**Лабораторная работа №2**

**Моделирование событий в динамических системах**

**Цель работы:**

1. Изучить способы задания событий в системе.
2. Изучить метод организации оптимизационного эксперимента.

**Практическое задание**

1. **Постановка задачи**

В качестве модели системы вновь рассмотрим тело массой m, удерживаемое упругой связью с жесткостью k. Рассмотрим воздействие на систему внезапно возникающей нагрузки следующего характера:

1. Единичный импульс: P(t)=J.
2. Линейно возрастающая нагрузка: P(t)=ct.
3. Гармоническое возбуждение: P(t)=P0.
4. **Моделирование событий**

Зададим поведение системы с помощью **Диаграммы состояний (**или **Стейтчарта).**

**Диаграмма состояний** позволяет графически задать пространство состояний алгоритма поведения объекта.

Диаграмма состояний рисуется с помощью следующих элементов:

* Начало диаграммы состояний
* Состояние



http://127.0.0.1:50771/help/topic/com.xj.anylogic.help/html/statecharts/Transition.files/image002.gif  Переход

http://127.0.0.1:50771/help/topic/com.xj.anylogic.help/html/statecharts/Initial%20State%20Pointer.files/image001.gif Указатель начального состояния

http://127.0.0.1:50771/help/topic/com.xj.anylogic.help/html/statecharts/Final%20state.files/image001.gif Конечное состояние

  Ветвление



http://127.0.0.1:50771/help/topic/com.xj.anylogic.help/html/statecharts/Shallow%20history%20and%20deep%20history%20states.files/image001.gif  Историческое состояние

Диаграммы состояний AnyLogic соответствуют стандарту UML. Они сохраняют графический вид, атрибуты и семантику выполнения, определенную в UML. Диаграмма состояний представляет собой состояния, соединенные переходами. Переходы могут сработать в результате заданного в качестве условия перехода события — это может быть истечение заданного таймаута, получение диаграммой состояний сообщения, выполнение заданного логического  условия и т.д. Срабатывание перехода приводит к переходу управления диаграммы состояний в то состояние, в которое ведет этот переход.

Из палитры **Диаграмма состояний** перетащим последовательно элементы **Начало** **диаграммы состояний**, задав в панели **Свойства** имя стейтчарта – **Колебания,** и элемент **Состояние,** задав имя S. Этот элемент описывает состояние системы до воздействия импульса.

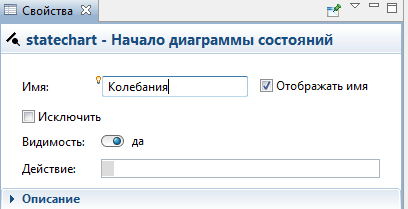


Рис 2.1

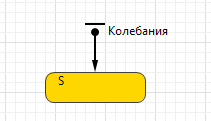


Рис. 2.2

Добавим элемент **Переход,** который будет срабатывать П**ри выполнении условия** достижения модельным временем определенного значения. В этот заданный с помощью функции time() (функция time() возвращает текущее значение модельного времени) момент, помимо силы упругости, на систему краткосрочно будет воздействовать дополнительная сила P=20 H. Это выражение записано в поле У**словие** окна свойств перехода (рис2.3 ). Именно так, стейтчарты cледят за событиями. При наступлении нужного события выполняется необходимое действие. И условие наступления события, и **Действие**, если необходимо, записываются операторами языка Java.

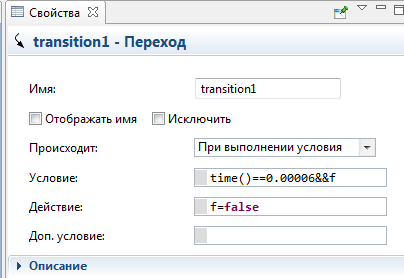


Рис. 2.3

Период действия силы P на систему описывается элементом **Событие** с именем IMP. Дополнительный параметр типа Boolean f, введен для того, чтобы **Событие** IMP выполнилось только один раз (рис2.4). Еще один **Переход** сработает по таймеру (рис.2.5) и вернет систему к прежнему состоянию.

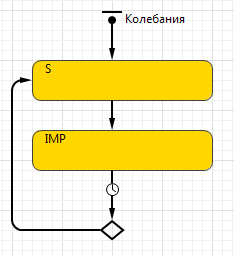


Рис. 2.4

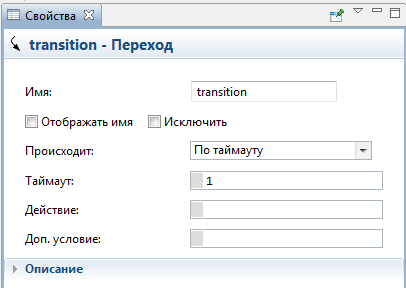


Рис. 2.5

Итак, поведение системы описывается стейтчартом с именем **Колебания**, с двумя состояниями. В одном состоянии описывается движение системы до импульса, в другом

– описывается движение во время импульса. В зависимости от состояния, в котором находится система, переменная Fupr будет описываться различными уравнениями.

Воспользуемся методом класса isStateActive( ). Если модель работает в состоянии S, то вызов метода Колебания.isStateActive(S) вернет значение true. Если модель

работает в состоянии IMP, то вызов этого метода вернет значение false. Откройте

свойства переменной Fupr и в соответствующее поле запишите следующее условное

выражение: Колебания.isStateActive(S) ? -k\*–: -k\*x+20

Запустите модель.

**Задайте поведение системы с помощью элемента Событие**

1. Перетащите элемент **Событие** http://127.0.0.1:50723/help/topic/com.xj.anylogic.help/html/statecharts/Add%20an%20Event.files/image001.gif из палитры **Основная** на диаграмму типа агентов.
2. После создания события,  можно изменить его имя в небольшом текстовом редакторе, который появится справа от элемента в графическом редакторе. Закончив ввод нового имени, нажмите Enter.
3. В панели **Свойства**, сконфигирируйте нужный тип события с помощью свойств, специфичных для данного типа события (рис.2.6)

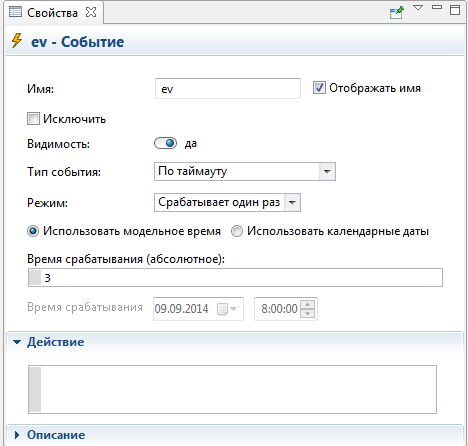
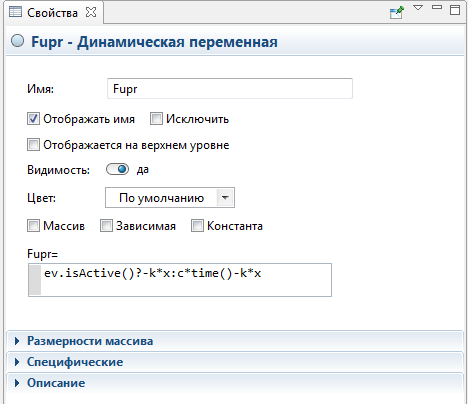


Рис. 2.6

Измените значение функции Fupr.



1. **Моделирование стохастики в системе**

**Задание для самостоятельной работы**

Описать целевую функцию, предложенную преподавателем и провести оптимизационный эксперимент.

**xЭксперимент Оптимизация**

Если необходимо изучить поведение модели при каких-то заданных условиях или улучшить производительность модели, найдя значения параметров, при которых достигается наилучший результат работы модели, то можно воспользоваться возможностью оптимизации модели AnyLogic.

Оптимизация модели AnyLogic заключается в последовательном выполнении нескольких прогонов модели с различными значениями параметров и нахождении оптимальных для данной задачи значений параметров.

В AnyLogic встроен оптимизатор OptQuest – лучший из предлагаемых сегодня оптимизаторов. Оптимизатор OptQuest автоматически находит лучшие значения параметров модели с учетом заданных ограничений. AnyLogic предоставляет удобный графический интерфейс для конфигурирования и отслеживания хода оптимизации.

Оптимизация состоит из нескольких последовательных прогонов модели с различными значениями параметров. Комбинируя эвристики, нейронные сети и математическую оптимизацию, OptQuest позволяет находить значения параметров модели, соответствующие максимуму или минимуму целевой функции, как в условиях неопределенности, так и при наличии ограничений.

**Чтобы оптимизировать модель**

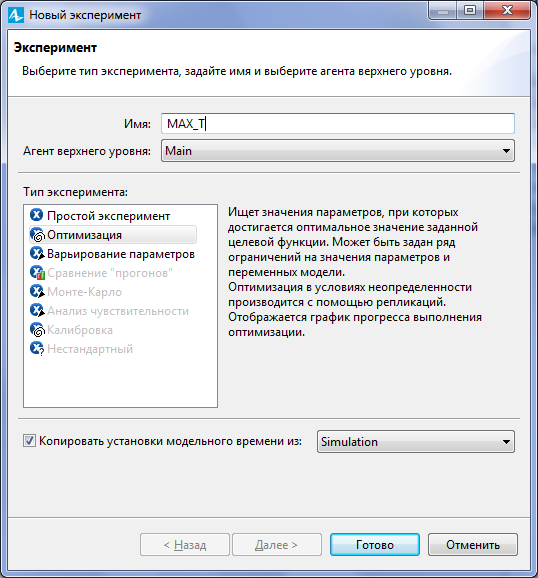
1. Создайте оптимизационный эксперимент.
2. Задайте целевой функционал (функцию, которую нужно минимизировать или максимизировать).
3. Задайте оптимизационные параметры (параметры, значения которых будут меняться).
4. Задайте ограничения, которые будут наложены на значения параметров и переменных.(опционально).
5. Задайте условия остановки прогона.
6. Задайте условия остановки оптимизации.
7. Запустите оптимизационный эксперимент.

Процесс оптимизации представляет собой итеративный процесс, который состоит в том, что:

* Оптимизатор OptQuest выбирает допустимые значения оптимизационных параметров и запускает модель с этими значениями.
* Завершив "прогон" модели, OptQuest вычисляет значение целевой функции на момент завершения.
* Оптимизатор анализирует полученное значение, изменяет значения оптимизационных параметров в соответствии с алгоритмом оптимизации и процесс повторяется заново.

**Чтобы создать оптимизационный эксперимент**

1. В панели **Проекты**, щелкните правой кнопкой мыши по элементу модели и выберите **Создать > Эксперимент** из контекстного меню. Появится диалоговое окно**Новый эксперимент**.
2. Выберите **Оптимизация** из списка **Тип эксперимента**.
3. Введите имя эксперимента в поле **Имя**.
4. Выберите корневой объект для этого эксперимента из выпадающего списка **Агент верхнего уровня**.
5. Если Вы хотите применить к создаваемому эксперименту временные установки другого эксперимента, оставьте установленным флажок **Копировать установки модельного времени из** и выберите эксперимент из расположенного справа выпадаюшего списка.
6. Щелкните мышью по кнопке **Готово**.



Целью оптимизации является нахождение таких значений параметров, при которых функция, выбранная в качестве целевого функционала, примет максимальное или минимальное значение. *Целевым функционалом* может быть математическое выражение, описывающее связь оптимизационных параметров, или результат какой-либо операции (например, прогона модели), входными параметрами которой являются параметры оптимизации. Функционал оптимизации задается целевой функцией и условием оптимизации. Условие оптимизации определяет, является ли целью оптимизации минимизация или максимизация значения целевого функционала.



* 1. Задайте Имя эксперимента

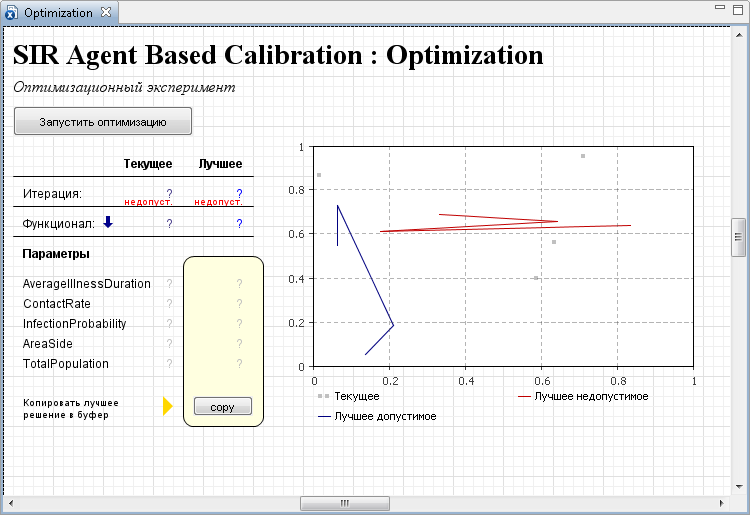
Поскольку AnyLogic генерирует для каждого эксперимента соответствующий Java класс, при задании имени эксперимента нужно руководствоваться правилами названия классов в Java. Пожалуйста, начинайте имя эксперимента с заглавной буквы.

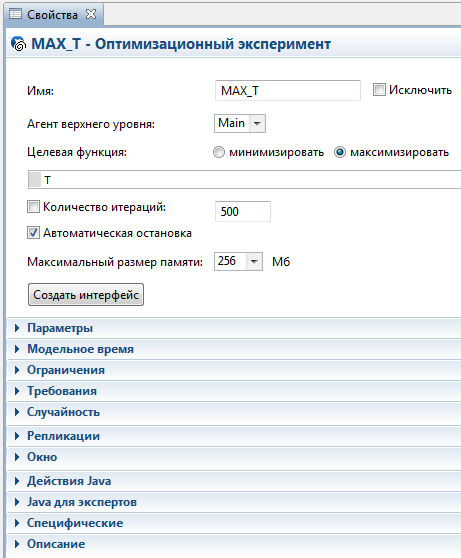
**Исключить** – Если опция выбрана, то эксперимент будет исключен из модели.

**Агент верхнего уровня** – Здесь задается агент верхнего уровня этого эксперимента.  Объект этого типа будет играть роль корня иерархического дерева объектов модели, запускаемой этим экспериментом.

* 1. Задайте целевую функцию **Целевая функция**, тип исследования (**минимизировать** или **максимизировать**). В качестве целевого функционала можно задать любое выражение Java, в том числе и арифметическое выражение или вызов метода. Поскольку это выражение будет рассматриваться в контексте корневого объекта оптимизационного эксперимента, то можно ссылаться в нем на переменные и параметры этого объекта (агент верхнего уровня доступен как root). Функцию сложного вида, лучше создать в агенте верхнего уровня явно с помощью функции и поместить вызов этой функции в поле **Целевая функция**. Например, root.T.

**Количество итераций** – Если опция выбрана, то оптимизация будет остановлена, если будет выполнено максимальное количесвто итераций, заданное в расположенном справа поле. **Автоматическая остановка** – Если опция выбрана, то оптимизация будет остановлена, если значение целевой функции перестанет существенно улучшаться (эта опция называется *автоостановом оптимизации*). **Создать интерфейс** – Кнопка создает для эксперимента пользовательский интерфейс по умолчанию.





**Оптимизационные параметры**

*Оптимизационный параметр* (называемый также переменной решения) – это параметр модели, значение которого необходимо оптимизировать.

Оптимизация модели AnyLogic заключается в последовательном выполнении нескольких прогонов модели с различными значениями оптимизационных параметров и нахождении тех значений параметров, при которых достигается оптимальное значение заданного целевого функционала.

**Типы оптимизационных параметров**

В процессе оптимизации оптимизационный параметр принимает значения из заданного интервала допустимых значений по правилам, принятым для данного типа параметра. AnyLogic поддерживает три типа оптимизационных параметров:

* Непрерывный параметр
* Дискретный параметр
* Перечисление

*Непрерывный параметр* может принимать любое значение из заданного интервала. Заданная точность параметров определяет минимальное значение, на которое может изменяться значение непрерывного параметра.

**Задание оптимизационных параметров**

Оптимизационные параметры задаются в таблице **Параметры** в соответствующей секции свойств оптимизационного эксперимента. Каждый параметр задается в отдельной строке таблицы.

**Чтобы сделать параметр эксперимента оптимизационным**

1. Выберите оптимизационный эксперимент в панели **Проекты**.
2. В секции **Параметры** панели **Свойства**, перейдите к той строке таблицы **Параметры**, которая содержит параметр, который Вы хотите сделать оптимизационным.
3. Щелкните мышью в ячейке**Тип** и выберите тип оптимизационного параметра, отличный от значения **фиксированный**. Список возможных значений будет меняться в зависимости от типа параметра: **набор значений**, **int**, **дискретный** для целочисленных параметров типа *int*; **непрерывный** и **дискретный** для вещественных параметров типа *double* и т.д.
4. Задайте диапазон допустимых значений параметра. Введите нижнюю границу диапазона в ячейке **Мин** и верхнюю - в ячейке **Макс**.
5. Для параметров **набор значений** и **дискретный** нужно также указать в ячейке **Шаг** величину шага (инкремента), с помощью которого будут определяться допустимые значения данного параметра (первое допустимое значение равно заданной нижней границе интервала, следующее равно сумме первого значения и заданного шага и т.д.).

**Модельное время**

**Использовать календарь** – Если опция выбрана, то начальное и конечное время моделирования задаются в виде календарных дат.

**Остановить** – Здесь нужно задать, когда моделирование должно быть остановлено. Если моделирование должно продолжалось бесконечно, пока пользователь его не остановит, нужно выбрать из выпадающего списка **Нет**. Если же необходимо, чтобы моделирование длилось до заданного времени или заданной даты, выбирается соответственно **В заданное время** или **В заданную дату**. В этом случае время остановки будет задаваться элементом управления **Начальное время/Начальная дата**.

**Начальное время** – [Доступно, если не выбрана опция **Использовать календарь**] Начальное время моделируемого интервала времени.

**Начальная дата** – [Доступно, если выбрана опция **Использовать календарь**] Начальная календарная дата моделируемого интервала времени.

**Конечное время** – [Доступно, если не выбрана опция **Использовать календарь**] Конечное время моделируемого интервала времени (количество единиц модельного времени, по прошествии которого модель будет остановлена).

**Конечная дата**  – [Доступно, если выбрана опция **Использовать календарь**] Конечная календарная дата моделируемого интервала времени.

**Дополнительные условия остановки оптимизации**  – Здесь можно задать любое количество дополнительных условий остановки оптимизации. Когда выполнится любое из этих условий, оптимизация будет остановлена.

# Ограничения и требования

AnyLogic позволяет задавать дополнительные условия оптимизации, накладывая ограничения на значения оптимизационных параметров и на получаемые оптимизатором решения

**Задание условий остановки оптимизации**

Чтобы оптимизатор мог успешно работать, обязательно нужно задать [условие остановки "прогона" модели](http://127.0.0.1:57292/help/topic/com.xj.anylogic.help/html/config/Setting%20Start%20and%20Stop%20Time.html). Эти настройки оптимизации задаются на странице  **Основные** панели свойств оптимизационного эксперимента.

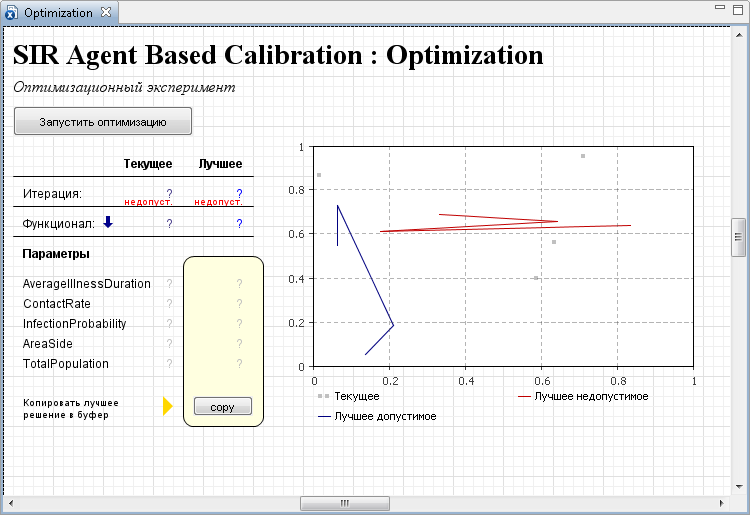
**Чтобы остановить оптимизацию после выполнения заданного числа итераций**

1. В панели **Проекты**, выделите нужный Вам эксперимент.
2. Перейдите в панель **Свойства**.
3. В разделе свойств **Условия остановки оптимизации** задайте количество итераций в поле **Количество итераций**. Оптимизация завершится, когда будет выполнено заданное количество итераций (если ранее не сработает условие автоматической остановки).

**Чтобы включить условие автоматической остановки**

1. В панели **Проекты**, выделите нужный Вам эксперимент.
2. Перейдите в панель **Свойства**.
3. В разделе свойств **Условия остановки оптимизации** установите флажок **Автоматическая остановка**.

Будьте внимательны при использовании режима автоматической остановки модели. В случае возникновения «оптимизационной пробки» (т.е., если находимые значения целевой функции будут улучшаться слишком медленно), оптимизация может быть остановлена задолго до того, как будет найдено оптимальное решение. Если Вы столкнетесь с такой проблемой, то Вам будет нужно увеличить точность целевого функционала, предложить оптимизатору другие значения оптимизационных параметров, или отключить режим авто-стопа.



**Интерфейс оптимизационного эксперимента**

Кнопка **Запустить оптимизацию** выполняет запуск оптимизационного процесса.

В правой части расположен график, визуально отображающий ход оптимизации. По оси X откладываются номера итераций, а по оси Y - **Текущее, Лучшее допустимое** и **Лучшее недопустимое** значения, найденные для каждой итерации.

Таблица, расположенная в левой части окна, отображает всю необходимую информацию о ходе оптимизационного процесса.

В столбце **Текущее** отображаются: номер последней завершенной итерации, значение целевой функции и значения параметров, при которых оно было получено на момент окончания этой итерации. В столбце **Лучшее** отображается та же информация для найденного решения, которое является оптимальным к текущему моменту: номер итерации, когда это значение было получено, собственно, само значение целевого функционала, и значения параметров, при которых оно было получено.

По завершении оптимизации это решение будет считаться оптимальным найденным решением. Вы можете экспортировать полученное решение в другие эксперименты модели, щелкнув по кнопке копировать и вставив его в дальнейшем в нужный Вам эксперимент с помощью кнопки **Вставить из буфера**, расположенной в секции **Параметры** панели **Свойства** этого эксперимента.