \tag{11}$$

Имеем два линейных дифференциальных уравнения. Тогда система (9) для случая учета сопротивления воздуха перепишется в виде:

$$\frac{dv_x}{dt} = -\frac{k}{m}v_x, \cdots \frac{dv_y}{dt} = -g - \frac{k}{m}v_y, \quad v_x = \frac{dx}{dt}, \quad v_y = \frac{dy}{dt}. \tag{12}$$

Математическая постановка соответствует задаче Коши для системы обыкновенных дифференциальных уравнений (12) с заданными начальными условиями (8).

## 2.4. Пример: Прыгающий мячик

**Постановка задачи моделирования**

Создать «с нуля» **S**-модель движения мячика, брошенного со скоростью 15 м/с вертикально вверх с высоты 10 м над твердой горизонтальной поверхностью(в более сложном условии: мячик брошен под углом к горизонту). При моделировании учесть сопротивление воздуха (*Resistance*) и эластичность мячика (*Elasticity*). В качестве примера использовать демонстрационную модель **`General Applications | Bouncing Ball Model`**, приведенную в **`Simulink Demos`**.

**Входные параметры**

Мячик будем считать материальной точкой массой **m**, положение которой совпадает с центром масс мячика. Движение происходит строго перпендикулярно горизонтальной поверхности в поле силы тяжести с постоянным ускорением в соответствии со вторым законом Ньютона.

В качестве параметров, описывающих состояние системы, будем использовать высоту **h** и скорость **v** центра масс мячика.

**y(t)=h(t)** – положение мячика над поверхностью в момент времени *t*;

**v(t)** – вертикальная скорость движения мячика в момент времени *t*;

**y(0)=h(0) = 10 м** – положение в начальный (нулевой) момент времени;

**v(0) = 15 м/с** – скорость в начальный (нулевой) момент времени;

**m = 1 кг** – масса мячика;

**g = 9.81 м/с$^2$** – ускорение свободного падения;

**k = 0.005 кг/с** – коэффициент сопротивления воздуха;

**El = –0.8** – коэффициент восстановления мячика, равный отношению скорости после удара к скорости до удара о поверхность.

### Математическая модель

Определим закон вертикального движения материальной точки массой **m** под действием силы тяжести и силы сопротивления воздуха, если известны начальная координата точки $\mathbf{h_0}$ и ее начальная скорость $\mathbf{v_0}$.

Данную задачу можно рассматривать как движение тела, брошенного под углом $\alpha_0 = 90°$ к горизонту (т.е. ***вертикально вверх***), в поле тяжести с учетом сопротивления воздуха.

Тогда из (12), (8) закон изменения скорости движущейся материальной точки с учетом сопротивления воздуха можно описать задачей Коши

$$\frac{dv}{dt} = -g - \frac{k}{m}v, \qquad v = \frac{dy}{dt}, \cdots v(0) = v_0, \cdots y(0) = y_0. \tag{2.1}$$

### Simulink-модель

Откроем окно библиотеки блоков **Simulink**, из которой будем брать заготовки блоков для модели. Создадим новую **S**-модель в отдельном окне и сохраним ее в файле **Ball.mdl**.

**Шаг 1**. Настройка параметров конфигурации S-модели.

- Выберем пункт меню **Simulation | Configuration Parameters…** окна **S**-модели и в разделе **Solver** установим параметры:

  **Stop time = 20**;

  **Type = Variable-step**;

  **Solver = ode45**;

  **Max step size = 0.01**.

**Шаг 2**. Подключение интеграторов для определения скорости и положения мячика в текущий момент времени (рис.2).

- На первом этапе будем считать, что сопротивление воздуха отсутствует, т. е. в системе (2.1) коэффициент **k = 0**.
- Начнем с ускорения свободного падения – блок **Sources | Constant**. Назовем его **Gravity** и настроим его параметр **Constant value = 9.81**.

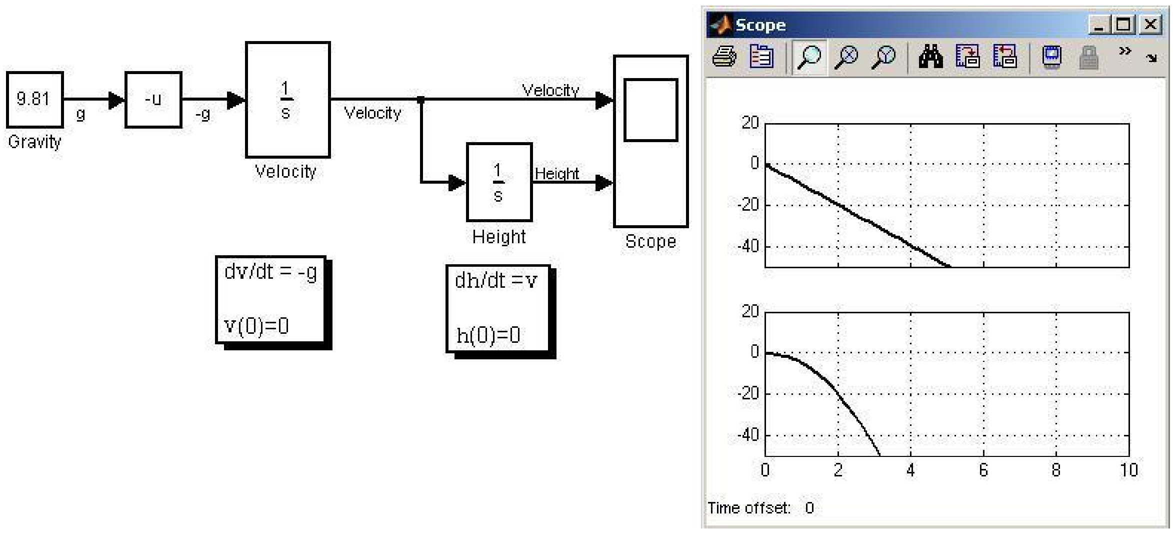



**Рис. 2.** Интеграторы, вычисляющие скорость и положение

- При помощи блока **Math Operations | Unary Minus** изменим знак ускорения свободного падения.
- Для моделирования скорости используем блок **Continuous | Integrator**. Назовем его **Velocity**. Соединим его вход с выходом блока **Unary Minus**.
- Для моделирования положения мячика используем блок **Continuous | Integrator**. Назовем его **Height**. Соединим его вход с выходом блока **Velocity**.
- Начальные условия **Initial Condition** в обоих интеграторах задаются внутри блоков и равны нулю, т. е. пока считаем **v(0) = 0**, **h(0) = 0**.
- Для контроля добавим в модель осциллограф **Sinks | Scope**, с помощью которого будем следить за изменением во времени интересующих нас параметров модели. Подсоединим его входы к выходам блоков **Velocity** и **Height**. Тем самым мы увидим, как изменяются скорость и положение мячика. Окно осциллографа можно открыть, дважды щелкнув по нему мышкой.
- Запустите модель, проанализируйте графики изменения скорости **v(t)** и положения **h(t)**, объясните полученный результат.

**Шаг 3**. Задание начальных условий вне интегратора (рис. 3).

- Добавим в модель два блока **Sources | Constant**, назовем их **v0** и **h0**. Установим их параметры **Constant value** равными **15** и **10** соответственно. На данном шаге эти блоки будут задавать начальную скорость **v(0)** и начальное положение **h(0)**.
- В настройках интегратора **Velocity** укажем, что начальное условие будет задаваться извне. Для этого установим параметр **Initial condition source = external**. У блока появится дополнительный вход $\mathbf{x_0}$, к которому подключим блок **v0**.
- Аналогично настроим параметры блока **Height**. К входу $\mathbf{x_0}$ подключим блок **h0**.
- Вновь запустите модель, проанализируйте графики изменения скорости **v(t)** и положения **h(t)**, объясните изменения.

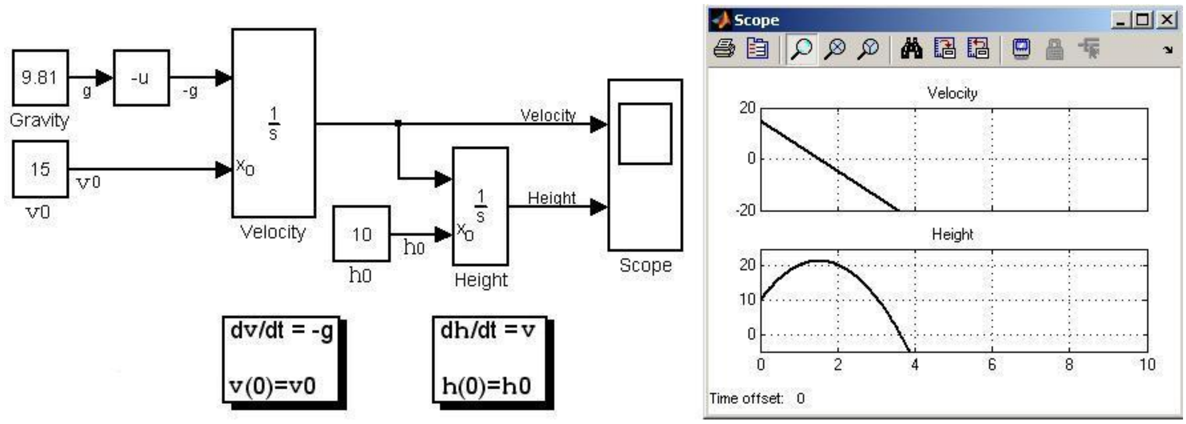



**Рис. 3.** Установка начальных значений **v0** и **h0**

**Шаг 4**. Моделирование удара мячика о поверхность (рис. 4).

- В блоке **Height** установим флажок **Limit output** и значение параметра **Lower saturation limit = 0**.
- В блоке **Velocity** установим параметр внешнего сброса **External reset = falling**. Тип внешнего управляющего сигнала обеспечивает сброс интегратора к начальному состоянию. На изображении блока появится дополнительный управляющий вход с условным обозначением типа управляющего сигнала ↯. Соединим данный вход с выходом блока **Height**.

- Теперь в модели учитывается удар мячика о поверхность, но его скорость после удара рассчитывается неверно: вместо изменения знака текущего значения скорости в момент удара, вновь задается начальное значение $\mathbf{v_0}$.

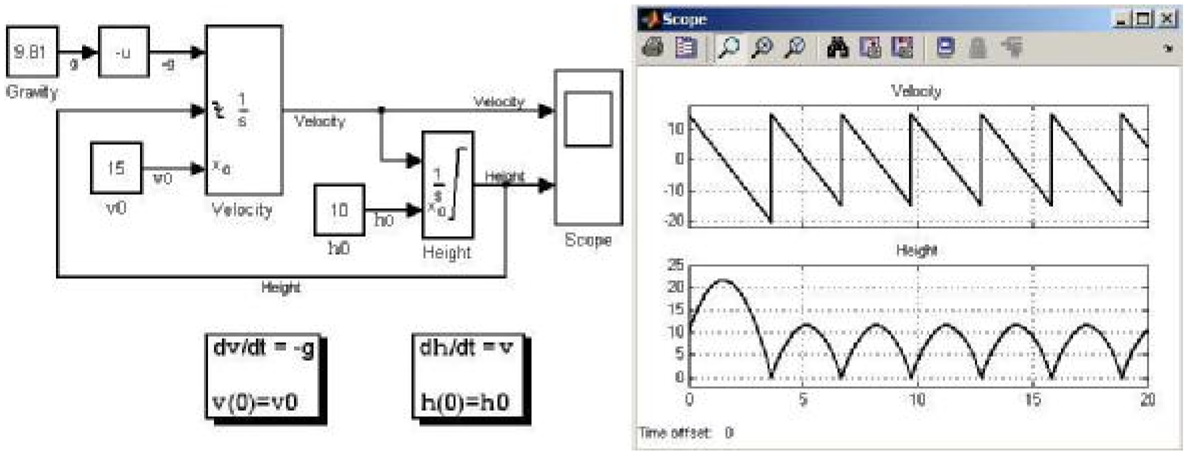



**Рис. 4.** Модель с учетом ударов мячика о поверхность

**Шаг 5**. Корректировка изменения скорости в момент удара о землю.

- Для задания ***правильной*** начальной скорости удалим из модели блок **v0** и подключим на вход $\mathbf{x_0}$ блока **Velocity** блок начальных условий **Signal Attributes | IC**. Назовем его **v0**. Установим значение параметра **Initial value = 15**.
- Добавим блок усилителя входного сигнала **Math Operations | Gain**, назовем его **Ball Elasticity** и установим его параметр **Gain = -1**. При помощи данного блока в дальнейшем будем изменять коэффициент восстановления мячика. Пока считаем мячик абсолютно упругим.
- В блоке **Velocity** установим флажок **Show state port**. Для изменения положений названия блока и нового выходного порта выделим блок **Velocity**, нажмем правую кнопку мыши и выберем пункт меню **Format > Flip Name**. Соединим появившийся выход с блоком **v0** через блок **Ball Elasticity**. Теперь текущее значение скорости в момент удара о поверхность изменяет знак (и величину, если мячик не является абсолютно упругим) и играет роль начальной скорости дальнейшего движения (рис. 5).

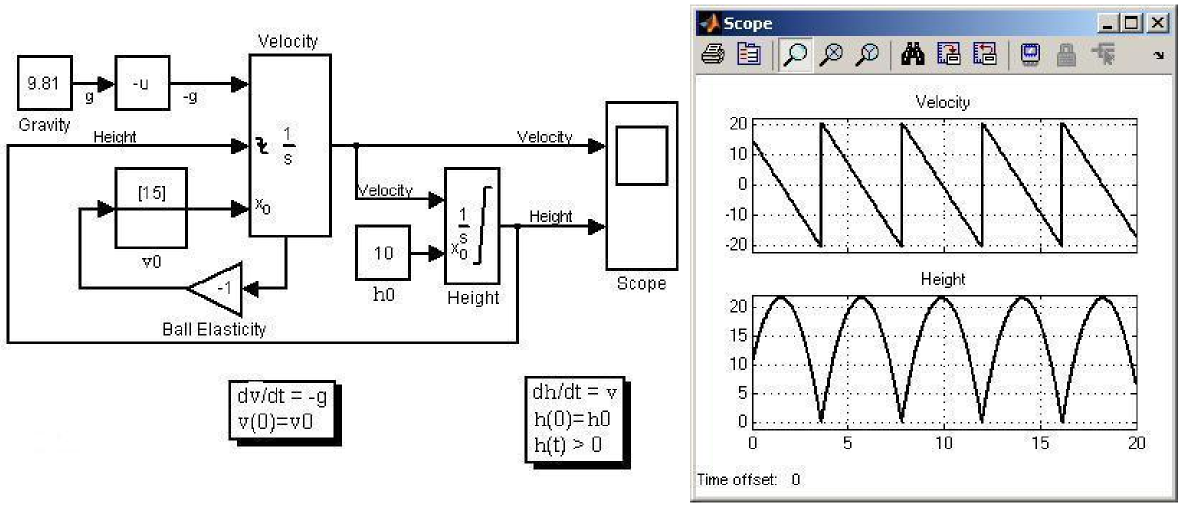



**Рис. 5.** Учет изменения скорости мячика в момент удара о поверхность

**Шаг 6**. Учет сопротивления воздуха и эластичности мячика (рис. 6).

- Заменим блок **Unary minus** на блок **Math Operations | Sum**, изменим его параметр **List of signs** на – –|.
- Добавим блок **Math Operations | Gain**. Назовем его **Resistance**. Через параметр **Gain** блока будем вводить в модель коэффициент сопротивления воздуха, деленный на массу мячика.
- Умножим выходной сигнал блока **Velocity** на значение блока **Resistance** и сложим результат с ускорением свободного падения **Gravity** при помощи блока **Sum**. Полученное со знаком «минус» значение подадим на вход блока **Velocity**.
- Установим значение коэффициента восстановления мячика **Ball Elasticity = –0.8**.
- Запустите **S**-модель и исследуйте полученные результаты.

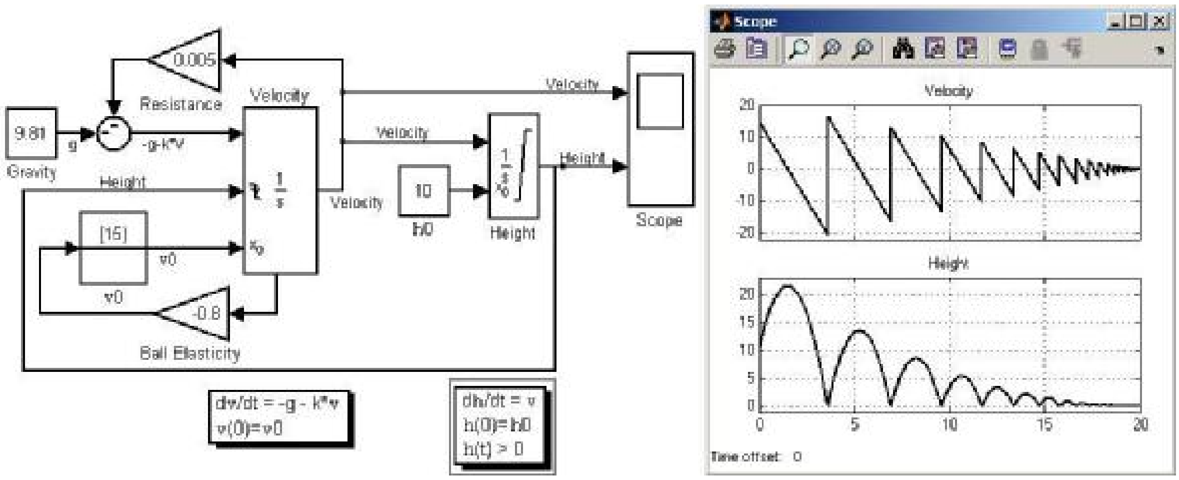



**Рис. 6. S**-модель прыгающего мячика

**Шаг 7**. Учет массы мячика. Самостоятельно измените модель для учета массы мячика согласно формуле (2.1).