

Scilab: Имитационное моделирование

То, чем мы собираемся заняться, называется компьютерным экспериментом в форме имитационного моделирования. Для этой цели в системе Matlab есть Simulink, а в Scilab – Xcos. Это инструмент для моделирования динамических процессов с помощью функциональных блоков и связей между ними. Проще говоря, математическая модель изучаемого объекта или процесса составляется из определенных блоков, а затем при помощи связей указывается, как данные перемещаются от одного блока к другому. Да, то же самое можно проделать, написав программу на встроенном языке, но если уже есть готовые блоки, так почему бы не воспользоваться ими? Кроме того, описание модели в виде диаграммы более наглядно и информативно.

Конечно, Xcos содержит не так много диаграмм, как Simulink, однако для моделирования многих объектов и процессов, а особенно для целей обучения он вполне подойдет.

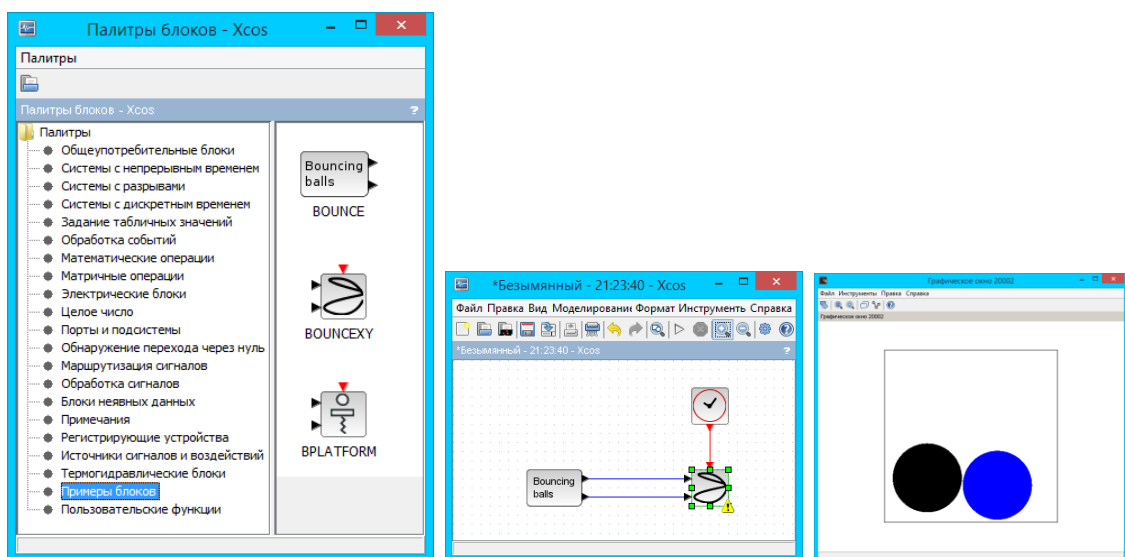


Рис. 1. Так выглядит графический редактор Xcos и древо палитр блоков.

Xcos содержит графический редактор, представляющий собой окно, в котором отображается диаграмма и отдельные окна палитр блоков. Все блоки сгруппированы в палитры. Вывести любую из них можно, выбрав последовательно *Инструменты > Визуальное моделирование Xcos*.

Блоки могут иметь различное число входов и выходов, называемых портами. Входной порт обозначается треугольником с углом, направленным в блок, а порт-выход – с углом, направленным из блока. Порты делятся на обычные (для ввода/вывода данных в/из блока), которые отображаются черным цветом, и активационные или управляющие (для ввода/вывода управляющей информации), отмечаемые красным цветом. Как правило, обычные порты расположены по бокам блока, а управляющие – сверху и снизу.

При составлении диаграммы вначале необходимо щелкнуть на желаемом блоке, а затем перенести его в окно диаграммы. Позиция блока не фиксирована, то есть впоследствии, если потребуется, вы сможете переместить блок, не нарушая связи и сохранив его атрибуты. Блоки также можно поворачивать и отражать: чтобы сделать это, щелкните на блоке правой кнопкой мыши и выберите в появившемся подменю *Формат* один из первых трех пунктов (*Повернуть, Зеркало, Отразить*).

После переноса блоков их необходимо соединить между собой. Для создания связи следует протянуть линию от треугольника выходного порта одного блока до треугольника входного порта другого. При этом будет создана прямая линия. Если вам необходимо создать ломаную, то каждую ее точку можно определить левым щелчком в пустой области, как при рисовании многоугольника. Каждый входной порт соединяется лишь с одним выходным портом. Если вам необходимо провести линии от одного блока ко многим, например, передать одно и то же значение на вход нескольких расчетных блоков или синхронизировать несколько генераторов, то сначала проводится одна связь, а затем от ее линии создаются ответвления. Для этого следует выполнить двойной щелчок в каком-либо месте связи и провести дополнительную ветвь. Альтернативный вариант – использовать блоки MUX и DEMUX (*Маршрутизация сигналов*).

Существует два вида связей: для передачи данных (черного цвета) и для передачи управляющей/временной информации (красного цвета). Пользователь может изменить цвета связей, но не цвет входа/выхода блока: он четко указывает необходимый тип связи. Следует также помнить, что обычный и управляющий порты соединить нельзя.

Почти все блоки имеют изменяемые параметры, определяющие обработку информации или поведение. Чтобы открыть окно с перечнем параметров, достаточно дважды щелкнуть на блоке левой кнопкой мыши.

Прежде чем переходить к созданию диаграммы, следует построить математическую модель, то есть записать уравнение или систему уравнений, описывающих процесс, а также определить начальные условия и внутренние константы. Xcos сможет решить поставленную перед ним задачу только при условии, что вы сами понимаете ее.

Рассмотрим такой пример: в коробке находится несколько упругих шариков, и в какой-то момент времени мы начинаем наблюдать за ними. Для этого случая модель уже построена и включена в демонстрационные примеры, поэтому в редакторе Xcos выберите последовательно пункты меню *Справка > Демонстрации Xcos > Xcos > Стандартные демонстрационные примеры > Скачущие шары*. После этого в редакторе откроется диаграмма, показанная на рис. 1.

Первый (имеющий вид красных часов) – это стандартный управляющий блок, без которого не обходится практически ни одна диаграмма. Он генерирует регулярные события, с заданным пользователем периодом. В данном случае он отсчитывает время, то есть подает команду на пересчет системы и отрисовку текущего состояния через определенные интервалы. Этот блок имеет всего два параметра: *Period* – период генерации событий и *Init time* – начальное значение времени.

Блок, с которым соединены часы, представляет собой инструмент анимации, подстроенный специально для данной задачи. У него много параметров, и большая их часть задается векторами, размерность которых совпадает с числом шариков. Они определяют цвета шариков, их радиусы и границы коробки.

Ну, а третий блок осуществляет расчеты и выдает вектора координат. У него тоже много параметров, которые задают массу, радиус шариков, коэффициент аэродинамики, ускорение свободного падения...

Запустите модель на выполнение (*Моделирование > Запустить*), и вы увидите, что происходит в неподвижной коробке с несколькими шариками. Хотите почувствовать разницу между Землей, Луной и Юпитером? Измените параметр *g* (*gravity*) и посмотрите,

как он влияет на поведение системы. Чтобы остановить выполнение, нажмите на красную кнопку *Остановить* в редакторе диаграмм (заметьте – не в окне с анимацией).

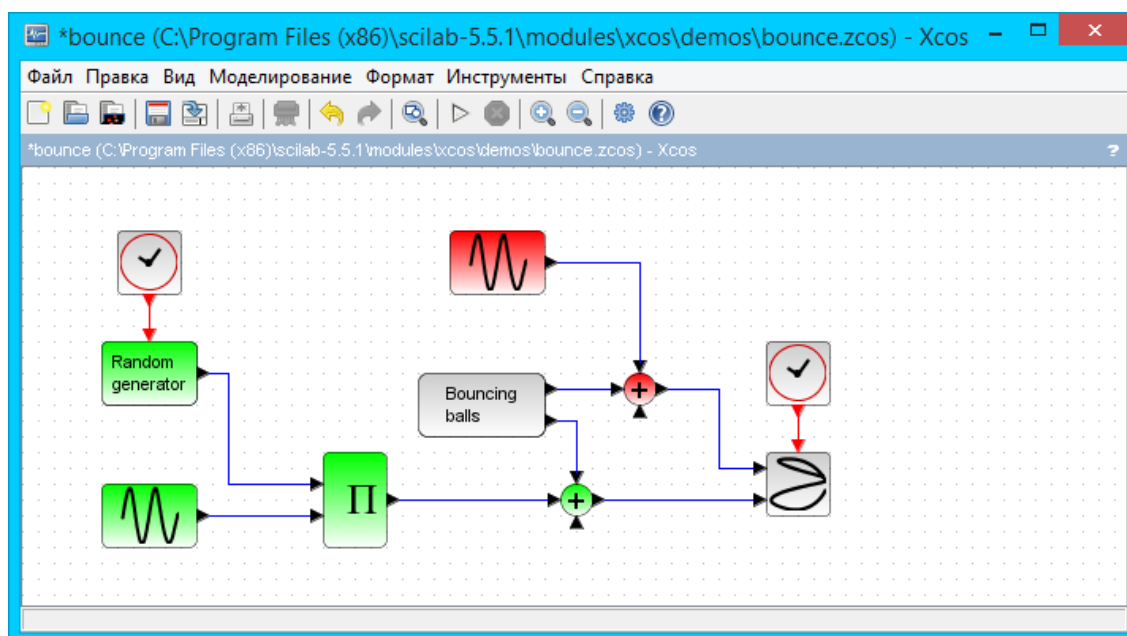


Рис. 2. Измененная диаграмма с заданным смещением коробки.

Пусть в нашей задаче будет всего один шарик и коробка должна двигаться. Попробуем изменить данную модель в соответствии с новыми условиями: коробка (и шарик) находятся в поле тяготения и коробка совершает синусоидальные колебания вдоль горизонтальной оси, а также периодически со случайной амплитудой вдоль вертикальной оси.

Добавим к открытой диаграмме описание смещения коробки, то есть синусоидальное изменение x-координаты шарика и синусоидальные изменения со случайной амплитудой y-координаты. Посмотрите результат на рис. 2.

Блок с красным фоном описывает смещение коробки по оси x и представляет собой генератор синусоидального сигнала с палитры *Источники сигналов и воздействий*. У него есть три параметра: *Абсолютная величина* – максимальная по модулю величина сигнала, в нашем случае это максимальное смещение коробки (установите 2); *Частота* – частота колебаний (выберем здесь значение 15); и *Фаза* – начальная фаза колебаний (укажите 0).

Блоки с зеленым фоном задают смещение коробки по вертикали. Первый из них – это просто генератор событий (красные часы), который используется для выдачи случайного значения амплитуды колебаний (установите значение *Period* в 0.2). Далее идет сам блок-генератор случайных чисел (*Random generator*). У него есть несколько интересных нам параметров, например, *flag* – флаг, определяющий как использовать параметры A и B:

- если $flag=0$, то A – это минимум, а максимальное значение равно $A + B$;
- если $flag=1$, то A – это среднее, а B – отклонение от среднего.

Установите *flag* равным 1, $A=0$, $B=0.5$. Ниже расположен знакомый нам генератор синусоиды. Укажите в нем следующие значения параметров: *Абсолютная величина* – 0.5, *Частота* – 5. Затем случайное значение амплитуды умножается на периодический сигнал (в блоке *PRODUCT* панели *Математические операции* установить параметр [1 1]) и прибавляется к координате y.

Теперь нам осталось задать параметры расчета смещения шарика. Дважды щелкаем на блоке *Bouncing balls* и устанавливаем следующие значения: $Mass=[1]$, $Radius=[1]$, $[xmin, xmax, ymin, ymax]=[-7;7;0;15]$, $xpos=0$, $xdpos=5$, $ypos=10$, $ydpos=10$.

Запустите модель на выполнение. Создание более точной модели требует детального анализа системы и вывода уравнений её состояния (не считая сборки диаграммы из блоков).

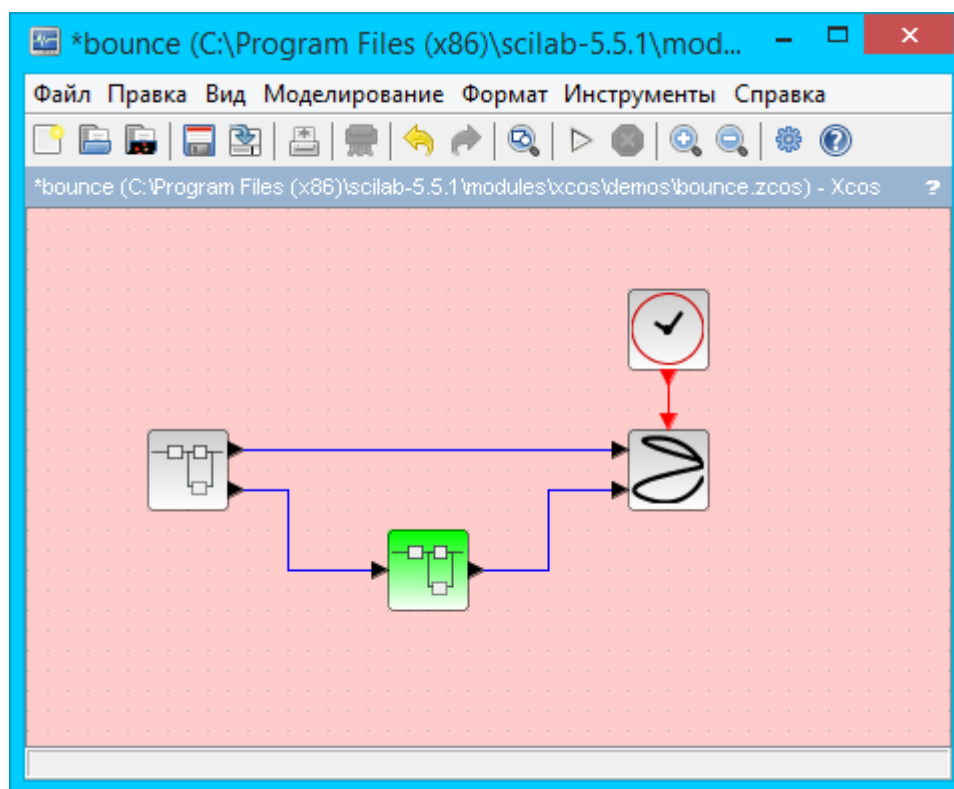


Рис. 3. Главная диаграмма с суперблоками.

Диаграммы не всегда можно уместить на одном листе: зачастую они бывают достаточно объемны, и их трудно отлаживать, а также обозревать. Некоторые блоки можно логически объединить в большие суперблоки, со своими входами и выходами. Для этой цели в Xcos существует специальный блок с именем *SUPER_f* (*Порты и подсистемы*). На рис. 3 приведен наш пример диаграммы для моделирования шарика в движущейся коробке, где части расчета координат x и y для шарика упакованы в суперблоки.

Создать суперблок можно двумя способами: обычным, т.е. просто перенести в диаграмму и затем наполнить его содержимым, и путем преобразования уже имеющейся части диаграммы. Последнее выполняется следующим образом: выделяем несколько блоков, а затем последовательно выбираем пункты меню *Правка > Selection to superblock*.

Чтобы увидеть и изменить содержимое суперблока, достаточно щелкнуть дважды на его иконке. При этом запускать модель на выполнение можно только из главного окна.

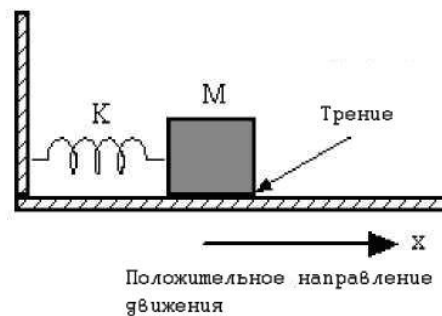
Каждый суперблок может содержать определенное число входных и выходных портов – как для данных, так и для управляющих сигналов. Таким образом, суперблок – это своего рода подпрограмма, и его можно отлаживать отдельно от основной части диаграммы, а затем вставить в нее уже готовым.

При работе с суперблоками входной порт данных определяется блоком IN_f (палитра *Порты и подсистемы*), а выходной – OUT_f. Для управляющих сигналов используются CLKINV_f и CLKOUTV_f. Каждый входной или выходной порт суперблока должен иметь уникальный для своей группы номер, т.е. все входные порты данных нумеруются с 1, все выходные также нумеруются с 1, и т.д.

Пример решения задачи

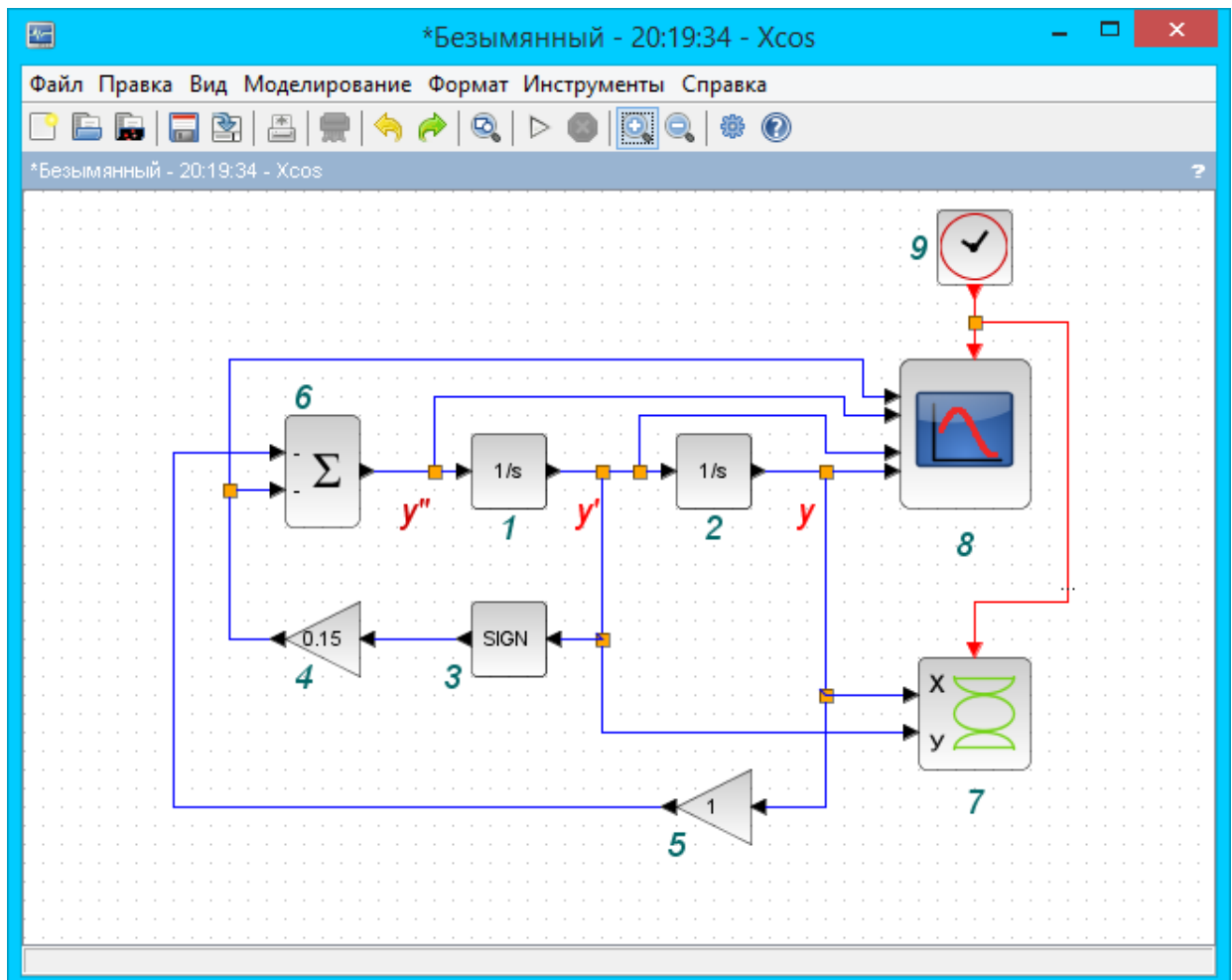
Блок с массой M перемещается по грубой поверхности с коэффициентом трения f . Постоянная сила сопротивления – $f \cdot M \cdot g$, где g – сила тяготения, и K – коэффициент упругости пружины.

Покажем движение блока на графике, если его перемещают на 1 метр вправо и отпускают.

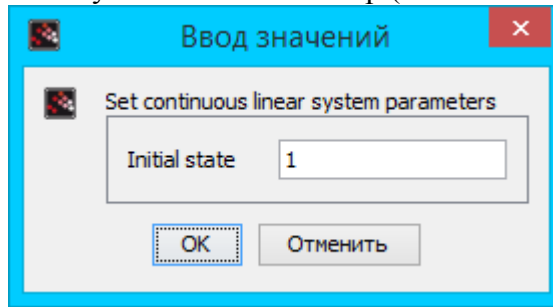


Такое движение можно описать уравнением:

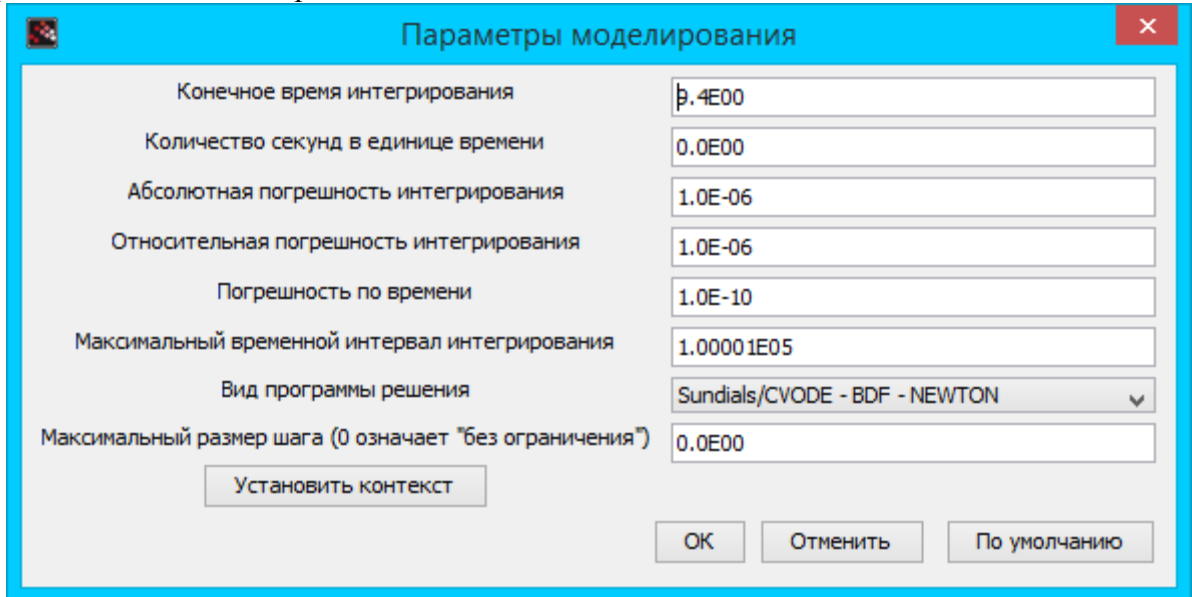
$$y'' = -y - 0.15 * \text{sign}(y')$$



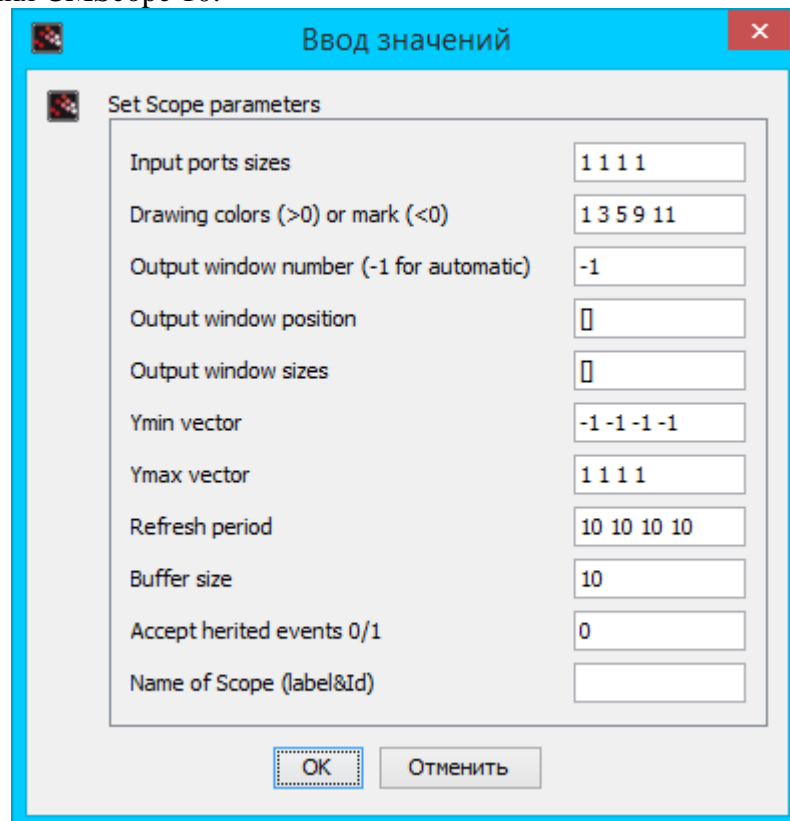
Начальное условие для положения установлено в 1 метр (в блоке интегратора 2):



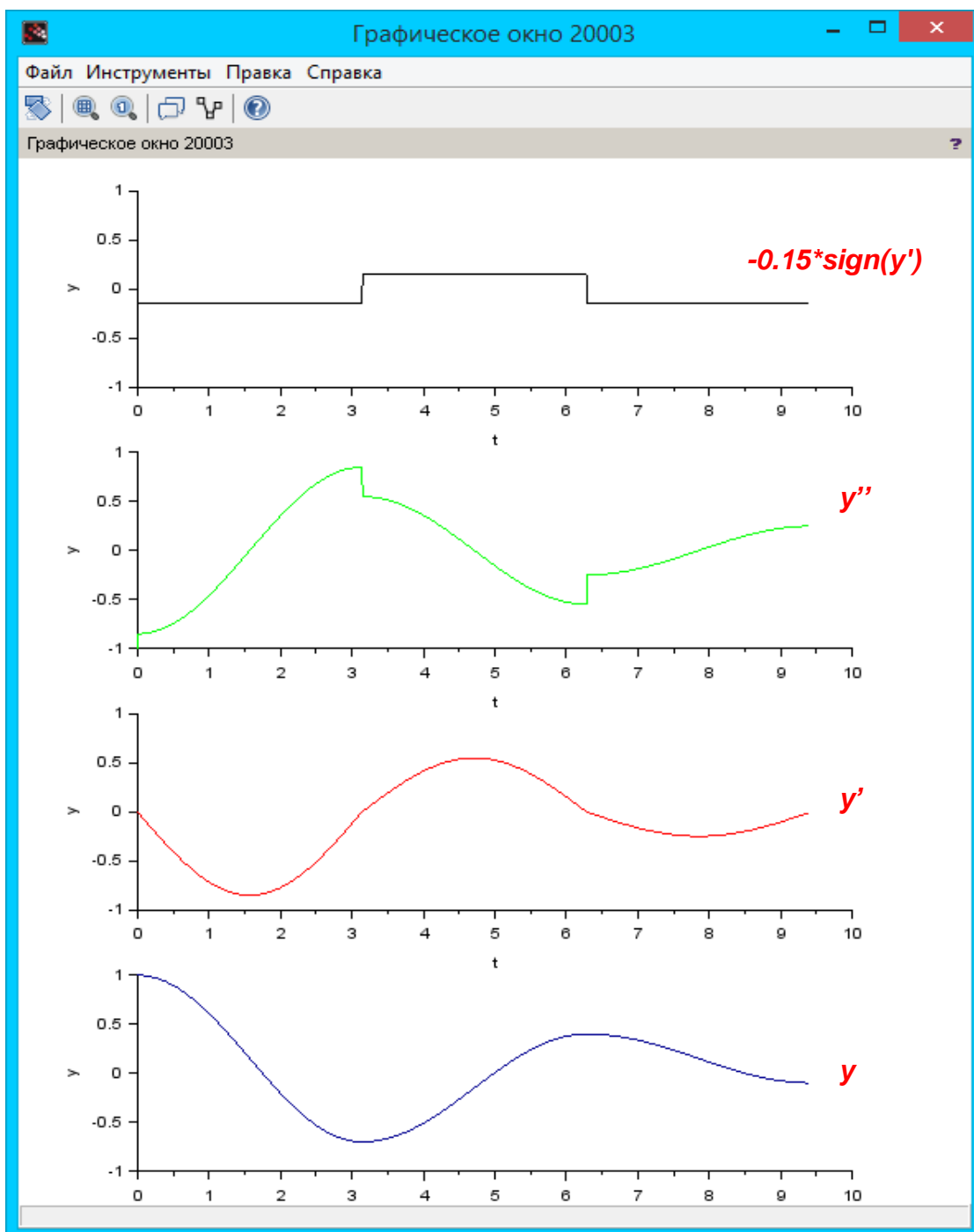
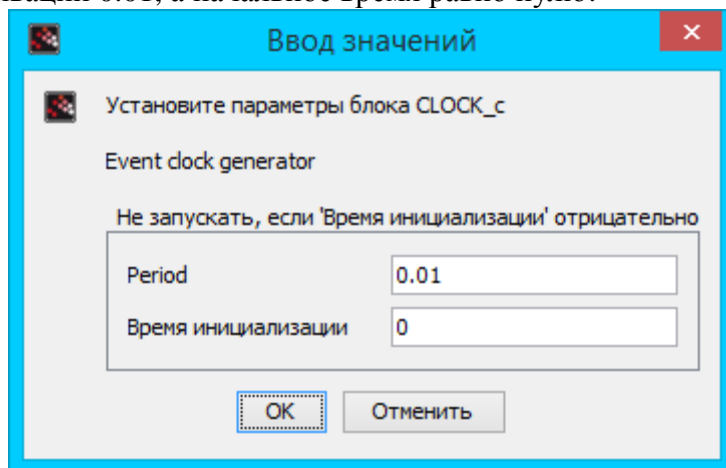
Время окончания интеграции 9.4:

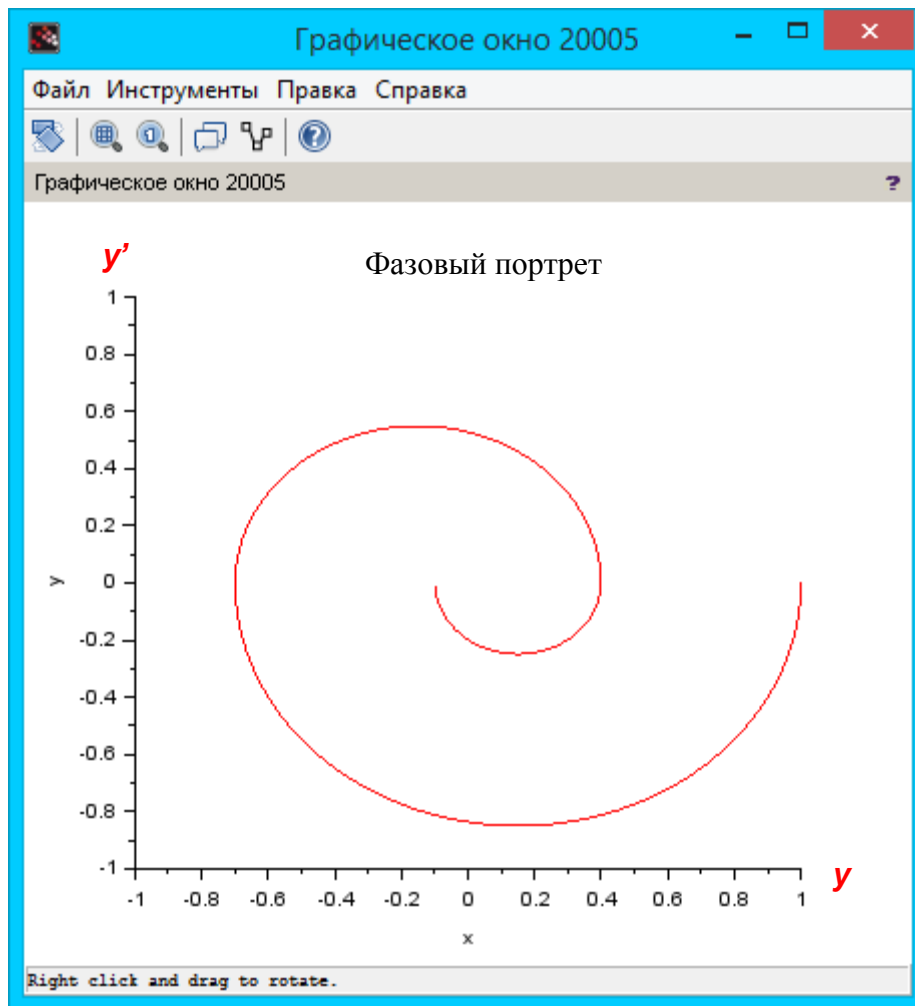


Время обновления CMScore 10:



Период часов активации 0.01, а начальное время равно нулю:





Комментарии:

Фазовый портрет отражает начальные величины $y = 1$ и $y' = 0$. Скорость в начале равна нулю, а ускорение отрицательно, таким образом, блок начинает перемещаться влево. Это означает, что постоянная сила трения, задаваемая блоком *б*, должна быть положительна и это так и есть, так как она отрицательна на входе (знак минус на входе блока *б*).

Если установить время интеграции больше, чем физически возможное время (здесь $3 \cdot \pi$ секунды) то программа входит в бесконечный цикл и после этого придется остановить модель вручную (кнопкой *Остановить* в редакторе диаграмм).