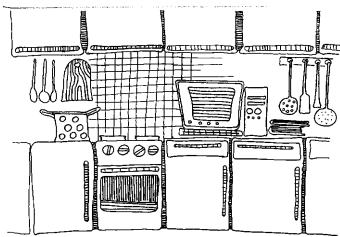


*Колесов Юрий Борисович  
Сениченков Юрий Борисович*

## КАК БЫСТРО И КАЧЕСТВЕННО "ПРИГОТОВИТЬ" ИНТЕРАКТИВНУЮ МОДЕЛЬ ДИНАМИЧЕСКОЙ СИСТЕМЫ



“Кухня в семейном быту представляет весьма существенный вопрос, для людей среднего достатка и вообще с ограниченными средствами, которые не могут держать повара или хорошо знающую свое дѣло кухарку.”

ПОЛНОЕ РУКОВОДСТВО для правильного ведения домашнего хозяйства, составленное Малоховской.  
С-Петербург, Издание П.А. Федорова, 1909

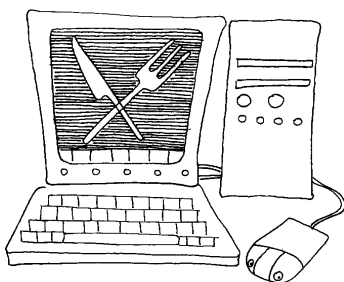
Существует два широко распространенных способа самому разработать интерактивную программную реализацию математической модели динамической системы. Первый. Самому пригласить профессионалов, заплатить им свои деньги и использовать то, что получится, по своему усмотрению. Второй. Выбрать простой с точки зрения обучения и использования программный продукт, освоить его и самостоятельно изготавливать нужные Вам приложения.

Цель настоящей статьи - убедить Вас, что программный комплекс MODEL

VISION 2.1, первый из семейства комплексов MODEL VISION, с успехом решает три повседневные для каждого преподавателя проблемы:

- подготовки сложного иллюстративного материала для урока,
- создания серии компьютерных лабораторных работ, выполняемых по схеме: “выбери нужные значения параметров” - “наблюдай поведение” - “объясни увиденное”,
- разработки сложных приложений с элементами исследования.

## ТЕХНОЛОГИЧЕСКАЯ ЦЕПОЧКА



“Для приготовления пищи, какъ бы ни была мала и скромна квартира, всегда отводится небольшая комната, называемая кухней. Въ ней устанавливается печь, обыкновенно плита, сложенная изъ кирпича и облицованная кафелемъ. Верхъ плиты состоитъ из...”

ПОЛНОЕ РУКОВОДСТВО для правильного ведения домашнего хозяйства, составленное Малоховской.  
С-Петербург, Издание П.А. Федорова, 1909

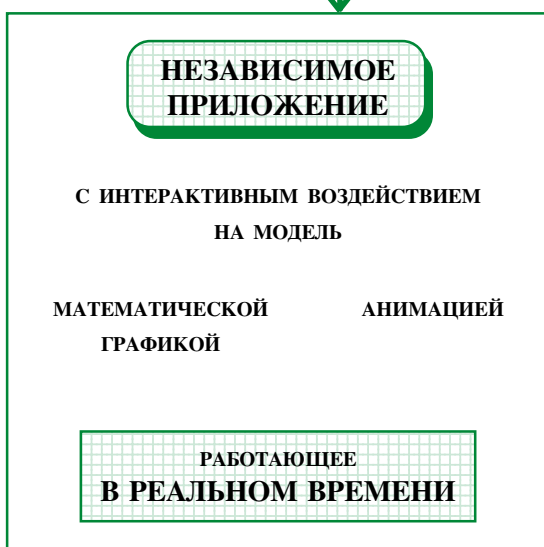
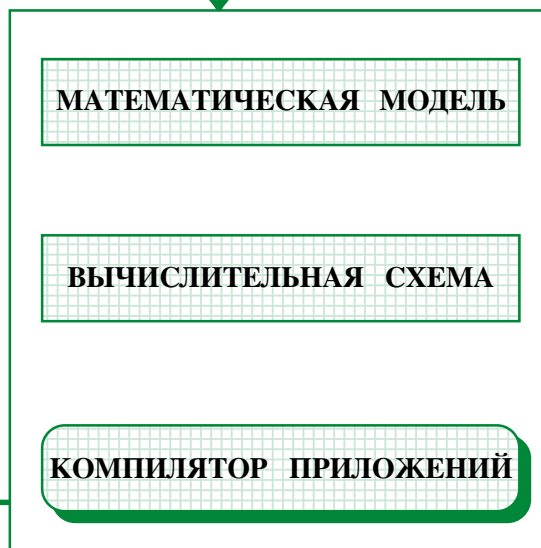


1. Графический язык Пользователя - разработчика приложений - содержит три компонента: язык функциональных блок-схем, язык карт состояний и язык для описания дифференциальных уравнений и функций поведения (MVW Editor).

2. Отдельно реализованная компонента - редактор Анимаций позволяет разрабатывать Динамические графические образы изучаемых объектов (MVW Animator).

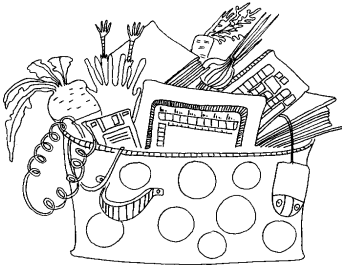
3. Описание Пользователем Структуры, Поведения моделируемого объекта и выбора Решателя Уравнений (из числа имеющихся в списке решателей) позволяет Пакету автоматически построить приложение.

4. При генерации кода выполняемой модели используется Borland Pascal. Созданное приложение может работать либо под управлением пакета, либо независимо от него.



5. Работающее независимо от MVW 2.1 приложение сохраняет возможность менять по ходу эксперимента параметры модели в ручном и автоматическом режимах, модифицировать окно вычислительного эксперимента, убирая или добавляя окна графиков, динамических таблиц, фазовых диаграмм и, если это необходимо, наблюдать за поведением модели в пошаговом режиме.

## ПОСТРОЕНИЕ МОДЕЛИ.



“В состав бульона входят: говядина, телятина, куры и рябчики, ребе ветчина. Что касается свинины, утокъ, гусей, баранины, поросенка и дичи, то бульонъ изъ нихъ менее вкусенъ.”

ПОЛНОЕ РУКОВОДСТВО для правильного ведения домашнего хозяйства, составленное Малоховской.  
С-Петербург, Издание П.А. Федорова, 1909

Установка MODEL VISION. MVW 2.1 поставляется в двух модификациях - в виде “русской” или “английской” версий.

“Русскую” версию рекомендуется использовать тем, чьи компьютеры работают под управлением русифицированной Windows.

Пользователям, получившим дискеты от разработчика, достаточно установить первую дискету и щелкнуть мышью по файлу “Install”. Сообщение об успешном завершении установки вашей программы говорит о том, что можно приступать к моделированию.

Пользователям, получившим пакет по сети INTERNET (<http://www.xjtek.com> или <http://dcn.nord.nw.ru/MV> или <http://www.hotfiles.com>), следует расположить оба файла DISK1 и DISK2 во временном каталоге (папке) и щелкнуть мышью по файлу “Install” (DISK1).

Простейшая модель. Математический маятник описывается дифференциальным уравнением

$$\frac{d^2\varphi}{dt^2} = -g/L * \sin(\varphi),$$

$$\varphi(0) = \varphi_0;$$

$$\frac{d\varphi}{dt} = \varphi'(0),$$

$g=9.81$ ;  $L$  - параметр. Начальные условия, как и длину маятника, будем считать параметрами, или входными величинами.

Описание структуры. Определение “простейшая” в данном случае означает, что наш объект моделирования рассматривается нами как объект простой структуры, то

есть не имеющий внутренних компонент. Функциональная блок-схема таких объектов представляется одним единственным прямоугольником, назовем его “Маятник”.

Вызовите программу MVW Editor. В меню действий выберите последовательно команды “Файл”, “Новый проект”. Перед Вами откроются два окна, одно из которых, а именно “Список Устройств”, будет содержать прямоугольник с вписанной в него буквой М (Main) и расположенным рядом стандартным названием “Device\_1”(Устройство\_1). Буква М обозначает, что данное устройство (пакет первоначально предназначался для инженеров, разработчиков аппаратных комплексов, поэтому все объекты моделирования называются устройствами) в данном проекте считается главным и расположено на верхнем уровне иерархии. В нашем случае оно единственное и поэтому главное. На будущее, если устройств несколько и Вам надо “сменить лидера”, выполните команду “Модель”, “Главное Устройство” и следуйте указаниям появившегося диалогового окна.

Прежде всего дадим нашему устройству осмысленное название “Маятник”. Выполним команды “Устройство”, “Изменить” и заполним поле “Идентификатор”. По умолчанию наше устройство уже отнесено пакетом к типу элементарных.

Для описания нашего объекта нужны переменные: время -  $t$ , угол -  $\varphi$ , производная угла по времени -  $d\varphi/dt$ , константа ускорения силы тяжести -  $g$ . Предположим, что помимо угла и его производной (переменные состояния) мы хотим вычислять и использовать как входные величины каких-

либо других устройств кинетическую энергию -  $EK=1/2 d\phi/dt$  и потенциальную  $V=g/L*(1-\cos(\phi))$  (выходные переменные). Предусмотрим возможность менять начальные значения угла и его производной, то есть будем считать начальные значения входными параметрами.

Среди всех перечисленных переменных одна, а именно время, играет в пакете особое значение. Пакет MODEL VISION самостоятельно моделирует течение непрерывного времени, для этого предусмотрена специальная переменная STime. В любом новом проекте главное устройство, создаваемое по умолчанию, с типовым названием "Device\_1", уже способно жить в этом непрерывном времени, демонстрируя поведение "ничего-не-делаю". Попробуйте прямо сейчас, еще не приступая к собственно проектированию, выполнить последовательность команд "Модель", "Создать и запустить" и (после компиляции модели) "Пуск". Вы увидите окно с образом вашего устройства и макетом часов, которые после щелчка по полю команды "Пуск" начнут отсчитывать время.

Итак, мы разделили наши переменные на **входные**:  $L$  - длину маятника,  $fi0$  - начальное значение угла,  $fip0$  - начальное значение производной угла, в их число входит константа -  $g$ ; **выходные**: кинетическую энергию  $EK=0.5*fip$ , ( $fip = d\phi/dt$ ) и потенциальную  $V=g/L*(1-\cos(fi))$ , ( $fi=\phi$ ); переменные состояния:  $fi$  (угол) и  $fip$  (производная угла).

Перенесите эту информацию на блок-схему. Установите курсор на прямоугольник, изображающий ваше устройство, и щелкните мышью. (Этому соответствует последовательность команд "Устройство", "Редактировать"). Перед вами откроются четыре новых окна - "Структура", "Интерфейс", "Процессы" и "Состояние". Перемещаясь

по окнам и делая их активными (установите курсор внутри окна и щелкните мышью), вы получаете возможность описывать входные и выходные переменные и константы ("Интерфейс"), переменные состояния ("Состояние"). Каждое окно имеет свою командную строку. Первое окно ("Структура") позволяет отображать и редактировать структуру устройств. Если устройство составное, то щелчок мыши внутри прямоугольника, изображающего внутреннее по отношению к данному устройству, "опустит" вас на один уровень иерархии вниз. Для того чтобы вернуться на покинутый уровень, необходимо сделать активным окно "Структура" и выполнить команды "Структура", "Вверх по иерархии структуры".

Второе окно ("Интерфейс") предназначено для описания входных и выходных переменных. Сделайте активным окно "Интерфейс". Выполните команду "Компонента" и выберите подкоманду "Новая". Заполните поля появившейся панели.

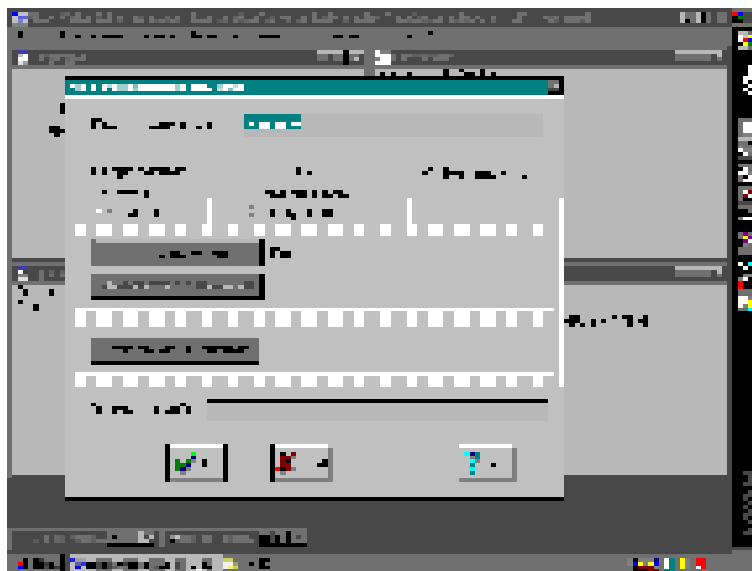


Рисунок 1.

При заполнении полей учтите, что, помимо деления переменных на входные, выходные и переменные состояния, каждая из переменных характеризуется типом и характером поведения во времени (Таблицы 1, 2).

Имя поля в окне	Тип переменной
REAL	Вещественная
INTEGER	Целая
BYTE	Байт
WORD	Слово
LONGINT	“Большие” целые числа
BOOLEAN	логическая
STRING	строка
ENUMERATED	перечислимая

Таблица 1.

Имя поля в окне	Характер поведения
Потенциальная	Определена для любого типа во всех временных точках отрезка “Время моделирования”
Импульсная	Принимает значение только перечислимого типа и только во временных точках, где ей присваивается новое значение. Во всех остальных точках отрезка “Время моделирования” не определена.

Таблица 2.

В результате описания всех переменных и констант мы получим (рис. 2):

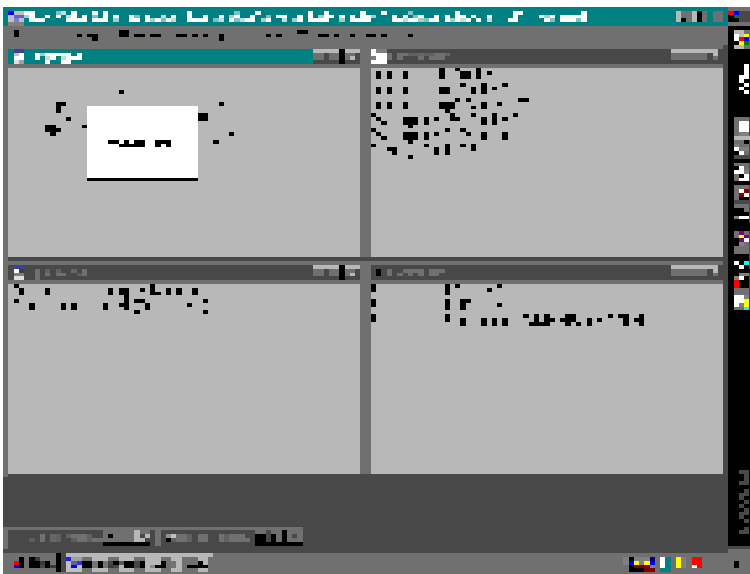


Рисунок 2.

Описание поведения. Поведение объекта моделирования, представленное с помощью функций, определенных в каждой временной точке, или же дифференциальных уравнений, решение которых определено опять же на заданном временном интервале, обычно называют непрерывным

поведением (то есть определенным на непрерывном времени). Интуитивно непрерывное поведение связывается

с непрерывной изменчивостью наблюдаемой величины. В то же время существуют процессы, где наблюдаемая величина, приняв определенное значение, сохраняет его на длительном временном промежутке. Более того, функция может принимать лишь небольшое количество различных значений заданного типа, например два (0 или 1) в зависимости от выполнения некоторых условий. Описать такую функцию можно, используя понятия состояния и события. Будем говорить, что функция, принимающая значение 0 на некотором

отрезке времени, находится в состоянии ОДИН, а если ее значение изменяется на 1 вследствие наступления некоторого события, то она переходит в состояние ДВА. Остается описать условия перехода функции из состояния ОДИН в состояние ДВА и обратно (определить события, приводя-

щие к смене состояния), а также указать начальное состояние. Непрерывное время как бы исчезло из такого описания, и, наблюдая за функцией, мы видим, как она при выполнении условий перехода скачком принимает то одно значение, то другое. Функция демонстрирует дискретное поведение. Так ведет себя светофор, автомат, наливающий газированную воду, реле.

В MVW 2.1 для описания непрерывного и дискретного поведения вводятся непрерывные и дискретные процессы.

Непрерывные процессы. Непрерывные процессы в MVW 2.1 могут быть представлены системами дифференциальных уравнений первого порядка и функциями от времени. Наше исходное уравнение второго порядка придется переписать в виде системы двух уравнений первого порядка

$$\begin{cases} \frac{df_i}{dt} = f_{ip} \\ \frac{df_{ip}}{dt} = -g / L * \sin(f_i) \end{cases}$$

и дополнить двумя функциями

$$EK = \frac{1}{2} (f_{ip})^2$$

$$V = g / L * (1 - \cos(f_i))$$

Сделайте активным окно “Процессы”. Выполните последовательно команды “Процесс”, “Новый”. В появившемся диалоговом окне подтвердите, что вы имеете дело с непрерывным процессом, и вы попадете в окно Редактора Дифференциальных уравнений и Функций. Окно содержит два независимых дочерних окна “Дифференциальные уравнения” и “Функции” (соответствующее

поле носит название “алгебраические уравнения”, так как предполагалось, что их действительно можно будет задавать). Последовательно, делая активным нужное окно, заполните поля окна “Дифференциальное уравнение” и “Функции”, как показано на рис. 3

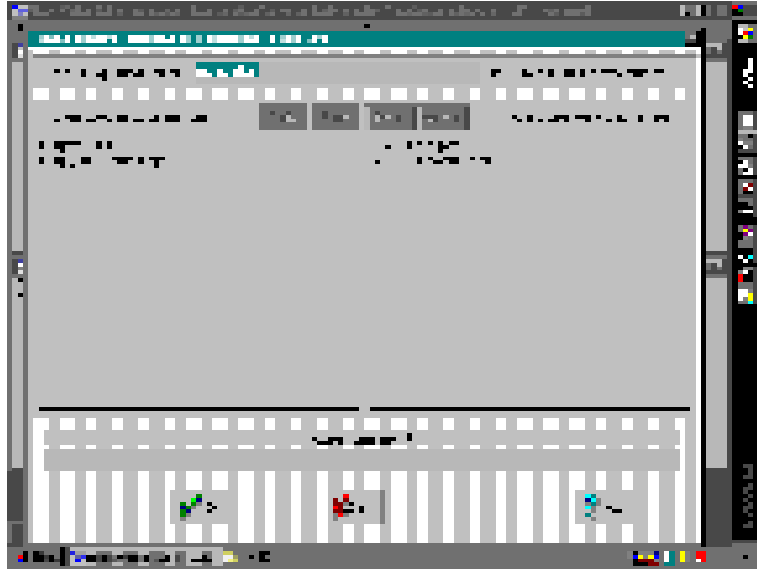


Рисунок 3.

Дискретные процессы. Для данной модели введение дискретного процесса носит несколько искусственный характер, но вполне приемлемо для наших учебных целей. Обратите внимание, у нас есть переменные состояния  $f_i$  и  $f_{ip}$  и переменные  $f_{i0}$  и  $f_{ip0}$ . Последние, хоть и описаны, имеют отличные от нуля ( $f_{i0}$ ) значения, но не используются для нахождения решения. Действительно, они ведь никак не связаны с переменными  $f_i$  и  $f_{ip}$ . Попытка присвоить значения  $f_i=f_{i0}$  и  $f_{ip}=f_{ip0}$  при описании непрерывных процессов приведет к абсурду с точки зрения Построителя Модели, так как с одной стороны  $f_i$  и  $f_{ip}$  - решения дифференциальных уравнений, а с другой - вы требуете, чтобы их значения всегда были бы постоянными и равнялись бы  $f_{i0}$  и  $f_{ip0}$ , соответственно. Во избежание противоречия опишем дискретный процесс “INIT” - “Присваивание начальных значений”. В

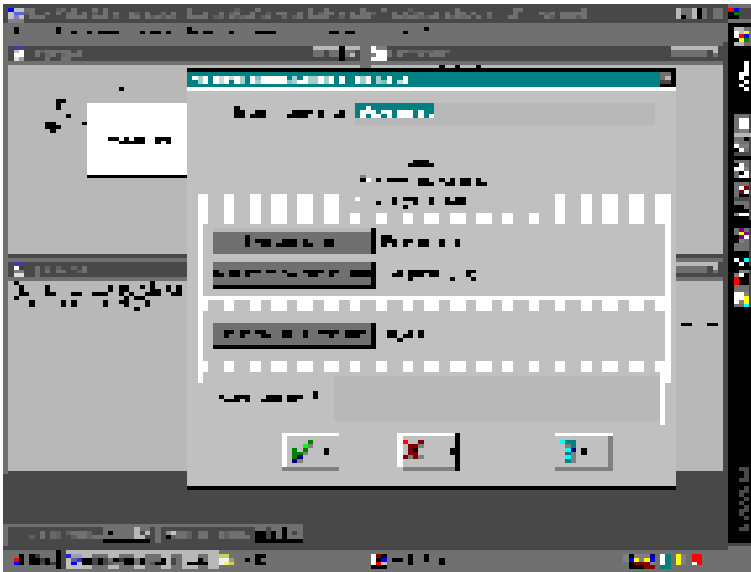


Рисунок 4.

некотором смысле это вырожденный процесс - нам даже не нужна функция, меняющая свои значения в состояниях ОДИН (до начала моделирования) и ДВА (на всем временном промежутке моделирования). Важно, что в момент переключения ( $Stime \geq 0$ ) мы сможем мгновенно присвоить переменным состояния нужные начальные значения, после чего они уже будут жить по законам, предписываемым им дифференциальными уравнениями.

Вернемся в окно “Состояние” и введем новую переменную `State_Name` перечислимого типа (Enumerated) с двумя значениями {ОДИН, ДВА} и присвоим ей начальное значение ОДИН. Эта переменная носит вспомогательный характер и нужна для нумерации узлов КАРТЫ СОСТОЯНИЯ (рис. 4).

Сделаем активным окно “Процессы”. Выполним команду “Процесс”, затем “Новый”, но теперь пометим процесс как дискрет-

ный и укажем, что для нумерации узлов необходимо использовать переменную `State_Name`. В окне “Состояние” появится новый процесс “INIT”. Его редактирование еще не завершено. Выделите строку, соответствующую вашему дискретному процессу, и щелкните по ней мышью. В открывшемся редакторе КАРТЫ СОСТОЯНИЯ сначала мышью расположите узлы “ОДИН” и “ДВА” в поле окна так, как вы считаете нужным, затем выполните команды “Переход”, “Новый”, ответьте

на вопросы и теперь уже выполняйте команды “Переход”, “Редактировать”. В появившемся меню выполните команду “Условие” и наберите булевское выражение  $Stime \geq 0$ , затем “Действия” и присвойте переменным `fi` и `fip` начальные значения  $fi:=fi0; fip:=fip0$ , используя “паскалевскую” нотацию для оператора присваивания (рис. 5).

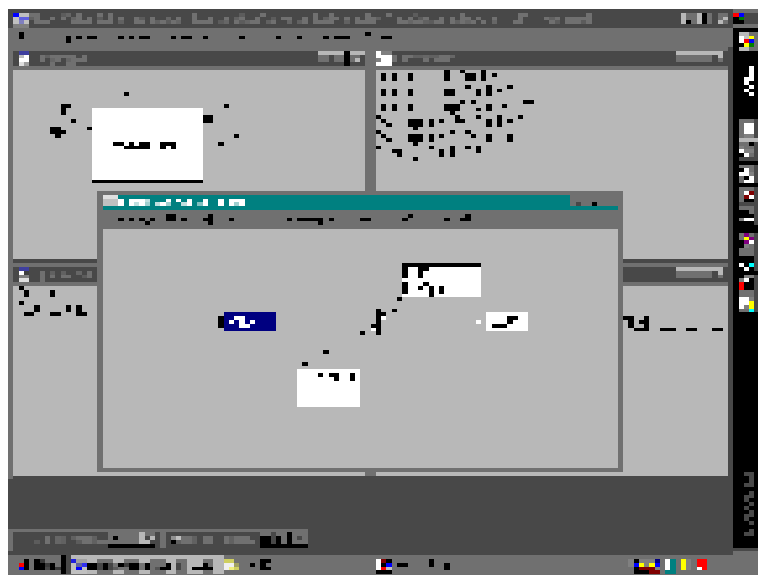
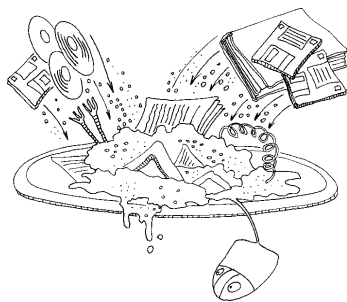


Рисунок 5.

## ОКНО ВЫЧИСЛИТЕЛЬНОГО ЭКСПЕРИМЕНТА.



“Известно, что мясо безъ приправы труднѣ переваривается желудкомъ, чемъ смешанное съ мучнистыми и вообще съ растительными веществами, а потому послѣднія составляютъ необходимую принадлежность почти всехъ мясныхъ блюдъ съ соусами, въ виде различныхъ гарнировъ”.

ПОЛНОЕ РУКОВОДСТВО для правильного ведения домашнего хозяйства, составленное Малоховской.  
С-Петербург, Изданіе П.А. Федорова, 1909

Теперь мы описали все, что нужно для автоматического построения программной реализации математического маятника. Выполните команды “Модель”, “Создать и запустить”, и пакет автоматически построит выполняемую модель, работающую под управлением программного комплекса. Это наиболее естественное решение, так как возможно нам придется корректировать модель. Для построения независимого приложения воспользуйтесь командой “Создать”.

Первоначально окно “Вычислительного эксперимента” содержит только внутреннее окно “Оборудование” (“Environment”). Это простейшее окно, автоматически поддерживающее диалог с пользователем. Достаточно щелкнуть мышью по стрелке, соответствующей входной переменной, и откроется окно редактирования начального значения переменной. Новое значение, отличное от заданного по умолчанию, справедливо только для одного опыта. При перезапуске модели (“Рестарт”) его нужно задавать заново.

Откроем три новых окна: окно “Фазовая диаграмма” и окно “Временная диаграмма” для переменных  $f_1$  и  $f_2$  и окно “Временная диаграмма” для переменных  $E_K$  и  $V$ .

Эти окна открываются единообразно. Выполняем команды “Окна”, “Открыть”, а далее выбираем нужный тип окна. Например, выбираем тип “Фазовая диаграмма” и говорим “ОК”. В появившемся диалоговом окне нажимаем кнопку “Доб(авить)”, выбираем устройство и принадлежащие ему переменные. Закрываем диалоговое окно, что предполагает автоматическое масштабирование.

Сами внутренние окна и их расположение в окне “Вычислительный эксперимент” можно запомнить. Выполните команды “Установки”, “Сохранить” и укажите имя файла с расширением “.stt”. В результате вы получите вот такое (или похожее) окно (рис. 6).

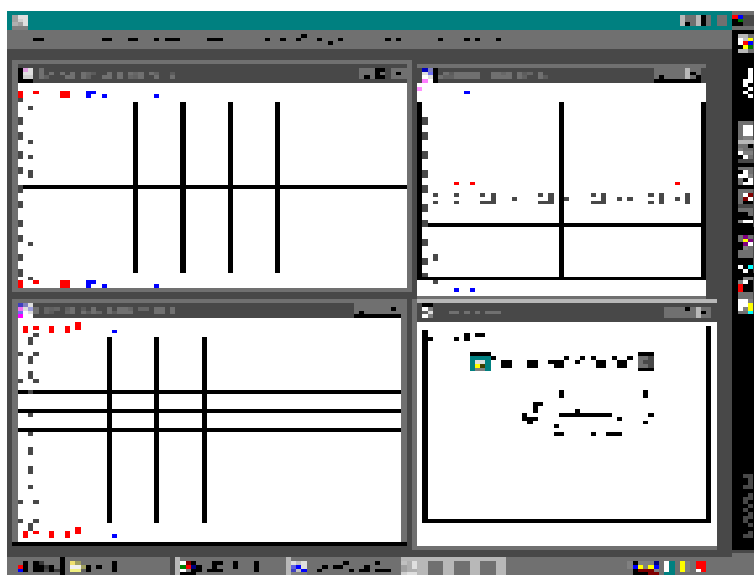


Рисунок 6.



Выполните команду “Пуск”, и вы увидите результаты моделирования (рис. 7).

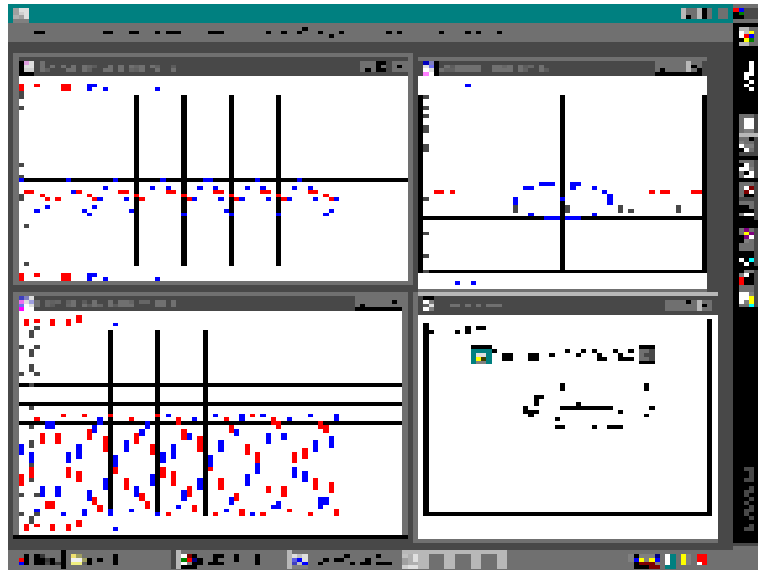


Рисунок 7.



“Хорошо приготовленный соус съ гарниром улучшает вкус и питательность мяса, при этом даже является возможным скрыть недостаток свежести провизии”.

ПОЛНОЕ РУКОВОДСТВО для правильного ведения домашнего хозяйства, составленное Малоховской. С-Петербург, Издание П.А. Федорова, 1909

Назначение этой цитаты - напомнить вам и себе, что легкость, с которой мы получили работающее приложение, может сыграть с нами злую шутку, если мы не будем тщательно проверять полученные результаты. В данном случае мы можем легко себя проверить, используя закон сохранения энергии -  $E=EK+V$ . Для этого необходимо добавить новую, не предусмотренную ранее формулу. Неужели ради этого нужно перетранслировать модель? Нет. Воспользуемся встроенным калькулятором. Выполните команды “Отлад-

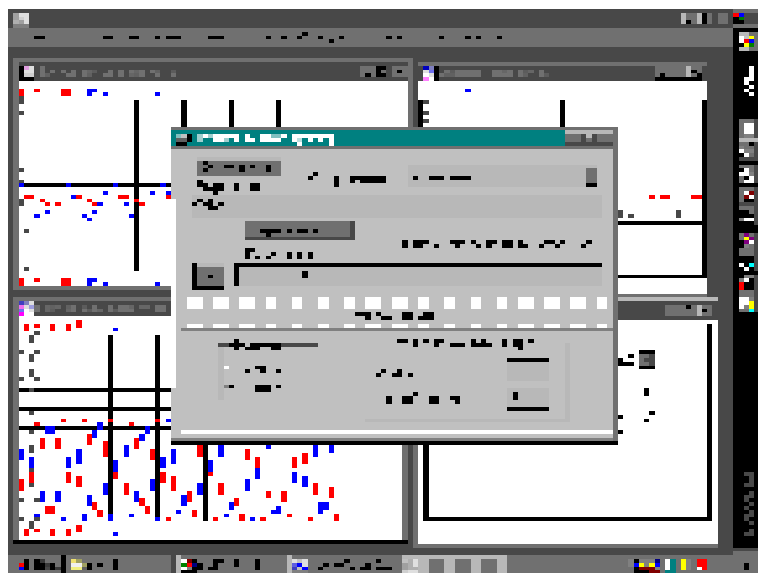


Рисунок 8.

ка”, ”Калькулятор” и напишите выражение, включающее любые переменные исследуемого устройства (рис. 8).

### **ВМЕСТО ЗАКЛЮЧЕНИЯ.**

Поверьте нам на слово, что разработка данной модели заняла всего десять минут реального времени. Нам кажется, что это неплохой результат. А Вам?

Третий, последний урок, мы посвятим разработке динамических образов.

*Колесов Юрий Борисович,  
кандидат технических наук,  
руководитель проекта MODEL  
VISION 2.1, старший научный  
сотрудник Центра Учебного и  
научного программного обеспечения  
ФТК СПбГТУ.  
Сениченков Юрий Борисович,  
кандидат физ.-мат. наук,  
доцент кафедры РВКС ФТК  
СПбГТУ.*

**НАШИ АВТОРЫ**