А.Л. КОРОЛЕВ

САМОСТОЯТЕЛЬНАЯ РАБОТА В ИНСТРУМЕНТАЛЬНОЙ СРЕДЕ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ ANYDYNAMICS

Учебно-методическое пособие

Министерство просвещения РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Южно-Уральский государственный гуманитарно-педагогический университет»

А.Л. КОРОЛЕВ

САМОСТОЯТЕЛЬНАЯ РАБОТА В ИНСТРУМЕНТАЛЬНОЙ СРЕДЕ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ ANYDYNAMICS

Учебно-методическое пособие

Челябинск
2023

УДК 681.4(021)
ББК 32.973:2-018я73
К 68

Королев, А.Л. Самостоятельная работа в инструментальной среде компьютерного моделирования ANYDYNAMICS: учебно-методическое пособие / А.Л. Королев; Министерство просвещения РФ; Южно-Уральский государственный гуманитарно-педагогический университет. – Челябинск: Изд-во ЮУрГГПУ, 2023. – 80 с. – ISBN 978-5-907790-78-0. – Текст: непосредственный.

Пособие содержит учебную информацию, а также подробные методические указания по самостоятельной работе в среде современного компьютерного пакета моделирования ANYDYNAMICS при построении компьютерных моделей. Материалы пособия будут использоваться в курсе «Компьютерное моделирование», 44.03.05 «Педагогическое образование» по профилям: «Информатика–Английский язык», «Математика–Информатика», «Физика с дополнительной специальностью». Настоящее пособие дополняет ранее опубликованное пособие по компьютерному моделированию:

Королев, А.Л. Компьютерное моделирование в инструментальной среде ANYDYNAMICS: лабораторный практикум / А.Л. Королев. – Челябинск: Изд-во ЮУрГГПУ, 2023.

Цель пособия – формирование у студентов умений разрабатывать компьютерные модели объектов на основе современной методологии моделирования с использованием современных технологий и программных средств в будущей профессиональной деятельности. Пособие построено на доступном для образовательных целей программном обеспечении.

ISBN 978-5-907790-78-0

Рецензенты:

В.Л. Дильман, д-р физ.-мат. наук, доцент

Г.Б. Поднебесова, канд. пед. наук, доцент

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Моделирование является общенаучным методом изучения законов окружающего мира, свойств объектов и систем самой различной природы. Моделирование как метод – мощный инструмент науки и техники. Любое исследование заканчивается построением определенной модели объекта или процесса. Таким образом, методология моделирования во многом совпадает с методологией исследований. Как показывает опыт, активное участие в моделировании вырабатывает более глубокое понимание сути протекающих процессов и наблюдаемых явлений.

Развитие компьютерных технологий предоставляет в профессиональной деятельности новые возможности в педагогической практике с максимальной степенью наглядности и оперативности получить и представить информацию о свойствах объектов и характере протекающих в них процессов. Применение компьютерных методов моделирования, проведение компьютерных экспериментов способствуют углублению и расширению знаний о процессах, протекающих в конкретных системах или объектах, существующих или проектируемых.

AnyDynamics – это высокопроизводительная среда для разработки компонентных моделей сложных динамических систем. AnyDynamics использует интуитивно понятный объектно-ориентированный язык моделирования высокого уровня, позволяющий быстро и качественно создавать сложные модели. **AnyDynamics** позволяет разрабатывать непрерывные, дискретные и дискретно-непрерывные модели и проводить с ними интерактивные вычислительные эксперименты.

AnyDynamics позволяет создавать модели многокомпонентных непрерывных, дискретных и гибридных (непрерывно-дискретных) систем.

Непрерывное поведение систем описывается с помощью дифференциально-алгебраических уравнений первого и второго порядка произвольной формы (в том числе неразрешенных относительно производных). Для описания дискретного и гибридного поведения используются визуальные карты поведения.

Дискретные действия записываются с помощью несложного алгоритмического языка, включающего хорошо известные базовые конструкции традиционных алгоритмических языков. Карта поведения позволяет задавать также план вычислительного эксперимента.

Для описания непрерывного поведения используется система уравнений, задаваемая в свободной форме. Свободная форма задания уравнений предполагает возможность:

Типовыми примерами использования AnyDynamics могут являться:

- моделирование механических систем;
- моделирование электрических цепей;
- имитационное моделирование;
- моделирование жидкостных гидравлических систем;
- моделирование газовых систем;
- моделирование систем управления;
- моделирование процессов макроэкономики;
- автоматическое построение выполняемой модели по создаваемому в интегрированной среде описанию;
- поддержка интерактивного и автоматизированного эксперимента с выполняемой моделью;
- возможность пошаговой отладки модели;
- широкий спектр средств проведения и визуализации интерактивного и автоматизированного эксперимента;
- 2D- и 3D-анимация;
- возможность моделирования непрерывных, дискретных и гибридных (непрерывно-дискретных) систем;
- возможность создания и использования библиотечных элементов;
- возможность использования выполняемой модели в качестве независимой программы (только в Rand Model Designer 8).

Новая среда моделирования AnyDynamics 8 – это свободно распространяемая версия Rand Model Designer 8 Standard Free (RMD). Она отличается от RMD 8 Professional только тем, что не позволяет создавать модели, встраиваемые в приложение.

Программный код выполняемой модели автоматически генерируется на основе математической модели и компилируется, что обуславливает высокую производительность при проведении вычислительных экспериментов. При автоматическом построении совокупной системы уравнений учитывается ее структура, уменьшается размерность и символьно разрешается часть уравнений, что в совокупности с использованием специальных численных методов дает возможность работать с большими системами уравнений (тысячи дифференциально-алгебраических уравнений) в том числе в режиме реального времени.

Имеются мощные средства демонстрации результатов модельных экспериментов, двухмерная и трехмерная анимация. Входной язык поддерживает возможность проведения «внутреннего» вычислительного эксперимента в ходе функционирования модели.

Моделирующая программа может быть, конечно же, написана непосредственно на каком-нибудь языке программирования с использованием готовых математических библиотек. Однако необходимо иметь достаточно высокую квалификацию в программировании и необходимые компетенции в области численных методов. Использование **AnyDynamics** существенно упрощает разработку компьютерной модели, а в ряде случаев программирование может быть исключено. Этот факт существенно расширяет круг специалистов, которые могут эффективно использовать AnyDynamics в своей практике, не имея должных компетенций в программировании и знаний в области вычислительной математики.

Если в создании компьютерной модели принимает участие много специалистов, хорошо известна проблема, возникающая при увольнении одного из них: если специалист увольняется, то соответствие модели и программы уходят вместе с ним, а оставшиеся в тексте программы комментарии часто понятны лишь их автору.

В AnyDynamics моделирующая программа генерируется автоматически по описанию модели и программных ошибок уже не содержит.

Ввод описания модели в удобной визуальной форме, генерация моделирующей программы, проведение вычислительных экспериментов, визуализация результатов – все это поддерживается специальными программными средствами, входящими в инструмент моделирования. Однако здесь кроется другая опасность. Мощность и наглядность современных языков визуального моделирования часто создают иллюзию простоты компьютерного моделирования.

Долгое время достаточно сильным препятствием в этом направлении была необходимость создания моделей средствами какой-либо системы программирования. В этом случае собственно моделирование отодвигалось на второй план, так как разработка модели начиналась специалистом в конкретной предметной области, затем математик формулировал математическую модель, а программист создавал компьютерную модель путем непосредственного программирования с получением в итоге программного продукта. Такой процесс требует привлечения узких специалистов и значительных затрат времени и средств, а также специфических знаний и умений.

В этом смысле компьютерное моделирование на основе специализированных инструментальных программных комплексов типа: MVS, RMD, AnyDynamics предоставляет возможность построить процесс моделирования, который будет принципиально отличаться тем, что модель создается средствами быстрой разработки, визуальными методами, с автоматическим выбором численных методов и генерацией исполняемого файла (*моделирование без программирования*). Компьютерная модель – это моделирующая программа, выполняемая на компьютере.

Следует использовать технологию компьютерного моделирования в учебном процессе по ряду дисциплин физико-математического и естественнонаучного циклов. Таким образом, инструментальные программные комплексы моделирования, дающие возможность конструирования моделей с наглядным представлением результатов при минимальной потребности в программировании, имеют особую ценность, так как наглядность – важное свойство учебного материала.

Визуализация – уникальная возможность компьютерной технологии моделирования, так как показать невидимое явление способны только компьютерные модели. Моделирование составляет неотъемлемую часть современной науки и техники, причем по важности моделирование приобретает первостепенное значение. Термин «моделирование» в большинстве случаев означает «компьютерное моделирование», так как применение компьютеров существенно расширило возможности и породило новые технологии и методики.

Цель настоящего пособия – формирование умений и навыков применения современных методов построения, реализации и исследования на основе компьютерных моделей объектов, процессов или систем разнообразной природы. Расширение представления студентов о моделировании как о методе научного познания; знакомство с методологией моделирования; обучение применению компьютера как средства познания в различных областях практической деятельности.

Таким образом, пособие позволяет решить следующие задачи подготовки специалистов:

- познакомить с современными методами и технологиями построения моделей и проведения модельных экспериментов средствами специализированных программных комплексов;
- обучить эффективному применению моделирования и модельного эксперимента;
- развить творческий потенциал будущего специалиста, необходимый для дальнейшего самообучения в условиях непрерывного развития и совершенствования информационных технологий;
- сформировать первоначальные навыки исследовательской деятельности.

В результате изучения представленного в пособии материала студенты должны быть способны использовать основные законы естественнонаучных дисциплин в профессиональной деятельности, применять методы математического моделирования, теоретического и экспериментального исследования, проводить моделирование процессов и систем. Результатом формирования компетенции в области моделирования будет являться готовность к участию в постановке задач проведения экспериментальных исследований; способность обосновывать правильность выбранной модели, сопоставляя результаты экспериментальных данных и полученных модельных решений в профессиональной деятельности.

Пособие предназначено для студентов направлений: 44.03.01 Педагогическое образование, профиль «Информатика»; 44.03.05 Педагогическое образование, профиль «Информатика (с дополнительной специальностью)».

Настоящее пособие является продолжением учебного пособия:
Королев, А.Л. Компьютерное моделирование в инструментальной среде ANYDYNAMICS: лабораторный практикум / А.Л. Королев. – Челябинск: Изд-во ЮУрГГПУ, 2023**.** Лабораторный практикум построен на доступном программном обеспечении.

В заключение следует отметить, что компьютерное моделирование изначально базируется на математике, но никак не заменяет математику, а лишь добавляет математике новые возможности при исследовании моделей. Более подробно о возможностях пакета компьютерного моделирования AnyDynamics можно узнать из документов, которые входят в состав программного комплекса: **«Руководства пользователя»** и **«Описание примеров»** (файлы help_ru.pdf и Demo.pdf).

ГЛАВА 1. МОДЕЛИ НЕПРЕРЫВНЫХ ПРОЦЕССОВ

1.1. Модель развития эпидемии

Предлагаемая ниже модель является учебной и не претендует на реальное отражение ситуации с «пандемией». Модель представляется в соответствии со статьей: Ляпцев А.В. Учебная модель развития эпидемии // Компьютерные инструменты в образовании. – 2020. – № 1. – С. 19–27.

В настоящее время в связи с развитием пандемии имеется достаточно много групп, занимающихся моделированием, анализом и предсказанием развития пандемии.

Цель данной работы – рассмотреть учебную модель и провести ее анализ. Учебная модель достаточно проста, чтобы в ней можно было разобраться, создать компьютерную модель и провести исследования. Такая модель должна содержать небольшое число параметров.

То, что излагается ниже, относится в основном к случаю, когда предпринимаемые «разумные» действия в процессе развития эпидемии исключаются, что делает её пригодной для описания развития в некоторой ограниченной области, заполненной некоторыми животными.

За основу взята известная модель Лотки–Вольтерра. Предполагается, что есть некоторая замкнутая область, населенная некоторыми особями, в которой по некоторым причинам появляется вирус. Возможно, вирус занесли со стороны. Зараженные вирусом особи при встрече со здоровыми собратьями заражают их, так что вирус начинает размножаться.

Обозначения: N общее число особей в области, $N_1(t)$ – число не заболевших особей, $N_2(t)$ – число больных в данный момент особей. Часть больных особей со временем выздоравливает и получает после выздоровления иммунитет, то есть уже не может заразиться. Обозначим число здоровых особей с иммунитетом через $N_3(t)$.

Число умерших к моменту времени t особей легко найти как разность $N - N_1(t) - N_2(t) - N_3(t)$.

Развитие эпидемии описывается системой обыкновенных дифференциальных уравнений:

$$\frac{dN_1}{dt} = -p \cdot N_1 \cdot N_2;$$
$$\frac{dN_2}{dt} = p \cdot N_1 \cdot N_2 - g \cdot N_2; \qquad (1)$$
$$\frac{dN_3}{dt} = a \cdot g \cdot N_2.$$

Уравнения описывают убыль незараженных особей и прибыль больных особей, параметр ***p*** характеризует вероятность заражения при встрече здоровой и больной особей. Параметр ***g*** характеризует длительность болезни и обратно пропорционален среднему времени течения болезни. В результате болезни относительная часть особей выздоравливает, что характеризуется параметром ***a*** в последнем уравнении. Систему уравнений (1) можно упростить, введя относительные значения переменных:

$$x = \frac{N_1}{N}; \;\; y = \frac{N_2}{N} -; \;\; z = \frac{N_3}{N}.$$

Также сделаем преобразование времени: $t' = g \cdot t$. Единица времени новой временной переменной соответствует средней продолжительности болезни. Поскольку для «обычных» вирусов время протекания болезни около недели, можно считать, что далее время измеряется в неделях. В результате небольших алгебраических преобразований получим следующую систему уравнений:

$$\frac{dx}{dt'} = -k \cdot x \cdot y,$$
$$\frac{dy}{dt'} = (k \cdot x - 1) \cdot y, \qquad (2)$$
$$\frac{dz}{dt'} = a \cdot y.$$

Производные берутся по новой временной переменной *t'*, а параметр *k* = *(Np)/g*. В этой системе уравнений взаимосвязанными являются два первых уравнения, решение которых определяется одним параметром *k*. Этот параметр зависит не только от «активности» особей, характеризуемой параметром *p*, но и от числа особей в рассматриваемой области *N.*

Систему уравнений (2) нужно решить при начальных условиях:

$$x(t = 0) = 1 - y_0,\ y(t = 0) = y_0,\ z(t = 0) = 0,$$

где y_0 – малая величина начальных зараженных особей ($y_0 = 0{,}0001$).

Нас интересует решение при $k > 1$. В противном случае, поскольку сомножитель $kx - 1$ ($x < 1$ по определению) отрицателен и число зараженных особей будет только убывать, и эпидемия не разовьется.

Создать проект по рис. 1.1.1.

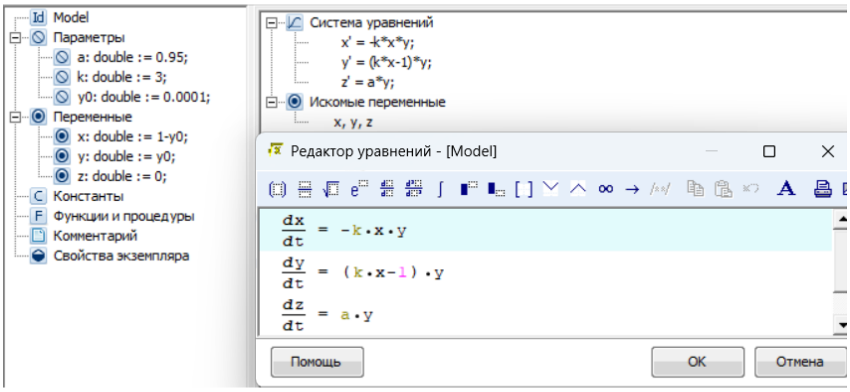


Рис. 1.1.1. AnyDynamics-проект развития эпидемии

Результат моделирования представлен на рис. 1.1.2.

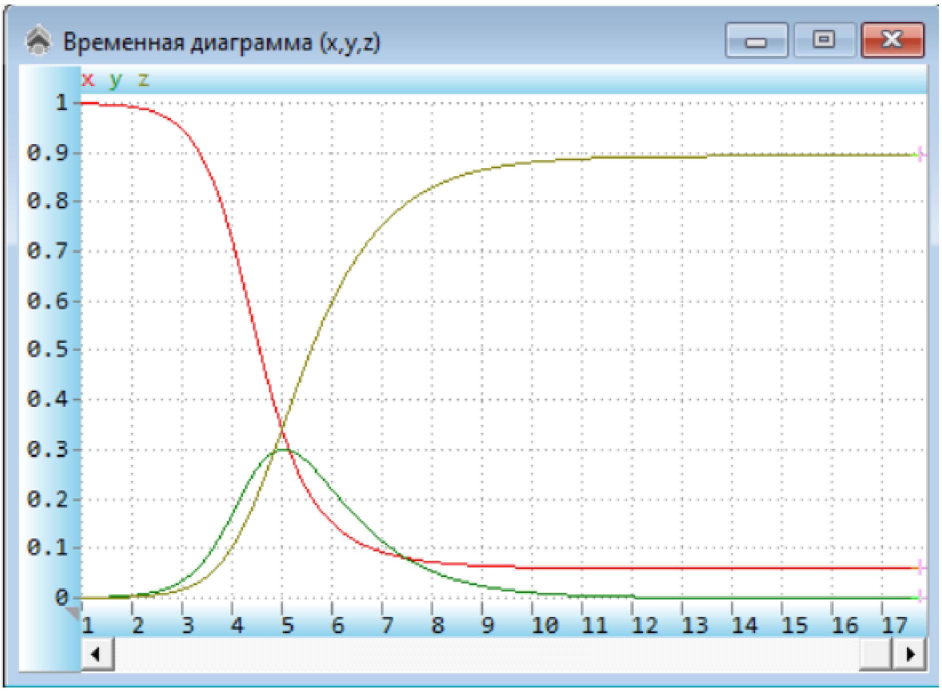


Рис. 1.1.2. Результат моделирования развития эпидемии при k = 3, a = 0,95

Выявить роль параметра k. Провести модельный эксперимент для $k = 1{,}1;\ 2;\ 5.$ Как зависит время эпидемии от значения k? Доказать, что уменьшение k при значениях, меньших $1{,}5$, «может принести больше вреда, чем пользы».

1.2. Модель системы «хищник–жертва»

Исследуется изолированная от прочих сторонних воздействий экологическая система, состоящая из двух видов: «хищник» и «жертва». Для «жертвы» имеется неограниченное количество корма. Смертность «жертв» обусловлена только воздействием со стороны «хищников».

Пусть x – количество «жертв», y – количество «хищников». Динамика взаимодействия между «жертвами» и «хищниками» описывается системой дифференциальных уравнений, которую называют уравнениями Лотки–Вольтерра:

Преобразование по методу неопределенных масштабов дает безразмерную систему дифференциальных уравнений:

$$\frac{dx}{dt} = x(1-y), \qquad \frac{dy}{dt} = y(x-k).$$

С начальными условиями: $x(t = 0) = x_0$, $y(t = 0) = x_0$, определяющими количество «жертв» и «хищников» в начальный момент времени. Коэффициент k является параметрам задачи.

Необходимо в среде AnyDynamics построить модель Лотки–Вольтерра и исследовать ее свойства.

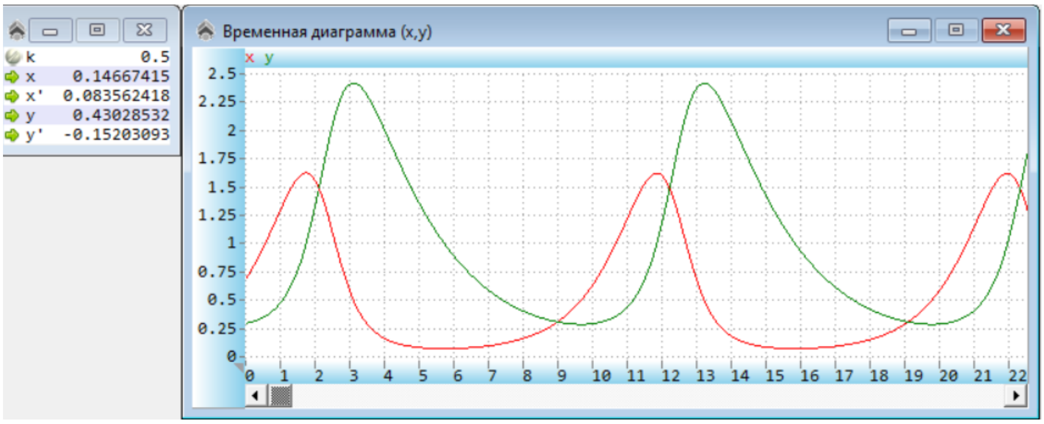


Рис. 1.2.1. Временная диаграмма процесса при $x(t = 0) = 0.7$, $y(t = 0) = 0.3$

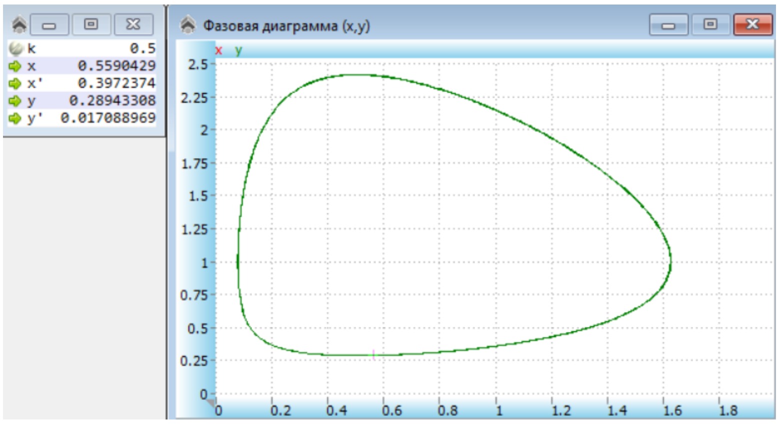


Рис. 1.2.2. Фазовая диаграмма процесса при *x(t = 0) = 0.7, y(t = 0) = 0.3*

При выполнении работы следует представить результаты моделирования в виде временной диаграммы *x(t), y(t)* и фазовой диаграммы *y(x)* (рис. 1.2.1, рис. 1.2.2).

На основе полученных результатов сделайте выводы о качественных закономерностях поведения системы «хищник–жертва» в зависимости от значений параметров. Установите возможные режимы разрушения системы вследствие исчезновения одного из видов. Кроме того, необходимо установить возможные стационарные состояния системы и проверить их устойчивость. С этой целью необходимо выявить такие начальные значения, которые дают решение, неизменное во времени. Один из таких режимов очевиден: *x(t = 0) = 0, y(t = 0) = 0,* а второй режим *x(t = 0) = k, y(t = 0) = 1.* Исследуйте поведение системы в том случае, когда начальные значения переменных мало отличаются от стационарных (исследование устойчивости).

1.3. Исследование модели «брюсселятор»

В данной работе необходимо провести исследование простейшей математической модели колебательной химической реакции, предложенной в 1968 году И.Р. Пригожиным и Р. Лефевром.

Колебательные химические процессы распространены в природе очень широко. Таковы многоступенчатые процессы биологического окисления углеводов, фотосинтеза. При построении математических моделей для этих объектов используют законы химической кинетики. Первый пример такой колебательной реакции – открытое в 1959 году Б.П. Белоусовым окисление лимонной кислоты броматом калия.

После открытия Б.П. Белоусовым периодической химической реакции, протекающей в одной фазе, И. Пригожин и его сотрудники предложили одну из наиболее эффективных математических моделей колебательных реакции – «брюсселятор» (от слов «Брюссель» и «осциллятор»).

«Брюсселятор» – основа описания диссипативных структур, возникающих в нелинейных неравновесных открытых системах различной природы – химической (периодические реакции, ведущие центры, спирали), биологической (биологические часы), физической (диссипативные структуры в твердых телах), экономической (колебания курса на бирже) и т.д.

Диссипативная система (или диссипативная структура, от лат. *dissipatio* — «рассеиваю, разрушаю») — это открытая система, которая оперирует вдали от термодинамического равновесия. Иными словами, это устойчивое состояние, возникающее в неравновесной среде при условии диссипации (рассеивания) энергии, которая поступает извне. Диссипативная система иногда называется ещё стационарной открытой системой или неравновесной открытой системой.

Неравновесная термодинамика – дисциплина, обладающая своим математическим аппаратом. Очень многие закономерности неравновесной термодинамики могут быть проиллюстрированы с помощью простых моделей, исследование которых приводит к системам нелинейных дифференциальных уравнений. Такой подход и используется в химии, где исследуются уравнения, полученные с помощью законов химической кинетики.

Простые модели химических реакций могут быть истолкованы в терминах, отличных от химических. Например, как модель Лотки–Вольтерра для системы «хищник–жертва».

Модель брюсселятора включает в себя следующие четыре реакции:

$$A \xrightarrow{k_1} X$$

$$B + X \xrightarrow{k_2} Y + D$$

$$2X + Y \xrightarrow{k_3} 3X$$

$$X \xrightarrow{k_4} E$$

Принятые допущения:

1. Все реакции протекают в открытой системе, т.е. возможен массообмен с окружающей средой.

2. Все реакции необратимы, обратных реакций нет.

Система является открытой, что позволяет удерживать ее вдали от состояния термодинамического равновесия. Это достигается тем, что поддерживается постоянными концентрации веществ *A* и *B* (это означает, что любой расход этих веществ может быть моментально компенсирован из окружающей среды).

Поэтому эти концентрации будут управляющими параметрами – изменяя их, мы можем влиять на поведение системы. *X* и *Y* являются промежуточными веществами, образуемыми в ходе реакций, зависимость их концентраций от времени является целью данного исследования. Вещества *D* и *E* являются конечными продуктами; предполагается, что они выводятся из системы и не влияют на кинетику реакций.

Помимо концентраций веществ *A* и *B*, есть еще ряд величин, характеризующих взаимодействие с окружающей средой. Константы скорости реакций k_1, k_2, k_3, k_4 являются функциями температуры и давления.

Наиболее важными свойствами рассматриваемой системы являются ее открытость и нелинейность. Система, в которой происходят химические реакции, по своей природе является диссипативной. Нелинейность задает третья из четырех реакций – это тримолекулярная реакция. Из двух молекул вещества *X* в результате взаимодействия с молекулой *Y* образуются три молекулы того же вещества *X*.

Это физико-химическая постановка задачи. Соответствующие дифференциальные уравнения для исследования скорости образования промежуточных веществ *X* и *Y* выписываются согласно закону действующих масс и выглядят следующим образом:

$$\frac{dX}{dt} = k_1 A - (k_2 B + k_4) X + k_3 X^2 Y$$

$$\frac{dY}{dt} = k_2 B X - k_3 X^2 Y$$

После преобразования к безразмерному виду получим следующую систему уравнений:

$$\frac{dx}{dt} = a - (b+1)x + x^2 y, \qquad \frac{dy}{dt} = bx - x^2 y.$$

Это нелинейная система обыкновенных дифференциальных уравнений с параметрами *a* и *b.* Решение этой системы при различных параметрах является кривая изменения концентраций *x* и *y* во времени.

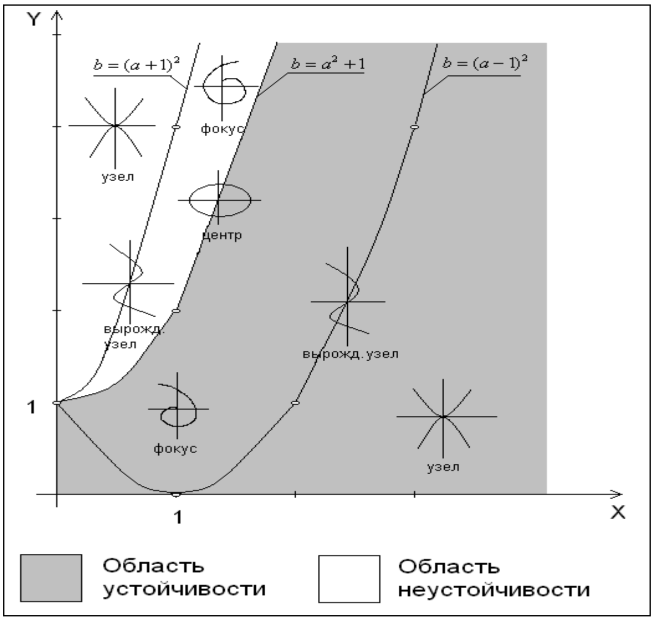


Рис. 1.3.1. Диаграмма устойчивости

На рис. 1.3.1 приведена графическая иллюстрация – диаграмма устойчивости. Средствами AnyDynamics построить модель системы (рис. 1.3.2).

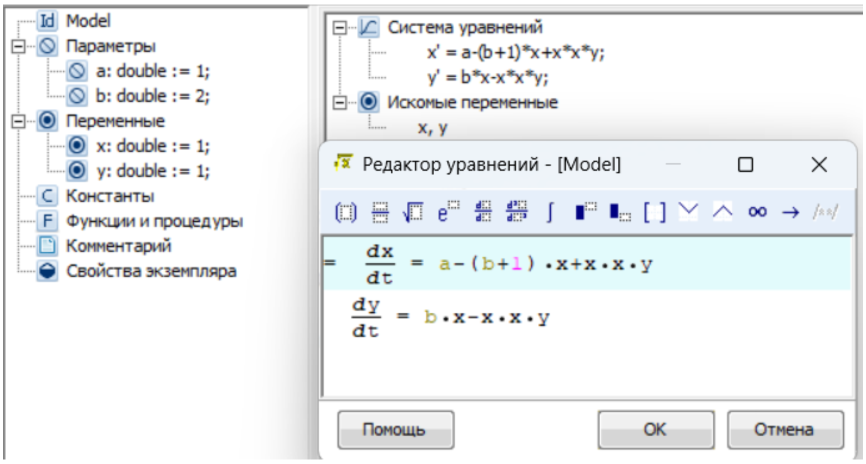


Рис. 1.3.2. AnyDynamics-модель «брюсселятора»

Исследовать влияние значений параметров и начальных условий на свойства системы. Использовать диаграмму устойчивости по рис. 1.3.1. Результаты моделирования отображать в виде временных и фазовых диаграмм (рис. 1.3.3, рис. 1.3.4).

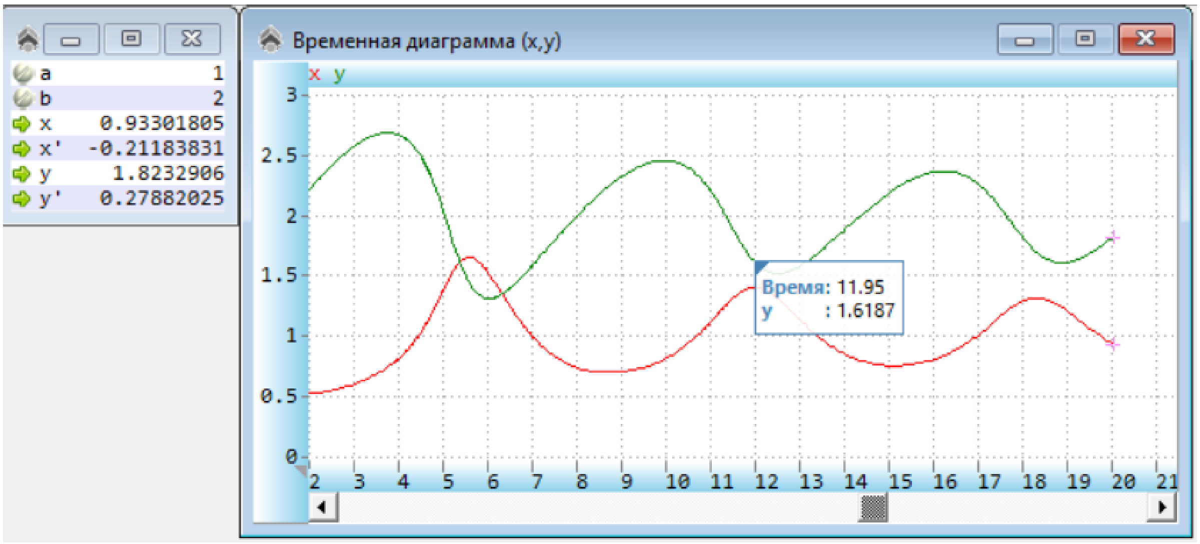


Рис. 1.3.3. Результаты моделирования: *x(t = 0) = 1, y(t = 0) = 1*

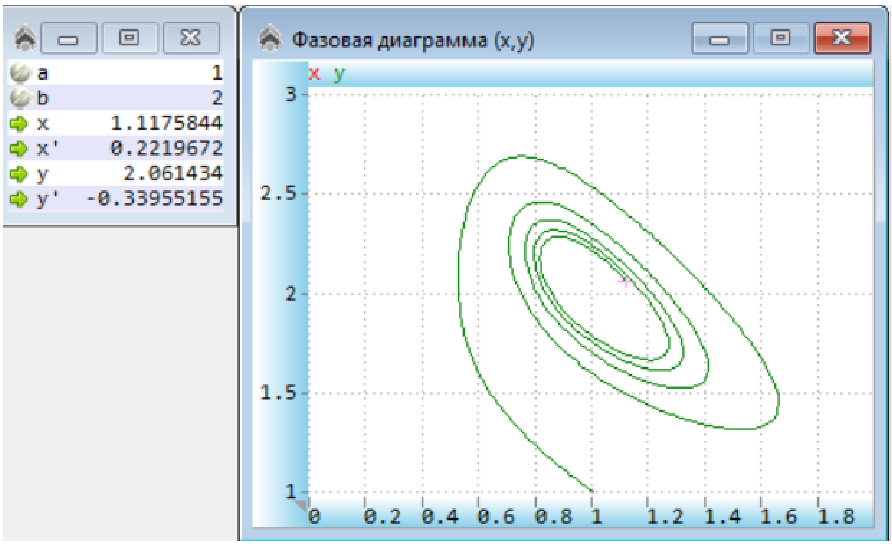


Рис. 1.3.4. Результаты моделирования: *x(t = 0) = 1, y(t = 0) = 1*

Дополнительные задания. В 1971 г. Лефлер и Николис предложили модель, описывающую колебательные процессы в химической реакции.

В среде AnyDynamics построить модель этой нелинейной химической реакции. Результаты моделирования отобразить в виде временных диаграмм *x(t), y(t)* и фазовой диаграммы *y(x), a = 0,5, b = 0,25.* Определить точки равновесия и исследовать их устойчивость.

$$\frac{dx}{dt} = a - (b+1)\cdot x + x^2 \cdot y, \qquad \frac{dy}{dt} = b \cdot x - x^2 \cdot y.$$

Начальные условия: *x(t = 0)= 2, y(t = 0)=3.*

В среде AnyDynamics построить модель (аттрактор Ресслера 1976 г.) химической реакции при значениях параметров: *a = 0,2; b = 2.*

Аттра́ктор (англ. *Attract* – привлекать, притягивать) – подмножество фазового пространства динамической системы, все траектории из некоторой окрестности которого стремятся к нему при времени, стремящемся к бесконечности.

Определить точки равновесия и исследовать их устойчивость. Результаты моделирования отобразить в виде временных диаграмм *x(t), y(t), z(t)* и фазовых диаграмм *y(x), z(x).*

$$\frac{dx}{dt} = -(y+z), \quad \frac{dy}{dt} = x + a\cdot y, \quad \frac{dz}{dt} = a + z\cdot(x-b).$$

Начальные условия: *x(t = 0) = 2,5; y(t = 0)=2; z(t = 0) = 1,5.*

В среде AnyDynamics построить модель аттрактора Э. Лоренца, полученный им в 1963 г. при моделировании метеорологических систем.

$$\frac{dx}{dt} = \sigma(x-z), \quad \frac{dy}{dt} = -yz + rx - y. \quad \frac{dz}{dt} = xy - bz.$$

Значения параметров для моделирования: σ *= 10; b = 2,66; r = 28.* Результат моделирования отобразить временными *x(t), y(t), z(t)* и фазовой диаграммой *z(y).* Начальные условия*: x(t = 0)=20; y(t = 0)=1; z(t = 0)=1.*

1.4. Моделирование связанных осцилляторов

Моделируемая система представляет собой систему связанных осцилляторов (рис. 1.4.1). Первоначально система выведена из состояния равновесия. Далее система будет совершать затухающие колебания.

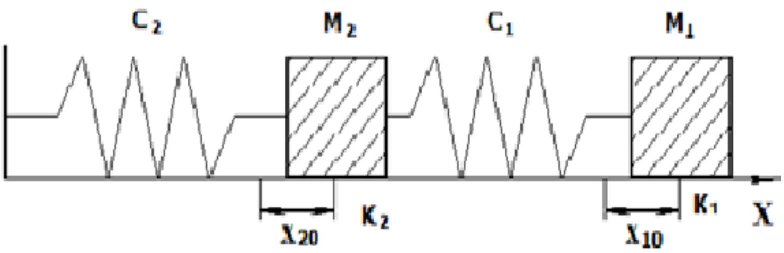


Рис. 1.4.1. Схема системы двух осцилляторов.
Начальное состояние – пружины растянуты

Исходная размерная модель динамики колебаний, записанная в виде системы дифференциальных уравнений, имеет следующий вид:

$$m_1 \frac{dV_1}{dt} = -c_1 \cdot (x_1 - x_2) - k_1 \cdot V_1; \quad \frac{dx_1}{dt} = V_1;$$

$$m_2 \frac{dV_2}{dt} = c_1 \cdot (x_1 - x_2) - c_2 \cdot x_2 - k_2 \cdot V_2; \quad \frac{dx_2}{dt} = V_2.$$

$$x_1(t=0) = x_{10}, \quad x_2(t=0) = x_{20}, \quad V_{1,2}(t=0) = 0.$$

где *m* – масса тела, *x* – координата тела (отклонение от положения равновесия), *V* – скорость движения тела, *c* – жесткость пружины, *k* – коэффициент трения. Значение координаты *x = 0* соответствует положению равновесия. Модель строится в размерной форме.

Таблица 1

Значения параметров системы осцилляторов

N	x_0	*c*	*k*	*m*
1	5.25	2.75	0.2	1.5
2	1	1.25	0.5	2.5

Построить временную диаграмму для $x_1(t)$ и $x_2(t)$, построить фазовую диаграмму $x_2(x_1)$, построить 3D-анимацию (рис. 1.4.2 – рис 1.4.4).

Рис. 1.4.2. Временная диаграмма процесса

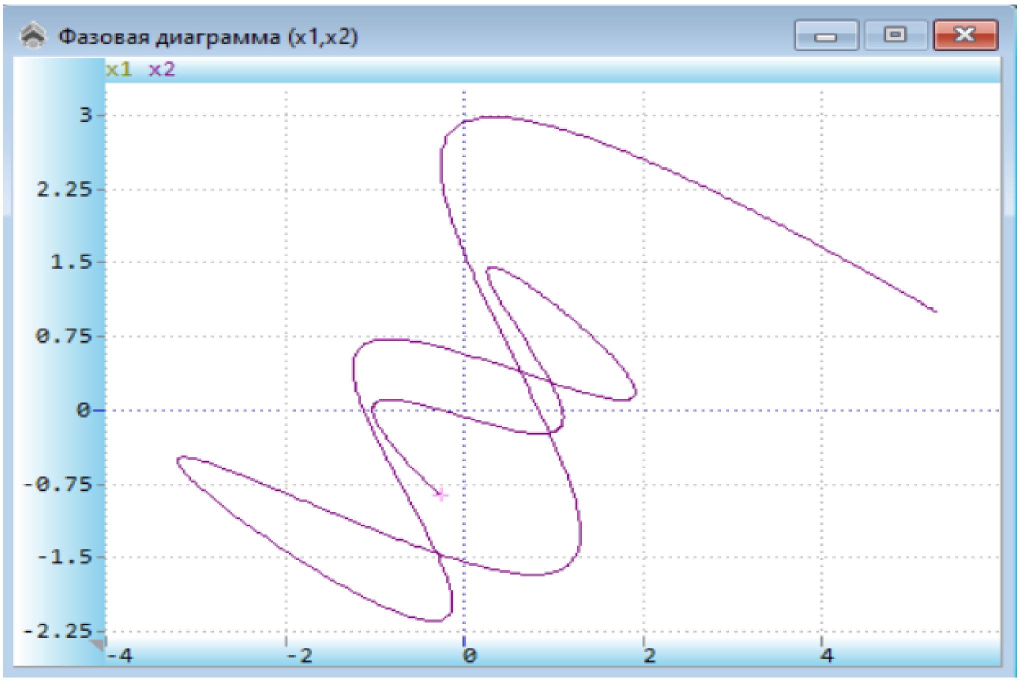


Рис. 1.4.3. Фазовая диаграмма процесса

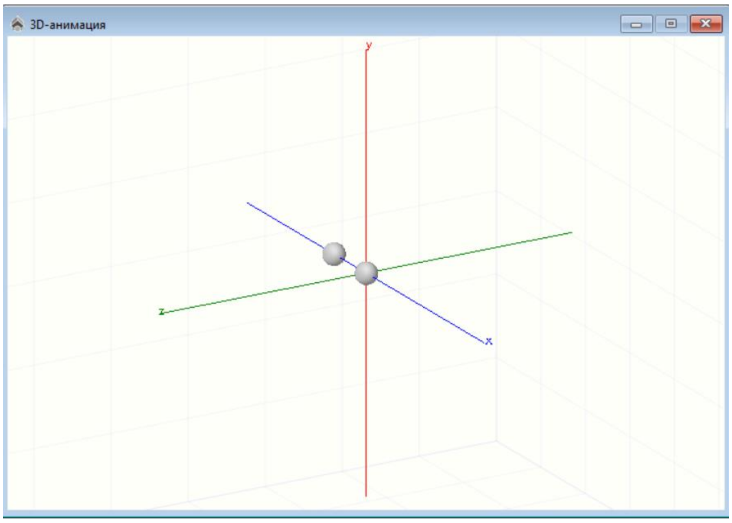


Рис. 1.4.4. 3D-анимация процесса

Установить влияние параметров на свойства системы путем варьирования их значений.

Создадим независимую от AnyDynamics исполняемую модель (файл с расширением exe), то есть независимое приложение. Для этого в главном окне проекта откроем пункт меню «Модель», выполним команду «Сохранить визуальную модель как…» (рис. 1.4.5).

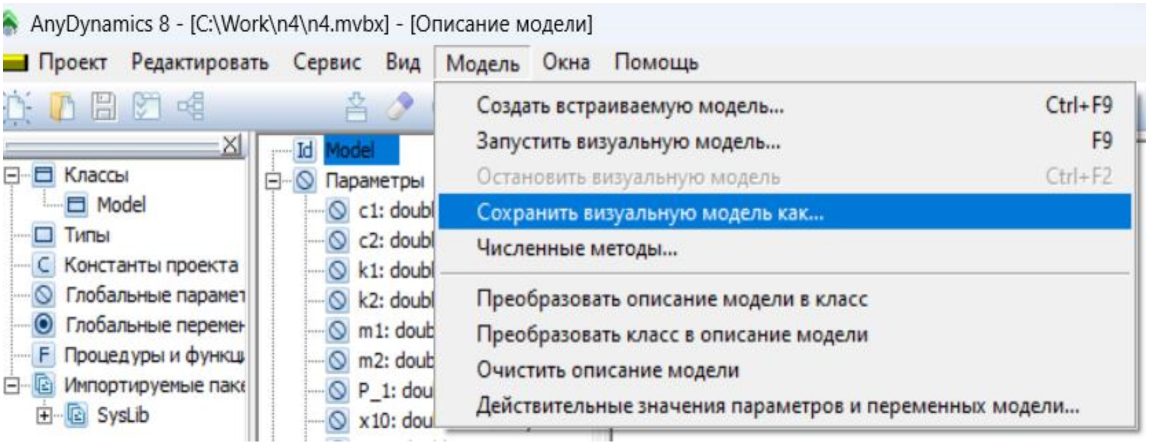


Рис. 1.4.5. Сохранение визуальной модели в виде приложения

В итоге в папке проекта N4 (рис. 1.4.6) образуется файл n4.exe.

Imports	08.10.2023 18:01	Папка с файлами	
Tmp	08.10.2023 18:09	Папка с файлами	
_mathmvs.dll	10.01.2023 13:45	Расширение при...	3 089 КБ
n4.bak	30.07.2020 12:50	Файл "BAK"	5 КБ
n4.dll	08.10.2023 18:05	Расширение при...	816 КБ
n4.exe	08.10.2023 18:09	Приложение	6 101 КБ
n4.ini	08.10.2023 18:03	Параметры конф...	2 КБ
n4.mvbx	04.10.2023 22:05	AnyDynamics Doc...	20 КБ
n4.mvbx.~~~	04.10.2023 22:05	Файл "~~~"	20 КБ
n4_em.bak	04.10.2023 22:05	Файл "BAK"	4 КБ
n4_em.ini	08.10.2023 18:03	Параметры конф...	4 КБ
n4_ide.ini	04.10.2023 22:05	Параметры конф...	1 КБ

Рис. 1.4.6. Рабочая папка проекта N4

1.5. Моделирование межвидовой конкуренции за общий ресурс

Пусть в экологической системе существует два вида, которые эксплуатируют общий жизненный ресурс (общая пища или территория существования) и находятся в конкурентной борьбе за его использование. Развитие каждого вида описывается модифицированным логистическим уравнением, которое учитывает взаимодействие видов за счет изменения экологической емкости окружающей среды. Модель построена для непрерывного процесса взаимодействия и имеет следующий вид:

$$\begin{cases} \dfrac{dx}{dt} = R_1 x\left(\dfrac{k_1 - x - \alpha y}{k_1}\right), \\ \dfrac{dy}{dt} = R_2 y\left(\dfrac{k_2 - y - \beta x}{k_2}\right). \end{cases}$$

Здесь R_1, R_2 – коэффициенты размножения, k_1, k_2 – параметры, характеризующие экологическую емкость среды для каждого вида соответственно. Коэффициент α – отражает влияние вида *y* на *x.* В свою очередь, коэффициент β отражает влияние вида *x* на вид *y*. Естественно, что при нулевых значениях α и β взаимодействие между видами отсутствует.

После приведения модели к безразмерному виду по методу неопределенных масштабов уравнения будут представлены соотношениями:

$$\frac{dx}{dt} = x(1 - x - e_1 y), \quad \frac{dy}{dt} = gy(1 - y - e_2 x).,$$

где $e_1 = \dfrac{\alpha k_2}{k_1}; \quad e_2 = \dfrac{\beta k_1}{k_2}; \quad g = \dfrac{R_2 k_1}{R_1 k_2}.$

Дополнив систему уравнений начальными условиями:

$x(t = 0) = x_0$, $y(t = 0) = y_0$, получим безразмерную модель взаимодействия двух популяций.

Необходимо провести исследование развития системы из двух популяций в зависимости от соотношения параметров.

Моделирование выполняется в среде AnyDynamics по аналогии с предыдущими работами. Динамику развития популяций необходимо отразить в виде временной диаграммы *x(t), y(t)*.

На основе данной модели необходимо проследить следующие варианты развития популяций:

1. $e_1 < 1, \ e_2 < 1$ с экологической точки зрения это означает, что межвидовая конкуренция слабее, чем внутривидовая. Вывод: возможно совместное суще-

ствование двух видов. Докажите путем выполнения компьютерных экспериментов, что независимо от начальных условий при неизменных значениях параметров с течением времени система приходит к одному и тому же состоянию.

2. $e_1 > 1,\ e_2 > 1$ межвидовая конкуренция является главным фактором, в зависимости от начальных условий выживает один из видов. В ходе экспериментов с моделью докажите это утверждение.

3. $e_1 > 1,\ e_2 < 1;\ \ e_1 < 1,\ e_2 > 1$ – один из видов вытесняет другой.

Все представленные варианты развития системы двух конкурирующих популяций необходимо подтвердить или опровергнуть в ходе выполнения численных экспериментов с моделью. Кроме того, необходимо установить возможные стационарные состояния системы и проверить их устойчивость. С этой целью необходимо выявить такие начальные значения $x(t = 0)=x_0$, $y(t = 0)=y_0$, которые дают решение неизменное во времени. Один из таких режимов очевиден: $x(t=0)=0$, $y(t=0)=0$.

Исследуйте поведение системы в том случае, когда начальные значения переменных мало отличаются от стационарных (исследование устойчивости).
Постройте в среде AnyDynamics компьютерную модель и проведите исследование поведения экологической системы, в которой три конкурирующих вида. Постройте временные и фазовые диаграммы *(x(t), y(t), z(t), y(x), z(x)).*

$$\frac{dx}{dt} = x\left(1 - x - e_1 y - r_1 z\right),\ \frac{dy}{dt} = gy\left(1 - y - e_2 x - r_2 z\right),\ \frac{dz}{dt} = uz(1 - z - e_3 x - r_3 y).$$

ГЛАВА 2. МОДЕЛИ С УПРАВЛЕНИЕМ КОМПЬЮТЕРНЫМ ЭКСПЕРИМЕНТОМ

2.1. Моделирование системы регулирования двигателя с управлением экспериментом

Системы регулирования предназначены для поддержания определенного состояния объекта. Рассмотрим систему регулирования частоты вращения двигателя постоянного тока (рис. 2.1.1). Задача системы – стабилизация частоты вращения двигателя постоянного тока при его включении или при действии внешней механической нагрузки.

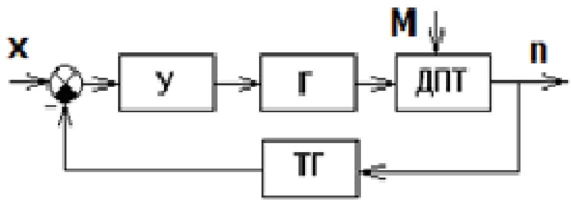


Рис. 2.1.1. Схема системы управления с обратной связью

Здесь *У* – усилитель; *Г* – генератор постоянного тока; *ДПТ* – двигатель постоянного тока; *ТГ* – тахогенератор (устройство, которое генерирует электрическое напряжение, пропорциональное частоте вращения); *M* – момент внешней нагрузки на валу двигателя; *n* – частота вращения вала двигателя, *x* – внешнее электрическое воздействие (при включении двигателя).

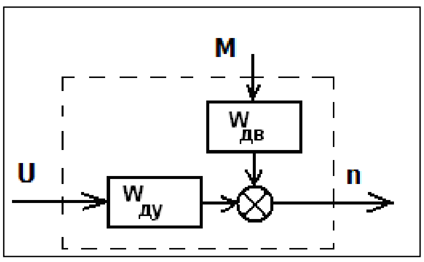


Рис. 2.1.2. Модель двигателя

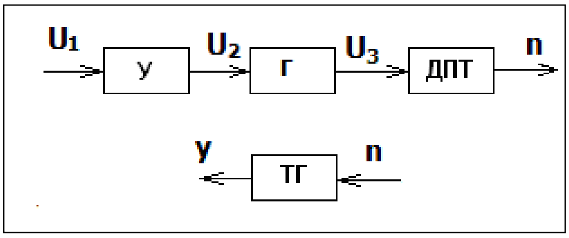


Рис. 2.1.3. Схема взаимодействия элементов системы

Математическая модель усилителя (рис. 2.1.3) представляется следующей зависимостью:

$$U_2 = k_u \cdot U_1 .$$

В свою очередь, математическая модель генератора имеет вид:

$$a_g \frac{dU_3}{dt} + b_g \cdot U_3 = k_g \cdot U_2 .$$

Математическая модель двигателя постоянного тока, описывающая поведение двигателя при изменении питающего напряжения, составляет следующее выражение:

$$a_d \cdot \frac{d^2 n}{dt^2} + b_d \cdot \frac{dn}{dt} + c_d \cdot n = k_d \cdot U_3 .$$

Математическая модель двигателя, описывающая поведение двигателя при изменении момента механической нагрузки на валу имеет вид:

$$a_d \cdot \frac{d^2 n}{dt^2} + b_d \cdot \frac{dn}{dt} + c_d \cdot n = k_d \cdot U_3 + k_m \cdot (a_m \frac{dM}{dt} + b_m \cdot M)$$

Модель тахогенератора имеет вид: $y = k_t \cdot n$. Тогда $U_1 = x - y$.

В работе строится математическая модель в виде общей системы уравнений, которая включает уравнения для всех элементов системы и соотношения для связей между элементами (рис. 2.1.4). Параметры системы регулирования представлены на рис. 2.1.4.

Построить AnyDynamics-модель системы регулирования (рис. 2.1.4). Построить временную диаграмму переходного процесса при включении двигателя; для включенного двигателя при возникновении нагрузки на двигатель.

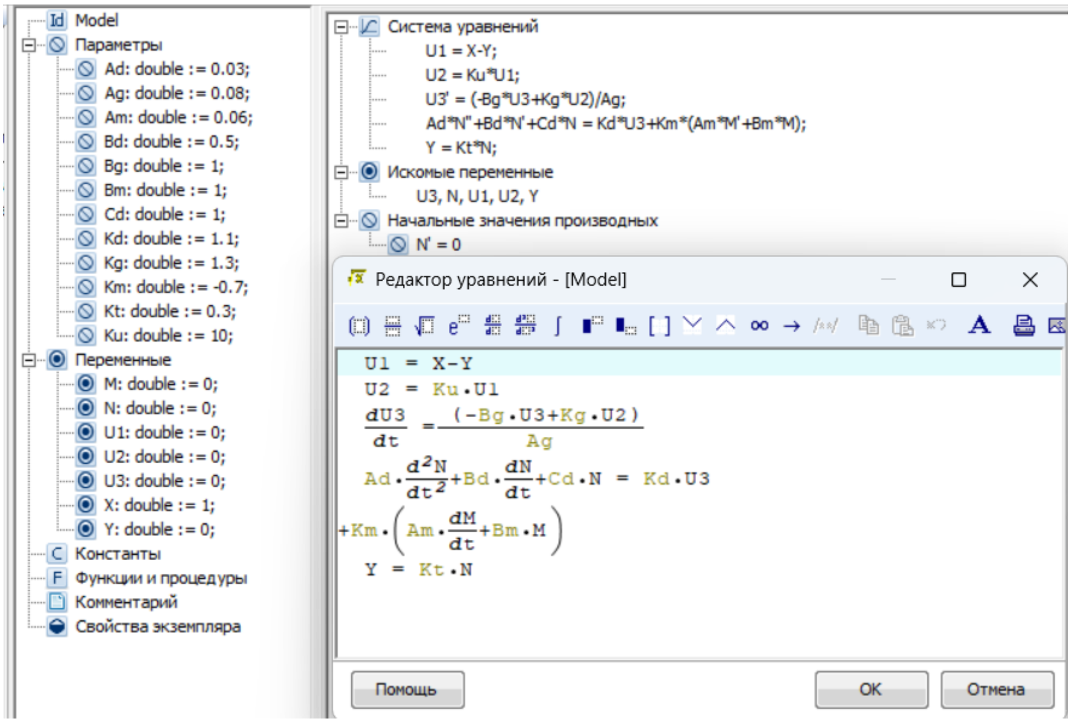


Рис. 2.1.4. AnyDynamics-модель системы стабилизации ДПТ

Для управления значением переменной *M* запустим модель и создадим «Новую 2D-анимацию» . Поместим на эту анимацию «Ползунок» из набора «Стандартные 2D-компоненты» (рис. 2.1.5).

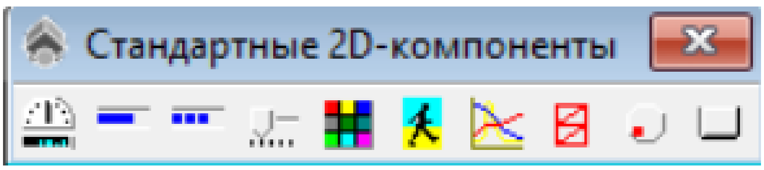


Рис. 2.1.5. Панель 2D-компонентов

Установим максимальное значение на ползунке равное ***10***. Методом «Drag and Drop» свяжем ползунок с переменной *M* и с помощью контекстного меню установим отображение значения *M*. Результат моделирования представлен на рис. 2.1.6.

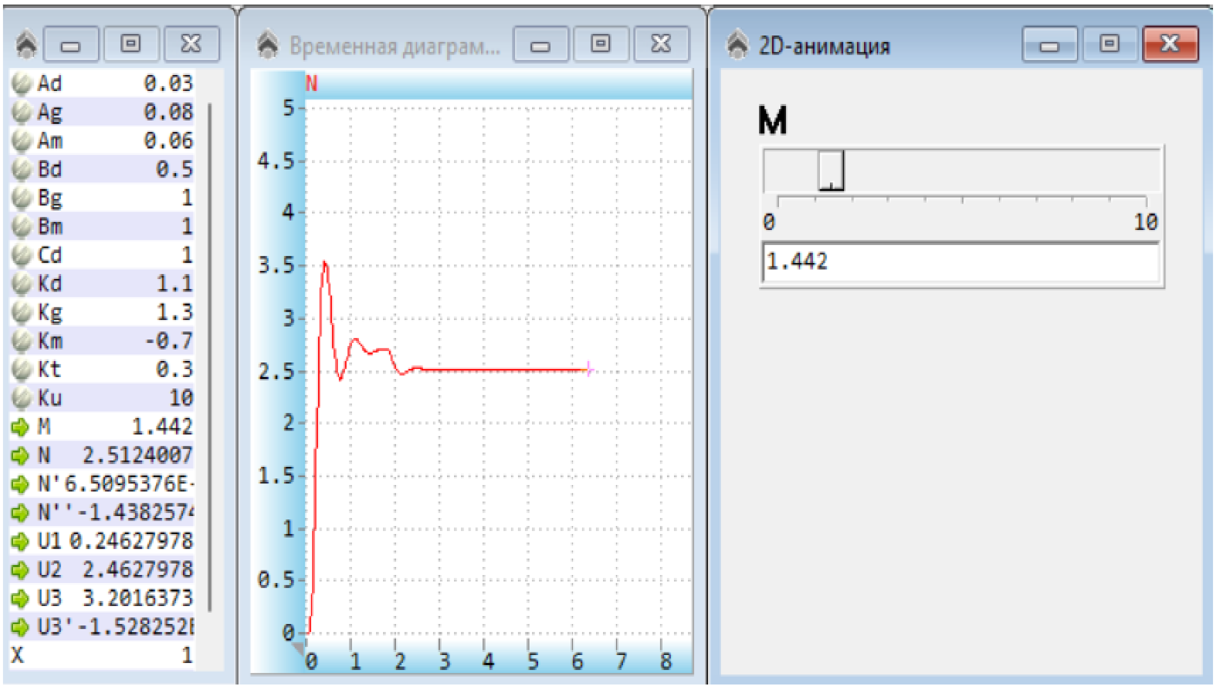


Рис. 2.1.6. Результат моделирования.
Временная диаграмма и ползунок управления

2.2. Модель с управлением и анимацией

Построить модель фонарика, состоящую из «лампочки» и «кнопки». При нажатии «кнопки» «лампочка» должна загораться. Данная модель – статическая, т.е. ее параметры не изменяются во времени. Срабатывание кнопки и зажигание света происходит в модельном времени мгновенно.

Решение задачи состоит в том, что модель необходимо создать одну переменную булевского типа с именем *«Вкл»*, связать ее с кнопкой и с окошком. В исходном состоянии, при отпущенной кнопке, значение переменной *«Вкл»* будет false (ложь). При нажатии кнопки значение переменной «*Вкл*» поменяется на true, (истина). Поскольку переменная «*Вкл*» будет связана и с окошком, то, получив значение true, она «зажжет» свет. При отпускании кнопки значение переменной «*Вкл*» станет false и фонарик «погаснет».

Запустить программу AnyDynamics. Создать новый проект (рис. 2.2.1). Дать проекту имя N6 и, нажав кнопку «обзор», поместить его, например, в папку C:\Work.

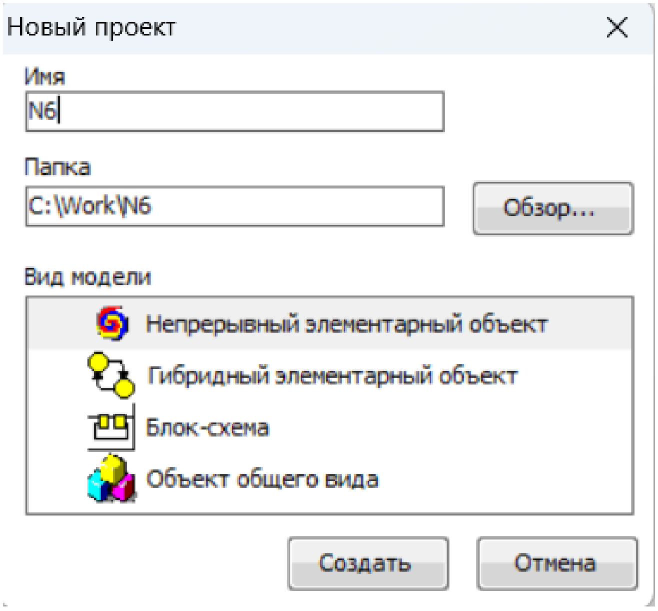


Рис. 2.2.1. Создание нового проекта

Нажать кнопку «Создать». После чего в главном окне проекта появится заготовка проекта (рис. 2.2.2):

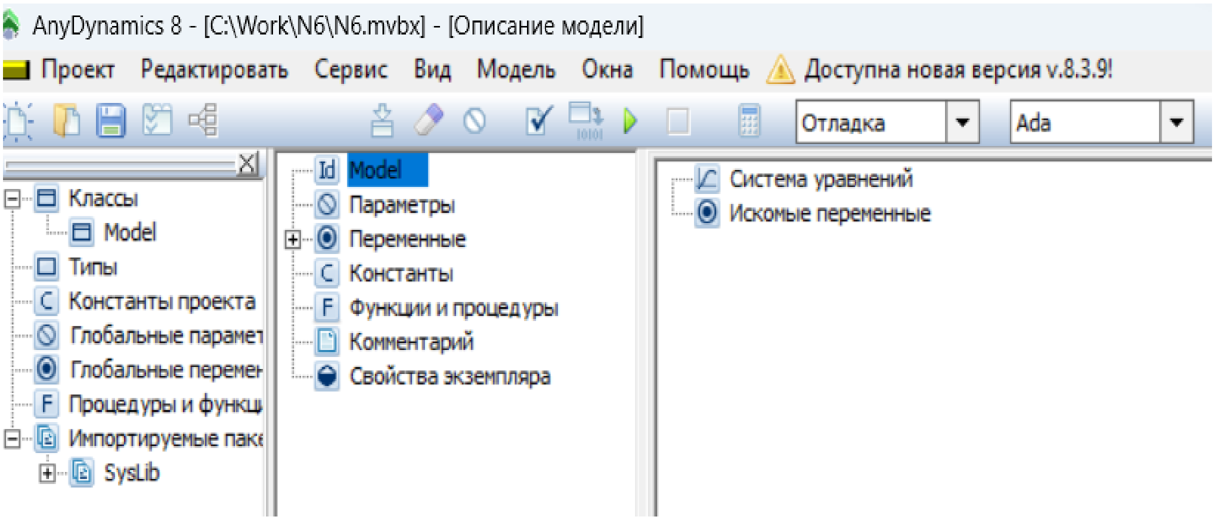


Рис. 2.2.2. Заготовка проекта

Следует добавить переменную булевского типа с именем *«Вкл».* Для добавления переменной следует щелкнуть правой кнопкой мыши по пункту «Переменные» и в контекстном меню выбрать пункт «Новая переменная». В появившемся окне задать имя переменной «*Вкл*», нажать кнопку «Тип» и выбрать boolean (рис. 2.2.3).

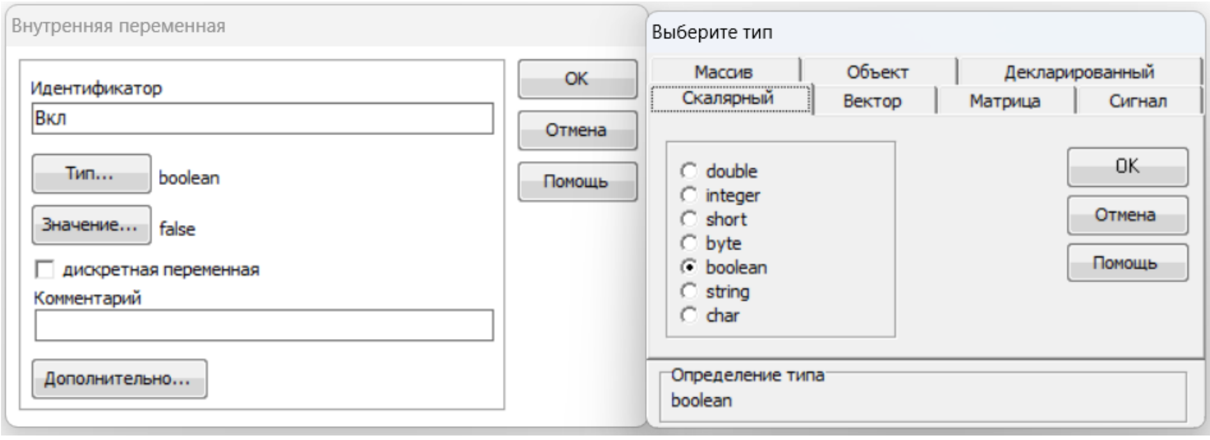


Рис. 2.2.3. Задание имени перемененной «*Вкл*»
и ее типа boolean с исходным значением false

После нажатия кнопки ОК в окне выбора типа, затем кнопки ОК в окне переменной, появится переменная «*Вкл*» типа boolean со значением false (рис. 2.2.4).

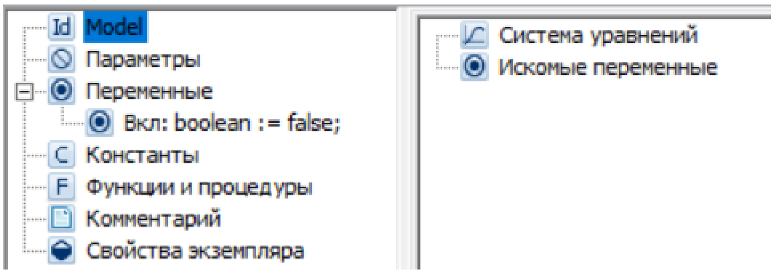


Рис. 2.2.4. Состояние проекта

Нажмите кнопку  «Запустить визуальную модель» на панели инструментов главного окна проекта. Вы увидите окно переменных. В окне переменных указана только что введенная переменная «*Вкл*» и ее исходное значение false (рис. 2.2.5). Создадим окно для 2D-анимации с помощью кнопки  «Новая 2D-анимация» (рис. 2.2.5).

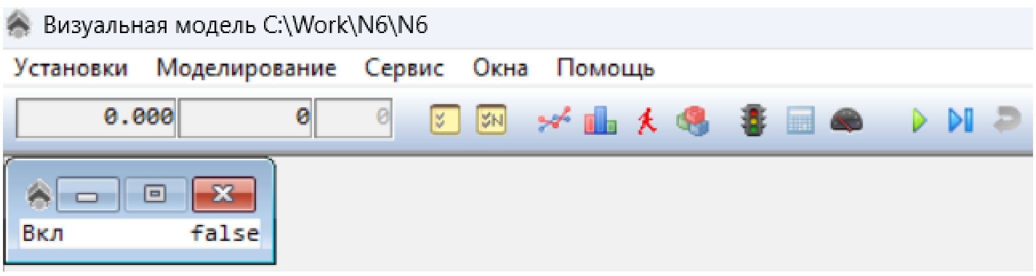


Рис. 2.2.5. Окно визуальной модели

В меню «Окна» выберем пункт «Новая 2D-анимация». В меню «Сервис» выберем «Стандартные 2D-компоненты» (рис. 2.2.6). Появится окно «Стандартные 2D-компоненты»:

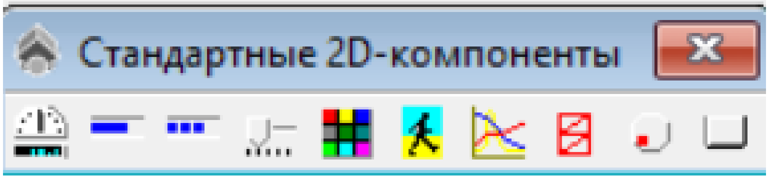


Рис. 2.2.6. Окно стандартных 2D-компонентов

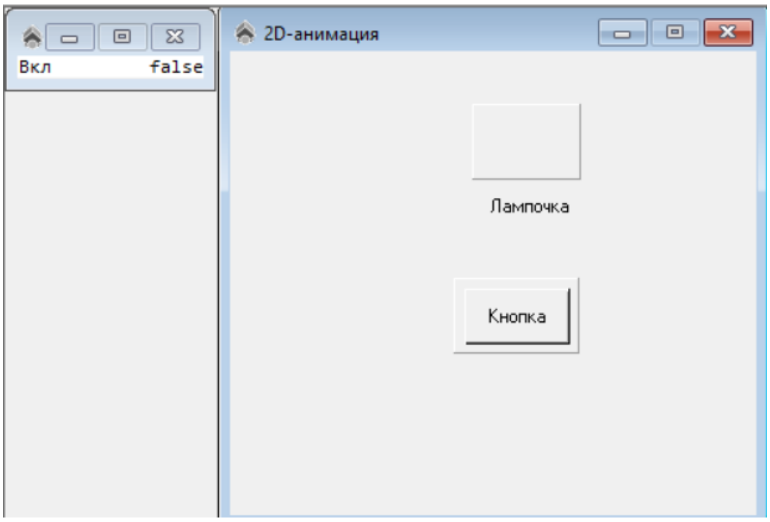


Рис. 2.2.7. Вид модели с 2D-анимацией

Для создания модели лампочки фонарика используем «Линейный индикатор сплошной», а для выключателя – «Кнопку». Нужно «перетащить» «Линейный индикатор сплошной» и «Кнопку» на поле «2D-анимация» (рис. 2.2.7).

Щелчок правой кнопкой над любым 2D-компонентом, выбор пункта меню «Надпись», позволяет подписать элементы в окне 2D-анимации (рис. 2.2.7).

Свяжем переменную «*Вкл*» с индикатором и кнопкой. Для этого перетащим переменную «*Вкл*» кнопку в окне 2D-анимации. И еще раз перетащим переменную «*Вкл*» на индикатор.

Щелкнем по кнопке, индикатор изменит цвет. Сделаем свет фонарика ярче: нужно правой кнопкой мыши щелкнуть на индикаторе, выбрать в меню «Цвет», в появившемся окне выбрать красный цвет и нажать ОК. Теперь, при нажатии кнопки, цвет индикатора будет красным (рис. 2.2.8).

Обратите внимание на то, что происходит со значением переменной «*Вкл*» в окне переменных при щелчке по кнопке переменная «*Вкл*» меняет свое значение с false на true. Поскольку эта переменная связана с индикатором, то и он изменяет состояние.

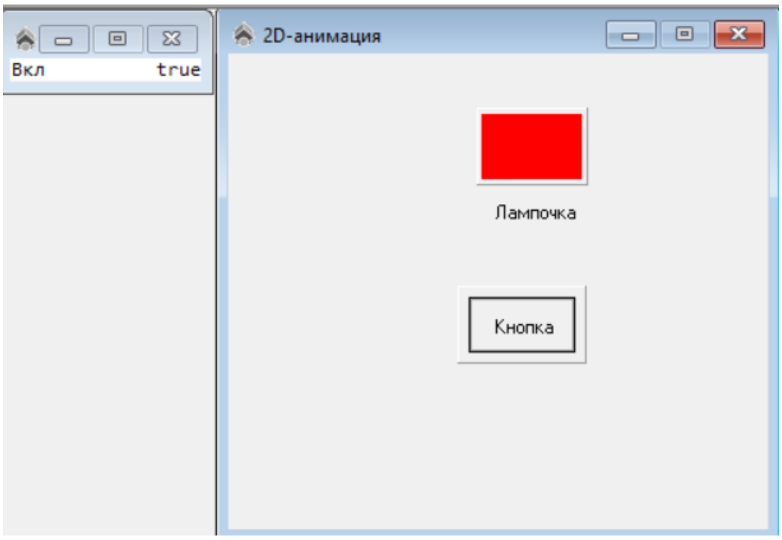


Рис. 2.2.8. Зажженный «фонарик»

В созданной модели фонарик светит только при нажатии кнопки. Исправим это. Щелкнем по кнопке правой клавишей мыши и в меню выберем пункт «Фиксация». Теперь, при щелчке по кнопке фонарика левой клавишей мыши, она остается утопленной, и фонарик светит постоянно. Чтобы его выключить, нужно щелкнуть по кнопке еще раз.

Остается выполнить сохранение проекта. На вопрос: «Сохранить текущие установки?», необходимо ответить «Да» и закрыть главное окно проекта.

Новая модель (продолжение проекта N6) будет, как и ранее, состоять из кнопки и «лампочки». При нажатии кнопки на лампочку должно подаваться напряжение от батарейки. Модель должна учитывать то, обстоятельство, что напряжение, которое обеспечивает батарейка, со временем снижается, и лампочка светит слабее. В этом и состоит динамика процесса: напряжение и связанная с ним яркость свечения лампочки зависят от времени.

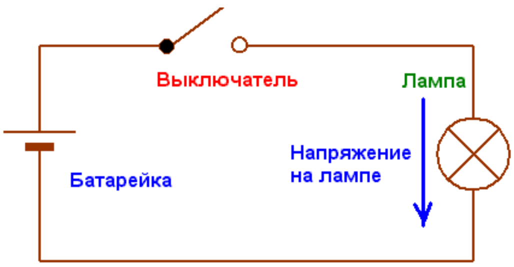


Рис. 2.2.9. Электрическая схема моделируемого объекта

Изменение напряжения батарейки со временем описывается уравнением:

$$U(t) = U_0 \cdot e^{-t/T_{\max}}, \qquad (1)$$

где: *U(t)* – напряжения батарейки – функция времени; U_0 – начальное напряжение у батарейки; *t* – время работы батарейки в фонарике во включенном состоянии; *T max* – срок службы батарейки. Это характерное время, в течение которого батарейка разрядится почти на две трети. У многих батареек для фонариков время *Tmax* составляет несколько часов. Таким образом, модель фонарика должна учитывать время работы батарейки и изменение ее напряжения в соответствии с формулой (1).

Используем проект N6 в качестве прототипа, потому что там уже выполнена значительная часть работы.

Переменная «*Вкл*» уже определена в проекте, добавим переменную *U*. Необходимо щелкнуть по пункту «Переменные», выбрать в выпавшем меню «Новая переменная» и в появившемся окне ввести имя переменной – *U*. Оставить тип double и значение *0*. Нажать ОК.

Таким же образом добавим параметр *Tmax* – срок службы батарейки, тип double, начальное значение *5* (часов). Для этого нужно щелкнуть по пункту «Параметры», выбрать «Новый параметр» и в появившемся окне ввести имя параметра –*Tmax* и его значение. Нажать ОК.

Параметр – это величина, которую, в отличие от переменной, нельзя менять в процессе работы модели, но можно изменить до запуска модели. Аналогично добавим переменную *U0* – начальное напряжение батарейки (*U0 = 1*). После добавлений окно модели примет вид: (рис. 2.2.10). Напоминаем, AnyDynamics различает строчные и прописные буквы в именах переменных и параметров.

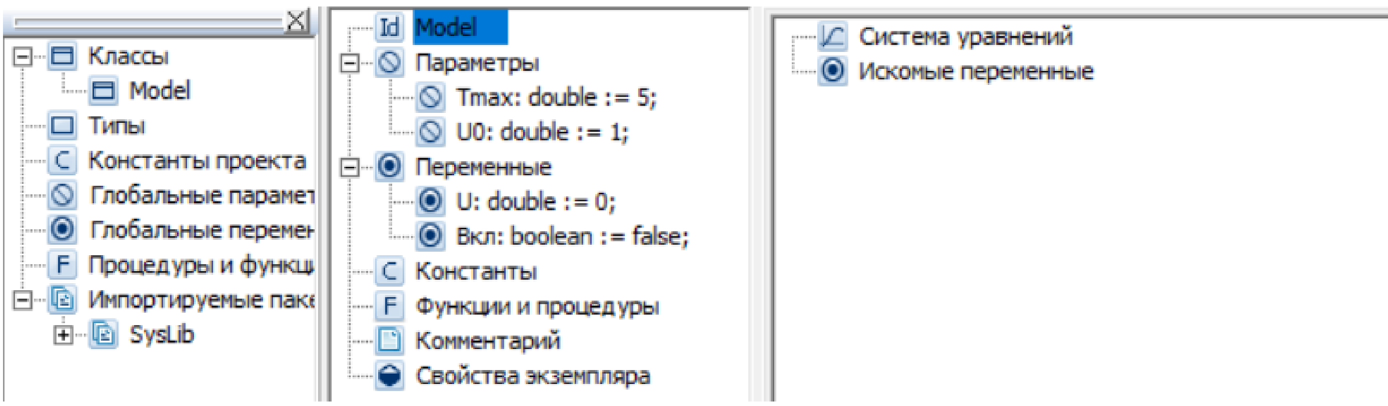


Рис. 2.2.10. Переменные и параметры модели

Введем уравнение, описывающее изменение напряжения батарейки с течением времени. Для этого нужно открыть окно «Система уравнений», в окне «Система уравнений» щелкнуть правой кнопкой по пункту «Изменить» и ввести уравнение (рис. 2.2.11, рис. 2.2.12).

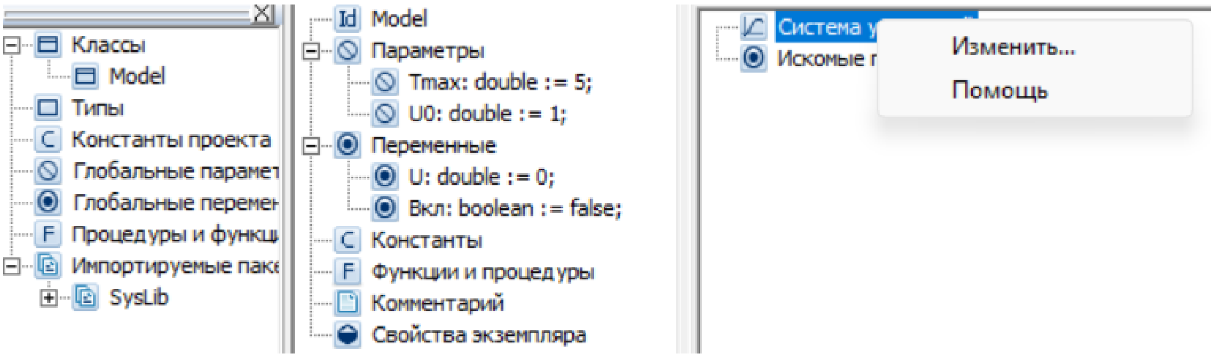


Рис. 2.2.11. Открытие редактора формул для ввода системы уравнений

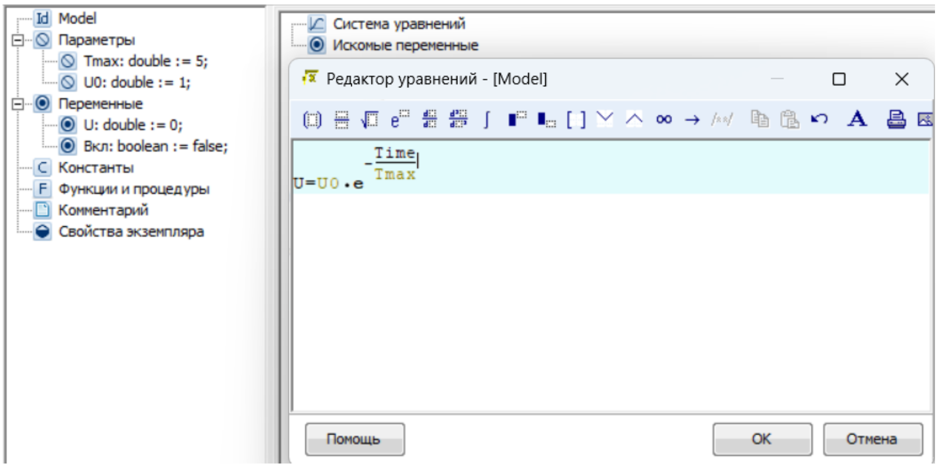


Рис. 2.2.12. Запись уравнения в редакторе формул

После чего щелкнуть по кнопке ОК. Окно системы уравнений примет вид по рис. 2.2.13.

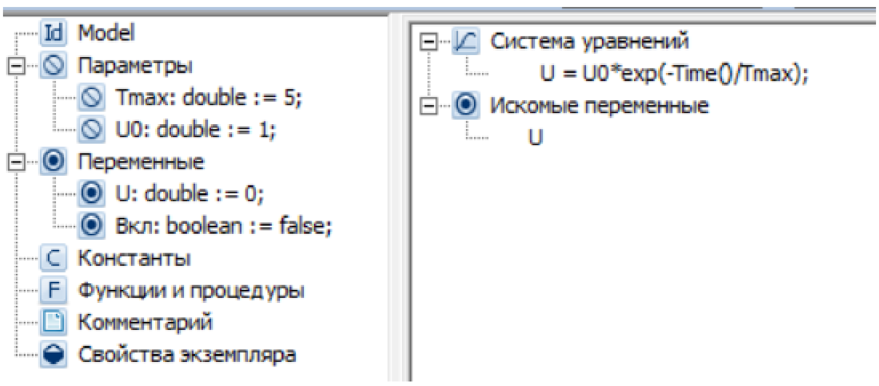


Рис. 2.2.13. Окончательный вид системы уравнений

Создадим визуальную модель. Для этого нужно, как и ранее, нажать на кнопку . С помощью кнопки «Новая диаграмма» добавим в модель временную диаграмму и перетащим на нее из окна переменных переменную *U* (временная диаграмма *U(t)*). Запустить выполнение визуальной модели можно нажав кнопку . В результате появится окно визуальной модели с результатами моделирования (рис. 2.2.14).

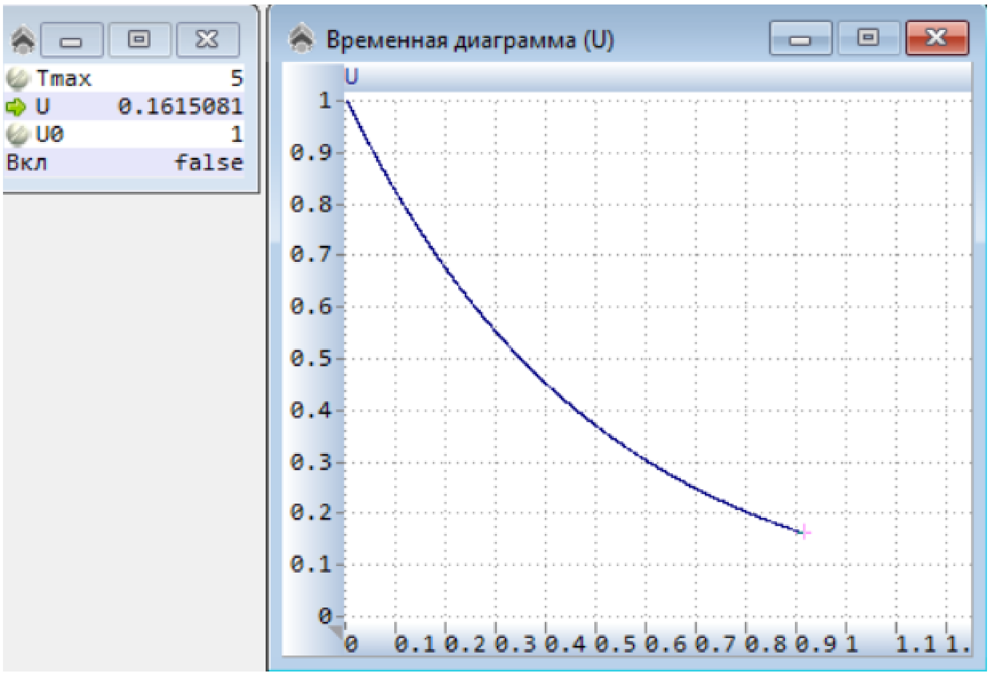


Рис. 2.2.14. Результат моделирования

ГЛАВА 3. ГИБРИДНЫЕ МОДЕЛИ

3.1. Гибридная модель фонарика

Значение переменной *U*, как видно из диаграммы (рис. 2.2.14), будет уменьшаться. Однако модель не реагирует на нажатие кнопки, напряжение изменяется непрерывно с момента запуска модели до окончания времени моделирования.

Однако объект моделирования должен проявлять гибридное поведение. В процессе функционирования должны быть два качественно различных состояния с непрерывным поведением. Это состояния: «Фонарик_светит» (лампочка светит) и «Фонарик_погашен» (лампочка не светит). Переход из одного состояния в другое происходит по событию: нажатие кнопки фонарика. Событие в программировании означает, что в процессе выполнения программы что-то произошло, что будет замечено и воспринято компьютером, и он должен на это как-то отреагировать.

Если событие влияет на состояние объекта, то до того, как событие случилось, модель AnyDynamics функционировала в соответствии с одними уравнениями, то после события модель должна подчиняться другим уравнениям. Например, до включения кнопки напряжение на лампочке равно нулю, то после «события» – включение кнопки, напряжение на лампочке изменяется в соответствии с формулой $U(t) = U_0 \cdot e^{-t/T_{\max}}$. Созданный проект этих событий не учитывает, поэтому лампочка горит непрерывно.

Создадим новый гибридный проект (рис. 3.1.1).

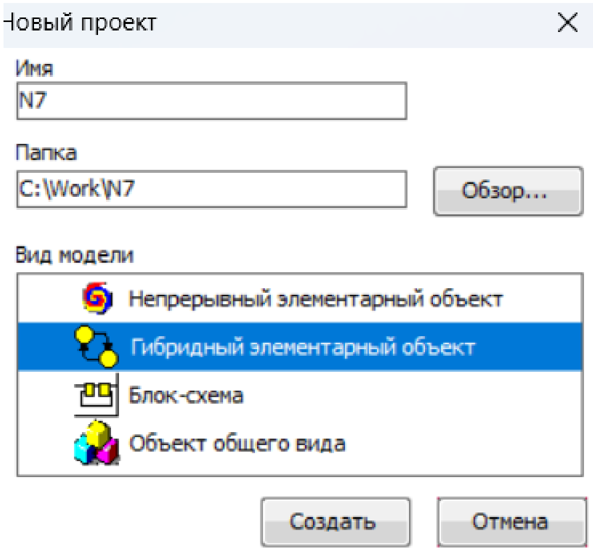


Рис. 3.1.1. Создание нового гибридного проекта

Добавим в модель параметры и переменные (рис. 3.1.2). Для модернизации модели введем дополнительную переменную с именем *«Светит»*, тип – boolean, начальное значение – false. Процедура добавления переменных и параметров была описана выше.

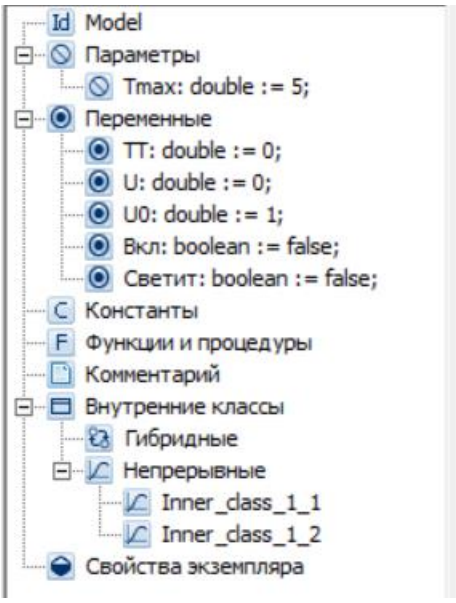


Рис. 3.1.2. Параметры и переменные проекта

Переменная *TT* является вспомогательной и при нажатии кнопки «запоминает» значение времени (*Time()*) ее нажатия. Значение *TT* необходимо для вычисления времени работы фонарика во включенном состоянии.

Модель, управляемая событиями, может быть наглядно представлена графически. В системе моделирования AnyDynamics для этого служит «Карта поведения». Она может графически отобразить и то, как модель реагирует на события, и то, как она себя ведет в промежутках между событиями. Программа AnyDynamics позволяет пользователю строить модель и программировать ее поведение, используя описание поведения модели в «Карте поведения». Она позволяет описать действия в переходах, во входных и выходных действиях для узлов, записывать условия срабатывания переходов, и действия, которые должна будет выполнить модель.

Карта поведения проекта с условиями срабатывания переходов и действиях в узлах представлена на рис. 3.1.3. Содержание «Inner_class_1_1» и «Inner_class_1_2», которые описывают непрерывное поведение в узлах «Карты поведения», представлено на рис. 3.1.4 и рис. 3.1.5.

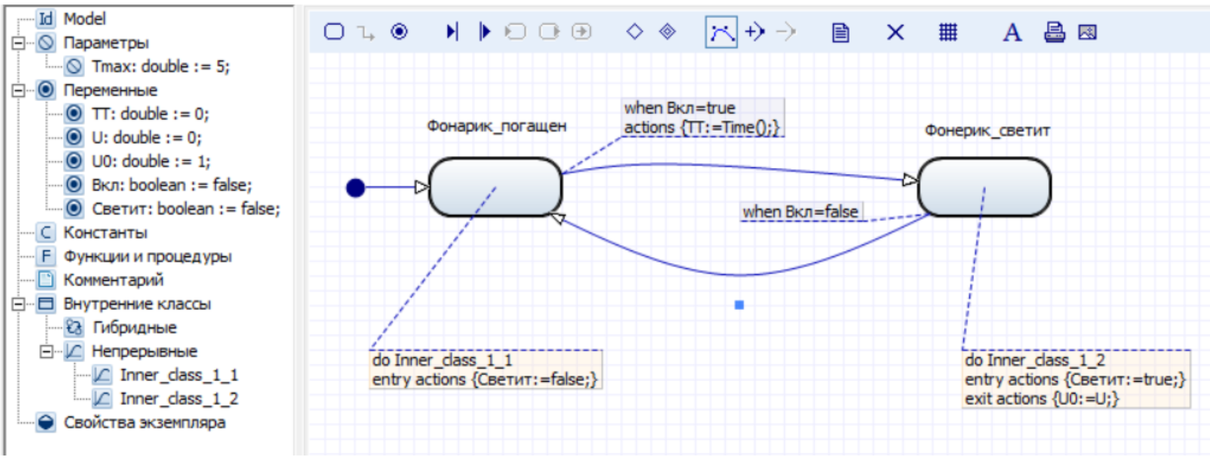


Рис. 3.1.3. Карта поведения проекта

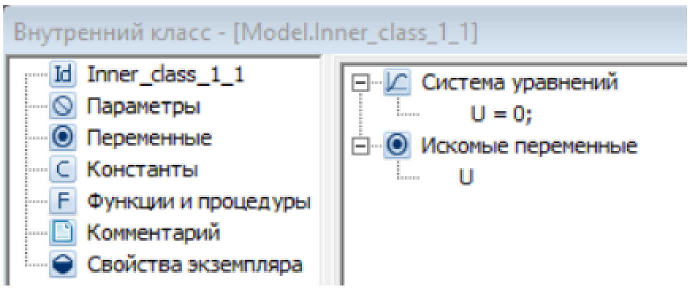


Рис. 3.1.4. Inner_class_1_1

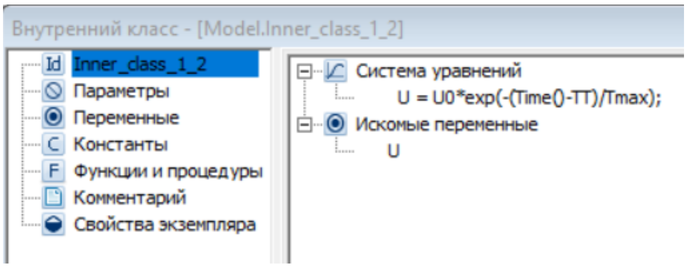


Рис. 3.1.5. Inner_class_1_2

Примечание. Редактор формул замечает синтаксические ошибки и при попытке сохранить и выйти сообщает об этом. Например, если забыть поставить в конце оператора точку с запятой (**;**), то редактор откажется сохранять такую запись и сообщит об этом.

Переходы от одного вида поведения модели к другому происходят вследствие и сразу после свершения **событий**, которые могут произойти при работе модели. Переходы на карте поведения представляются линиями, связывающими узлы. В нашем проекте два события. Первое – это нажатие на кнопку фонарика («Фонарик_светит»). Второе – это отпускание этой кнопки («Фонарик_погашен»). И то и другое события будут происходить при щелчке по кнопке.

Напряжение батарейки уменьшается только, когда фонарик светит, поэтому напряжение рассчитывается по формуле $U(t) = U_0 \cdot e^{-(Time()-TT)/T_{\max}}$.

Модель, которую мы построили ранее, не предусматривала отключение фонарика. Для того чтобы при новом включении изменение напряжения начиналось с величины, которое оно имело при предыдущем отключении, добавим действие на выходе из узла «Фонарик_светит». Необходимо записать: *U0:=U* (рис. 3.1.3).

Создадим 2D-анимацию и свяжем переменные с объектами 2D-анимации. Добавим «Линейный индикатор», который будет отражать свечение лампочки. Теперь свяжем переменную *U* с линейным индикатором «Напряжение», переменную *«Вкл»* свяжем с кнопкой, переменную и проверим работу модели (рис. 3.1.6). Действительно, теперь лампочка реагирует на нажатие кнопки, а напряжение уменьшается только во время свечения лампочки.

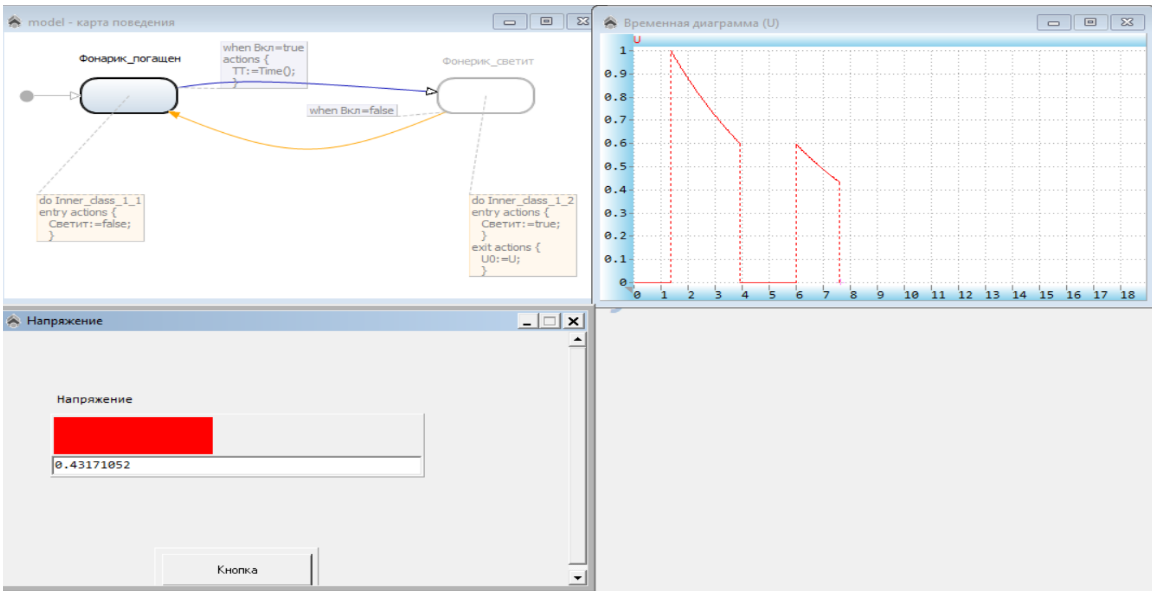


Рис. 3.1.6. Результат моделирования

Переменная *U* при запуске модели должна изменяться во времени в соответствии с формулой (1), если переменная *«Вкл»* имеет значение true.

Запустить модель можно нажатием кнопки .

В меню появившегося главного окна модели выберем пункт «Новая 3D-анимация» (рис. 3.1.7). Начальное состояние окна 3D-анимации представлено на рис. 3.1.8.

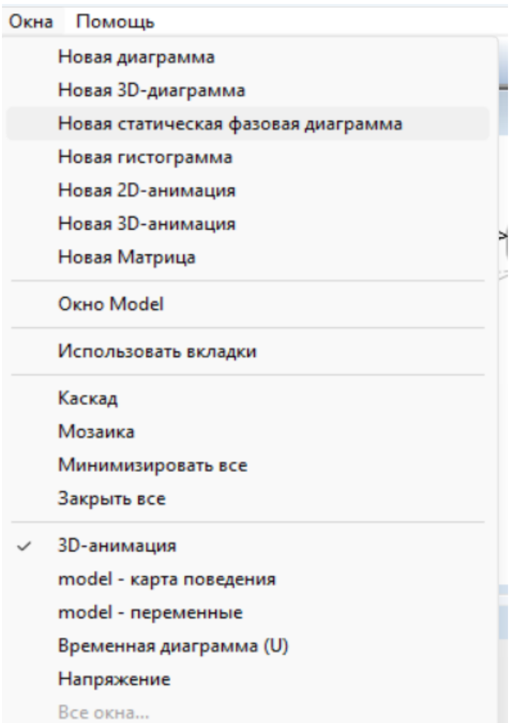


Рис. 3.1.7. Меню для создания новой 3D-анимации

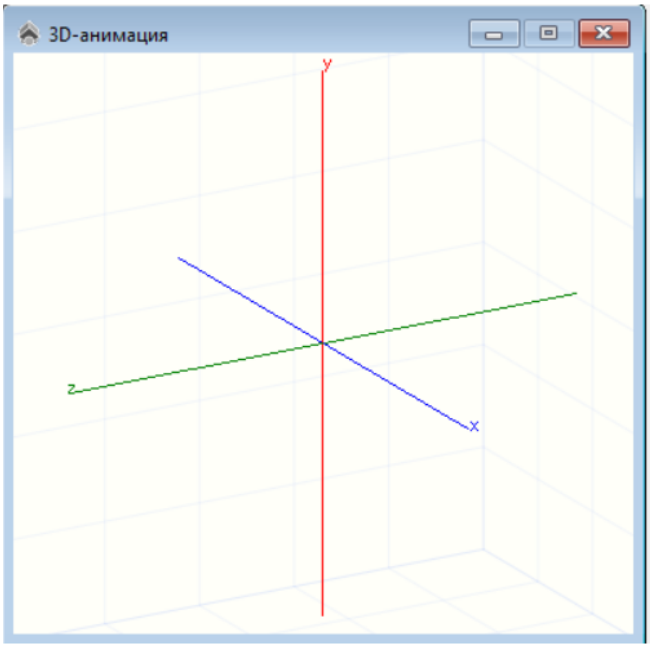


Рис. 3.1.8. Начальное состояние окна 3D-анимации

3D-анимация позволяет видеть на экране монитора движущиеся объекты, размеры и форма которых может также меняться в соответствии с изменением переменных модели. В этой работе мы создадим такую модель.

Особенность 3D-анимации состоит в том, что движущиеся объекты здесь воспроизводятся не просто как мультипликация. 3D-анимация, созданная в AnyDynamics, изображает движущиеся и изменяющиеся объекты строго в соответствии с точными уравнениями и управляющими событиями. Это позволяет проводить физические эксперименты с моделью, в которые может вмешиваться экспериментатор.

Для создания 3D-анимации необходимо щелкнуть на окне «3D-анимация» правой кнопкой и в появившемся меню выбрать пункт «Параметры 3D-анимации» (рис. 3.1.8). Появится окно «Параметры 3D-анимации». (рис. 3.1.9).

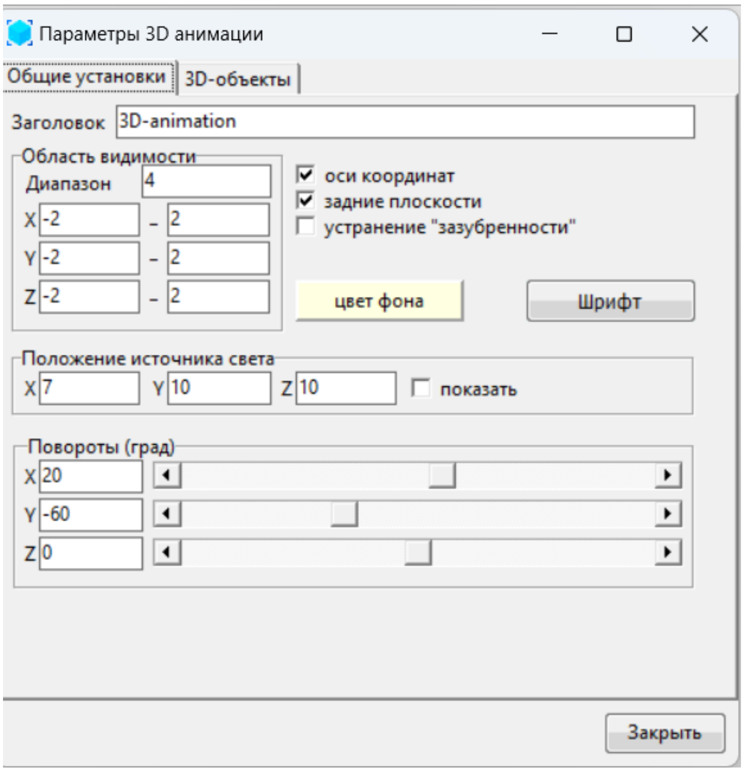


Рис. 3.1.9. Общие установки 3D-анимации

Анимация будет составлена из простейших фигур: цилиндра и конуса, которые будут изображать корпус фонарика, и конуса желтого цвета, для изображения света фонарика. В окне «Параметры 3D-анимации» (рис. 3.1.9) перейти на вкладку «3D-объекты». Щелкнуть по кнопке «Добавить цилиндр» и ввести его свойства в таблице, появившейся в правой части окна свойств (рис. 3.1.10).

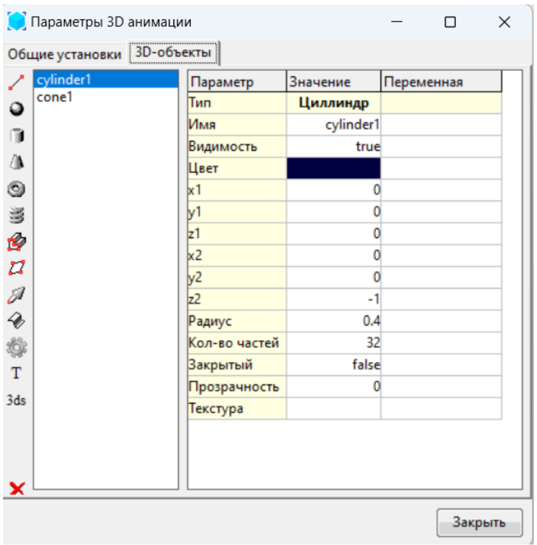


Рис. 3.1.10. Добавление цилиндра – корпуса фонарика

Аналогично добавим конус и зададим его параметры (рис. 3.1.11).

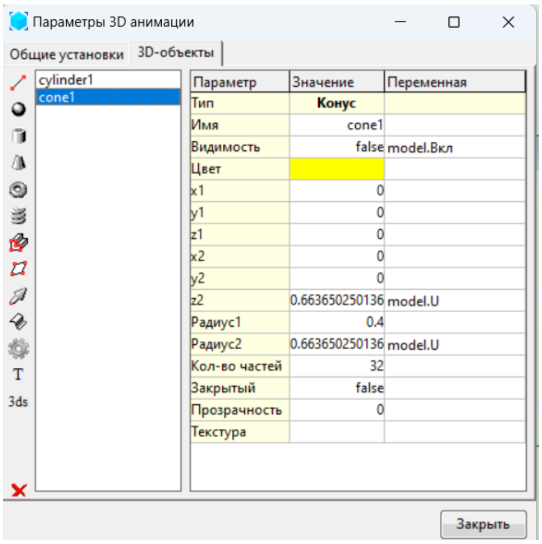


Рис. 3.1.11. Свойства светового конуса фонарика

Щелкнем по кнопке «Добавить конус» на вкладке «Объекты» в окне «Параметры 3D-анимации» задать его свойства (рис. 3.1.11). Добавим конус, который будет изображать свет и зададим его свойства.

Проведем связывание переменных со свойствами примитивов в окне «3D-анимации». В 3D-анимации изменяться со временем будет высота конуса света. Переменные проекта будут связаны со свойствами конуса «cone1». Перетащить и бросить переменную «*Вкл*» в строку «Видимость» раздела «Переменная» свойств объекта «cone1», а переменную *U* в строки z2 и Радиус2.

Закрыть окно «Параметры 3D-анимации». Создание 3D-анимации завершено. Можно приступать к тестированию модели.

Если все сделано правильно, то при запуске модели и включении фонарика, в окне «3D-анимации» появляется сноп желтого света, а при выключении фонарика свет исчезает. Фонарик можно поворачивать и рассматривать с разных сторон, схватив левой клавишей мыши и сдвигая ее. Результат моделирования представлен на рис. 3.1.12.

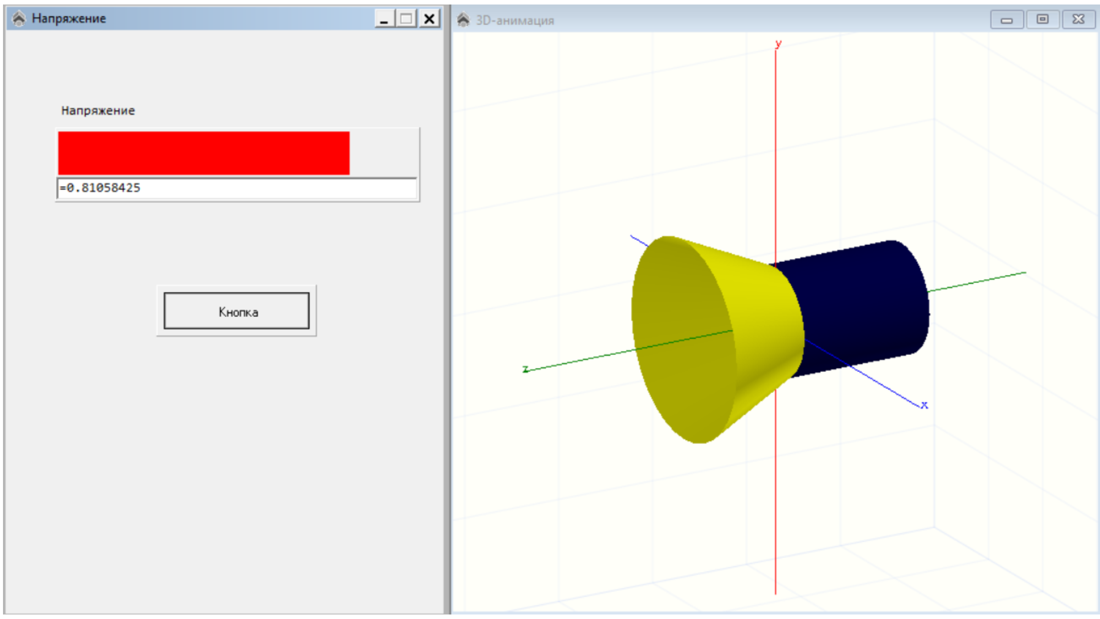


Рис. 3.1.12. Результаты моделирования с 3D-анимацией

3.2. Динамически управляемая модель электрической цепи

Имеется схема электрической цепи (рис. 3.2.1), состоящая из конденсатора, индуктивности, активных сопротивлений и лампочки. При замыкании выключателя образуется замкнутая электрическая цепь, которая определенным образом реагирует на напряжение батареи. По цепи течет электрический ток.

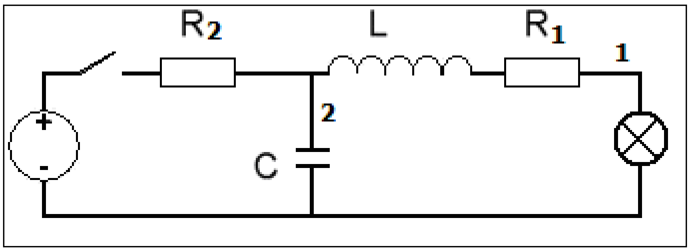


Рис. 3.2.1. Электрическая цепь

Математические модели элементов цепи (законы Ома) имеют вид:

$$u_R = R \cdot i, \quad u_L = L \cdot \frac{di}{dt}, \quad C \cdot \frac{du_C}{dt} = i .$$

Связь между элементами в замкнутой цепи описывается соотношениями (законы Кирхгоффа):

$$E = u_{R1} + u_{R2} + u_L; \quad E = u_{R2} + u_C; \quad i = i_1 + i_2.$$

В разомкнутой цепи связь между элементами цепи следующая:

$$u_C = u_{R1} + u_L; \quad i_1 + i_2 = 0.$$

Математическая модель электрической цепи при первоначальном замыкании выключателя имеет вид:

$$u_{R1} = R \cdot i_1, \quad u_{R2} = R \cdot i \;, \quad \frac{di_1}{dt} = \frac{u_L}{L}, \quad i_2 = C \cdot \frac{du_C}{dt},$$
$$i = i_1 + i_2, \quad u_L = E - u_{R1} - u_{R2}, \quad u_C = E - u_{R2}.$$

$$i_1(t=0) = 0; \;\; i_2(t=0) = 0; \;\; u_C(t=0) = 0; \;\; u_L(t=0) = 0; \;\; u_{R1}(t=0) = 0 \;\; u_{R2}(t=0) = 0.$$

Здесь $i_{1,2}$ – ток в соответствующей ветви цепи (рис. 3.2.1), u_c – напряжение на конденсаторе, u_L – напряжение на индуктивности, u_{R1}, u_{R2} – напряжения на активных сопротивлениях, R_1, R_2 – величина сопротивлений, L – индуктивность катушки, C – емкость конденсатора.

При размыкании цепи ее математическую модель составляют зависимости:

$$u_{R1} = R \cdot i_1, \quad \frac{di_1}{dt} = \frac{u_L}{L}, \quad \frac{du_C}{dt} = \frac{i_2}{C}, \quad i_2 = -i_1, \quad u_L = u_C - u_{R1}.$$

Начальное состояние в этом случае соответствует состоянию цепи в момент размыкания выключателя. При дальнейшем замыкании или размыкании выключателя начальное состояние цепи определяется ее предыдущим состоянием.

Построить AnyDynamics-модель электрической цепи, управляемую выключателем (кнопка).

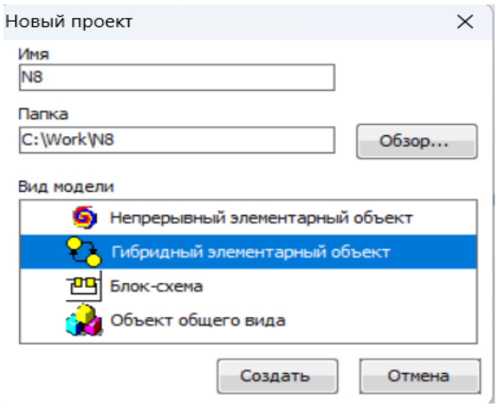


Рис. 3.2.2. Создание нового гибридного проекта

Работа выполняется в среде AnyDynamics. Создать проект с именем N8 и поместить его в доступную для записи папку, например, C:\Work (рис. 3.2.2). В результате моделирования отобразить изменение напряжений в цепи временной диаграммой.

Первоначальное состояние проекта N8 представлено на рис. 3.2.3.

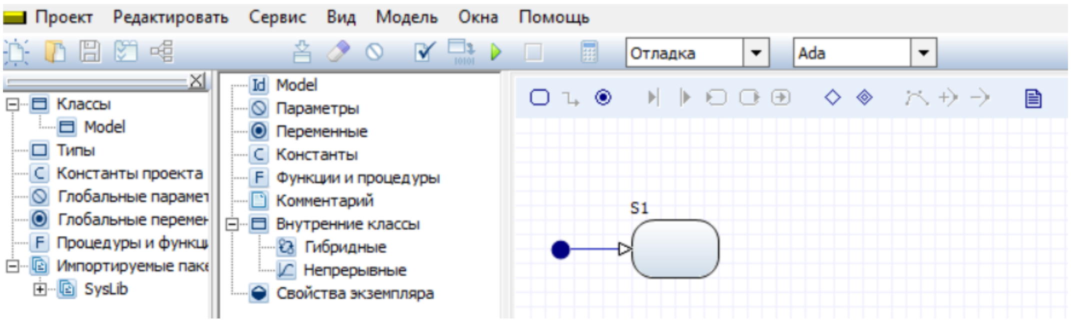


Рис. 3.2.3. Первоначальное состояние проекта N8

Необходимо добавить переменные, параметры и их значения (рис. 3.2.4).

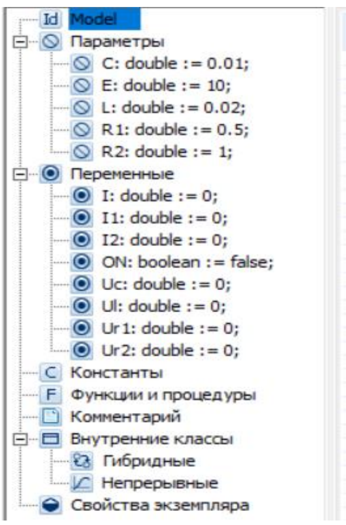


Рис. 3.2.4. Параметры и переменные проекта

Переменная «*ON*» играет роль выключателя (*ON = true* – цепь замкнута; *ON = false* – цепь разомкнута). Переменная *«ON»* – булевского типа.

Создадим математическую модель электрической цепи. Новая модель будет отражать изменение электрических параметров цепи во времени. Модель должна учитывать то обстоятельство, что при выключении цепи, поведение системы и ее математическая модель меняются. Изменение поведения отображается с помощью «Карты поведения», которая представлена на рис. 3.2.5.

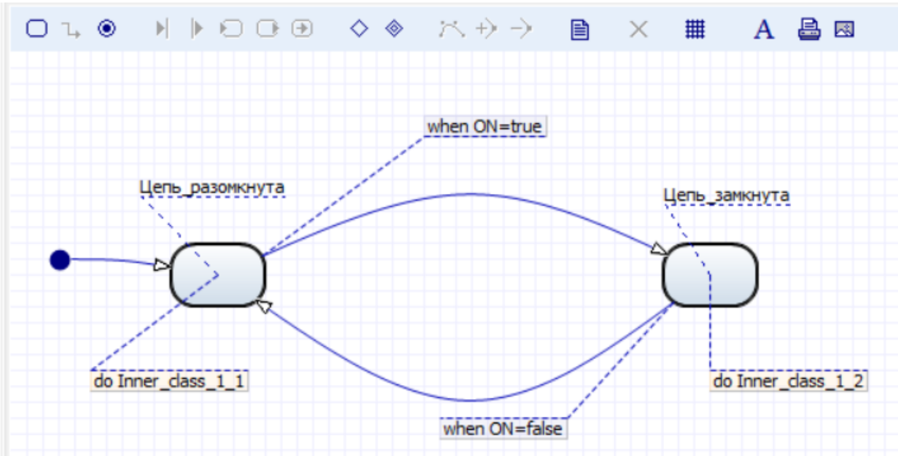


Рис. 3.2.5. Готовая карта поведения

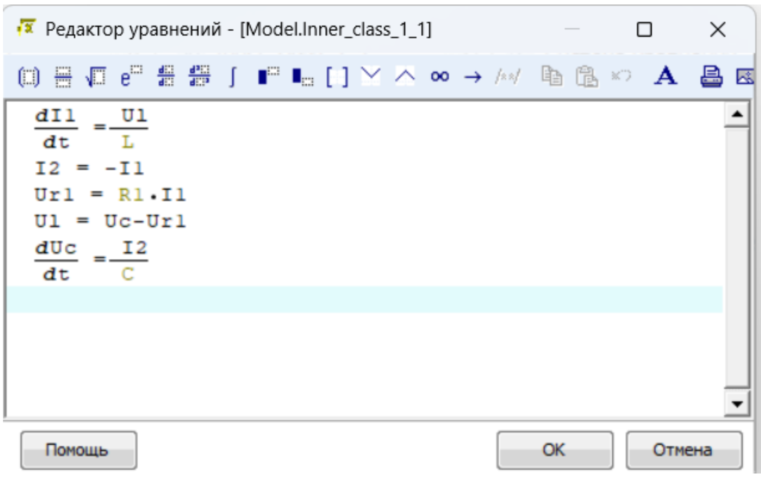


Рис. 3.2.6. Уравнения для разомкнутой цепи. Inner_class_1_1

Уравнения, описывающее непрерывное изменение переменных проекта с течением времени в узлах карты поведения, представлены на рис. 3.2.6 и рис. 3.2.7.

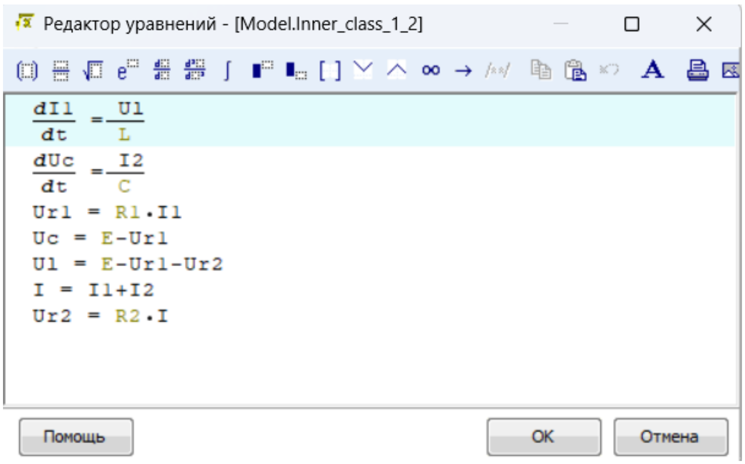


Рис. 3.2.7. Уравнения для замкнутой цепи. Inner_class_1_2

Решение задачи управления электрической цепью состоит в том, что необходимо создать такую «Карту поведения», которая обеспечит смену поведения модели цепи при ее замыкании и размыкании. Такая карта представлена на рис. 3.2.5. Переходы из состояния «Цепь_разомкнута» в состояние «Цепь_замкнута» и обратно происходят в зависимости от значения переменной *«ON»*.

Запустим визуальную модель**.** Для этого нужно, как и ранее, нажать на кнопку  в главном окне проекта. В результате появится окно «Визуальной модели» и окно «Переменных и параметров» (рис. 3.2.8).

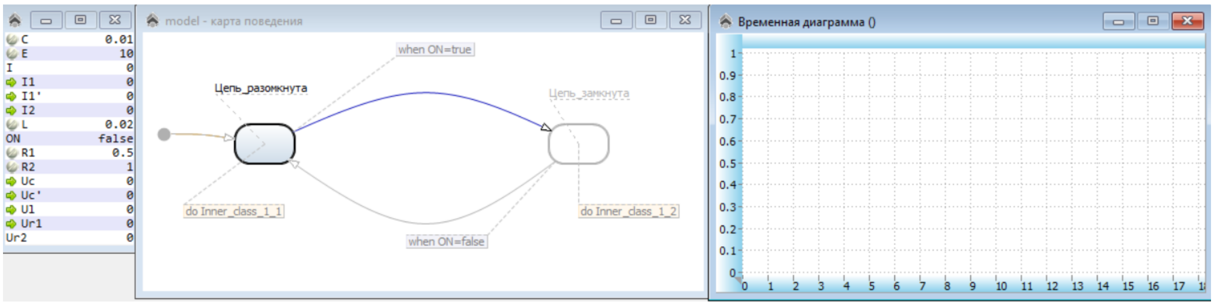


Рис. 3.2.8. Визуальная модель

В окне переменных среди прочих указана переменная *«ON»* и ее исходное значение false (рис. 3.2.8). Создадим окно для 2D-анимации. В пункте меню «Окна» выберем подпункт «Новая 2D-анимация». В меню «Сервис» выберем «Стандартные 2D-компоненты». Появится окно (рис. 3.2.9).

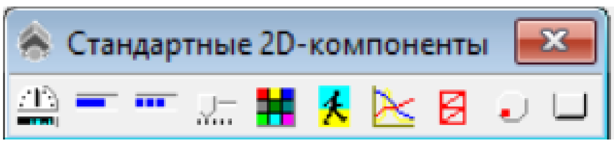


Рис. 3.2.9. Окно стандартных 2D-компонентов

Следует поместить в окно «2D-анимация» компонент «Кнопка» (рис. 3.2.10). Переменную «*ON*» необходимо связать с кнопкой путем перетаскивания. Теперь при нажатии кнопки переменная «*ON*» будет менять свое значение, и карта поведения будет менять состояние электрической цепи (замыкать и размыкать)

Запустите модель (кнопка). При нажатии кнопки переменная «*ON*» изменит свое значение, и цепь перейдет в новое состояние. Аналогичным образом добавьте в окно «2D-анимация» линейный индикатор и свяжите его с переменной **«***ON***».** Теперь при нажатии кнопки произойдет замыкание цепи.

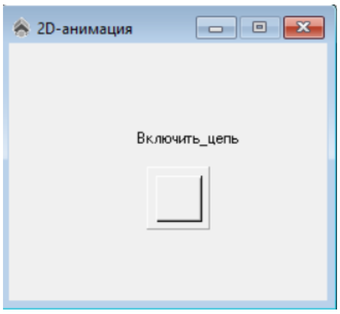


Рис. 3.2.10. Кнопка управления проектом

Переход цепи в новое состояние представлен на рис. 3.2.11.

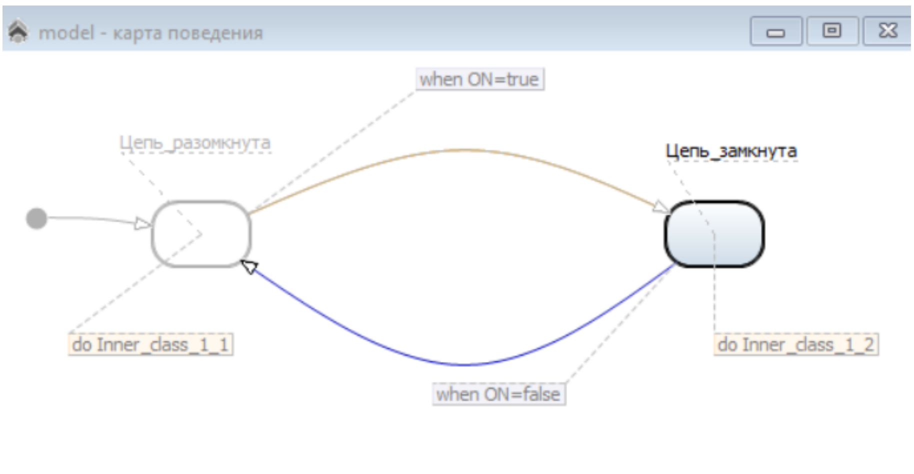


Рис. 3.2.11. Замыкание цепи при нажатии кнопки.
Переход в состояние «Цепь_замкнута»

В заключение создания модели необходимо добавить временные диаграммы, которые будут отражать изменение переменных *Ul, Uc* во времени (рис. 3.2.12).

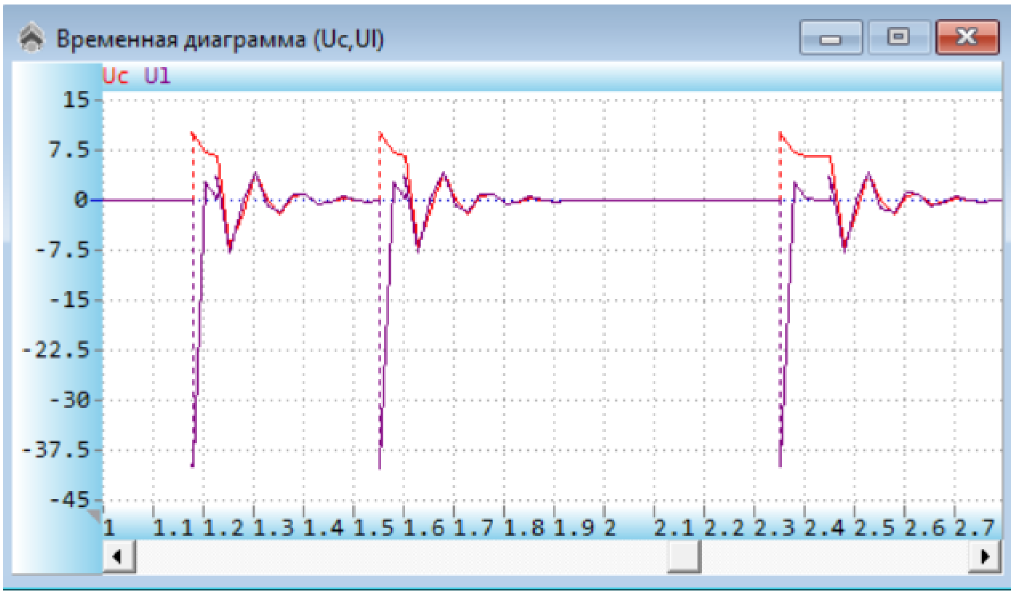


Рис. 3.2.12. Временная диаграмма изменения напряжения

На основе созданной модели исследуйте свойства электрической цепи в зависимости от ее параметров (R_1, R_2, C, L). В окне «2D-анимация» (рис. 3.2.10) добавить индикатор (лампочка), который будет менять цвет при замыкании цепи.

3.3. Моделирование колебаний маятника

Моделируемая система (математический маятник) представляет собой материальную точку (шарик достаточно малого размера), прикрепленную к нерастяжимому и невесомому стержню длиной ***L***, другой конец которого шарнирно закреплен в начале системы координат (рис. 3.3.1). Потери на трение в системе отсутствуют.

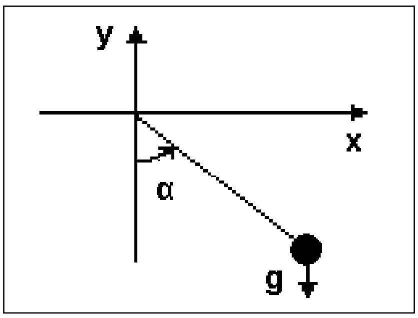


Рис. 3.3.1. Схема объекта моделирования

Динамика колебаний маятника в соответствии с основными законами механики определяется дифференциальными уравнениями:

$$\frac{d\alpha}{dt} = \omega, \quad \frac{d\omega}{dt} = \frac{-g \cdot \sin\alpha}{L},$$

с начальными условиями $\alpha(t=0) = \alpha_0, \omega(t=0) = \omega_0$: $\alpha_0 = -\frac{\pi}{2} \quad \omega_0 = 0$.

Модель содержит параметр *L* – длина стержня, константу *g* – ускорение свободного падения, переменные *α* – угол отклонения маятника и *ω* – угловая скорость колебаний. Так как AnyDynamics не имеет греческого алфавита, то вместо *α* будем использовать переменную *a*, а вместо ω переменную *w*. Для представления движения маятника средствами 3D-анимации потребуются дополнительные переменные – координаты *x* и *y* материальной точки:

$$x = L \cdot \sin(a), \quad y = -L \cdot \cos(a).$$

Средствами AnyDynamics построить модель маятника и установить влияние параметров системы на ее свойства.

После запуска AnyDynamics выполните команду главного меню «Проект\Новый…». Создадим новый проект с именем N9. Для модели выберем вид «Непрерывный элементарный объект» (рис. 3.3.2).

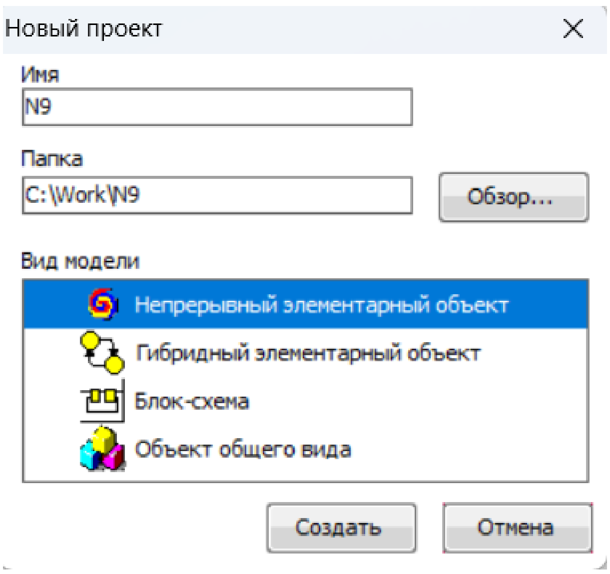


Рис. 3.3.2. Создание нового проекта

Выберите путь к папке проекта. Введите имя проекта и нажмите кнопку «Создать» (рис. 3.3.2). В данной папке будет создан файл базы данных проекта N9.mvbx. На рис. 3.3.2 для создания проекта выбрана папка Work на диске **C:**, имя проекта – «N9». При создании модели выберите доступную вам на запись папку.

После этого появятся «заготовка» проекта (рис. 3.3.3).

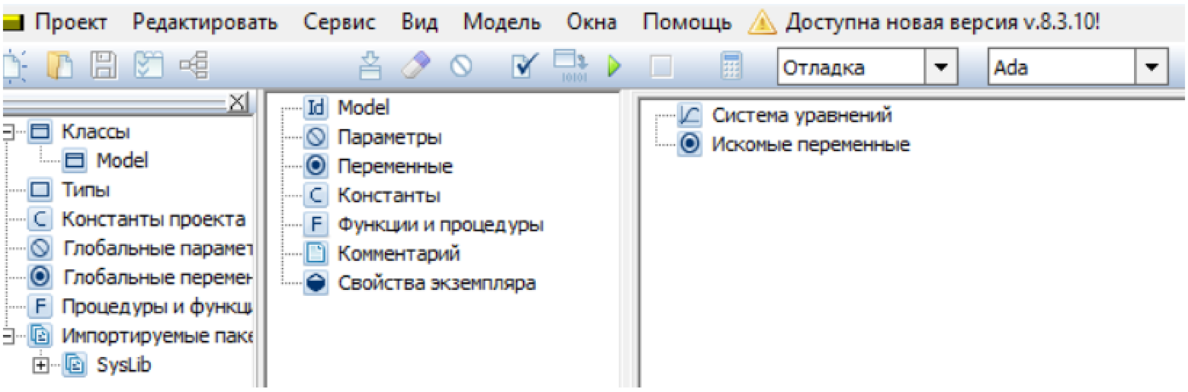


Рис. 3.3.3. «Заготовка» проекта

Модель маятника – это модель непрерывной системы. Для ее построения подходит вид моделей, создаваемый по умолчанию при открытии нового проекта. В эту модель необходимо добавить соответствующие переменные, параметры, константы и уравнения (рис. 3.3.4).

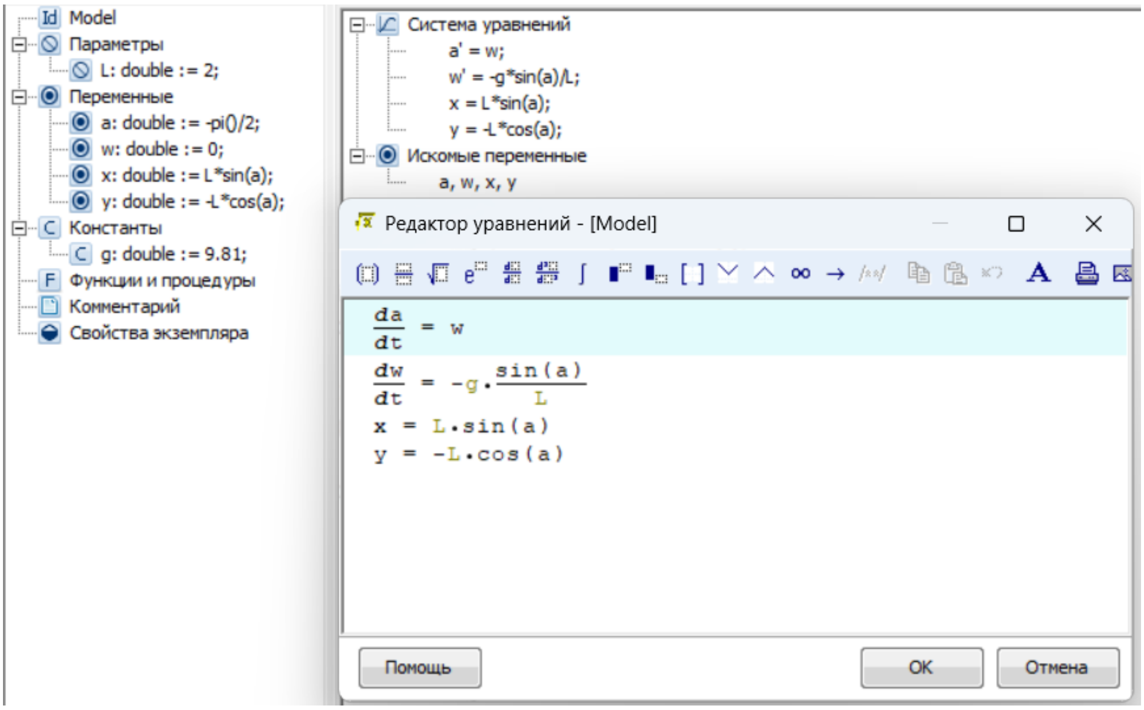


Рис. 3.3.4. Заполненное окно модели

В окне модели на узле «Система уравнений» и команды «Изменить» контекстного меню (рис. 3.3.4) вызываем «Редактор уравнений», который позволяет вводить математические выражения в виде, который близок к математической форме записи. С помощью этого редактора вводим необходимые уравнения. Специальный знак производной $\frac{d}{dt}$ вводится или с помощью кнопок на панели инструментов (рис. 3.3.4).

Запуск и создание визуальной модели производится с помощью кнопки . На рис. 3.3.5 показано главное окно визуальной модели, на котором помещены временные диаграммы: a*(t), w(t)* и *x(t), y(t).* Временные диаграммы создаются после нажатия кнопки «Новая диаграмма»  и перетаскивания переменных *a, w; x, y*.

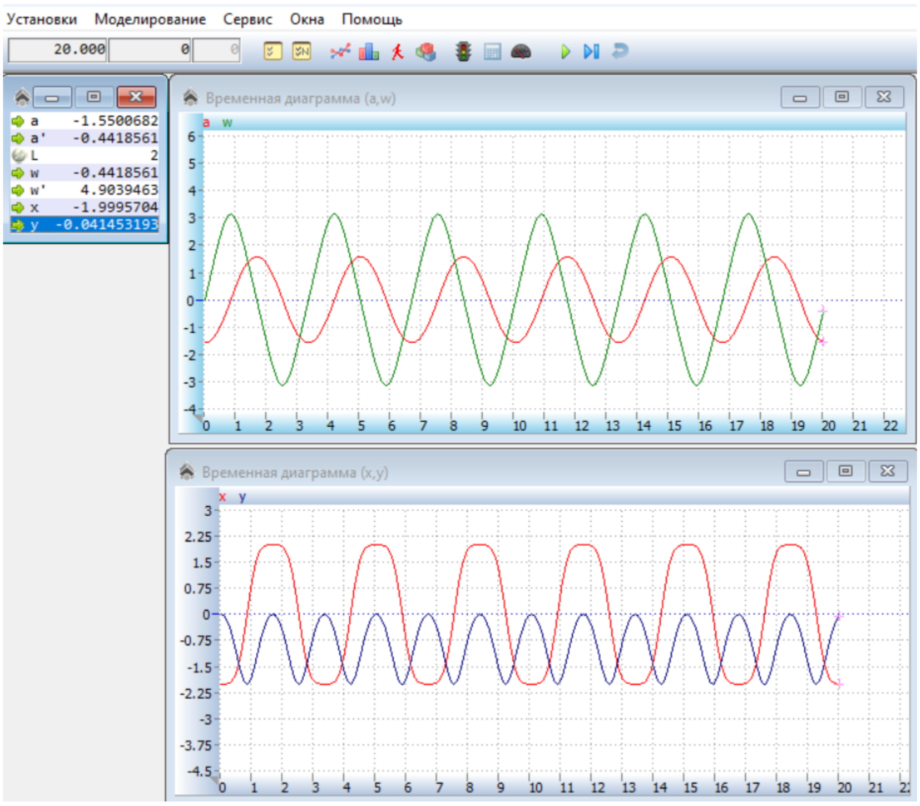
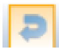


Рис. 3.3.5. Окно визуальной модели с диаграммами

Визуальная модель является многооконным приложением. В левой части инструментальной панели отображается текущее значение модельного времени (начальное значение = 0). Для модели автоматически открывается окно переменных. Обратите внимание на то, что его параметры и переменные приняли указанные значения.

Запустим выполнение модели с помощью кнопки . При этом начнет изменяться модельное время и значения фазовых переменных. Останов выполнения модели производится с помощью кнопки . Возврат модели в начальное состояние производится с помощью кнопки . Модельное время снова будет равно *0*.

Может оказаться, что уравнения решаются так быстро, что вы просто не успеваете ничего заметить. С помощью кнопки  вызовите диалог редактирования установок. На станице «Выполнение» переключите параметр «Соотношение модельного и реального времени» из положения «Так быстро как возможно» в положение «число» (по умолчанию это 1.000E+00), то есть моделирование в реальном масштабе времени. Изменяя это число, вы можете ускорять или замедлять прогон модели (рис. 3.3.6).

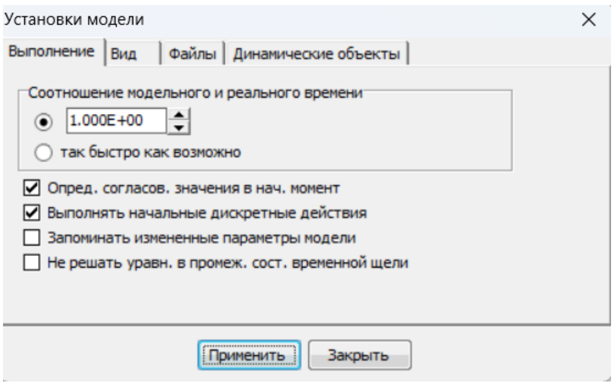


Рис. 3.3.6. Установки модели

Построим фазовую диаграмму, т.е. график зависимости *w(a)*. Для этого создадим новую диаграмму, перетащим в нее переменные *w* и *a*, а затем правой клавишей мыши откроем на ней контекстное меню и выполним команду «Настройка параметров диаграммы». В появившемся диалоге настроек укажем с помощью двойного щелчка мышью в поле X, что по оси абсцисс откладываются значения переменной *a* (рис. 3.3.7). Запустив модель, получим следующий график (рис. 3.3.8) – фазовую диаграмму.

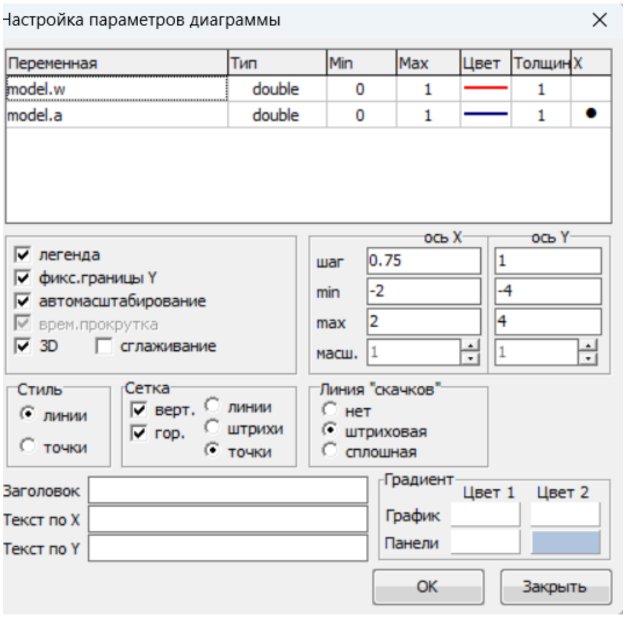


Рис. 3.3.7. Настройка параметров диаграммы

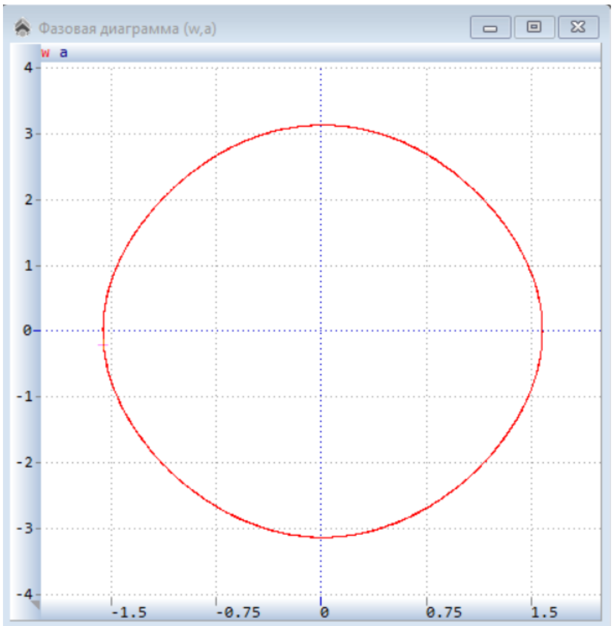


Рис. 3.3.8. Фазовая диаграмма *w(a)*

Можно получить больше информации из непосредственного наблюдения поведения трехмерного изображения моделируемой системы. В визуальной модели для этого предназначено окно 3D-анимации. Создать его можно с помощью команды главного меню «Окна» – «Новая 3D-анимация» (рис. 3.3.9).

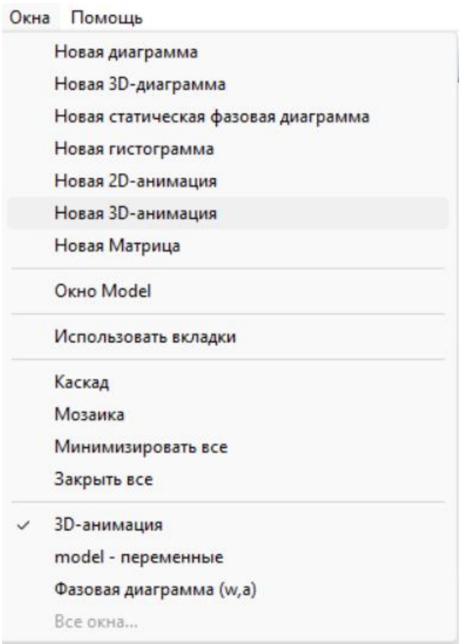


Рис. 3.3.9. Создание новой 3D-анимации

Окно 3D-анимации (рис. 3.3.10) позволяет строить динамические трехмерные модели, используя совокупность трехмерных примитивов (линия, шар, цилиндр, конус и т.д.), параметры, которых связываются со значениями соответствующих переменных модели.

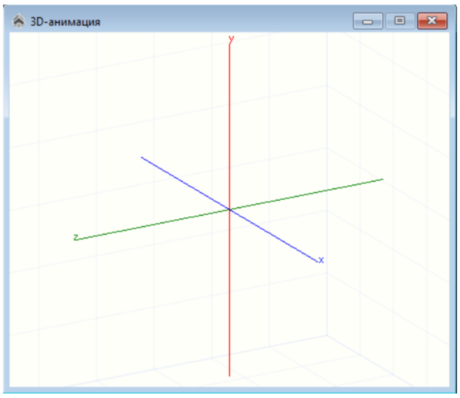


Рис. 3.3.10. Начальное состояние окна 3D-анимации

С помощью команды «Параметры» контекстного меню вызовем диалог редактирования свойств 3D-анимации. В данной модели нам понадобится только два стандартных объекта – отрезок (line) и сфера (sphere).

Один конец линии должен всегда находиться в начале координат (параметры *x1 = 0, y1 = 0*), а координаты второго конца (параметры *x2, y2*) должны изменяться в соответствии со значением переменных ***X*** и ***Y*** маятника. Для задания этого соответствия «перетащим» необходимые переменные из окна переменных и бросим их в колонке «Переменная» соответствующих параметров отрезка. Аналогичным образом этим же переменным x, *y* мы сопоставляем координаты центра сферы (параметры *x1, y1*) (рис. 3.3.11).

После чего достаточно запустить модель, и вы увидите качающийся маятник (рис. 3.3.12). В любой момент вы можете изменить точку наблюдения, нажав левую клавишу мыши и перемещая ее с прижатой клавишей. Таким образом, вы можете рассматривать колебания маятника сверху, снизу и т.д.

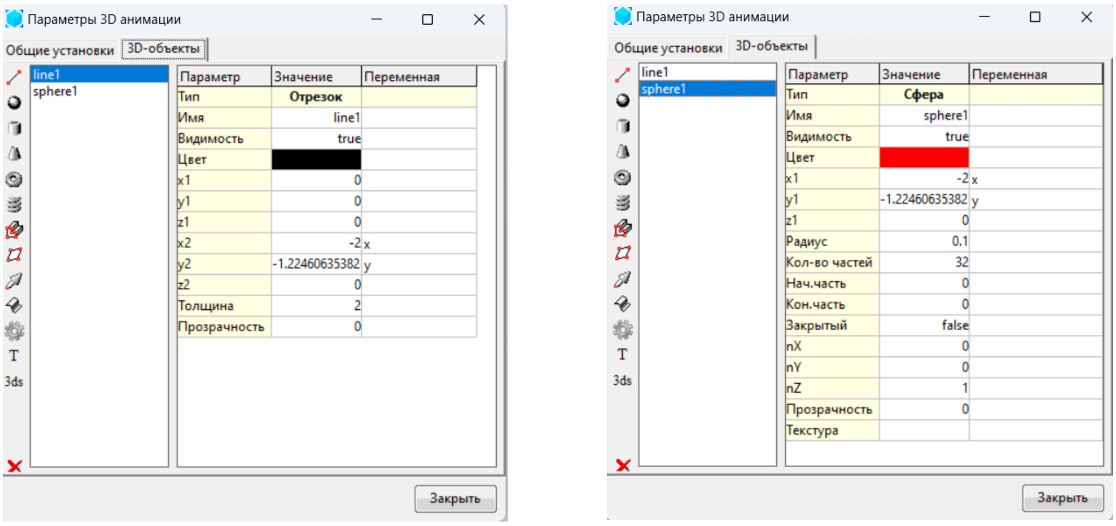


Рис. 3.3.11. Параметры 3D-объектов

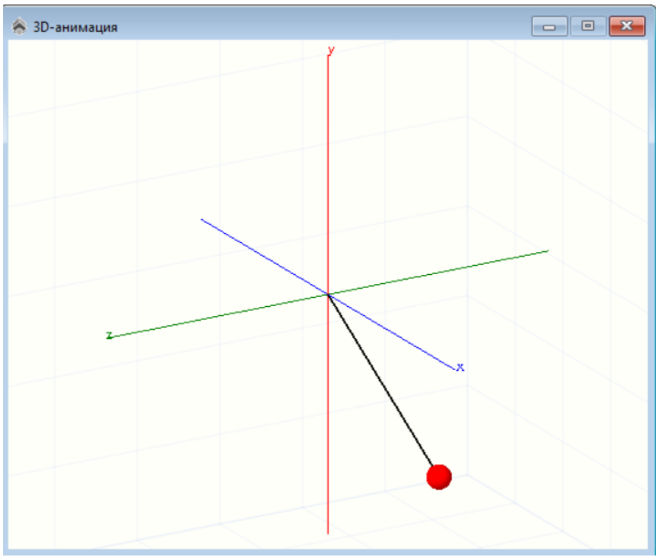


Рис. 3.3.12. Представление результатов моделирования средствами 3D-анимации

Провести компьютерные эксперименты и установить влияние параметров объекта на период колебаний маятника.

3.4. Построение модели отрывающегося маятника

Данная модель будет построена по аналогии с моделью «Моделирование колебаний маятника». Построить модель маятника, который отрывается от нити в определенной фазе колебаний. Моделируемая система является гибридной, т.к. она имеет три различных непрерывных поведения: «Колебания», «Полет» и «Останов».

Новый проект создадим на основе разработанного ранее проекта «N9.mvbx». Новый проект N10 имеет вид «Гибридный элементарный объект» (рис. 3.4.1).

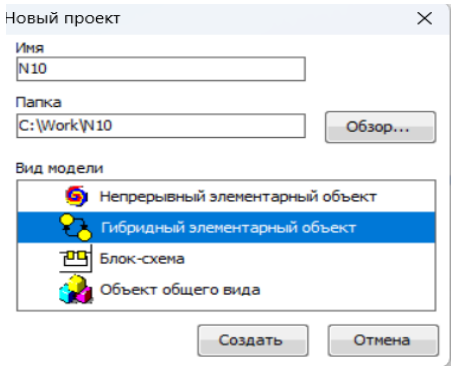


Рис. 3.4.1. Создание нового проекта

Новая модель – модель изолированной системы с несколькими различными качественными состояниями: «Колебания», «Полет», «Останов». Каждое из состояний подчиняется своим законам. Для описания переходов из одного состояния в другое потребуется создание «Карты поведения».

Добавим в модель параметры, переменные и константы, которые будут необходимы в дальнейшем (рис. 3.4.2).

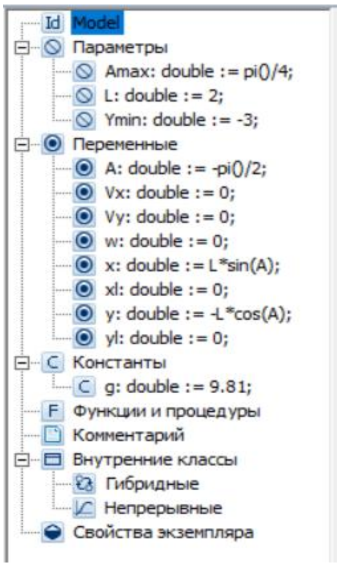


Рис. 3.4.2. Параметры, переменные и константы проекта

Новые параметры: *Ymin, Amax*, новые переменные: *Vx, Vy, xl, yl*.
Создадим **«**Карту поведения**»** (рис. 3.4.3).

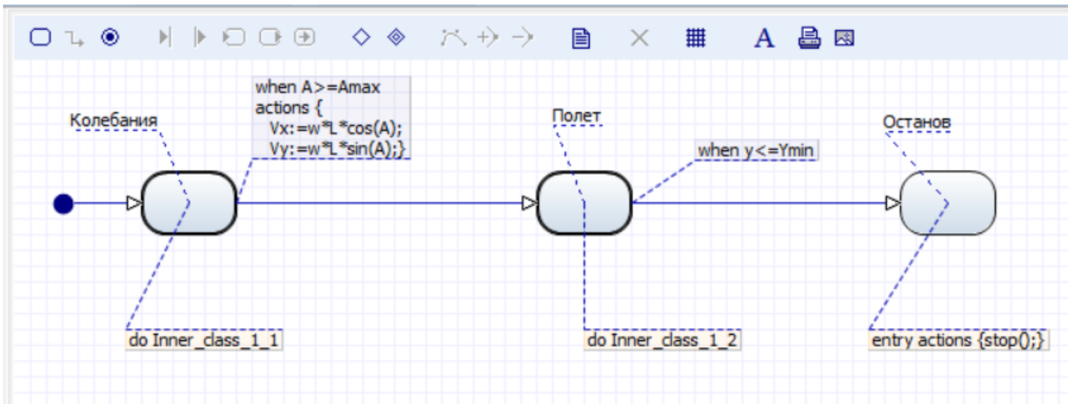


Рис. 3.4.3. Карта поведения

Непрерывное поведение в узлах «Колебания» и «Полет» задается уравнениями в «Inner_class_1_1» и «Inner_class_1_2» (рис.3.4.4 и рис. 3.4.5).

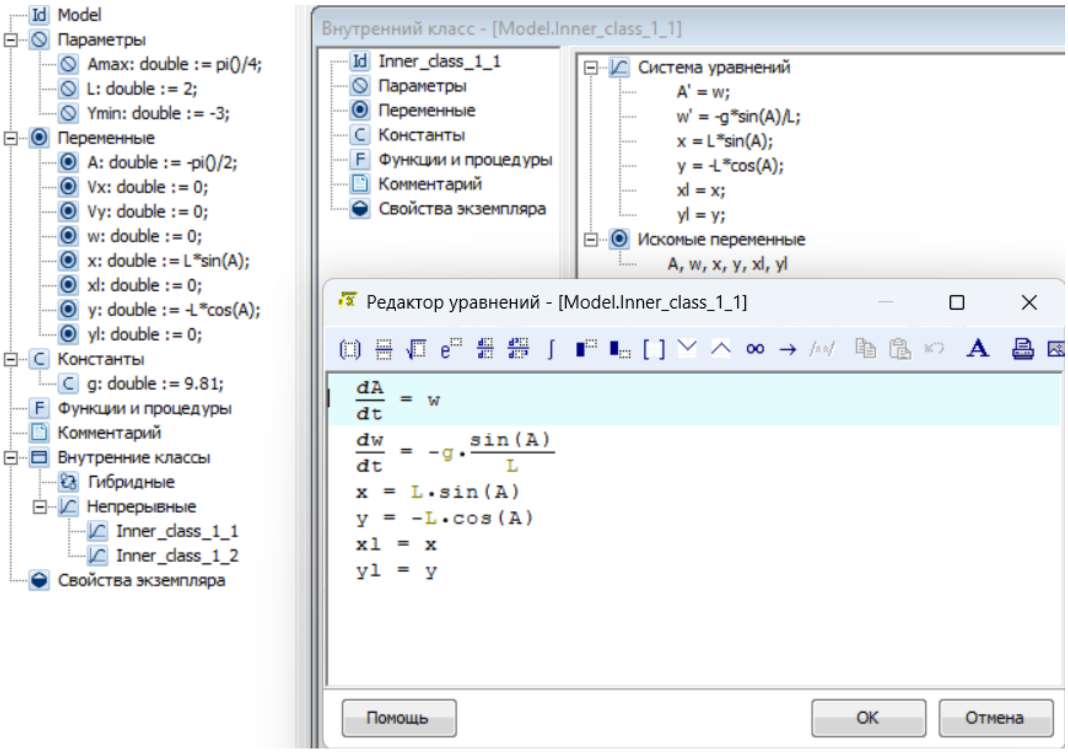


Рис. 3.4.4. Inner_class_1_1

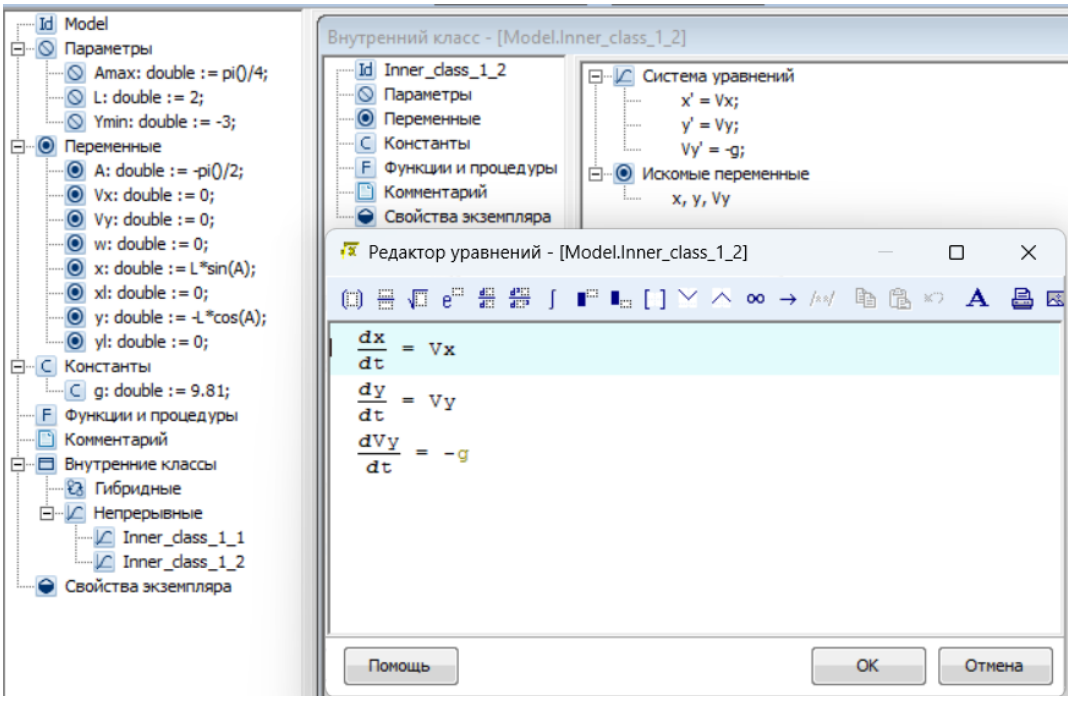


Рис. 3.4.5. Inner_class_1_2

Результат моделирования и фазовая диаграмма *y(x)* представлены на рис. 3.4.6.

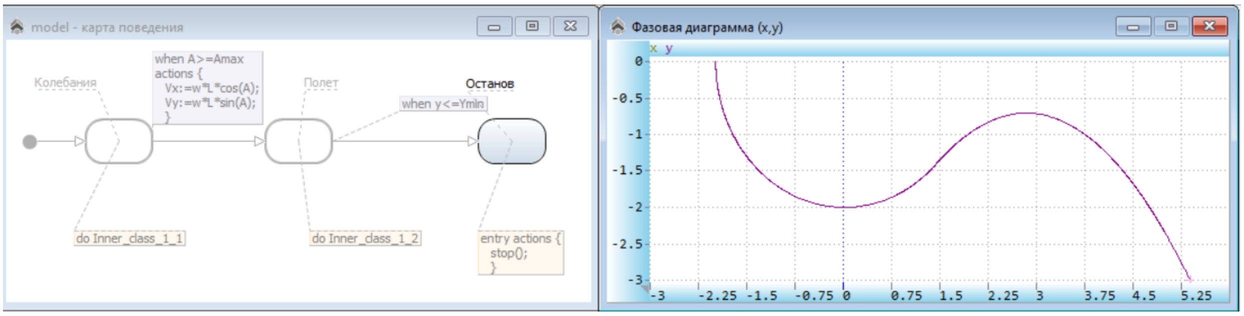


Рис. 3.4.6. Результат моделирования колебаний и отрыва маятника

Узел «Останов» содержит в своих входных действиях вызов предопределенной процедуры Stop(). Действия в переходе, входные действия и условия срабатывания перехода задаются с помощью контекстного меню.

Самостоятельно создайте 3D-анимацию колебаний и отрыва маятника. Для обеспечения корректной работы 3D-анимации в систему уравнений колебаний введены переменные: *xl* и *yl*, которые являются координатами свободного конца стержня маятника. Переменные *x*, *y* являются координатами шарика (рис. 3.4.7).

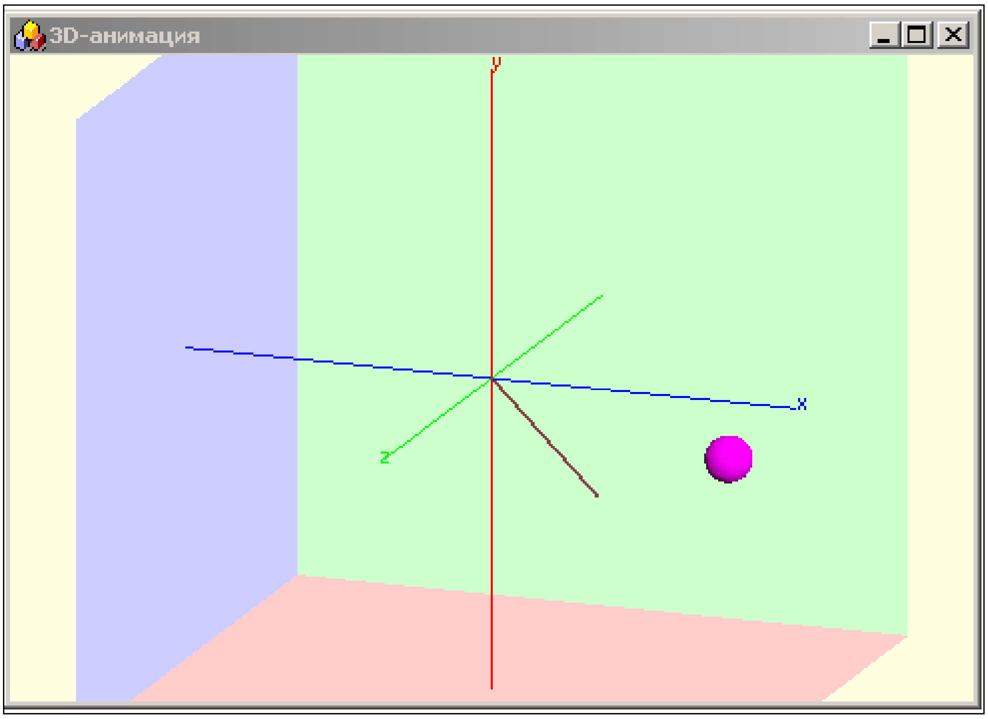


Рис. 3.4.7. 3D-анимация поведения маятника

Выполнить серию компьютерных экспериментов с построенной моделью, в которых следует изменить условия срабатывания переходов.

3.5. Управление пушечным огнем

В данной работе построим модель «Орудийный выстрел». Каждый выстрел орудия характеризуется начальной скоростью снаряда и углом наведения орудия. Выстрел выполняется по команде «Огонь». По команде «Отбой» орудие заканчивает участие в сражении. Создадим компьютерную модель, которая позволяла бы задать конкретные параметры стрельбы, подавать команды «Огонь» и «Отбой», менять угол стрельбы и наблюдать траекторию полета снаряда.

С точки зрения динамики полета снаряд, в первом приближении, будем считать материальной точкой, брошенной под углом α к горизонту с заданной начальной скоростью V_0. Сопротивлением воздуха пренебрегаем. Последнее допущение позволяет использовать простые зависимости, которые хорошо известны из школьного курса физики:

$$x(t)=x_0+V_x\cdot t \quad y(t)=y_0+V_y\cdot t$$

$$V_x=V_0\cdot\cos(\alpha) \quad V_y=V_0\cdot\sin(\alpha)-g\cdot t.$$

Угол стрельбы выбирается наводчиком, то есть при каждом залпе может быть задано новое конкретное значение.

Построить AnyDynamics-модель (рис. 3.5.1) орудийного выстрела, в которой предусмотрены средства управления огнем и отображается траектория полета снаряда. Модель будет гибридной с именем N12 (рис. 3.5.1).

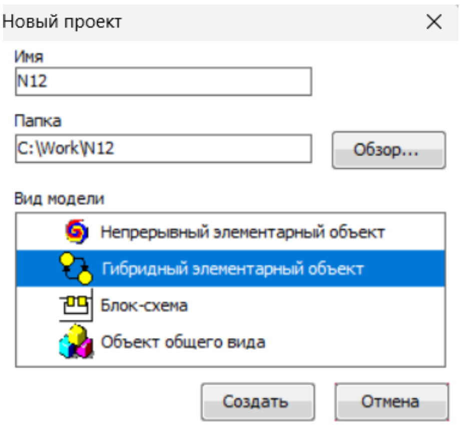


Рис. 3.5.1. Новый проект

Запустите программу AnyDynamics и создайте новый проект с помощью команд основного меню «Проект/Новый». Имя проекта – «N12».

Зададим имена переменных, которые будут характеризовать свойства объекта. Для описания полета снаряда нам понадобятся переменные – угол стрельбы *Alpha*, координата *x*, координата *y*, горизонтальная скорость *Vx*, вертикальная скорость *Vy*, начальная скорость V_0. Полный набор констант и переменных проекта представлен в таблице 3.5.1.

Таблица 3.5.1

Константы и переменные проекта

Имя проекта N12	
Константы	**Значение**
g	9.81
x0	0
y0	0

Переменные	**Начальное значение**
X	x0
Y	y0
Vx	V0 cos(Alpha)
Vy	V0 sin(Alpha)
V0	100
Alpha	Rad(Alpha0)
Alpha0	45°
К_огонь	false
К_отбой	false

Константы в экспериментах никогда не будут менять своего значения. Переменной *Alpha* задано начальное значение *rad(Alpha0)****.*** Функция *rad* переведет значение *Alpha0* из градусной меры в радианы. Так как *Alpha0* в экспериментах с моделью будет меняться, то пересчитываются все переменные, зависящие от *Alpha0*. Полное описание значений констант и переменных представлено на рис. 3.5.2.

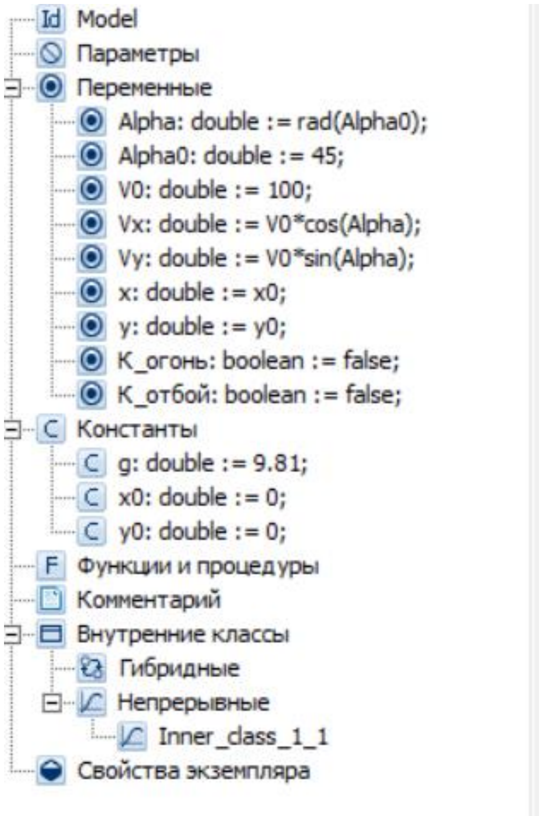


Рис. 3.5.2. Константы и переменные проекта

Рассмотрим поведение объекта. Модель, которую предстоит построить, будет отображать сложное поведение. Полет снаряда начинается по команде «К_огонь» и заканчивается приземлением снаряда. После чего орудие произвольно долго ожидает либо новой команды «К_огонь» и повторяет залп, либо команды «К_отбой». Такое поведение называется гибридным.

Поведение орудия состоит из длительного ожидания команды, при этом никакие переменные не меняются. Полет снаряда продолжается до момента приземления ($y <= 0$ *и* $Vy < 0$). При полете снаряда переменные *x, y, Vx, Vy* меняют свои значения, подчиняясь известным законам движения (уравнения представлены выше).

Поведение орудия можно отобразить графически с помощью карты поведения. «Карта поведения» – это схема, на которой изображаются длительные действия в виде узлов и стрелок, указывающих последовательность смены поведения (рис. 3.5.3).

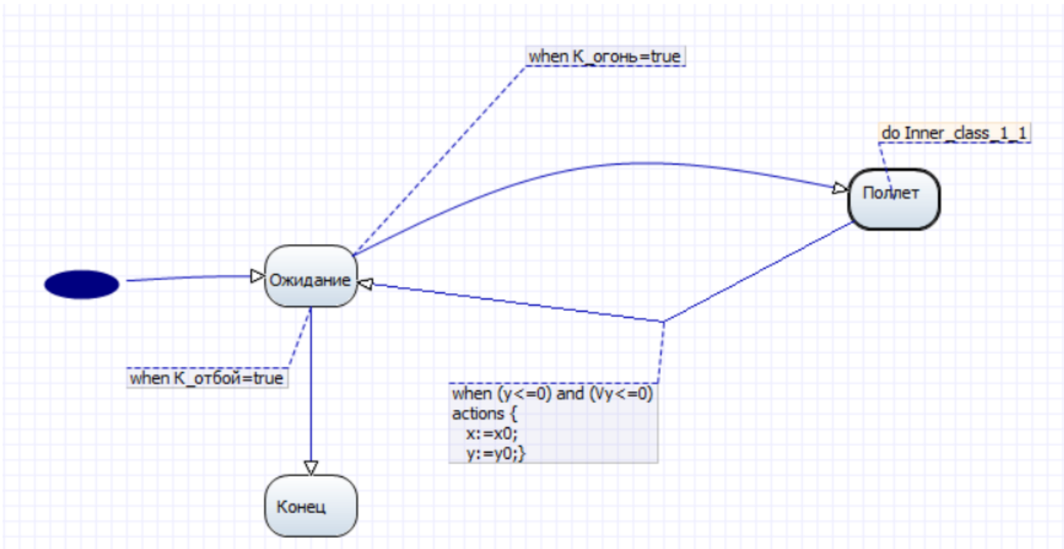


Рис. 3.5.3. Карта поведения

Длительные действия (узлы) поименованы «Полет», «Ожидание», «Конец». Рядом со стрелками переходов записаны условия срабатывания переходов, т.е. окончания длительных действий.

Каждому длительному действию предпишем законы. Необходимо также перечислить все условия окончания длительных действий и все подготовительные операции, необходимые для начала новых длительных действий (таблица 3.5.2).

Таблица 3.5.2

Условия окончания действий

Условия окончания действия	**Подготовительные операции к новому длительному состоянию**
Полет снаряда**: y <=0 И Vy <0**	**Ожидание:** занять исходную позицию **(x0,y0)**
Ожидание: **«К_Огонь=true»**	**Полет:** вычислить значения ***Vx,Vy*** и **Alpha**
Ожидание: «**К_Отбой= true**»	**Конец**: stop

Введем уравнения модели (рис. 3.5.4).

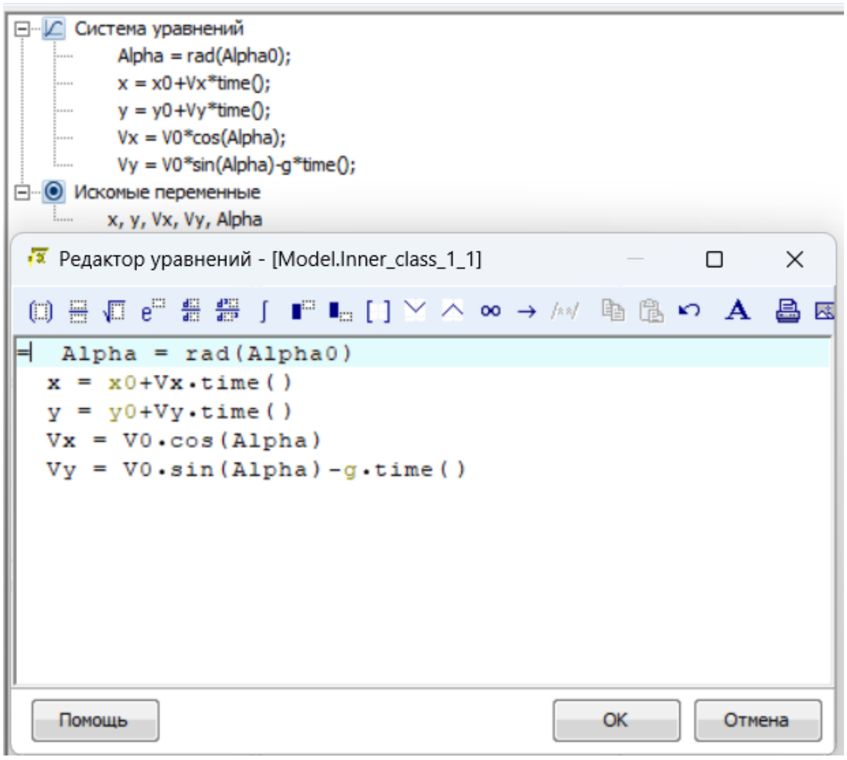


Рис. 3.5.4. Система уравнений полета снаряда (Inner_class_1_1)

Результаты выстрелов для разных значений *Alpha0* отображены на рис. 3.5.5.

Рис. 3.5.5. Результат моделирования
(траектории при разных значениях *Alpha0*)

Управление стрельбой можно создать средствами 2D-анимации (рис. 3.5.6). Кнопки следует связать с переменными *«К_огонь»* (кнопка «Огонь») и *«К_отбой»* (кнопка «Отбой»), а ползунок связать с переменной *Alpha0*.

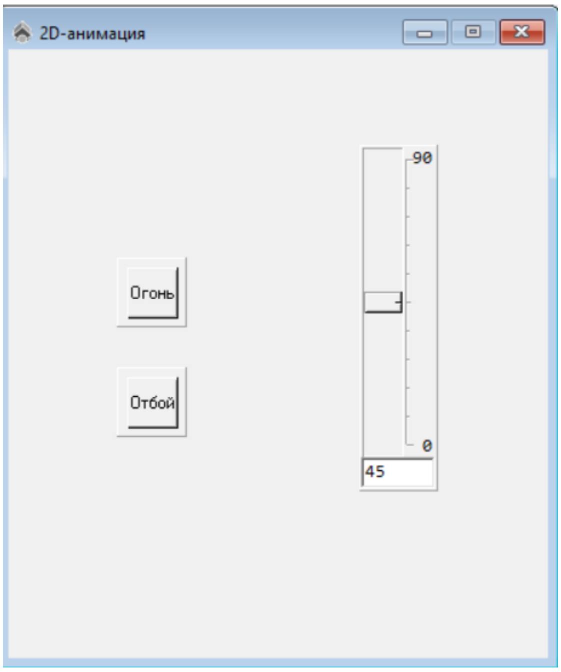


Рис. 3.5.6. Средства управления моделью

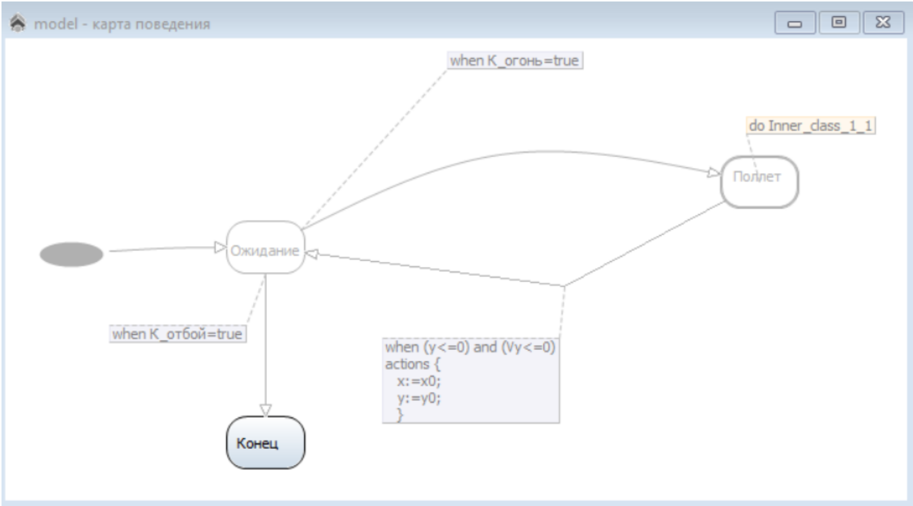


Рис. 3.5.7. Состояние модели после нажатия кнопки «Отбой»

Нажатие кнопки имитирует подачу команды. При нажатии кнопки «Отбой» модель перейдет в состояние по рис. 3.5.7. Кроме того, для управления дальностью стрельбы введен ползунок, который позволит плавно изменять угол стрельбы (значение переменной *Alpha0*) для попадания в цель.

Проведите эксперименты с моделью, изменяя угол стрельбы, добейтесь поражения заданной цели методом пристрелки (методом проб и ошибок, допустимое отклонение 5 метров).

3.6. Моделирование дискретного развития популяции

Во многих нелинейных динамических системах (биологические, метеорологические, экономические системы и т.п.) можно обнаружить развитие неустойчивости. Известно, что нелинейная система в случае ее неустойчивого поведения может либо выйти на новое устойчивое состояние, либо перейти в режим автоколебаний, либо перейти в режим хаотического поведения. На достаточно продолжительных промежутках времени при развитии хаотического поведения состояние систем не предсказуемо.

В 1977 году развитие хаоса было обнаружено при моделировании динамики биологических популяций. В некоторых случаях численность популяции представляет собой дискретную функцию времени. Такие популяции развиваются путем полной смены поколений. Подобные закономерности имеют место, например, у растений и насекомых.

Если предположить, что численность данного поколения зависит от численности предыдущего поколения, то для описания динамики численности популяции можно применить уравнение вида:

$$M = F(N).$$

Здесь M и N – численности двух последовательных поколений популяции (N – «родители», M – «потомки»). Причем «родители» и их «потомки» не живут во одно и тоже время, т.е. происходит полная смена поколений. На следующем этапе «потомки» становятся «родителями» и т.д. Модель развития подобной популяции может быть получена как дискретный аналог логистического уравнения:

$$M = RN(1 - N / k).$$

Здесь R – коэффициент размножения, k – экологическая емкость среды, N – численность поколения «родителей», M – численность поколения «потомков». Следуя данной модели, численность потомков пропорциональна численности родителей и относительной части свободного жизненного пространства. Логистическое уравнение отражает процесс внутривидовой конкуренции, при этом при малой численности внутривидовая конкуренция практически отсутствует, а при максимальной численности она становится определяющим фактором.

В среде AnyDynamics построить компьютерную модель развития популяции и исследовать свойства модели развития популяции при различных значениях R.

Требуется выявить параметры системы, при которых могут возникать: монотонное, колебательное приближение численности популяции к состоянию равновесия, автоколебания, хаотическое поведение системы. Доказать, что в режиме хаотического поведения малое изменение начального состояния приводит к существенному изменению дальнейшего поведения.

Рассмотреть варианты развития систем с различными значениями параметра R при прочих равных условиях. Для режима хаотического поведения необходимо сравнить развитие систем с близкими начальными состояниями (так называемый «эффект бабочки»). Малое воздействие на неустойчивый объект приводит к большим изменениям в его поведении.

Создадим новый проект с именем N13. Вид проекта – «Гибридный элементарный объект».

Введем параметры, переменные и их значения (рис. 3.6.1).

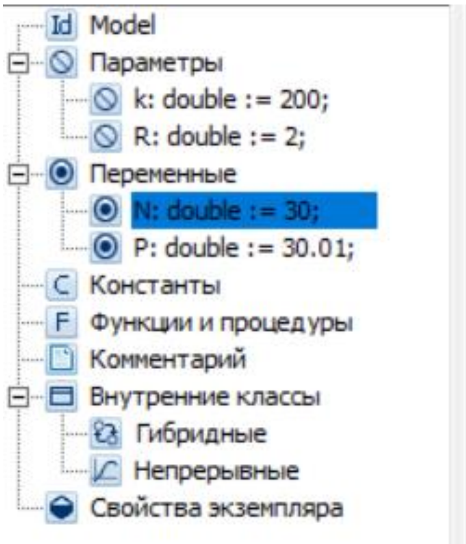


Рис. 3.6.1. Параметры и переменные проекта

Создадим **«Карту проведения»** (рис. 3.6.2).

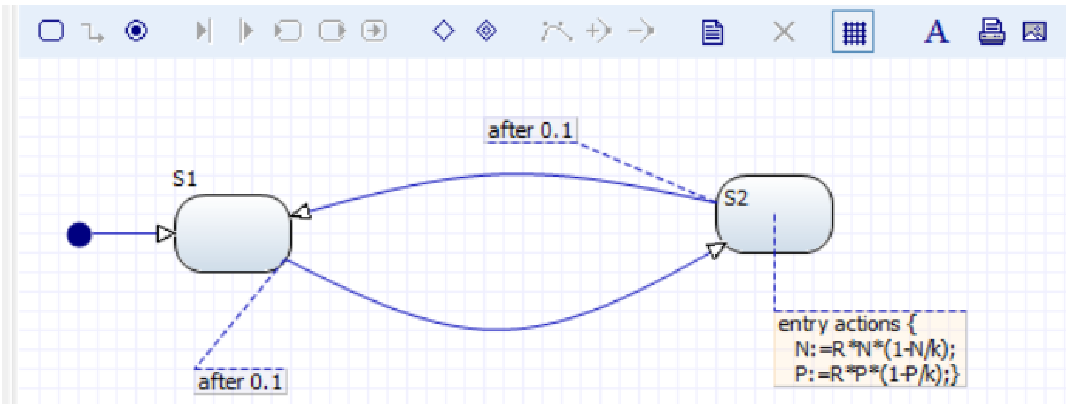


Рис. 3.6.2. Карта поведения модели

В узле *S1* установлено пустое поведение.

При *R = 2* имеет место монотонный рост численности популяции (рис. 3.6.3).

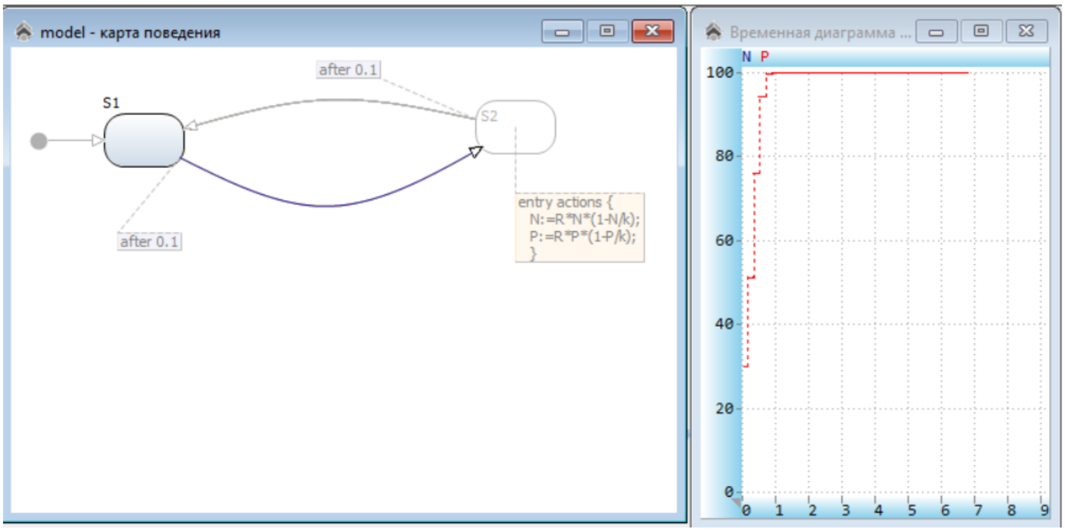


Рис. 3.6.3. Рост популяции при *R = 2*

При *R = 4* наблюдается хаотичное поведение (рис. 3.6.4).

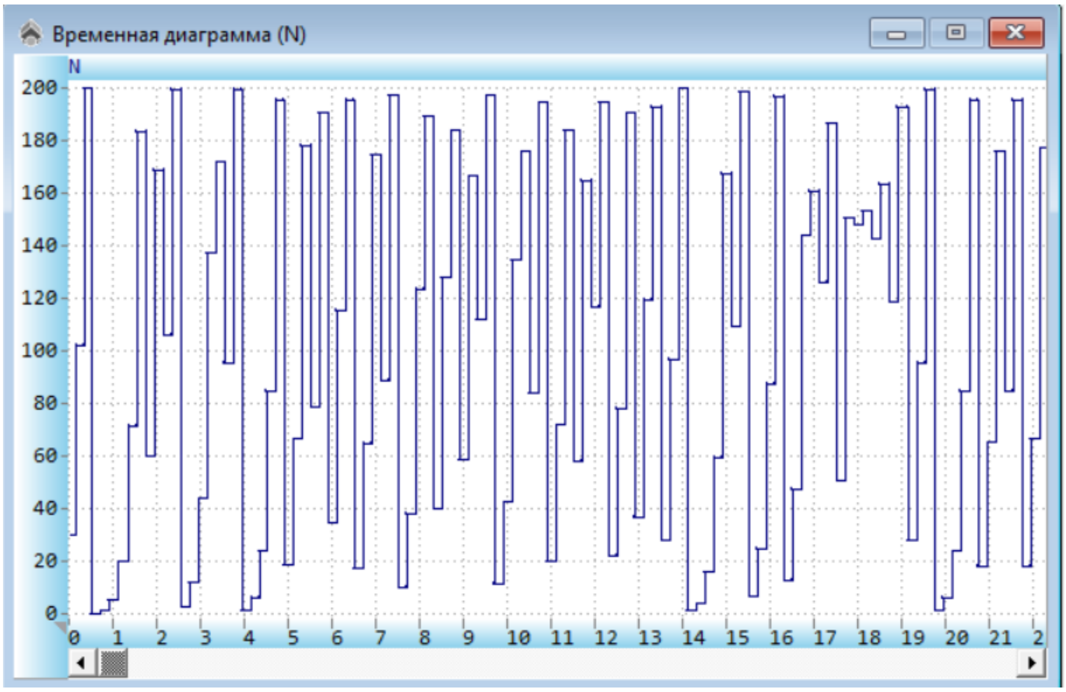


Рис. 3.6.4. Рост популяции при *R = 4*

При хаотичном поведении популяции (рис. 3.6.4) реализуется неустойчивое поведение (рис. 3.6.5). С течением времени состояния, которые мало отличаются в начальной численности, существенно различаются. Таким образом, моделирование неустойчивых систем на длительном промежутке времени дает неадекватные результаты.

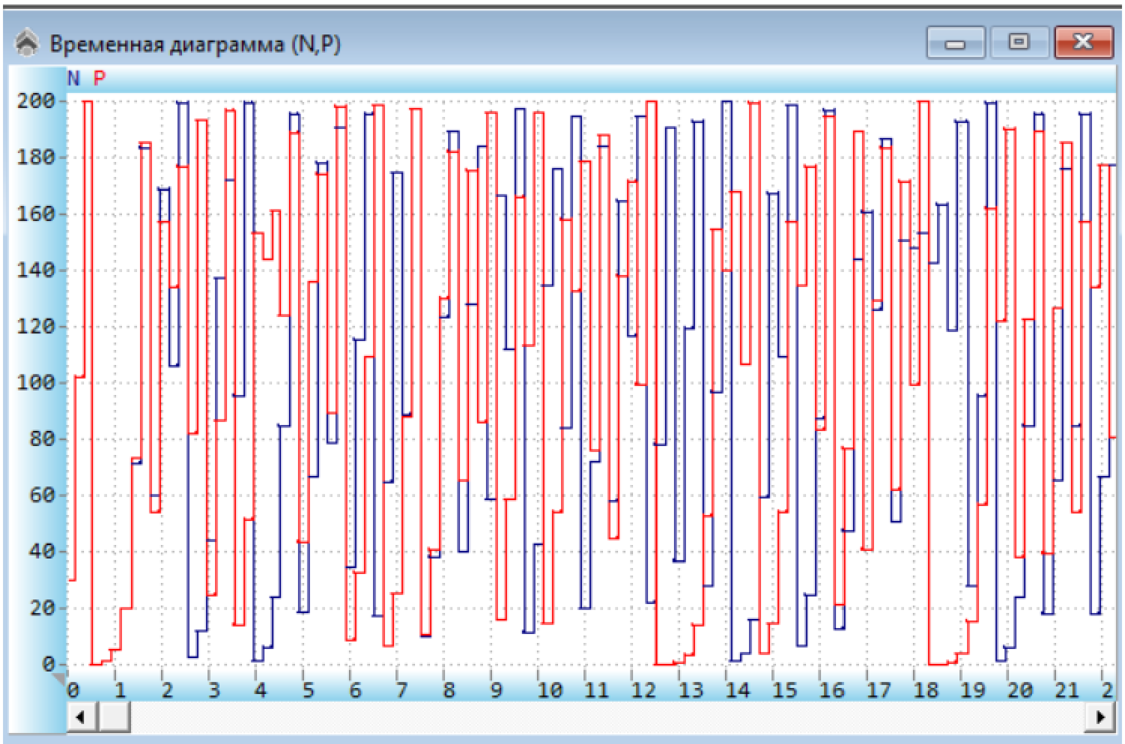


Рис. 3.6.5. Рост популяций при *R = 4* на длительном интервале времени, которые слабо отличаются в начальной численности

Глава 4. Моделирование случайных процессов

4.1. Моделирование случайного блуждания

В данной работе моделируется случайное изменение состояния объекта. Состояние объекта характеризуется двумя переменными *X* и *Y*. Изменение состояния объекта связано с изменением значений этих переменных, которое происходит случайно и дискретно. Каждая переменная может: остаться неизменной, либо изменить свое значение на *+1* или на *–1*.

Таким образом, возможна реализация 9 событий, которые образуют полную группу:

$$X, Y;\; X, Y+1;\; X, Y-1;$$
$$X+1, Y;\; X+1, Y+1;\; X+1, Y-1;$$
$$X-1, Y;\; X-1, Y+1;\; X-1, Y-1.$$

Вероятность изменения каждого параметра задана: *PXa, PXb, PXc* **;** *PYa, PYb, PYc*. Причем *PXa + PXb + PXc = 1*, *PYa + PYb + PYc = 1*. Таким образом, для каждого параметра возможны следующие случайные события: событие *A* – параметр остался неизменным; событие *B* – параметр изменился на *+1*; событие *C* – параметр изменился на *–1*. Изменение значения одного параметра происходит независимо от другого.

Построить средствами AnyDynamics компьютерную модель реализации случайного блуждания (рис. 4.1.1–4.1.3). Исследовать поведение объекта.

Создадим новый проект с именем N11 (рис. 4.1.1).

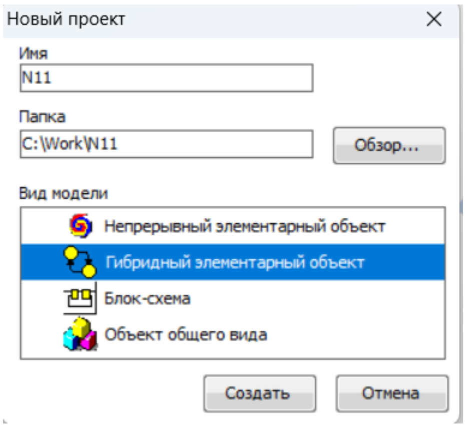


Рис. 4.1.1. Новый проект

Зададим параметры, переменные и их значения (рис. 4.1.2).

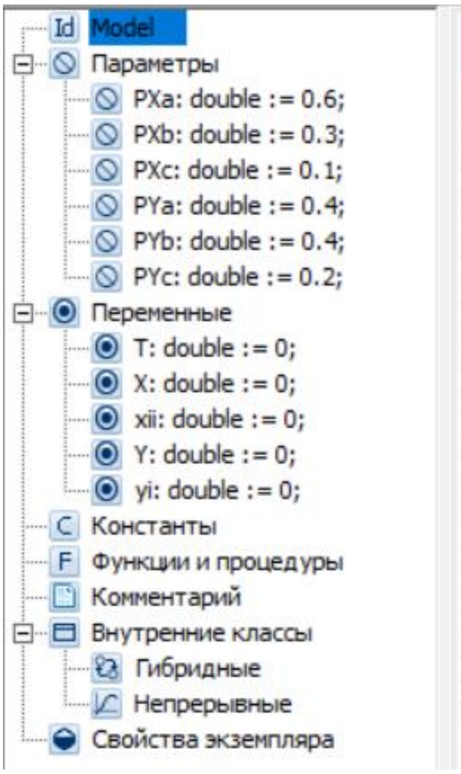


Рис. 4.1.2. Параметры и переменные модели

Создадим карту поведения (рис. 4.1.3). Исходными данными для модели является вероятности реализации событий *A, B, C*; *PXa, PXb, PXc* ; *PYa, PYb, PYc*. Начальные значения: $X = 0$, $Y = 0$. При моделировании случайных событий *X* и *Y* использовать отдельный датчик случайных чисел с равномерным законом распределения (uniform (0,1)).

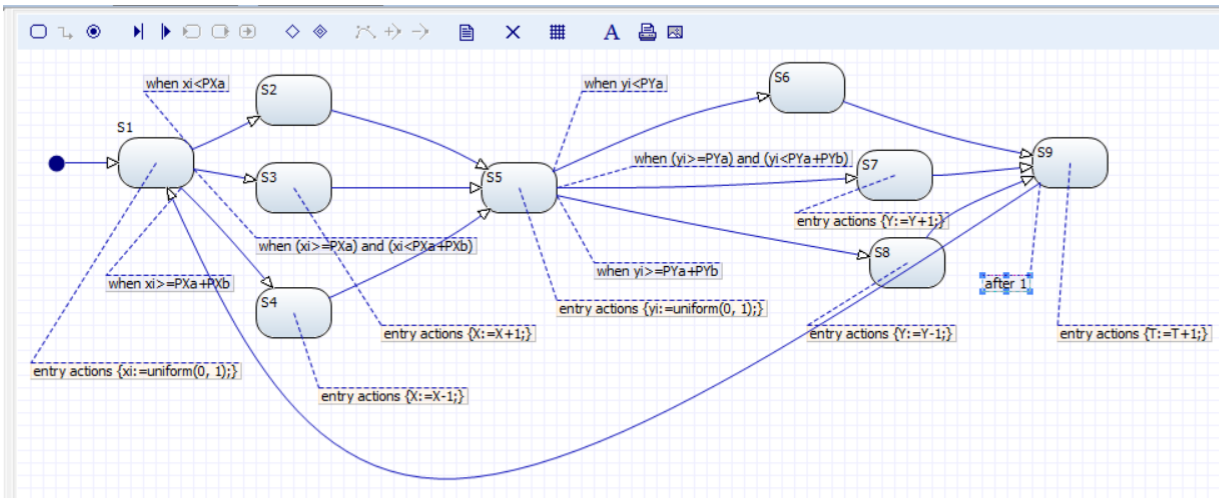


Рис. 4.1.3. Карта поведения для модели случайного блуждания

Проведите серии экспериментов при разных значениях вероятностей событий и понаблюдайте, как изменяется диаграмма (рис. 4.1.4).

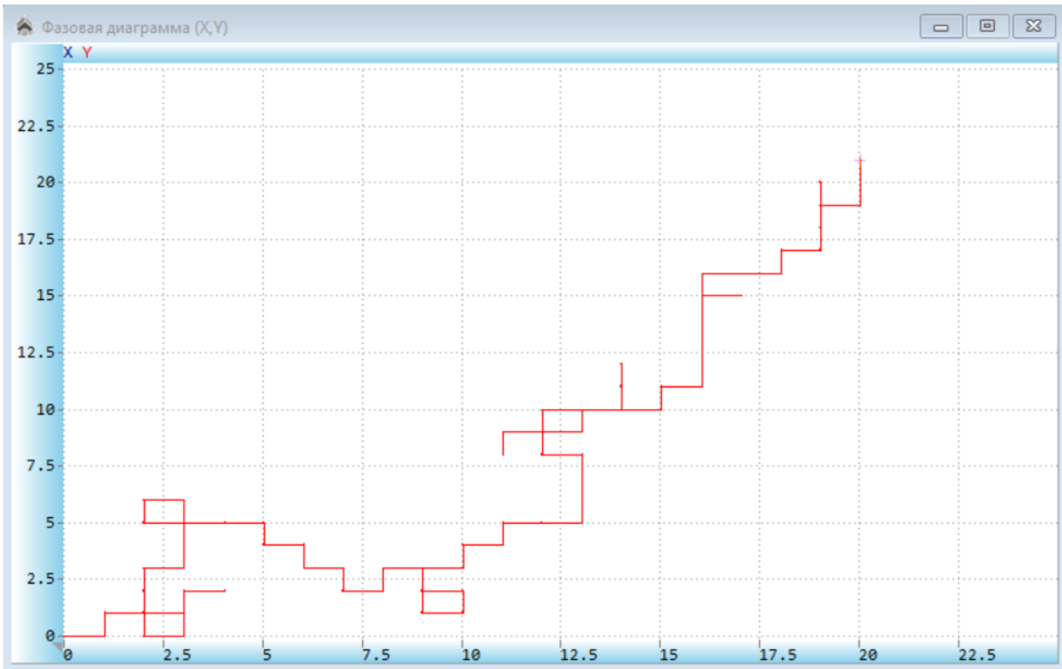


Рис. 4.1.4. Результат моделирования: траектория блуждания

4.2. Генерация случайной величины

Создадим проект, в котором будет генерироваться случайная величина с заданными законом распределения. Проект содержит переменные, которые указаны на рис. 4.2.1.

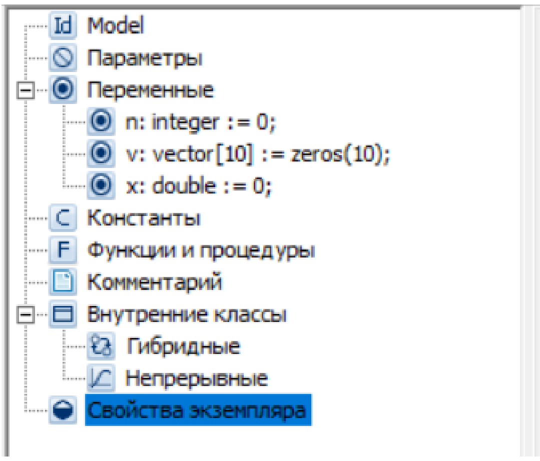


Рис. 4.2.1. Переменные проекта

Переменная *V* является вектором. В модели компоненты этого вектора получат значения сгенерированной случайной величины. Первоначально они будут иметь нулевые значения (zeros).

Карта поведения данной модели представлена на рис. 4.2.2.

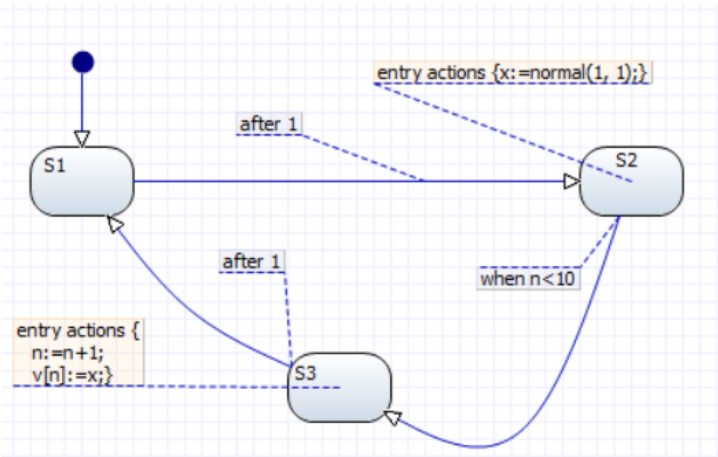


Рис. 4.2.2. Карта поведения

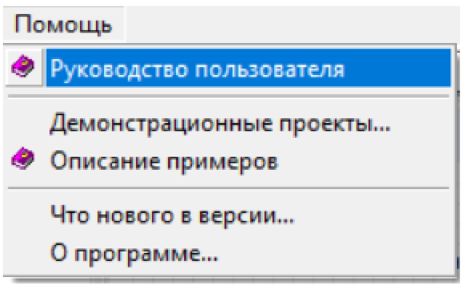


Рис. 4.2.3. Раздел «Помощь»

В узле *S1* установлено пустое поведение. В узле *S2* происходит генерация значения случайной величины с равномерным законом распределения и значениями от 0 до 1 (функция uniform (0,1)).

В AnyDynamics имеются функции генерации случайных величин и с другими законами распределения (смотри «Руководство пользователя», рис. 4.2.3.).

В узле *S3* заполняется очередной компонент вектора *V.* После создания визуальной модели поместим на ее поле гистограмму (копка ). На гистограмму из окна переменных перетащим вектор *V* и запустим модель. В установках модели (рис. 4.2.4) выберем режим **«**Так быстро, как возможно».

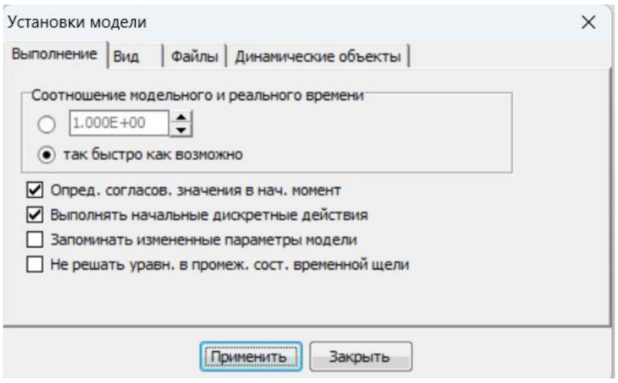


Рис. 4.2.4. Установки модели

Результат моделирования представлен на рис. 4.2.5 и рис. 4.2.6.

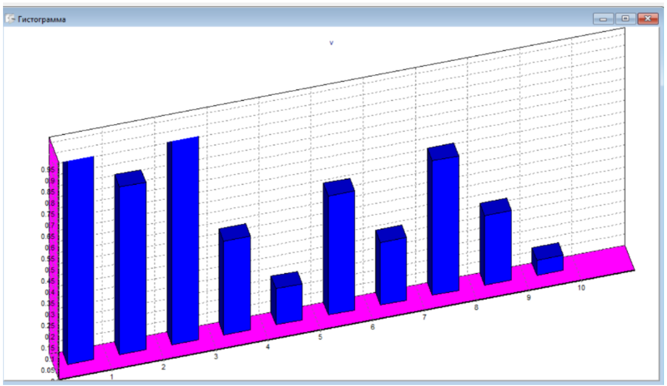


Рис. 4.2.5. Гистограмма в 3D-изображении

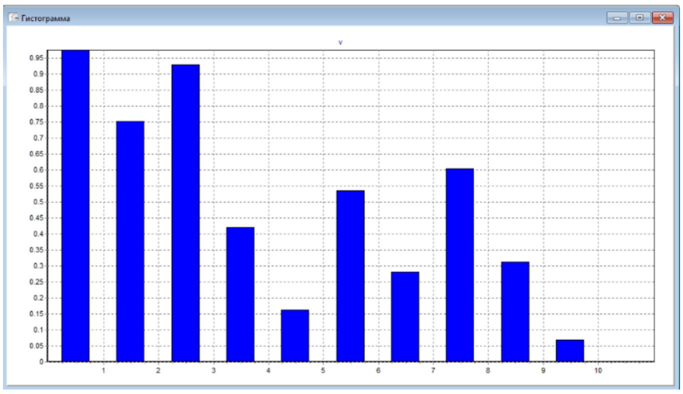


Рис. 4.2.6. Гистограмма в плоском изображении

ЗАКЛЮЧЕНИЕ

Лабораторный практикум позволяет студентам, будущим учителям информатики, математики–информатики или физики–информатики познакомиться и эффективно использовать программный комплекс **AnyDynamics** в будущей профессиональной деятельности на уроках информатики, математики или физики, как при проведении занятий по теме «Моделирование» в курсе «Информатика», так и при использовании моделей для демонстрации при изучении наук физико-математического или естественнонаучного профиля в университете и школе.

Лабораторный практикум посвящен современному компьютерному пакету быстрой разработки моделей, который не требует программирования и глубоких знаний в области численных методов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Бенькович, Е.С. Практическое моделирование динамических систем / Е.С. Бенькович, Ю.Б. Колесов, Ю.Б. Сенюченков. – Санкт-Петербург: БХВ-Петербург, 2002. – 464 с. – ISBN 5-94157-099-6. – Текст: непосредственный.
2. Королев, А.Л. Компьютерное моделирование в инструментальной среде ANYDYNAMICS. Лабораторный практикум / А.Л. Королев. – Челябинск: Изд-во ЮУрГГПУ, 2023.– 62 с. – ISBN 978-5-907790-59-9. – Текст: непосредственный.
3. Королев, А.Л. Компьютерное моделирование объектов процессов и систем: учебное пособие / А.Л. Королев, Н.Б. Паршукова. – Челябинск: Издательство ЮУрГГПУ, 2020. – 329 с. – ISBN 978-5-907409-15-6. – Текст: непосредственный.
4. Тарасевич, Ю.Ю. Математическое и компьютерное моделирование. Вводный курс / Ю.Ю. Тарасевич. – Москва: Едиториал УРСС, 2019. – 149 с. – ISBN 5-354-00913-8. – Текст: непосредственный.

Учебное издание

Королев Александр Леонидович

САМОСТОЯТЕЛЬНАЯ РАБОТА В ИНСТРУМЕНТАЛЬНОЙ СРЕДЕ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ ANYDYNAMICS

Учебно-методическое пособие

ISBN 978-5-907790-78-0

Работа рекомендована РИС ЮУрГГПУ
Протокол №29 от 2023 г.

Издательство ЮУрГГПУ
454080, г. Челябинск, пр. Ленина, 69
Редактор Е.М. Сапегина
Технический редактор Н.А. Усова

Подписано в печать 03.11.2023 г.
Объём 2 уч.-изд. л. (9,3 усл. печ. л.)
Тираж 100 экз.
Формат 60×84/8
Заказ №

Отпечатано с готового оригинал-макета в типографии ЮУрГГПУ
454080, г. Челябинск, пр. Ленина, 69