



АРМ ЕСА
Руководство пользователя

АРМ ЕСА
Расчет электрических цепей
Версия 17
Руководство пользователя

Научно-технический центр «Автоматизированное Проектирование Машин»
141070, Россия, Московская область, г. Королёв, Октябрьский бульвар 14, оф. 6
тел.: +7 (495) 120-58-10.
Наш адрес в Интернете: <http://www.apm.ru>, e-mail: com@apm.

Авторские права © 1989 – 2019 Научно-технический центр «Автоматизированное проектирование машин». Все права защищены. Все программные продукты НТЦ «АПМ» являются зарегистрированными торговыми марками центра. Названия и марки, упомянутые в данном руководстве, являются зарегистрированными торговыми марками их законных владельцев.
Отпечатано в России.

Оглавление

1	Принципы формирования моделей	3
1.1	Интерфейс программы	3
1.2	Просмотр результатов расчета	7
1.3	Математические выражения и переменные окружения	9
1.4	Глобальные настройки программы	10
1.5	Использование среды программирования Julia	10
2	Имитационные схемы	11
2.1	Основные принципы формирования имитационных моделей	11
2.2	Стандартные элементы	12
2.3	Подсистемы	15
2.4	Программируемые элементы и источники	19
2.5	Проведение расчетов	26
3	Электрические схемы	28
3.1	Основные принципы формирования моделей электрических цепей	28
3.2	Стандартные элементы	29
3.3	Подсистемы	34
3.4	Программируемые элементы и источники	38
3.5	Проведение расчетов	40
	3.5.1 Статический расчет	40
	3.5.2 Гармонический расчет	40
	3.5.3 Динамический расчет	41

Глава 1

Принципы формирования моделей

1 Интерфейс программы

Окно программы состоит из следующих основных элементов (рисунок 1) :

- 1) Общее меню и панель инструментов для работы с документом.
- 2) Закладки открытых документов.
- 3) Меню и панель инструментов для работы с документом.
- 4) Панель управления масштабом активного документа.
- 5) Панель элементов для добавления на схему.
- 6) Графическое представление схемы (вид).
- 7) Панель результатов расчетов.
- 8) Область вывода информационных сообщений.
- 9) Индикатор прогресса расчета.

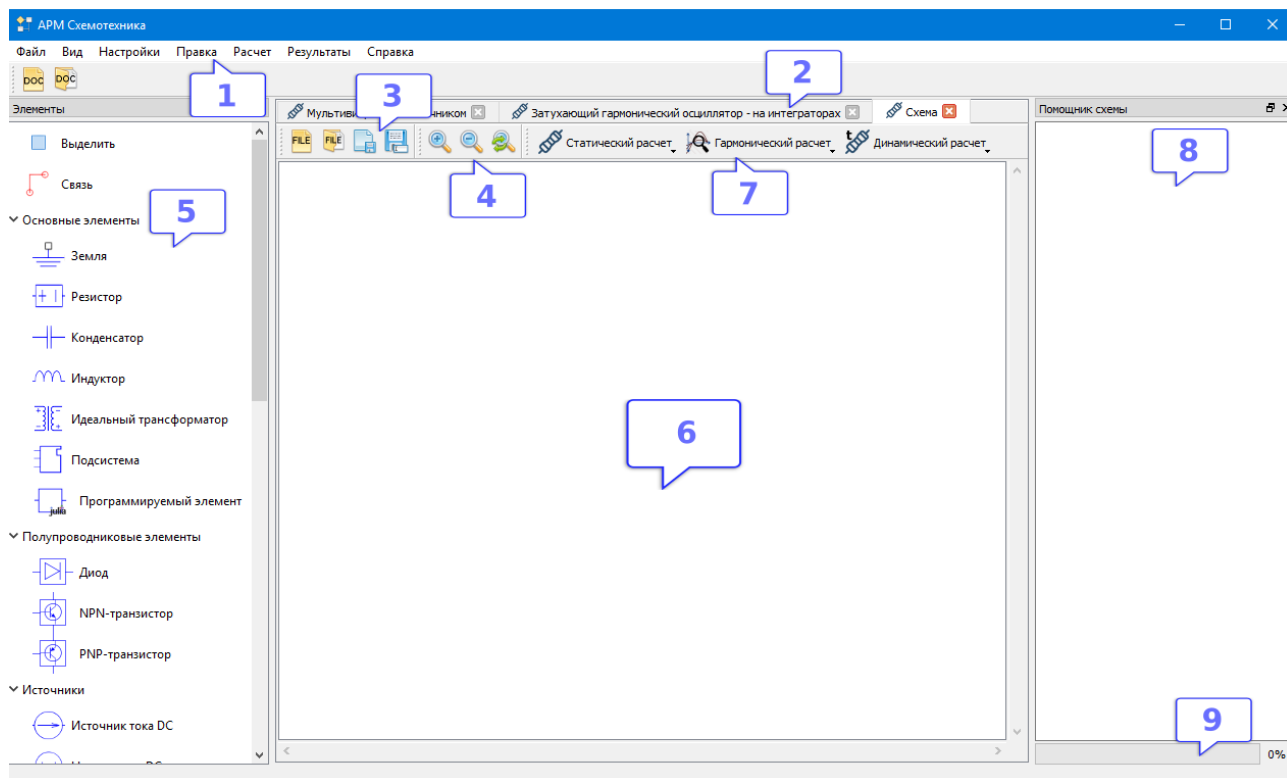


Рисунок 1 – Интерфейс программы

Для создания новой схемы в новой вкладке необходимо выбрать «Файл - Новая схема», либо кнопку «Создать документ» на основной панели инструментов (рис. 1, п.1). После этого будет предложено выбрать один из возможных типов модели: «Электрическая цепь», «Имитационная схема». Для открытия новой схемы в уже имеющейся вкладке требуется нажать кнопку «Создать схему» в панели инструментов документа. В этом случае новый документ будет иметь тот же тип, что и ранее открытый в этой вкладке.

Элементы на панели элементов (по умолчанию располагается в правой части рабочего окна) сгруппированы по категориям. Для добавления элемента в модель требуется нажатием кнопки в дереве элементов активировать режим добавления соответствующего элемента, после чего нажать левой кнопкой мыши на любое место в документе. Основные составляющие элемента представлены на рисунке 2.

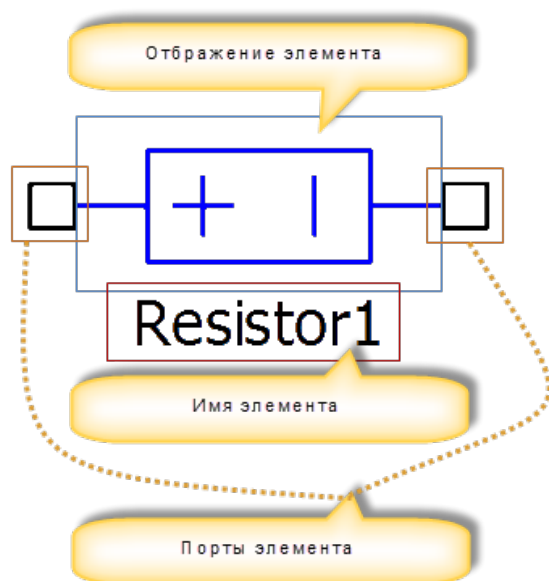


Рисунок 2 – Структура элемента

Область отображения элемента служит для репрезентации графического отображения, а также является объектом, с которым связаны все действия, выполняемые над элементом (перемещение объекта по виду, выделение, изменение свойств). Область имени элемента служит для задания пользователем каждому элементу уникального имени, которое присваивается по умолчанию программой, при этом не допускается двух одинаковых имен. И наконец, порты элемента, являются средством соединения элемента с другими элементами.

После добавления элемента на схему, пользователь может изменить его размер или повернуть на произвольный угол. Для этого необходимо на панели элементов нажать кнопку «Выделить» и использовать маркеры вокруг элемента при наведении мыши рисунок 3. При клике на элементе происходит его выделение рамкой.

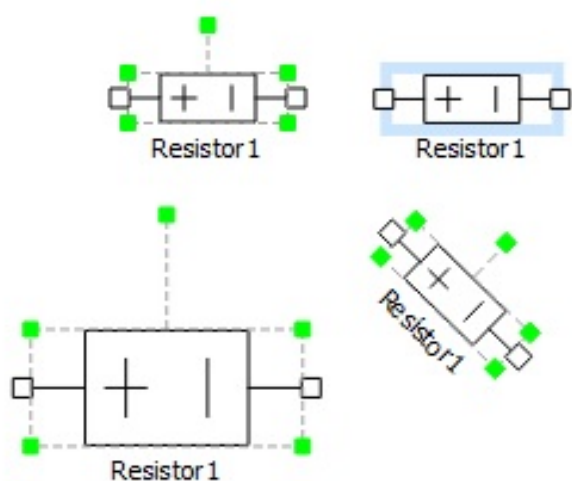


Рисунок 3 – Преобразование элемента

Также для каждого элемента схемы нажатием правой кнопки мыши пользователь может вызвать контекстное меню (рисунок 4, состоящее из следующих основных пунктов:

- 1) Контекстное меню элемента состоит из следующих пунктов:

- 2) «Повернуть по часовой стрелке» - поворот элемента на 90° по часовой стрелке относительно текущего положения.
- 3) «Повернуть против часовой стрелки» - поворот элемента на 90° против часовой стрелки относительно текущего положения.
- 4) «Отразить относительно вертикали» - зеркальное отражение элемента относительно вертикальной линии, проходящей через центр элемента.
- 5) «Отразить относительно горизонтали» - зеркальное отражение элемента относительно горизонтальной линии, проходящей через центр элемента.
- 6) «По умолчанию» - сброс всех действий пользователя по изменению размеров элемента, поворотов и отражений.
- 7) «Удалить» - удаление элемента.
- 8) «Цвет элемента и цвет фона элемента» - позволяют пользователю настроить цветовое оформление элементов схемы. Пользователь может выбрать один из зарезервированных цветов, используя соответствующий пункт меню или указать произвольный цвет, используя пункт «Выбрать...».
- 9) «Результаты на схеме» и «Показать результаты» отображает различные результаты (в зависимости от вида расчета), либо непосредственно на схеме, либо в отдельном окне.
- 10) «Свойства» - задание физических свойств элементов схемы.

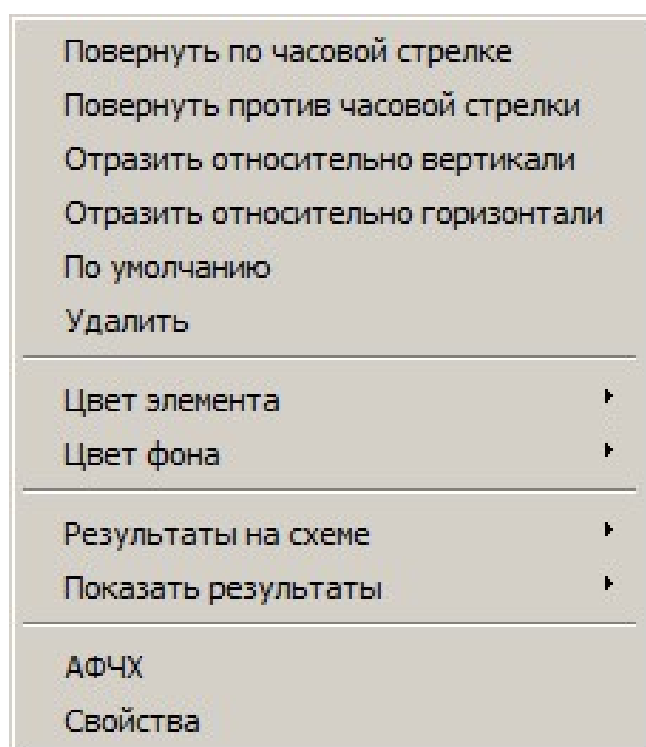


Рисунок 4 – Контекстное меню элемента

Для каждого элемента схемы, после выбора пункта «Свойства» в контекстном меню или двойного щелчка левой кнопкой мыши на элементе, появится диалоговое окно для задания физических характеристик этого элемента (рисунок 5). Значением свойства элемента может быть число с плавающей точкой, либо математическое выражение, включающее переменные окружения (см. 1.3).

Два элемента можно соединить связью между собой, используя порты подключения элемента (далее просто порт), которые изображены черными квадратиками по краям элемента. Нажав на кнопку «Связь», необходимо нажать и удерживать левую кнопку мыши в порте первого элемента и перетащить связь в порт второго элемента, после чего отпустить кнопку мыши. Связь, у которой хотя бы один из концов не соединен ни с каким элементом, обозначается на схеме красным цветом. В нормальном состоянии связь отрисовывается черным цветом (рисунок 6).

По умолчанию связь строится по кратчайшему расстоянию между двумя точками. В случае активации режима «Ортогональные связи» в меню «Настройки» связь будет строиться таким образом, чтобы минимизировать пересечение с элементами схемы (рисунок 7). Каждый из сегментов связи можно перетащить вручную в режиме «Выделить».

Помимо связей между двумя элементами можно создать связи между элементом и другой связью и между двумя связями (рисунок 8). Для этого пользователь в режиме «Связь» должен начать создавать новую

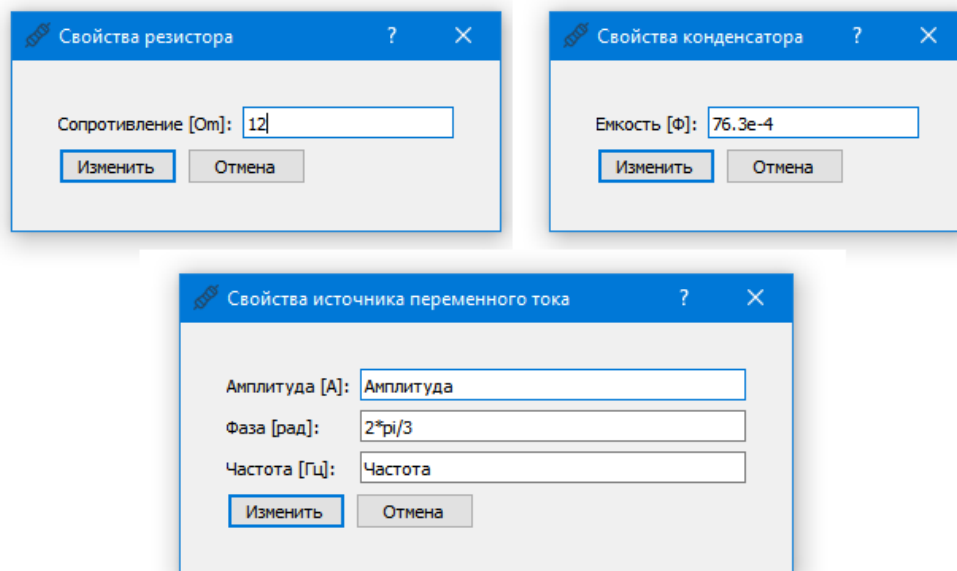


Рисунок 5 – Свойства элементов

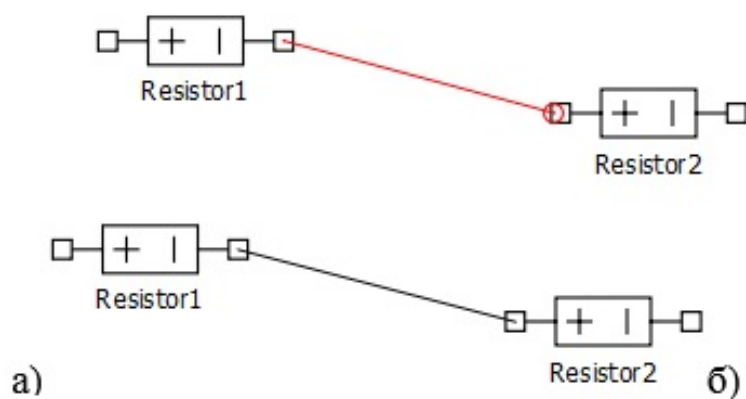


Рисунок 6 – Обозначение незавершенных и завершенных связей

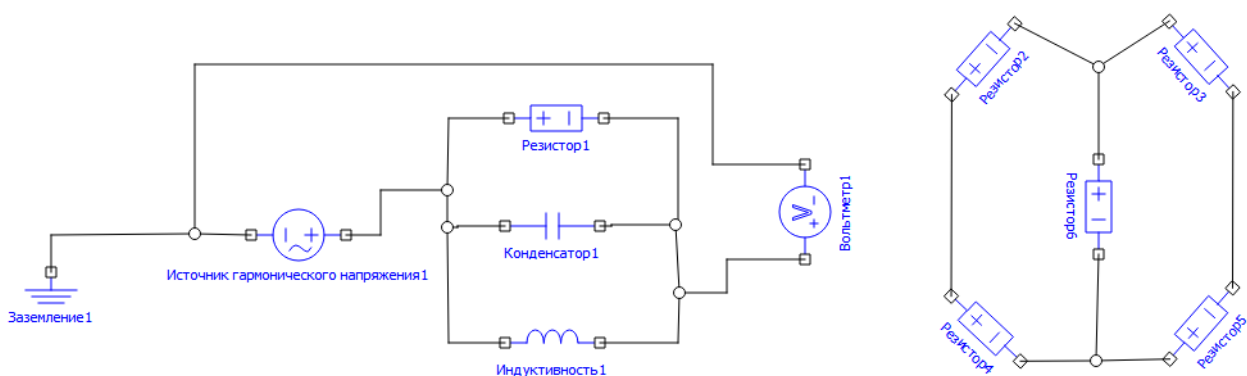


Рисунок 7 – Виды связей: ортогональные связи (слева) и «резиновые» связи (справа)

связь, начиная не с порта элемента, а с любой точки уже существующей связи и закончить ее в порте элемента или на другой связи.

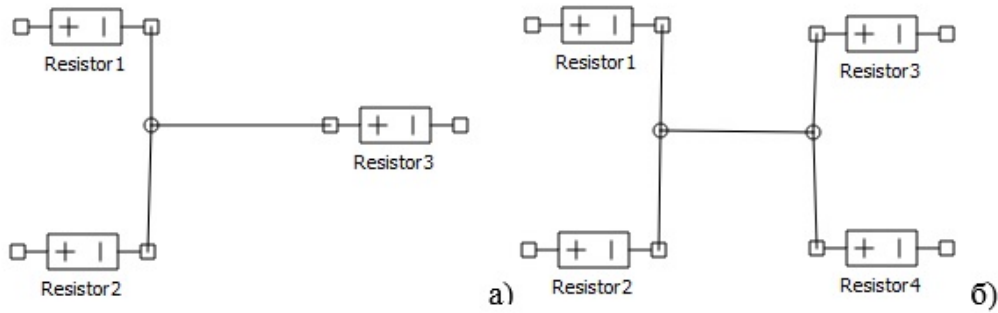


Рисунок 8 – Построение связи от другой связи

2 Просмотр результатов расчета

Для активации отображения результата на схеме необходимо выбрать соответствующий пункт в одном из меню результатов на панели инструментов документа (рисунок 9).

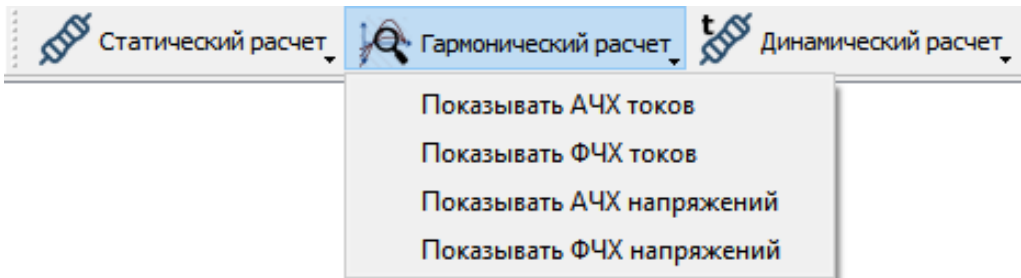


Рисунок 9 – Меню отображения результатов на схеме документа

Отображение результатов на схеме происходит при наведении указателя мыши на соответствующий элемент схемы (узел, элемент или связь). Способ отображения результата зависит от вида расчета (рисунок 10-11). Для определенных результатов дополнительная информация выводится в поле «Помощник схемы».

Для некоторых типов моделей и видов расчета могут быть дополнительные формы представления результатов, доступные через меню «Результаты» главного горизонтального меню. Подробности см. в разделе соответствующих типов моделей.

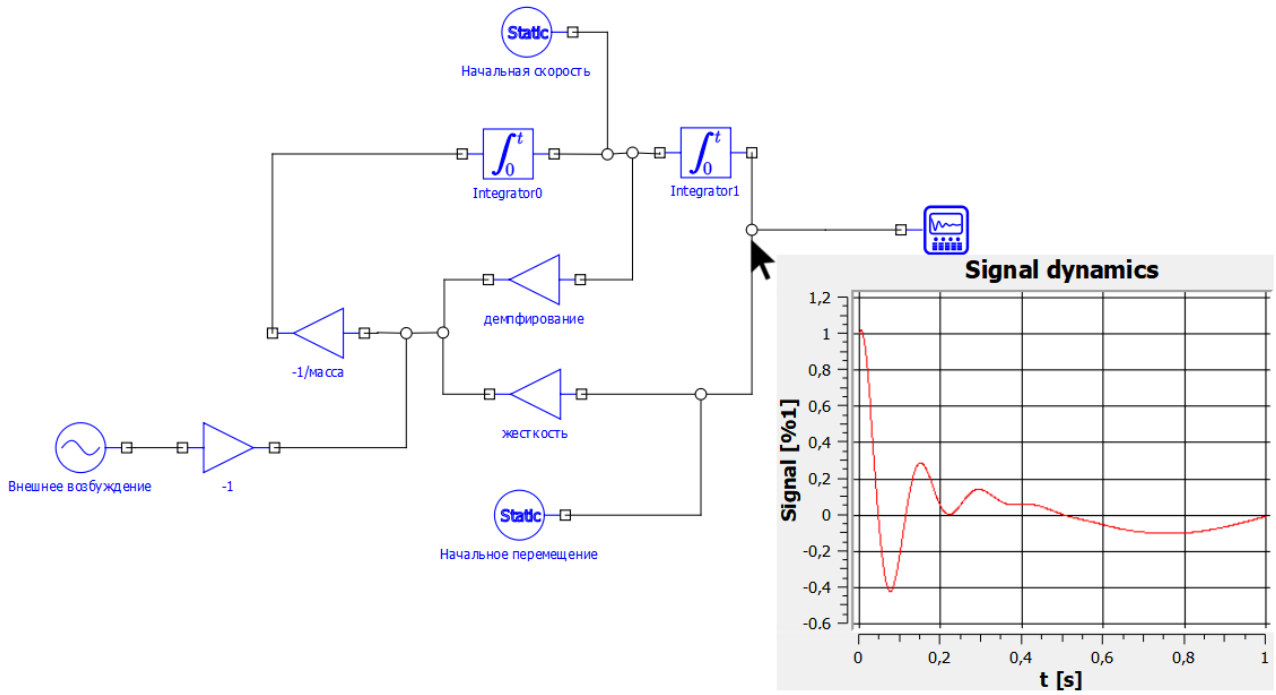


Рисунок 10 – Результат динамического расчета имитационной модели в форме графика

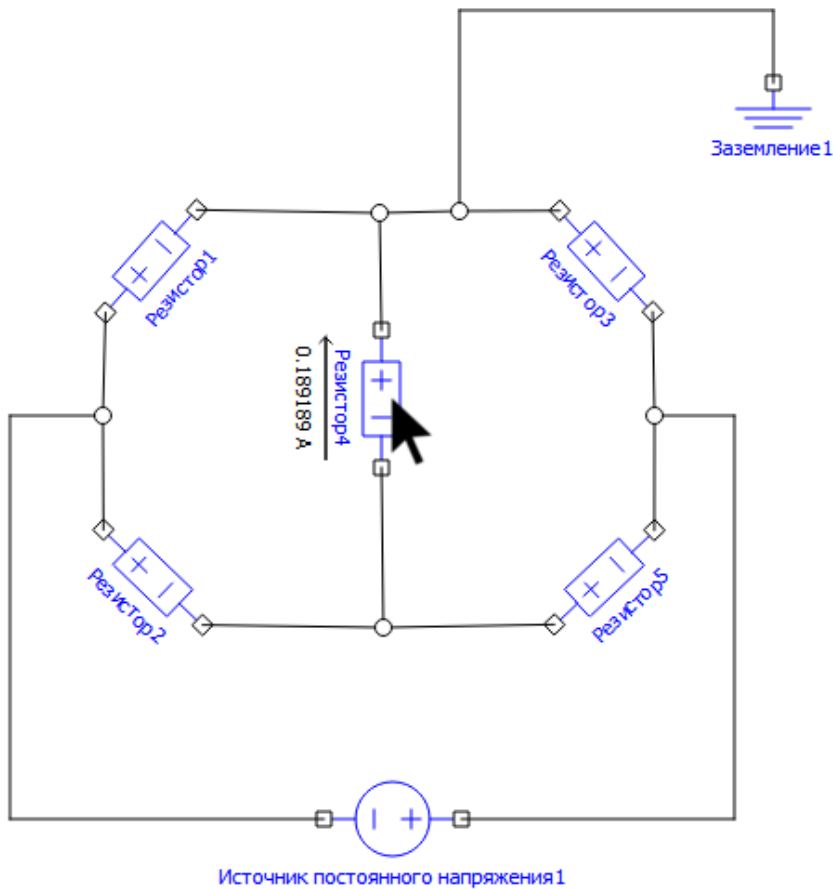


Рисунок 11 – Результат статического расчета электрической цепи. На схеме отображается величина электрического тока через резистор

3 Математические выражения и переменные окружения

В качестве значений для свойств элементов допускается использование математических выражений, включающих операции сложения (+), вычитания (−), умножения (*), деления (/) и возведения в степень (). Допускается использование следующих констант и функций:

e	2.71828182845904523536
log2e	1.44269504088896340736
log10e	0.434294481903251827651
ln2	0.693147180559945309417
ln10	2.30258509299404568402
pi	3.14159265358979323846
pi2	1.57079632679489661923
pi4	0.785398163397448309616
sqrt2	1.41421356237309504880
abs	возвращает абсолютную величину целого числа
acos	арккосинус
asin	арксинус
atan	арктангенс
atan2	арктангенс с двумя параметрами, $atan2(x, y) = atan\left(\frac{x}{y}\right)$
ceil	округление до ближайшего большего целого числа
cos	косинус
exp	вычисление экспоненты
floor	округление до ближайшего меньшего целого числа
fmod	вычисление остатка от деления нацело для чисел с плавающей точкой
ldexp	умножение числа на целую степень двух, $ldexp(x, n) = x \cdot 2^n$
log	натуральный логарифм
log10	логарифм по основанию 10
pow	результат возведения x в степень y
sin	синус
sinh	гиперболический синус
sqrt	квадратный корень
tan	тангенс
tanh	гиперболический тангенс
acosh	гиперболический арккосинус
asinh	гиперболический арксинус
atanh	гиперболический арктангенс
cbrt	кубический корень
copysign	возвращает величину, абсолютное значение которой равно x, но знак которой соответствует знаку y
erf	функция ошибок
erfc	дополнительная функция ошибок
exp2	значение числа 2, возведённого в степень x, 2^x
expm1	значение функции $exp(x - 1)$
fdim	вычисление положительной разницы между x и y, $\max(x - y, 0)$
fma	значение функции $(x * y) + z$
max	
min	наименьшее значение среди x и y
hypot	гипотенуза, $\sqrt{x^2 + y^2}$
ilogb	экспонента числа с плавающей точкой, конвертированная в int
lgamma	натуральный логарифм абсолютного значения гамма-функции
int	округление до ближайшего целого
lround	округление до ближайшего целого в направлении от нуля
log1p	натуральный логарифм $1 + x$
log2	логарифм по основанию 2
logb	целочисленная часть логарифма x по основанию 2
nextafter	следующий ближайшее представимое для x (по направлению к y)
remainder	вычисляет остаток от деления согласно стандарту IEC 60559
round	округление до целого
tgamma	гамма-функция
trunc	отбрасывание дробной части

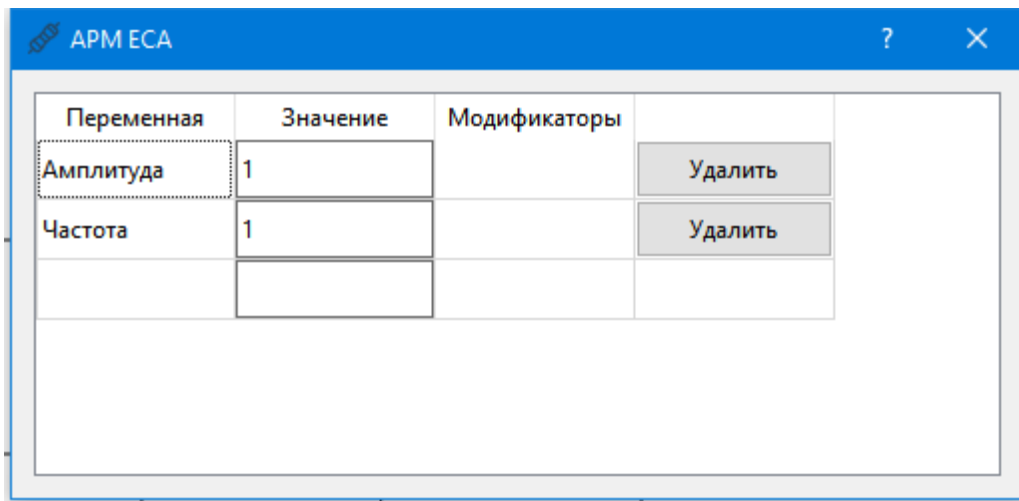


Рисунок 12 – Редактор окружения

В математических выражениях допускается использование т.н. *переменных документа*. Редактор окружения доступен через меню «Правка - Настроить окружения». Именем переменной может быть любая строка. Значением переменной должно быть число с плавающей точкой или символьное выражение (рисунок 12).

Если имя переменной удовлетворяет стандартным соглашениям о наименовании переменных, то есть начинается с буквы и содержит только буквы, цифры и технические пробелы, то обращение к значению переменной в выражении может осуществляться по ее имени. Например, $Амплитуда * exp(3)$ и $sin(Частота_пульсаций1)$ являются корректными выражениями. Если имя переменной не удовлетворяет стандартному соглашению, то для обращения к ее значению необходимо заключить имя в фигурные скобки:

$sin(\{Частота_пульсаций [Гц]\})$.

В случае, если выражение содержит не определенную в окружении переменную, при попытке проведения расчета будет выведено сообщение об ошибке с указанием имени неизвестной переменной. В случае синтаксической ошибки в выражении соответствующее сообщение будет выведено при попытке изменить значение свойства элемента или переменной.

4 Глобальные настройки программы

Глобальные настройки программы доступны через пункт меню «Настройки - Параметры». В окне настроек пользователь может указать поведение программы по умолчанию, а также скорректировать системные пути. Системные пути содержат перечень путей к директориям и файлам, используемым APM ECA, разделенным в соответствии с принятым в ОС соглашением (точкой с запятой для ОС Windows и двоеточием для ОС Linux).

– *Библиотеки элементов*. Содержит перечень директорий, в которых ищутся дополнительные элементы (Подробно см. в соответствующем разделе документации моделей).

– *Исполняемая среда Julia*. Содержит перечень директорий, в которых ищется исполняемая среда Julia 0.5. Значение по умолчанию $\$AppLocalData/Julia-0.5.2;\$PROGRAMFILES/Julia-0.5.2$. В случае, если исполняемая среда не найдена ни в одной из перечисленных директорий, соответствующее расширение будет отключено до следующего запуска программы.

Изменения глобальных настроек вступают в силу при следующем запуске программы.

5 Использование среды программирования Julia

В случае, если на компьютере пользователя установлена среда программирования Julia 0.5 (www.julialang.org) и путь к ней присутствует в системной переменной «Исполняемая среда Julia» (см. раздел 1.4), программа APM ECA предоставляет возможность использования данной среды программирования для определения дополнительных элементов. Взаимодействие со средой Julia осуществляется через меню «Правка - Редактировать исходный код». Формат описания элементов специфичен для каждого вида моделей, подробно см. в соответствующем разделе документации по типу модели.

Описание синтаксиса и библиотеки стандартных функций см.

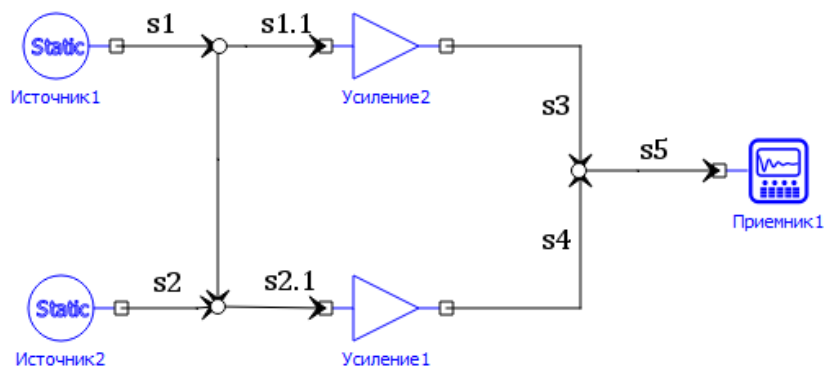
<https://docs.julialang.org/en/release-0.5/>

Глава 2

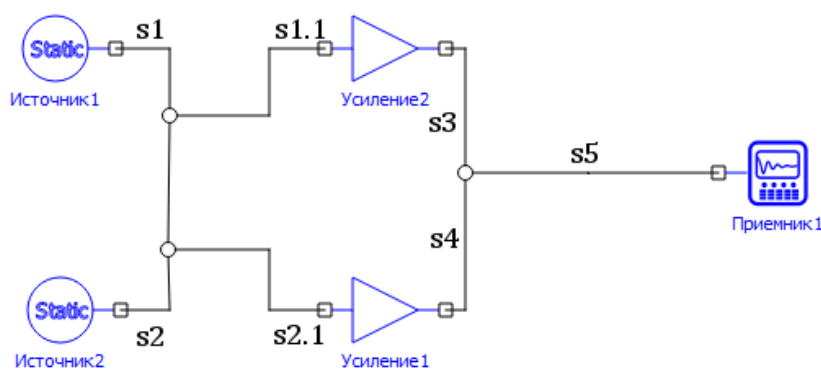
ИМИТАЦИОННЫЕ СХЕМЫ

1 Основные принципы формирования имитационных моделей

Имитационные модели состоят из элементов, ненаправленных и направленных связей (рис. 1). Каждый элемент определяется входами X , выходами Y и функцией $F, Y = F(X)$, либо уравнением $G(X, Y) = 0$. Каждая связь является носителем сигнала; в месте соединения связей происходит суммирование сигналов. Направленные связи обычно используются для разделения сигнала от элемента на несколько направлений.



(a) направленные связи



(b) ненаправленные связи

Рисунок 1 – Сравнение направленных и ненаправленных связей между элементами имитационных схем

Схемы рис. 1а и 1б иллюстрирует работу направленных связей, значения сигналов приведены в таблице 1.

Для проведения расчета динамики имитационной модели требуется, чтобы в схеме отсутствовали элементы, имеющие ни к чему не подсоединенный вход или выход. Допускается, чтобы в модели присутствовало несколько топологически не связанных частей (рис. 2). Связи, у которых хотя бы один конец не подключен

Таблица 1 – Сравнение работы направленных и ненаправленных связей

Сигнал	Значение на схеме 1a	Значение на схеме 1b
s_1	Источник1	
s_2	Источник2	
$s_{1.1}$	s_1	$s_1 + s_2$
$s_{2.1}$	$s_1 + s_2$	
s_3	$s_{1.1} \cdot \text{Усиление}_2$	
s_4	$s_{2.1} \cdot \text{Усиление}_1$	
s_5	$s_3 + s_4$	

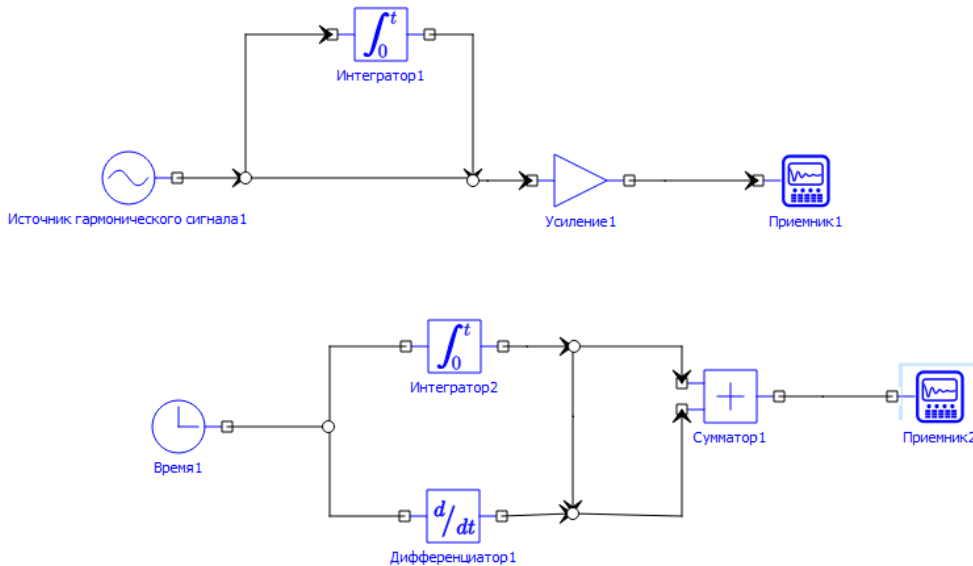


Рисунок 2 – Корректная схема, содержащая две топологически не связанные части

к элементу схемы (*незавершенные связи*, рис. 3), отображаются красным цветом. Наличие незавершенных связей допускается при проведении расчета, однако является нежелательным.

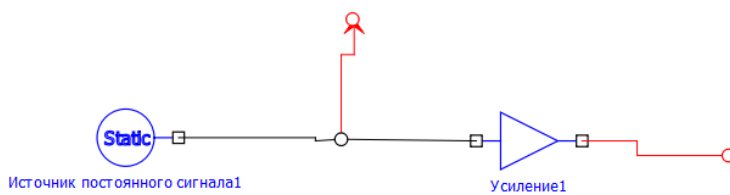


Рисунок 3 – Схема, содержащая незавершенные связи

2 Стандартные элементы

Название элемента	Параметр	Значение по умолчанию	Описание
Основные элементы			
Начальное значение	Начальное значение	0	$y(t) = \begin{cases} \text{«Начальное значение»} & \text{при } t = 0 \\ x & \text{иначе} \end{cases}$
Приемник			Используется для вывода результатов и «закрывания» модели
Приемник с выводом в файл	Имя файла		Используется для вывода результатов в текстовый файл
Дифференциатор			$y(t) = \frac{dx}{dt}$
Интегратор			$y(t) = \int_0^t x(\tau) d\tau$
Усиление	Коэффициент усиления	1	$y = \text{«Коэффициент усиления»} \cdot x$
Сумматор			$y = x_1 + x_2$
Вычитание			$y = x_1 - x_2$
Модуль			$y = x $
Мультипликатор			$y = x_1 \cdot x_2$
Делитель			$y = \frac{x_1}{x_2}$
Обращение			$y = \frac{1}{x}$
Подсистема	Опционально		Предназначен для загрузки ранее созданных моделей в текущую модель (см. 2.3)
Программируемый элемент	Опционально		Использует определенные пользователем типы элементов, описанные на языке Julia (см. раздел 2.4)
Источники сигнала			
Источник постоянного сигнала	Значение	0	$y(t) = \text{«Значение»}$
Время			$y(t) = t$
	Амплитуда	1	
Источник гармонического сигнала	Фаза [рад]	0	$y(t) = \text{«Амплитуда»} \cdot \sin(\text{«Частота [Гц]»} \cdot 2\pi + \text{«Фаза [рад]»})$

¹ При использовании данного элемента требуется избегать обращения в ноль второго входа

<p>Источник переменного сигнала</p>	<p>Частота [Гц]</p>	<p>1</p>	<p>Использует задаваемую пользователем формулу для формирования сигнала $y(t)$</p>
<p>Программируемый источник сигнала</p>	<p>Вычисляемое выражение</p>	<p>Использует среду Julia для формирования сигнала $y(t)$ (см. 2.4)</p>	<p>Использует задаваемую пользователем формулу для формирования сигнала $y(t)$</p>
<p>Инструменты формирования подсистем (подробнее см. 2.3)</p>			
<p>Источник внешнего сигнала</p>			
<p>Вывод сигнала</p>			
<p>Системы управления</p>			
<p>ПИД-регулятор</p>	<p>Коэффициент пропорциональности</p>	<p>1</p>	<p>Использует формулу для формирования сигнала $y(t)$</p>
	<p>Интегральный коэффициент</p>	<p>1</p>	
	<p>Дифференциальный коэффициент</p>	<p>1</p>	
<p>Пользовательские элементы</p>			

3 Подсистемы

Имитационная модель может быть оформлена для включения в качестве составного элемента в другие модели с помощью элементов группы «Инструменты формирования подсистем». Для загрузки подсистемы в текущую модель используется элемент «Подсистема» группы «Основные элементы», в свойствах которого необходимо указать путь к загружаемому документу. Программа автоматически определит тип загружаемой модели и сформирует элемент с нужным количеством входов и выходов. В случае загрузки имитационной схемы в качестве подсистемы на базе элементов «Источник внешнего сигнала» формируются входы подсистемы, элементы «Вывод сигнала» формируют выходы подсистемы (рис. 4).

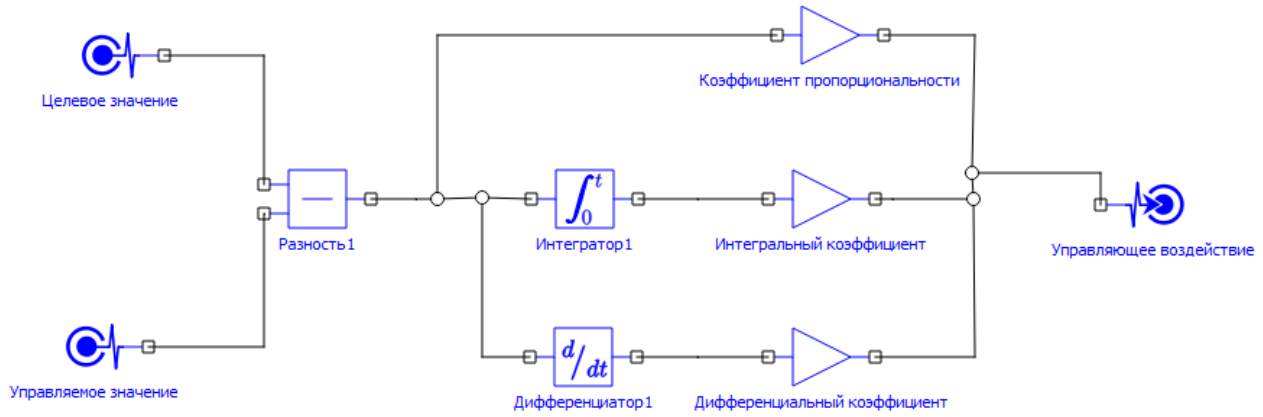


Рисунок 4 – ПИД-регулятор как пример подсистемы

Для удобства использования входам и выходам подсистемы могут быть даны имена, совпадающие с названием соответствующих элементов «Вывод сигнала» и «Источник внешнего сигнала». Имена входов и выходов подсистемы отображаются при увеличении схемы (рис. 5). Также обозначения портов подсистемы можно посмотреть во вкладке «Обозначения» в свойствах элемента. В той же вкладке указан URL загруженного документа. Для элементов, входящих в основную или пользовательскую библиотеку элементов, используется специальный режим доступа `lib://`.

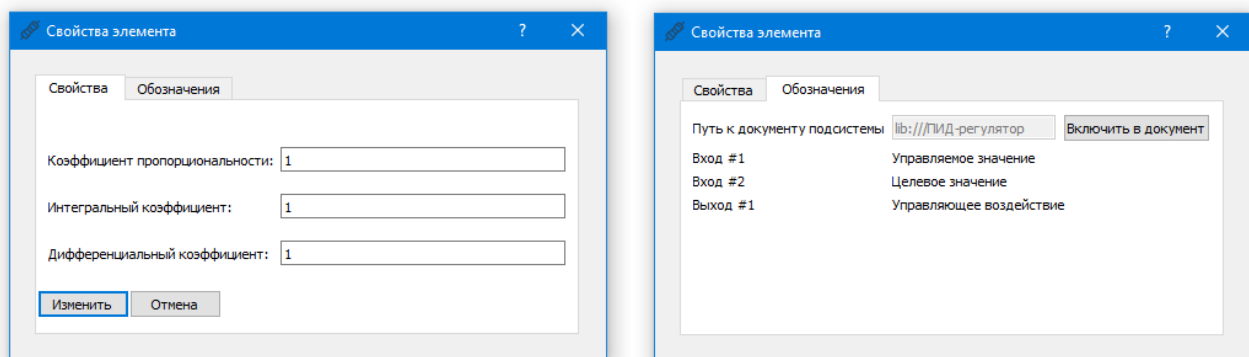
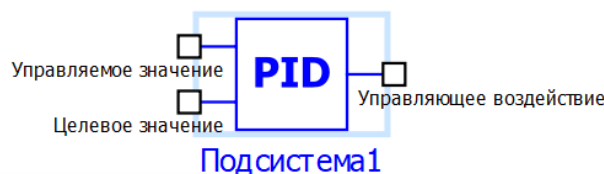


Рисунок 5 – Загруженная в документ подсистема

По умолчанию подсистема загружается по внешней ссылке, т.е. при сохранении документа, включающего в себя подсистему, содержание подсистемы не сохраняется вместе с ним, а указывается только место распо-

ложения документа подсистемы². Данное поведение удобно для отдельного редактирования различных компонент сложной модели. В том случае, когда не предполагается последующее редактирование подсистемы модели, возможно ее полное включение в текущий документ путем нажатия на кнопку «Включить в документ» во вкладке «Обозначения» в свойствах подсистемы. При выполнении данного действия документ подсистемы полностью копируется в документ текущей модели. Для подсистем, включенных в текущий документ, используется специальный режим доступа doc://.

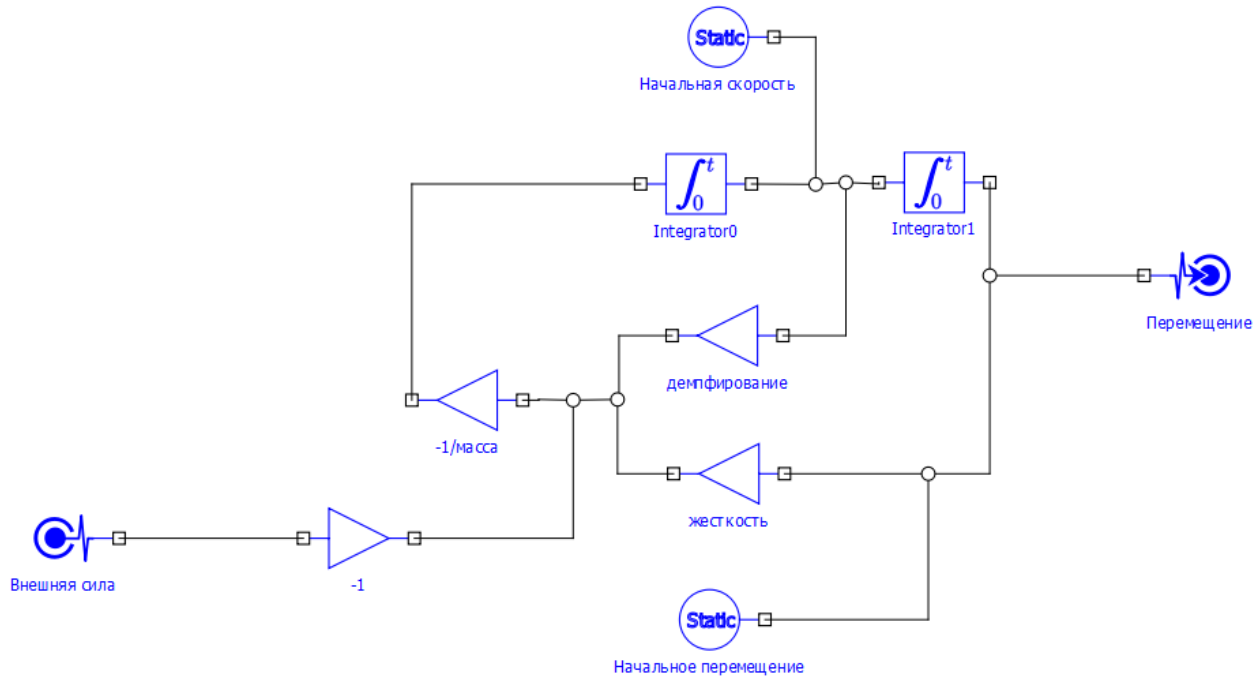


Рисунок 6 – Гармонический осциллятор как пример подсистемы

Основные настройки подсистемы доступны через меню «Правка - Экспортируемые параметры системы» и разделены на следующие категории:

1) *Экспортируемые элементы.* Любой элемент, присутствующий на схеме, может быть помечен как экспортируемый. Свойства экспортируемых элементов при загрузке подсистемы отображаются как свойства подсистемы. Меню настроек данной вкладки разделена на две части: верхняя таблица содержит перечень элементов, помеченных как экспортируемые, и их обозначения - имена, под которыми они будут фигурировать в свойствах подсистемы; снизу - таблица доступных для экспорта элементов. Допускается экспортировать только элементы, имеющие настраиваемые параметры (рис. 7). С помощью кнопок в левой части таблицы экспорта можно установить порядок, в котором должны перечисляться свойства подсистемы, а также удалить элемент из списка экспортируемых.

2) *Входы.* Содержит таблицу элементов схемы типа «Источник внешнего сигнала» и позволяет редактировать порядок, в котором будут представлены входы подсистемы (сверху вниз).

3) *Выходы.* Содержит таблицу элементов схемы типа «Вывод сигнала» и позволяет редактировать порядок, в котором будут представлены входы подсистемы (сверху вниз).

4) *Переменные.* Допускается экспортировать переменные, объявленные в документе. Экспортируемая переменная при загрузке подсистемы отображается как свойство подсистемы. Вкладка содержит перечень переменных документа и элементы управления, позволяющие отметить переменную как экспортируемую или внутреннюю (рис. 8).

5) *Локальные расчеты.* Позволяет отметить локальные расчеты (см. 2.5) элементов схемы как экспортируемые. Экспортируемые расчеты будут отображаться как локальные расчеты при загрузке подсистемы, название расчета указывается в колонке «Экспортируемый расчет».

6) *Настройки отображения.* Позволяет указать графическое обозначение (иконку) подсистемы. В качестве иконки используется изображение в формате SVG рекомендуемого размера 40x40px. По умолчанию для подсистем используются иконки, показанные на рис. 9.

²В случае, когда документ подсистемы и документ включающей ее модели находятся на одном разделе жесткого диска, используются относительные пути. В противном случае в документе сохраняется абсолютный путь к файлу подсистемы. Рекомендуется размещать документы подсистем в том же каталоге, что и использующие их модели

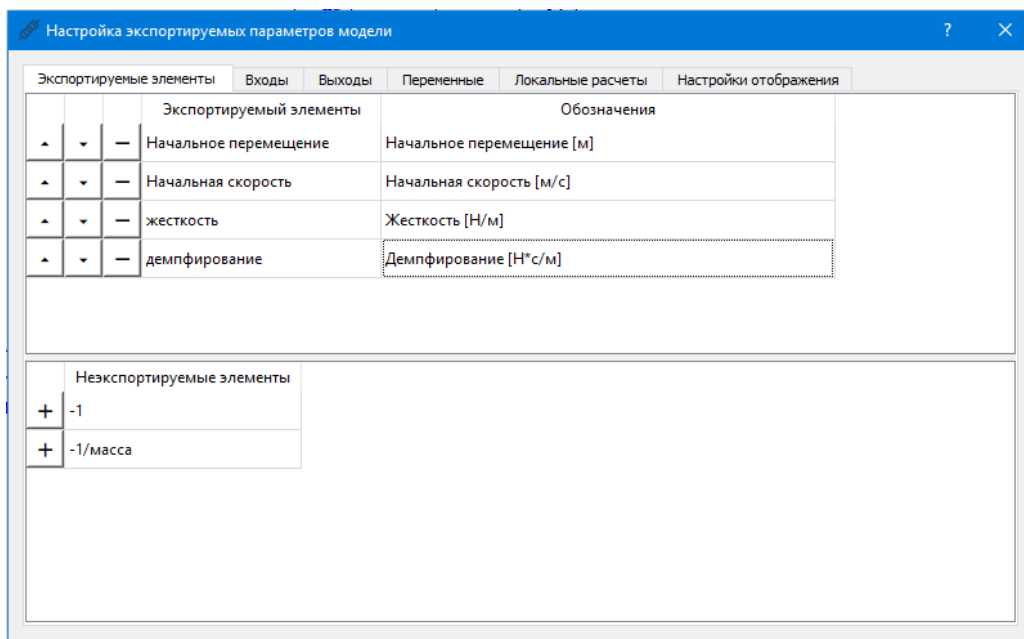


Рисунок 7 – Экспортируемые элементы подсистемы

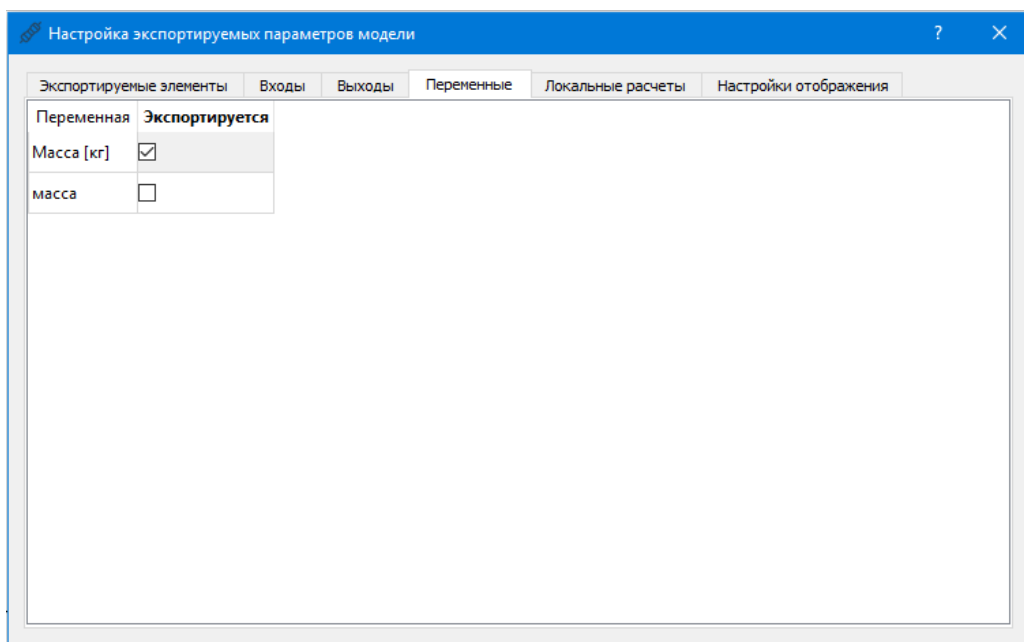
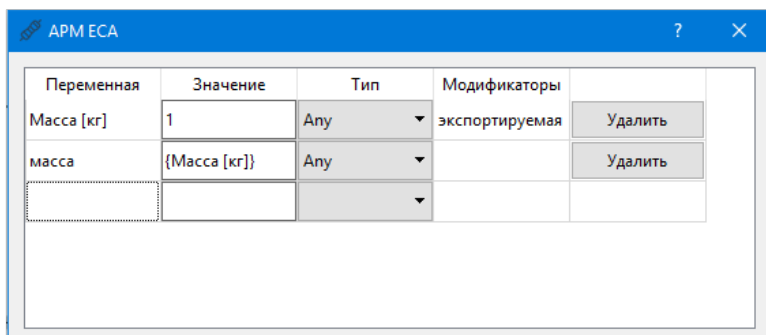
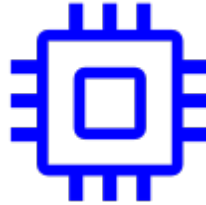


Рисунок 8 – Экспорт переменных окружения



(a) имитационная подсистема



(b) электрическая подсистема

Рисунок 9 – Иконки по умолчанию для подсистем



Затухающий гармонический осциллятор1

Свойства	Обозначения
Масса [кг]:	1
Начальное перемещение [м]:	1
Начальная скорость [м/с]:	10
Жесткость [Н/м]:	1000
Демпфирование [Н*с/м]:	10

Рисунок 10 – Загруженная подсистема гармонического осциллятора

4 Программируемые элементы и источники

Коллекцию доступных элементов схемы можно существенно расширить, используя встраиваемую среду Julia. Для определения необходимых программных сущностей используется встроенный редактор кода, вызываемый через меню «Правка - Редактировать исходный код».

В имитационных схемах доступна возможность определять следующие типы элементов:

Программируемые источники сигнала

Для создания источника сигнала Julia используется элемент «Программируемый источник сигнала» группы «Источники сигнала». В свойствах элемента необходимо указать имя функции одной переменной $f(t)$, t – время, возвращающей значение типа *Float64*, определенной в текущем окружении Julia (включающем функции стандартной библиотеки и функции, определенные в текущем документе).

Пример: *создание источника пилообразных импульсов с единичной амплитудой и частотой*

```
## Модуль встроенного кода Julia (www.julialang.org)
```

```
function Source(t::Float64)
    t = mod(t, 1)
    if t < 0.5
        return 2*t
    else
        return 2 - 2*t
    end
end
```

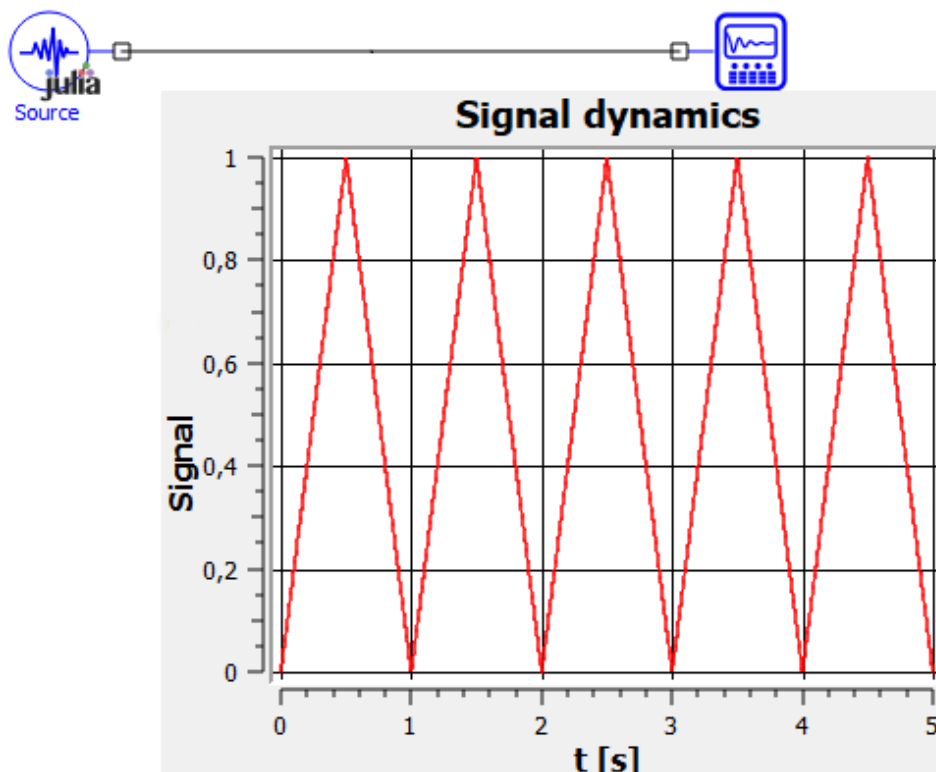


Рисунок 11 – Программируемый источник пилообразного сигнала

Для создания элемента, поведение которого описывается средствами языка Julia используется элемент «Программируемый элемент» группы «Основные элементы». В свойствах элемента необходимо указать тип элемента, определенный в окружении Julia.

Функциональные элементы

Элементы данной категории наследуются от абстрактного типа *SignalFuncElement*. Минимальное определение функционального элемента имеет следующий вид:

```
type FuncElementExample <: SignalFuncElement
  a1::Float64
  a2::Float64
  FuncElementExample() = new(1.0, 1.0)
end

function numPorts(e::FuncElementExample)
  return (2, 1)
end

function (e::FuncElementExample)(t::Float64, Inputs::Array{Float64})
  y = (Inputs[1]^e.a1 + Inputs[2]^e.a2)*sin(2*pi*t)
  return [y]
end
```

В данном примере определен элемент с именем *FuncElementExample*, имеющий два входа и один выход. Функциональный элемент обязательно должен содержать определение конструктора без параметров, задающий значение свойств элемента по умолчанию. Количество портов элемента задается путем определения перегрузки функции *numPorts*, принимающей объект созданного типа и возвращающий пару («Количество входов», «Количество выходов»). Описание поведения элемента задается через перегрузку оператора вызова для созданного типа, принимающего два аргумент: текущее время $t :: Float64$ и вектор значений, поданных на вход элемента $Inputs :: Array\{Float64\}$. Оператор должен возвращать вектор значений, подаваемых на выход элемента. Важно: *результат обязательно должен иметь тип $Array\{Float64\}$, даже если элемент имеет единственный выход*. В показанном примере выходом является гармонический сигнал, амплитуда которого определяется по формуле $x_1^{a_1} + x_2^{a_2}$, где x_1 и x_2 – входы элемента, a_1 и a_2 – свойства элемента.

Элементы, определяемые уравнением

В случае, когда отсутствует явная зависимость выходов элемента от его входов, возможно определить элемент через уравнение $G(Inputs, Outputs) = 0$. Элементы данной категории наследуются от абстрактного типа *SignalEquationElement*. Минимальное определение элемента имеет следующий вид:

```
type EqElementExample <: SignalEquationElement
  a1::Float64
  a2::Float64
  EqElementExample() = new(2.0, 1.5)
end

function numPorts(e::EqElementExample)
  return (2, 2)
end

function (e::EqElementExample)(t::Float64, Inputs::Array{Float64}, Outputs::Array{Float64})
  f1 = log( e.a1 + Inputs[2]*Outputs[1] ) + Outputs[1]
  f2 = log( e.a2 + Inputs[1]*Outputs[2] ) + Outputs[2]
  return [f1; f2]
end
```

Определение элемента, наследуемого от абстрактного типа *SignalEquationElement*, полностью повторяет определение функционального элемента за исключением того, что в перегрузке оператора вызова присутствует третий аргумент $Outputs :: Array\{Float64\}$, содержащий значения выходов. Результатом

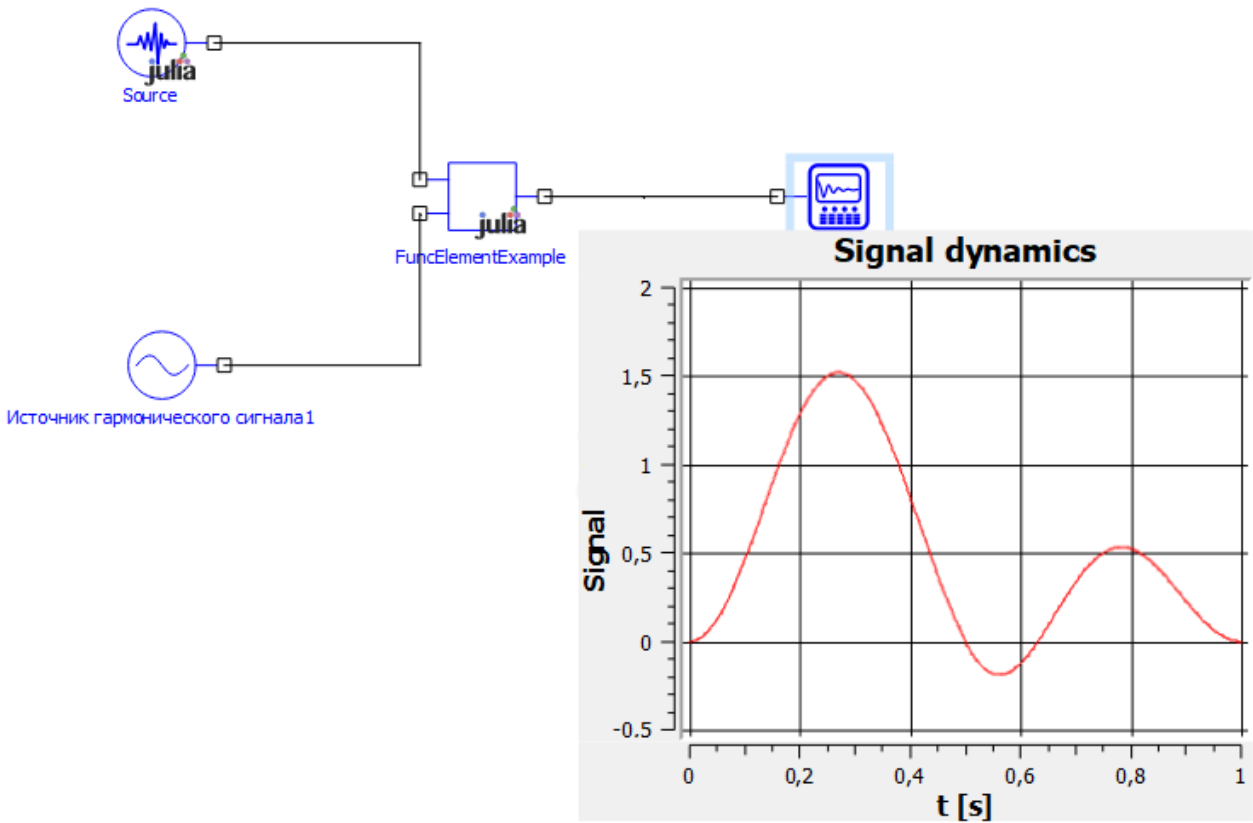


Рисунок 12 – Пример функционального элемента Julia

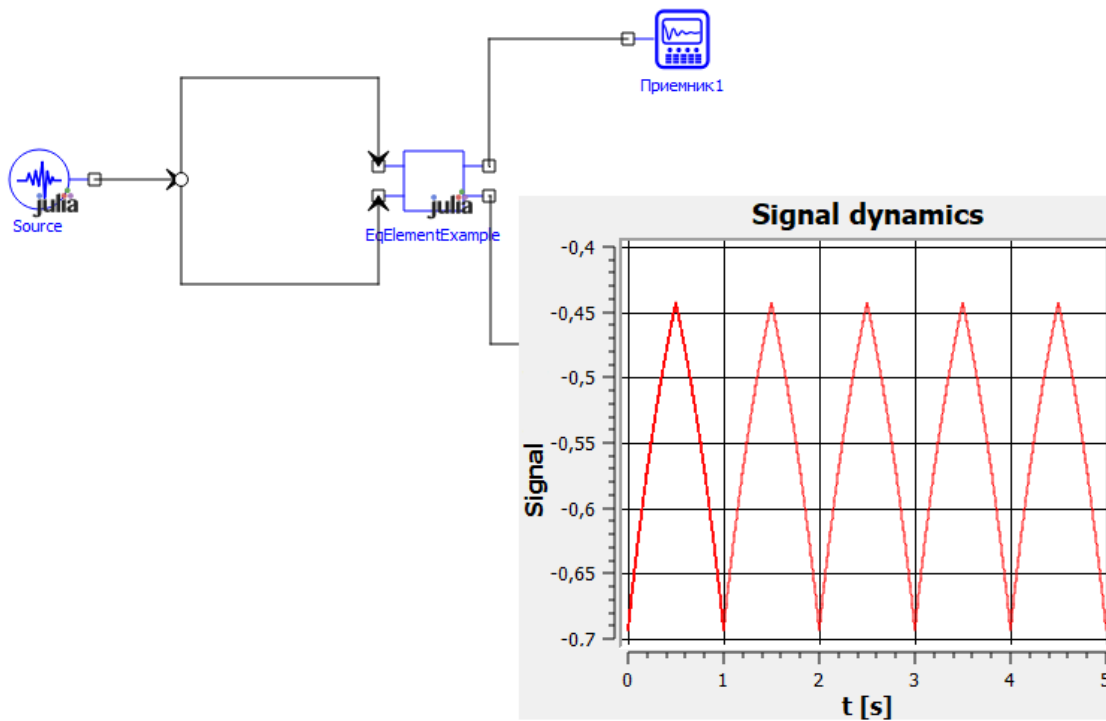


Рисунок 13 – Пример элемента Julia, заданного уравнением

перегруженного оператора должен быть массив размерности равной количеству выходов элемента и содержащий значение $G(Inputs, Outputs)$ – невязку векторного уравнения.

Линейные элементы и предварительно вычисляемые параметры

В ряде случаев удобно (или более эффективно) определять поведение элемента непосредственно перед проведением расчета, когда все параметры установлены. Для этого вместо определения указанных выше форм оператора вызова необходимо определить оператор вызова без аргументов, который возвращает значения типа `SignalFuncElement`, `SignalEquationElement` или `SignalLinearElement`. Последний имеет следующее определение:

```
type SignalLinearElement <: SignalElement
    A::Array{Float64,2}
    b::Array{Float64}
    SignalLinearElement(A::Array{Float64,2}, b::Array{Float64}) = new(A, b)
end

function (e::SignalLinearElement)(t::Float64, Inputs::Array{Float64})
    return e.A * Inputs + e.b
end
```

Тип `SignalLinearElement` предназначен для описания линейных функциональных элементов, выходы которых определяются по формуле $Outputs = A \cdot Inputs + b$. В качестве примера приведем описание элемента, поведение которого определяется формулой $Y = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} X + \begin{bmatrix} x \\ y \end{bmatrix}$, где α , x и y – параметры элемента:

```
type Func1ElementExample <: SignalFuncElement
    alpha::Float64
    x::Float64
    y::Float64
    Func1ElementExample() = new(0.5*pi)
end

function numPorts(e::Func1ElementExample)
    return (2, 2)
end

function (e::Func1ElementExample)()
    A = [cos(e.alpha) -sin(e.alpha); sin(e.alpha) cos(e.alpha)]
    b = [e.x; e.y]
    return SignalLinearElement(A, b)
end
```

Преимущество такой реализации по сравнению со стандартной перегрузкой оператора вызова заключается в том, что в данном случае матрица и вектор вычисляется единственный раз в момент запуска расчета, в то время как при обычной реализации вычисления проводятся на каждой итерации расчета. Оператор вызова без аргументов имеет смысл перегружать тогда, когда необходимо провести трудоемкий расчет параметров элемента перед непосредственным запуском основного расчета. При этом возвращаемым значением оператора может быть исходный вызываемый объект.

Использование модулей и пакетов Julia

Программа использует среду Julia, установленную на компьютере пользователя и, следовательно, при редактировании исходного кода модели можно использовать любой из дополнительно загружаемых модулей и пакетов (<https://pkg.julialang.org>). Пользователь также может создавать собственные программные модули, доступные только внутри программы APM ECA. Для этого модули необходимо разместить в каталоге `julia` основной директории программы (по умолчанию при установке программы используется директория `%ProgramFiles%/APM ECA`).

```

# Созданный пользователем модуль
module MyModule

export something

function something
...
end

end # of module

# Исходный код документа
## Модуль встроенного кода Julia (www.julialang.org)

using MyModule

type MyType <: SignalFuncElement
...
end

...

function (e::MyType)(t::Float64, Inputs::Array{Float64})
    y = something(Inputs[1])
    return [y]
end

```

Определение локальных расчетов для элементов Julia

С элементами, описанными на языке Julia, также как и со стандартными элементами имитационных схем, могут быть связаны определенные типы расчетов (т.н. *локальные расчеты*, см. 2.5). Для описания локального расчета требуется определить объект типа, унаследованного от абстрактного типа *Calculation*:

```

type Тестовый_Расчет <: Calculation
параметр1::Float64
параметр2::Float64
Тестовый_Расчет() = new(1, 1)
end

```

По умолчанию название расчета совпадает с именем типа расчета. Чтобы это изменить, требуется перегрузить для созданного типа расчета функцию *Base.show*:

```

function Base.show(io::IO, e::Тестовый_Расчет)
print(io, "Тестовый расчет")
end

```

Для описания действий, которые необходимо произвести в данном локальном расчете, необходимо перегрузить оператор вызова:

```

function (a::Тестовый_Расчет)(e::MyType)
...
end

```


Аргументом оператора является элемент, для которого данный расчет должен быть произведен. Результатом оператора должно являться значение типа, унаследованного от *ResultsObject*. В текущей версии APM ECA поддерживаются следующие типы результатов:

- двумерный график в декартовых координатах

```
type Cartesian2D <: ResultsObject
xLabel::String
X::Array{Float64,1}
yLabel::String
Y::Array{Float64,1}
Cartesian2D(xLabel::String, X::Array{Float64,1}, yLabel::String, Y::Array{Float64,1}) = new(xLabel, X,
Cartesian2D(X::Array{Float64,1}, Y::Array{Float64,1}) = new("X", X, "Y", Y)
end
```

- параметрическая двумерная кривая в декартовых координатах

```
type CartesianParametric2D <: ResultsObject
xLabel::String
X::Array{Float64,1}
yLabel::String
Y::Array{Float64,1}
tLabel::String
T::Array{Float64,1}
CartesianParametric2D(xLabel::String, X::Array{Float64,1}, yLabel::String, Y::Array{Float64,1}, tLabel:
CartesianParametric2D(X::Array{Float64,1}, Y::Array{Float64,1}, T::Array{Float64,1}) = new("X", X, "Y",
end
```

- двумерный график в полярных координатах

```
type Polar2D <: ResultsObject
phiLabel::String
Phi::Array{Float64,1}
rLabel::String
R::Array{Float64,1}
Polar2D(phiLabel::String, Phi::Array{Float64,1}, rLabel::String, R::Array{Float64,1}) = new(phiLabel, P
Polar2D(Phi::Array{Float64,1}, R::Array{Float64,1}) = new("угол", X, "расстояние", Y)
end
```

Список локальных расчетов, связанных с типом элементов Julia, задается с помощью перегрузки функции *calculations*, принимающей в качестве единственного аргумента элемент, для которого должен быть проведен расчет и возвращающей список объектов-расчетов для данного элемента.

Пример: *локальный расчет, осуществляющий вывод зависимости между входом и выходом программируемого элемента в указанном диапазоне значений*

```
## Модуль встроенного кода Julia (www.julialang.org)
```

```
type LogElement <: SignalFuncElement
    основание::Float64
    LogElement() = new(exp(1.0))
end

function numPorts(e::LogElement)
    return (1, 1)
end

function (e::LogElement)(t::Float64, Inputs::Array{Float64})
    return [ log(e.основание, Inputs[1]) ]
end
```

```

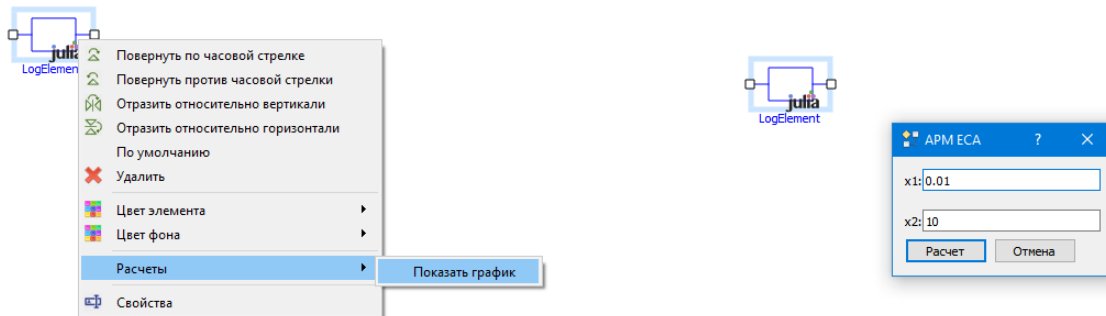
type LogElementCalculation <: Calculation
x1::Float64
x2::Float64
LogElementCalculation() = new(0.01, 10.0)
end

function Base.show(io::IO, e::LogElementCalculation)
print(io, "Показать график")
end

function (a::LogElementCalculation)(e::LogElement)
X = [x for x in a.x1:0.1:a.x2]
Y = map(x -> e(0.0, [x])[1], X)
return Cartesian2D("X", X, string("log(", e.основание, ", x)"), Y)
end

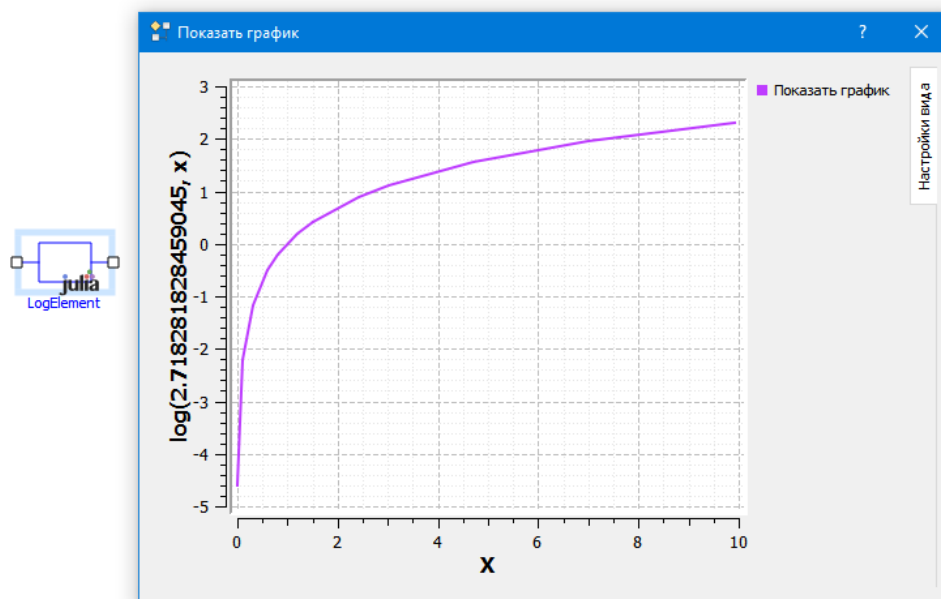
function calculations(e::LogElement)
return [LogElementCalculation()]
end

```



(a) меню локальных расчетов

(b) параметры расчета



(c) результаты расчета

Рисунок 14 – Выполнение локального расчета программируемого элемента Julia

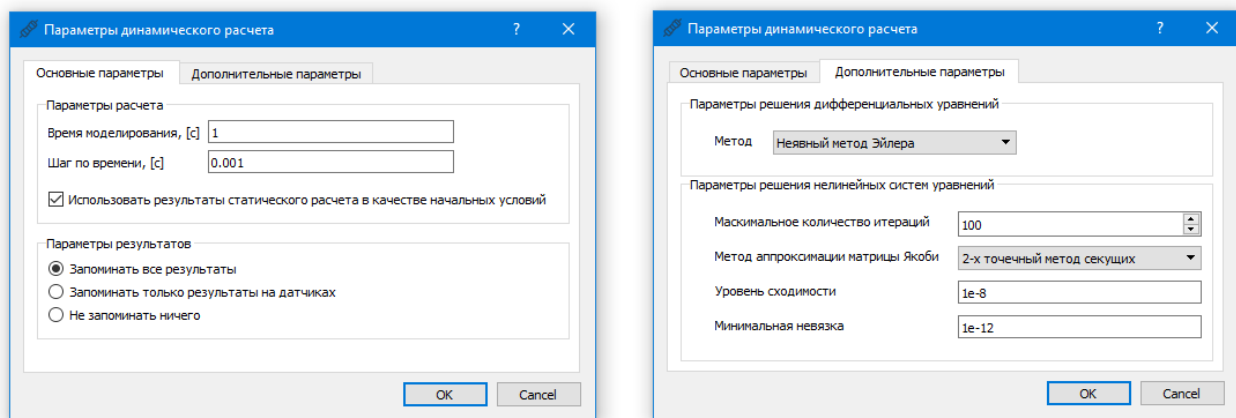
5 Проведение расчетов

Для имитационных моделей доступно два типа расчетов:

- динамический расчет, предназначен для симуляции поведения системы во времени;
- локальные расчеты, связанные с определенными элементами. Предназначены для вычисления характеристик элементов схемы и автоматической настройки их параметров. В некоторых случаях проведение локальных расчетов требует проведения одного или нескольких динамических расчетов модели.

Локальные расчеты специфичны для конкретных элементов и могут быть вызваны через их контекстное меню, пункт «Расчеты».

Выполнение динамического расчета доступно через меню «Расчет - Динамический расчет». В диалоговом окне «Параметры динамического расчета» имеются две вкладки: «Основные параметры» (рис. 18a) и «Дополнительные параметры» (рис. 18b).



(a) основные параметры

(b) дополнительные параметры

Рисунок 15 – Параметры динамического расчета имитационных моделей

В основных параметрах предлагается установить время моделирования и величину шага расчета. Параметр «Шаг по времени» определяет расстояние между точками расчетной сетки, в которых значение искомого сигнала должно быть вычислено. В случае необходимости программа автоматически уменьшает величину шага дискретизации для достижения необходимой точности, однако *любая рассчитываемая траектория гарантированно проходит через все точки расчетной сетки*. Включение опции «Использовать результаты статического расчета в качестве начальных условий» приводит к решению задачи статики (нахождения равновесного состояния системы без учета динамических элементов типа дифференциаторов и интеграторов) в момент времени $t = 0$. В случае отключения данной опции в начальный момент времени значения во всех связях модели принимаются равными нулю.

При проведении динамического расчета предлагается выбрать один из трех вариантов сохранения результатов:

- запомнить все результаты. При активации данного режима по завершении расчета будут доступны сигналы во всех связях модели;
- запоминать только результаты на датчиках. В данном режиме будут доступны только результаты в связях, примыкающих к элементам типа «Приемник»;
- не запоминать ничего. Активация данного режима предполагает, что по завершении расчета ни один сигнал не может быть показан на рабочей области. Данную опцию следует использовать только для проверки корректности больших моделей с целью экономии памяти вычислительной машины, либо когда требуются только результаты, выводимые непосредственно в процессе расчета, например посредством элемента «Приемник с выводом в файл».

Во вкладке «Дополнительные параметры расчета» доступны более тонкие настройки процедуры расчета, включающие выбор метода дискретизации дифференциально-интегральных уравнений, метода решения возникающих в процессе моделирования систем нелинейных алгебраических уравнений.

В программе APM ECA доступны следующие методы дискретизации дифференциально-интегральных операторов:

- неявный метод Эйлера.

Для методов решения систем нелинейных алгебраических уравнений можно указать следующие параметры:

– максимальное количество выполняемых итераций, по достижению которого расчет считается неудачным. Данную величину, как правило, не следует делать слишком большой, поскольку большое количество выполняемых итераций часто свидетельствует о необходимости уменьшить шаг по времени. Исключение составляют модели, в которых наблюдаются резкие скачки или разрывы производных;

– метод аппроксимации матрицы Якоби. Определяет способ приближения неизвестной части матрицы $\frac{\partial G}{\partial X}$ уравнения $G(X) = 0$, описывающего состояние системы в определенный момент времени.

Доступны следующие варианты аппроксимации:

- 2-х точечный метод секущих. Является наиболее точной численной аппроксимацией, однако требует наибольшего количества вспомогательных вычислений. Рекомендуется только в случае, когда использование метода аппроксимации, установленного по умолчанию приводит к невозможности получения результата установленной точности.
- конечно-разностная аппроксимация в начальной точке. Наименее затратна в вычислительном плане, однако наименее точна и может приводить к расхождению расчета при большом количестве требуемых итераций;
- метод Бройдена (первого порядка). Требует незначительно больше вычислительных ресурсов по сравнению с конечно-разностной аппроксимацией в начальной точке, однако обладает большей скоростью сходимости и устойчивостью. Используется по умолчанию.

– уровень сходимости ϵ . Определяет максимальное значение приращения между итерациями, при котором процесс решения системы нелинейных уравнений может считаться сошедшимся. Решение X_{i+1} считается допустимым приближением истинного решения только при выполнении условия $\|X_{i+1} - X_i\| < \epsilon$. Рекомендуется устанавливать значение данного параметра большим, чем величину минимально допустимой невязки, однако чрезмерно большое значение данного параметра может приводить к появлению ложных скачков в решении в случае наличия широких областей малого изменения функции $G(X)$. При появлении неоправданно резких изменений в решении следует уменьшить величину данного параметра;

– минимальная невязка δ . Определяет наибольшее допустимое отклонение функции $G(X)$ от нуля. Иначе, X_i принимается в качестве приближенного решения уравнения $G(X) = 0$ только если $G(X_i) < \delta$. Значение данной величины не должна быть меньше погрешности вычисления $G(X)$, что следует учитывать при использовании программируемых элементов. Не рекомендуется устанавливать значение минимальной невязки меньше $1e - 9$.

Глава 3

Электрические схемы

1 Основные принципы формирования моделей электрических цепей

Основными элементами электрических схем являются электрические элементы (далее просто элементы), узлы и связи (рис. 1). Основу для формирования математической модели электрической схемы составляет первый закон Кирхгофа, утверждающий, что в каждый момент времени сумма токов в любом из узлов схемы равна нулю. Расчет электрической схемы позволяет определить значения электрического потенциала в каждом из узлов схемы и величину электрического тока, проходящего через каждый из элементов и связей. Важно: *поскольку значения электрического тока в цепи определяются разностями потенциалов в узлах, электрическая схема должна включать в себя как минимум один элемент типа «Земля», определяющий узел, потенциал в котором берется в качестве точки отсчета. При отсутствии заземленных узлов попытка провести расчет электрической схемы приведет к возникновению ошибки. Выбор заземленного узла не влияет на результаты расчета.*

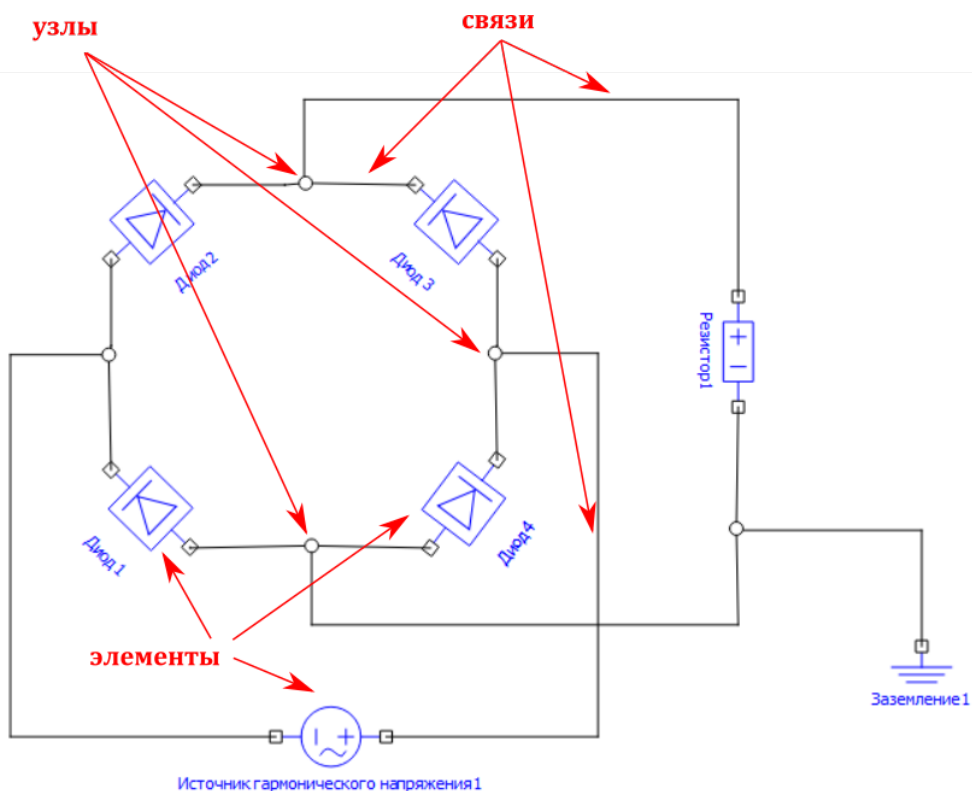
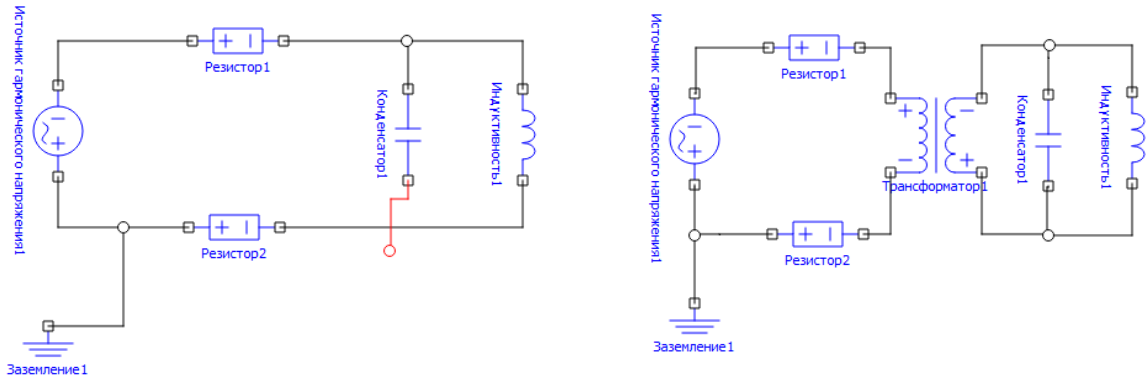


Рисунок 1 – Пример электрической схемы

Для проведения расчета требуется, чтобы в схеме отсутствовали элементы, у которых хотя бы один из портов не связан с другими элементами схемы. Допускается, чтобы в одном документе присутствовало несколько топологически не связанных электрических компонент, однако в этом случае каждая компонента должна иметь хотя бы один заземленный узел. Важно: *Элемент типа «Идеальный трансформатор» не создает физическую, но не топологическую связь между двумя электрическими цепями. Если две*

части электрической модели связаны между собой через идеальный трансформатор, то в каждой из этих частей должен присутствовать заземленный узел. Программа не имеет встроенной проверки топологической связности схемы, вследствие чего отсутствие заземления в одной из частей схемы не приведет к возникновению сообщения об ошибке, однако может привести к ошибке расчета или к получению некорректных результатов. На рис. 2а и 2б приведены примеры некорректных схем.

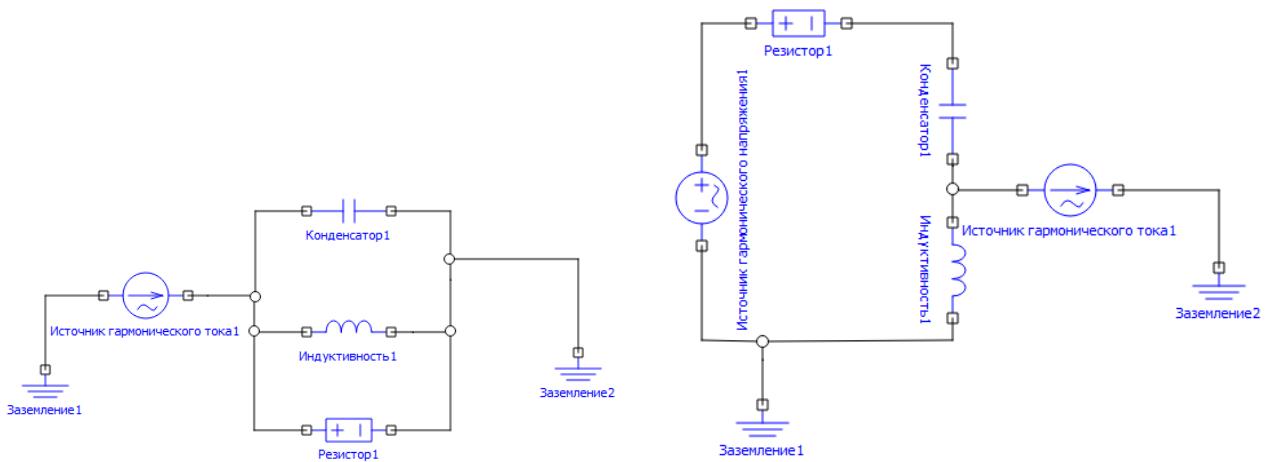


(а) присутствует не подключенный порт элемента

(б) отсутствует заземление в правой цепи трансформатора

Рисунок 2 – Примеры некорректных электрических схем

В случае присутствия в схеме источника тока с заземленным портом, что обычно отражает наличие связи рассматриваемого участка электрической цепи с другими электрическими системами, необходимо добавить заземление к любому другому узлу внутреннего участка цепи.



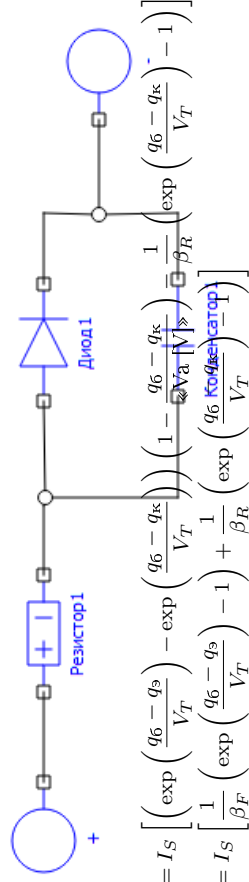
(а) ток приходит из внешнего участка электрической цепи в LC-контур

(б) ток уходит из LC-контура во внешний участок электрической цепи

Рисунок 3 – Схемы с источником тока, требующие дополнительного заземления

2 Стандартные элементы

Название элемента	Параметр	Значение по умолчанию	Описание
условные обозначения: q_{index} – величина потенциала в узле, подключенном к порту <i>index</i> , I_{index} – величина тока, выходящего из элемента через порт <i>index</i>			
Основные элементы			
Земля		$q = 0$	
Резистор	Сопротивление [Om]	1	$I_+ = \frac{q_+ - q_-}{\text{«Сопротивление [Om]»}}$
Конденсатор	Емкость [Ф]	1	$I_+ = \text{«Емкость [Ф]»} \frac{d(q_+ - q_-)}{dt}$
Индуктивность	Индуктивность [Гн]	1	$I_+ = \frac{1}{\text{«Индуктивность [Гн]»}} \int_0^t (q_+ - q_-) d\tau$
Идеальный трансформатор	Коэффициент трансформации	1	$\frac{q_{L+} - q_{L-}}{q_{R+} - q_{R-}} = \frac{I_{R+}}{I_{L+}} = \text{«Коэффициент трансформации»}$
Подсистема	Опционально		Предназначен для загрузки ранее созданных моделей в текущую модель (см. 3.3)
Программируемый элемент	Опционально		Использует определенные пользователем типы элементов, описанные на языке Julia (см. раздел 3.4)
Полупроводниковые элементы			
Диод	Ток насыщения [A]	1e-14	$I_+ = \text{«Ток насыщения [A]»} \cdot \left(\exp\left(\frac{q_+ - q_-}{V_T}\right) - 1 \right)$, V_T – термическое напряжение при температуре 298.15K
Диод (подсистема)	Ток насыщения [A]	1e-14	
	Сопротивление перехода [Om]	0.01	
NPN-транзистор	Емкость перехода [Ф]	5e-12	
	Ток насыщения [A] $\equiv I_S$	1e-14	$I_K = I_S \left[\left(\exp\left(\frac{q_6 - q_9}{V_T}\right) - \exp\left(\frac{q_6 - q_K}{V_T}\right) \right) \left(1 - \frac{q_6 - q_K}{2V_A} \right) + \frac{1}{\beta_R} \left(\exp\left(\frac{q_6 - q_K}{V_T}\right) - 1 \right) \right]$
	Коэффициент усиления на эмиттере в прямом направлении $\equiv \beta_F$	100	$I_6 = I_S \left[\frac{1}{\beta_F} \left(\exp\left(\frac{q_6 - q_9}{V_T}\right) - 1 \right) + \frac{1}{\beta_R} \left(\exp\left(\frac{q_6 - q_K}{V_T}\right) - 1 \right) \right]$
	Коэффициент усиления на эмиттере в обратном направлении $\equiv \beta_R$	1	$I_9 = -I_K - I_6$
	V_A [V]	200	



Ток насыщения [A] 1e-14

Коэффициент усиления на эмиттере в прямом направлении $\equiv \beta_F$ 100

Коэффициент усиления на эмиттере в обратном направлении $\equiv \beta_R$ 1

NPN-транзистор (подсистема)

V_a [V] 200

Сопротивление коллектора [Ом] 0.01

Сопротивление эмиттера [Ом] 0.0001

Сопротивление базы [Ом] 1

Емкость перехода коллектор-база 1e-12

Емкость перехода эмиттер-база 1e-12

$$I_{\kappa} = -I_S \left[\left(\exp \left(\frac{q_s - q_6}{V_T} \right) - \exp \left(\frac{q_{\kappa} - q_6}{V_T} \right) \right) \left(1 - \frac{q_{\kappa} - q_6}{\ll V_a [V] \gg} \right) - \frac{1}{\beta_R} \left(\exp \left(\frac{q_{\kappa} - q_6}{V_T} \right) - 1 \right) \right]$$

$$I_6 = -I_S \left[\frac{1}{\beta_F} \left(\exp \left(\frac{q_s - q_6}{V_T} \right) - 1 \right) + \frac{1}{\beta_R} \left(\exp \left(\frac{q_{\kappa} - q_6}{V_T} \right) - 1 \right) \right]$$

$$I_3 = -I_{\kappa} - I_6$$

PNP-транзистор

Коэффициент усиления на эмиттере в обратном направлении $\equiv \beta_R$ 100

V_a [V] 200

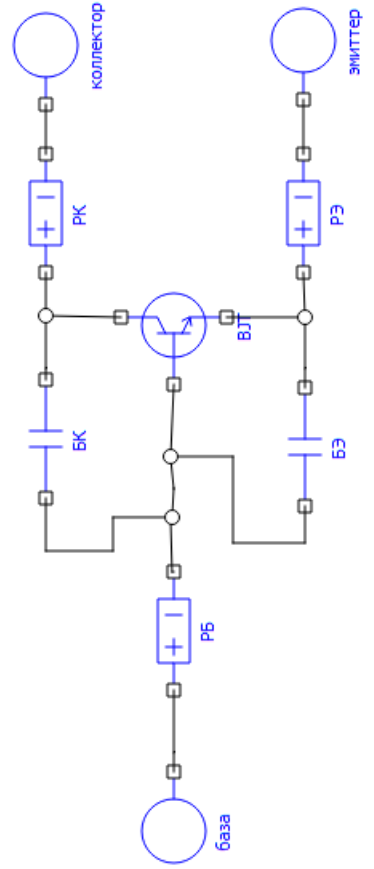
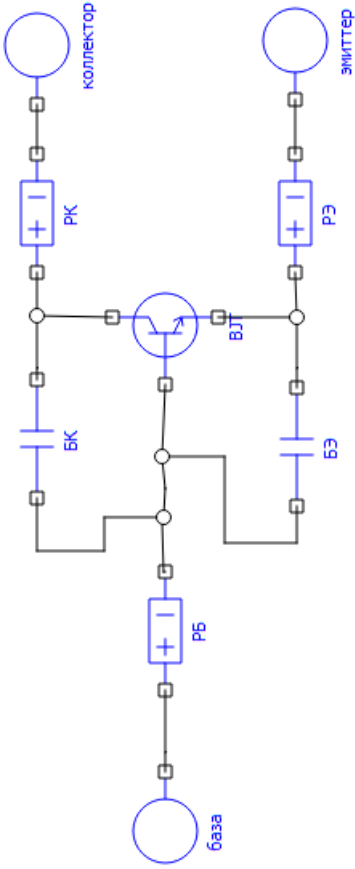
Ток насыщения [A] 1e-14

Коэффициент усиления на эмиттере в прямом направлении $\equiv \beta_F$ 100

Коэффициент усиления на эмиттере в обратном направлении $\equiv \beta_R$ 1

PNP-транзистор (подсистема)

V_a [V] 200



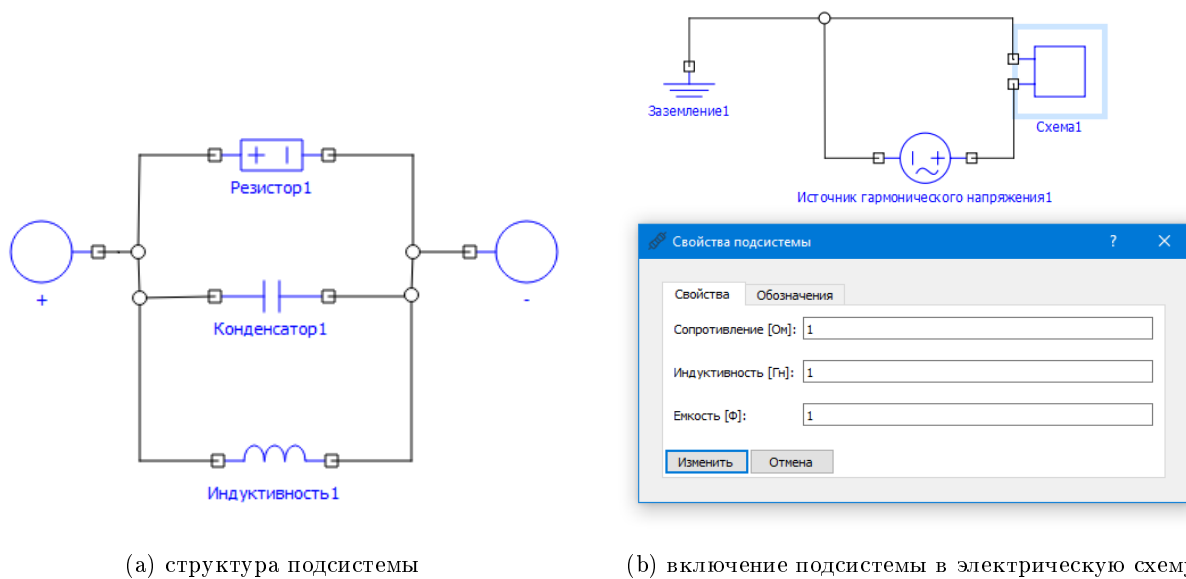
Сопротивление коллектора [Ом]	0.01		
Сопротивление эмиттера [Ом]	0.0001		
Сопротивление базы [Ом]	1		
Емкость перехода коллектор-база	1e-12		
Емкость перехода эмиттер-база	1e-12		
Источники			
Источник тока DC	Сила тока [A]	1	$I_+ = \text{«Сила тока [A]»}$
Напряжение DC	Напряжение [V]	1	$q_+ - q_- = \text{«Напряжение [V]»}$
Источник тока AC	Амплитуда [A]	1	
	Фаза [рад]	0	$I_+ = \text{«Амплитуда [A]»} \sin(2\pi t \cdot \text{«Частота [Гц]»} + \text{«Фаза [рад]»})$
	Частота [Гц]	60	
	Амплитуда [A]	1	
Напряжение AC	Фаза [рад]	0	$q_+ - q_- = \text{«Амплитуда [A]»} \sin(2\pi t \cdot \text{«Частота [Гц]»} + \text{«Фаза [рад]»})$
	Частота [Гц]	60	
Источник переменного тока	Вычисляемое выражение		Использует задаваемую пользователем формулу для определения значения $I_+(t)$
Источник переменного напряжения	Вычисляемое выражение		Использует задаваемую пользователем формулу для определения значения $q_+(t) - q_-(t)$
Программируемый источник тока			Использует среду Julia для формирования для определения значения $I_+(t)$ (см. 3.4)
Программируемый источник напряжения			Использует среду Julia для формирования для определения значения $q_+(t) - q_-(t)$ (см. 3.4)
Измерительные приборы			
Вольтметр			Выводит значение $q_+ - q_-$ в рабочей области схемы
Вольтметр с выводом в файл	Путь к файлу		Выводит значение $q_+ - q_-$ текстовый файл в процессе расчета
Амперметр			Выводит значение I_+ в рабочей области схемы

Амперметр с выводом в файл	Путь к файлу	Выводит значение I_+ текстовый файл в процессе расчета
Инструменты формирования подсистем (подробнее см. 3.3)		
Контакт		Определяет точку возможного подключения других электрических цепей
Управляемый источник тока		I_+ задается внешним процессом
Управляемое напряжение		q_+ – q_- задается внешним процессом
Датчик напряжения		q_+ – q_- передается во внешний процесс
Датчик тока		I_+ передается во внешний процесс

3 Подсистемы

Электрическая схема может быть оформлена для включения в качестве составной части в другую электрическую или имитационную схему с помощью элементов группы «Инструменты формирования подсистем». Оформленную специальным образом схему можно загрузить в текущую электрическую схему с помощью элемента «Подсистема» из группы «Основные элементы». В электрическую схему допускается загружать только другие электрические схемы.

Элементы типа «Контакт» используются для оформления электрической схемы как возможной части других электрических схем. Узел, к которому подключен элемент типа «Контакт», при загрузке данной схемы в другую электрическую схему будет преобразован в порт электрического элемента, имена элементов типа «Контакт» преобразуются в имена соответствующих портов электрического элемента (рис. 4). Имена входов и выходов подсистемы отображаются при увеличении схемы. Также обозначения портов подсистемы можно посмотреть во вкладке «Обозначения» в свойствах элемента. В той же вкладке указан URL загруженного документа. Для элементов, входящих в основную или пользовательскую библиотеку элементов, используется специальный режим доступа `lib://`.



(а) структура подсистемы

(б) включение подсистемы в электрическую схему

Рисунок 4 – Пример использования электрической подсистемы

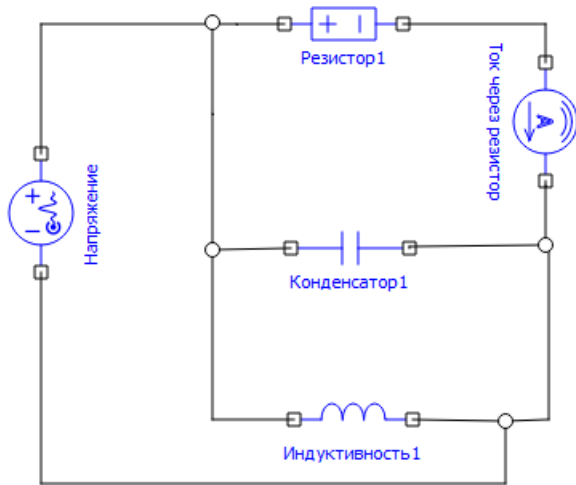
По умолчанию подсистема загружается по внешней ссылке, т.е. при сохранении документа, включающего в себя подсистему, содержание подсистемы не сохраняется вместе с ним, а указывается только место расположения документа подсистемы¹. Данное поведение удобно для раздельного редактирования различных компонент сложной модели. В том случае, когда не предполагается последующее редактирование подсистемы модели, возможно ее полное включение в текущий документ путем нажатия на кнопку «Включить в документ» во вкладке «Обозначения» в свойствах подсистемы. При выполнении данного действия документ подсистемы полностью копируется в документ текущей модели. Для подсистем, включенных в текущий документ, используется специальный режим доступа `doc://`.

Для использования электрической системы в имитационных моделях используются элементы «Управляемый источник тока», «Управляемое напряжение», «Датчик тока» и «Датчик напряжения». При загрузке электрической схемы в имитационную модель управляемые источники преобразуются во входы элемента имитационной модели, датчики – в выходы (рис. 5).

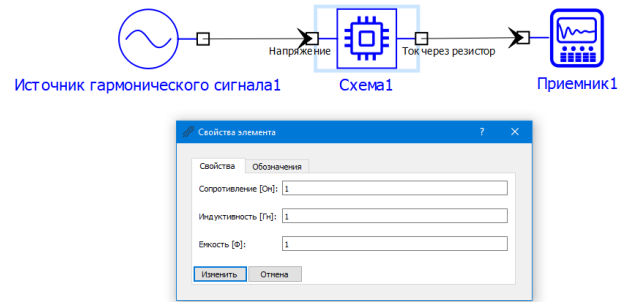
Важно: обычно при создании подсистем для электрических систем нет необходимости в заземлении какого-либо узла за исключением случаев, когда в подсистеме содержится электрическая цепь, топологически не связанная ни с одним из контактов. Формально заземление может присутствовать на любом из уровней вложенности схемы, однако строго рекомендуется указывать заземленный узел только на верхнем уровне модели. Напротив, при оформлении электрической схемы с целью включения ее в состав имитационной модели, указание заземленного узла является необходимым, в противном случае попытка расчета приведет к возникновению ошибки.

Основные настройки подсистемы доступны через меню «Правка - Экспортируемые параметры системы» и разделены на следующие категории:

¹ В случае, когда документ подсистемы и документ включающей ее модели находятся на одном разделе жесткого диска, используются относительные пути. В противном случае в документе сохраняется абсолютный путь к файлу подсистемы. Рекомендуется размещать документы подсистем в том же каталоге, что и использующие их модели



(a) структура подсистемы



(b) включение подсистемы в имитационную схему

Рисунок 5 – Пример использования электрической подсистемы

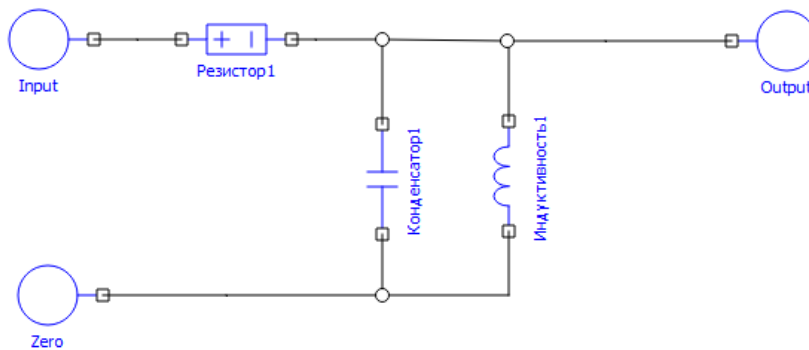


Рисунок 6 – LC-фильтр как пример подсистемы

1) *Экспортируемые элементы.* Любой элемент, присутствующий на схеме, может быть помечен как экспортируемый. Свойства экспортируемых элементов при загрузке подсистемы отображаются как свойства подсистемы. Меню настроек данной вкладки разделена на две части: верхняя таблица содержит перечень элементов, помеченных как экспортируемые, и их обозначения - имена, под которыми они будут фигурировать в свойствах подсистемы; снизу - таблица доступных для экспорта элементов. Допускается экспортировать только элементы, имеющие настраиваемые параметры (рис. 7). С помощью кнопок в левой части таблицы экспорта можно установить порядок, в котором должны перечисляться свойства подсистемы, а также удалить элемент из списка экспортируемых.

2) *Контакты.* Содержит таблицу элементов схемы типа «Контакт» и позволяет указать положение соответствующего порта подсистемы: слева, справа, снизу или сверху, а также устанавливать порядок, в котором будут представлены порты подсистемы (сверху-вниз, слева-направо).

3) *Переменные.* Допускается экспортировать переменные, объявленные в документе. Экспортируемая переменная при загрузке подсистемы отображается как свойство подсистемы. Вкладка содержит перечень переменных документа и элементы управления, позволяющие отметить переменную как экспортируемую или внутреннюю (рис. 8).

4) *Настройки отображения.* Позволяет указать графическое обозначение (иконку) подсистемы. В качестве иконки используется изображение в формате SVG рекомендуемого размера 40x40px.

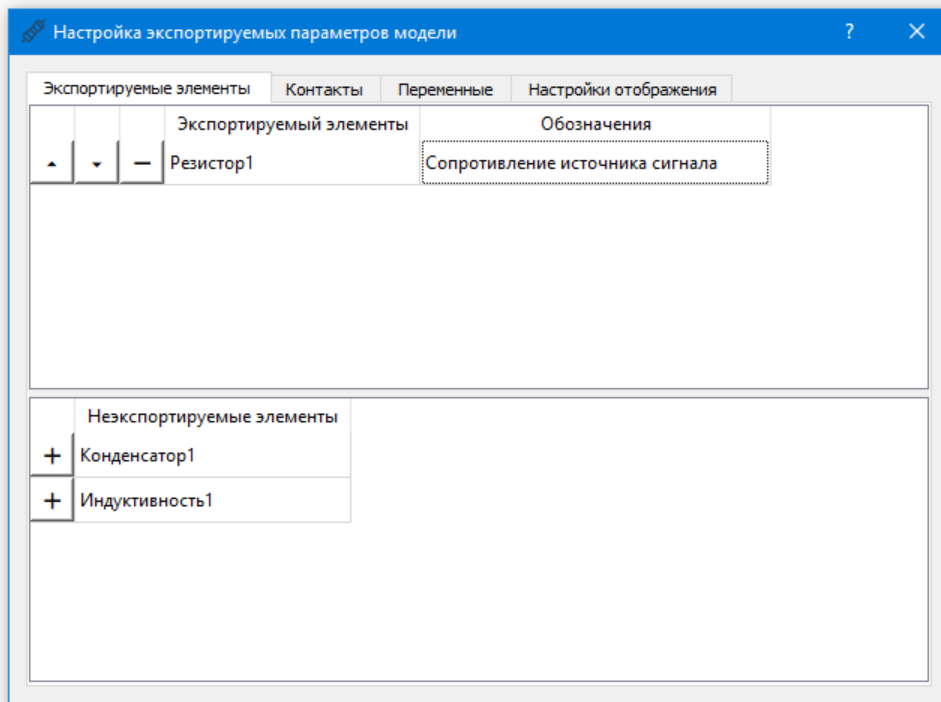


Рисунок 7 – Экспортируемые элементы подсистемы

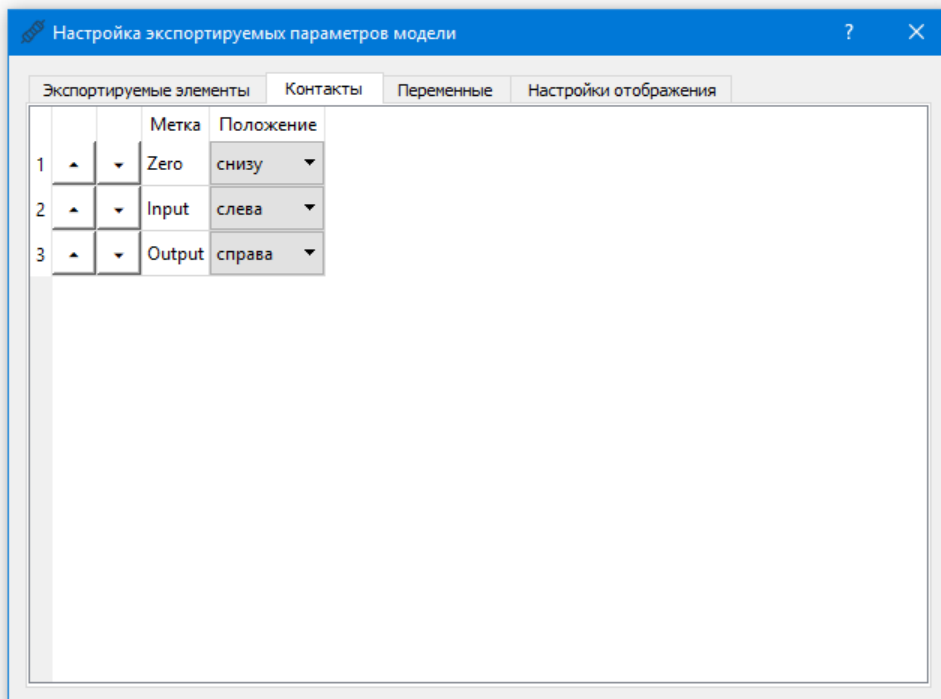


Рисунок 8 – Контакты подсистемы подсистемы

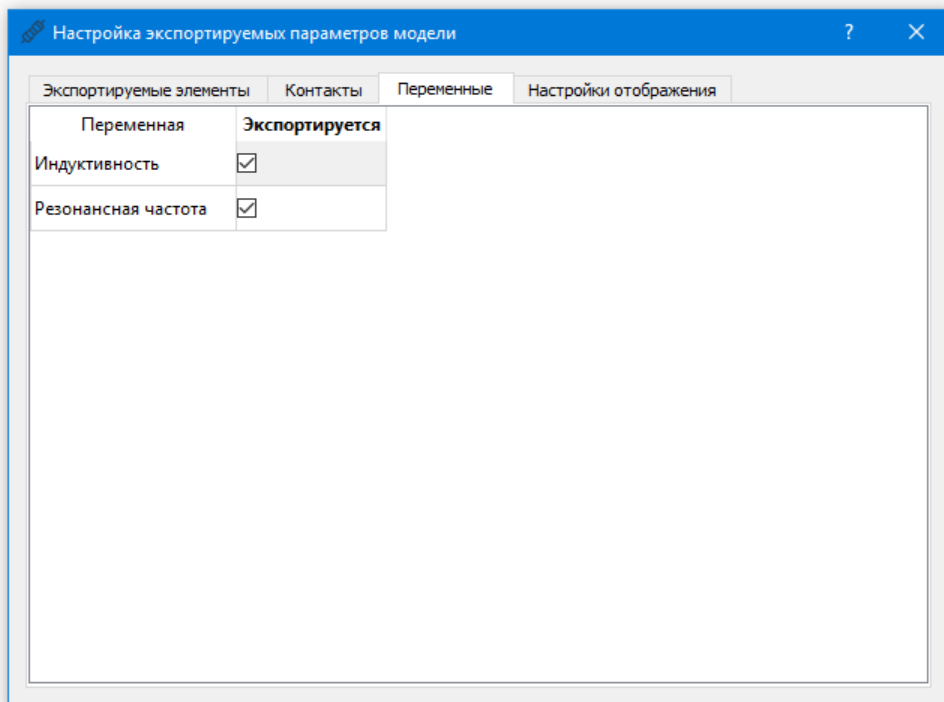


Рисунок 9 – Экспорт переменных окружения

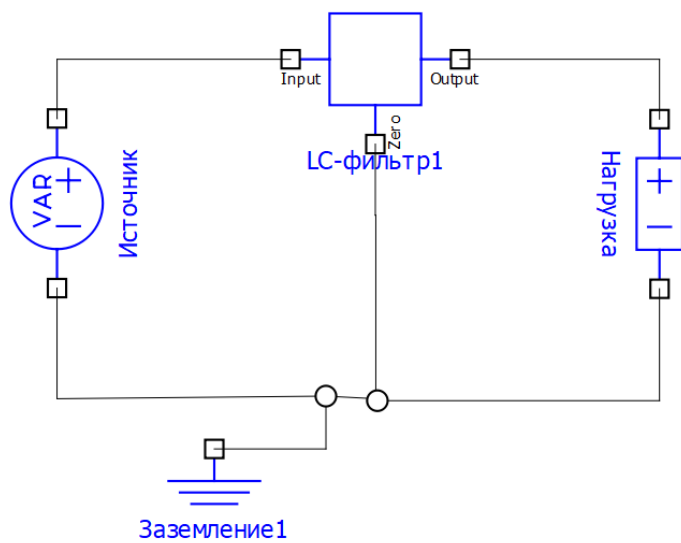


Рисунок 10 – Загруженная подсистема LC-фильтра

4 Программируемые элементы и источники

Коллекцию доступных элементов электрической схемы можно существенно расширить, используя встраиваемую среду Julia. Для определения необходимых программных сущностей используется встроенный редактор кода, вызываемый через меню «Правка - Редактировать исходный код».

Программируемые источники

Элементы «Программируемый источник напряжения» и «Программируемый источник тока» доступны в группе «Источники» панели элементов. В свойствах этих элементов необходимо указать имя функции одной переменной, определенной в текущем окружении Julia (включающем функции стандартной библиотеки и функции, определенные в текущем документе) $f(t)$, t – время, возвращающей число типа *Float64*.

Пример: *создание источника напряжения прямоугольных импульсов с единичной амплитудой и частотой*

```
## Модуль встроенного кода Julia (www.julialang.org)

function Source(t::Float64)
    t = mod(t, 1)
    if t <= 0.5
        return 1.0
    else
        return 0.0
    end
end
```

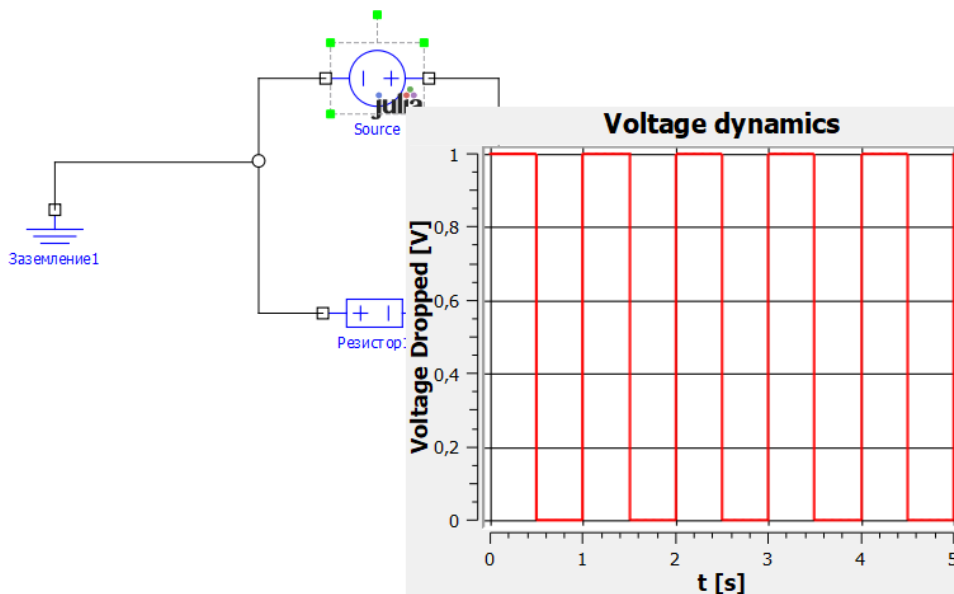


Рисунок 11 – Программируемый источник импульсного напряжения

Программируемые элементы

Для создания электрических элементов, поведение которых описывается средствами среды Julia, используются элементы типа «Программируемый элемент» группы «Основные элементы». Минимальное определение функционального элемента имеет следующий вид:

```
type ElectricElementExample <: ElectricalElement2
    Номинальное_сопротивление::Float64
```

```

Степень::Float64
ElectricElementExample() = new(1.0, 1.0)
end

function (e::ElectricElementExample)(q::Array{Float64})
    U = q[1] - q[2]
    R = (abs(U)^e.Степень + 1) * e.Номинальное_сопротивление
    I = U / R
    return [-I; I]
end

```

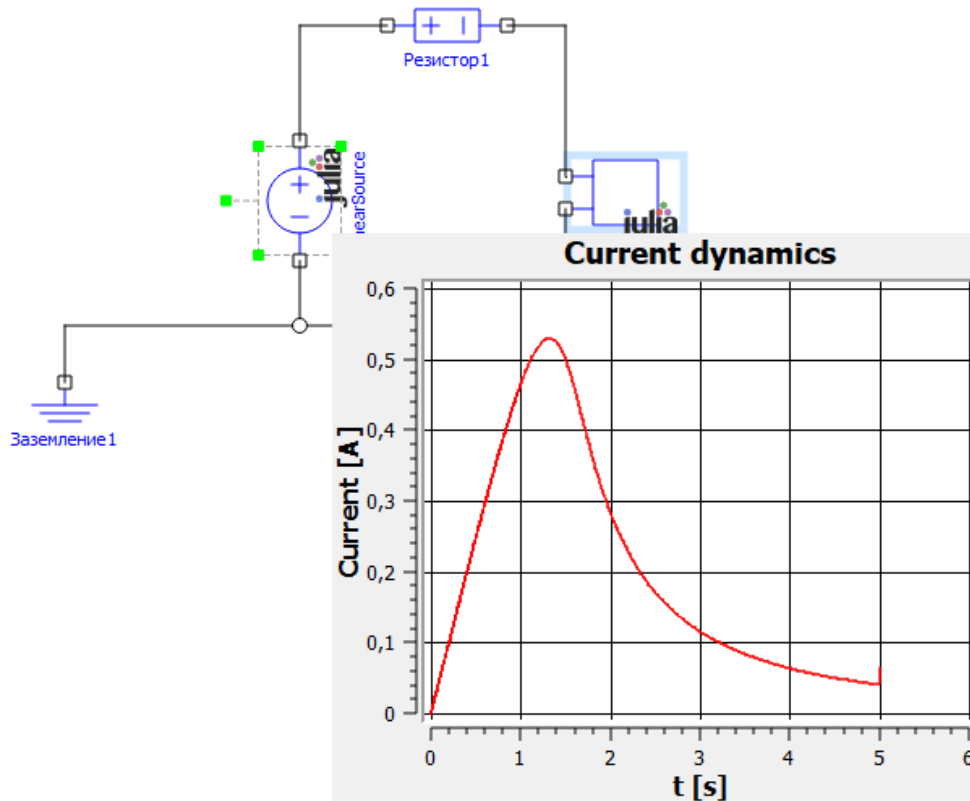


Рисунок 12 – Пример программируемого элемента

В данном примере определен двухпортовый элемент с именем *ElectricElementExample*, имеющий два свойства: *Номинальное_сопротивление* и *Степень*. Программируемый элемент обязательно должен содержать конструктор без аргументов, задающий значения свойств элемента по умолчанию. Вольт-амперная характеристика элемента определяется уравнением $I = \frac{U}{R(U)}$, $R(U) = (|U|^\alpha + 1) \cdot R_0$, где R_0 и α – параметры элемента. Количество портов элемента определяется типом, от которого данный элемент наследуется. Любой программируемый электрический элемент должен наследоваться от абстрактного типа *ElectricalElement*. Данный тип является родительским по отношению к абстрактным типам *ElectricalElement2*, *ElectricalElement3*, ..., *ElectricalElement9*, определяющих двухпортовый, трехпортовый и т.д. электрический элемент. В рассматриваемом примере тип *ElectricElementExample* наследуется от типа *ElectricalElement2* и, следовательно, имеет два порта. Порты элемента нумеруются сверху-вниз. Для определения поведения элемента в форме зависимости токов через его порты от потенциалов узлов цепи, к которым элемент подключен, требуется перегрузить для созданного типа оператор вызова, принимающий в качестве единственного аргумента вектор узловых потенциалов. Результатом оператора должен быть вектор токов через порты элементов. Ток имеет положительное значение при «вытекании» в порт элемента и отрицательное значение при «вытекании». Важно: *размерность вектора токов должна совпадать с размерностью вектора узловых потенциалов. Значения вектора токов должны удовлетворять условию $\sum_{i=1}^N I_i = 0$. Несоблюдение данного условия приведет к ошибочным результатам.*

Использование модулей и пакетов Julia

Программа использует среду Julia, установленную на компьютере пользователя и, следовательно, при редактировании исходного кода модели можно использовать любой из дополнительно загружаемых модулей и пакетов (<https://pkg.julialang.org>). Пользователь также может создавать собственные программные модули, доступные только внутри программы APM ECA. Для этого модули необходимо разместить в каталоге *julia* основной директории программы (по умолчанию при установке программы используется директория %ProgramFiles%/APM ECA).

```
# Созданный пользователем модуль
module MyModule

export something

function something(U)
...
end

end # of module

# Исходный код документа
## Модуль встроенного кода Julia (www.julialang.org)

using MyModule

type MyType <: ElectricalElement2
...
end

...

function (e::MyType)(q::Array{Float64})
    I = something(q[1] - q[2])
    return [I; -I]
end
```

5 Проведение расчетов

Расчеты характеристик электрической цепи доступны через меню «Расчет».

5.1 Статический расчет

Позволяет рассчитать мгновенные значения токов и узловых потенциалов в начальный момент времени. Результаты статического расчета доступны в табличной форме через меню «Результаты – Показать диалог результатов», вкладка «Статический расчет» (рисунок 13).

Результаты статического расчета также можно представить в графическом виде на схеме. Для активации графического отображения результатов необходимо выбрать соответствующий пункт в меню «Статический расчет» на панели инструментов документа. Доступно две опции: «Показывать все токи» и «Показывать все падения напряжений». При активации первого пункта при наведении на связь или на графический элемент отображается направление электрического тока и его абсолютное значение (рисунок 14). При активации второй опции при наведении на двухпортовый графический элемент значение падения напряжений на этом элементе выводится в окно Помощника схемы.

5.2 Гармонический расчет

Проведение гармонического расчета доступно только в случае, когда в электрической цепи присутствуют только линейные электрические элементы и гармонические источники тока и напряжения. Перед выпол-

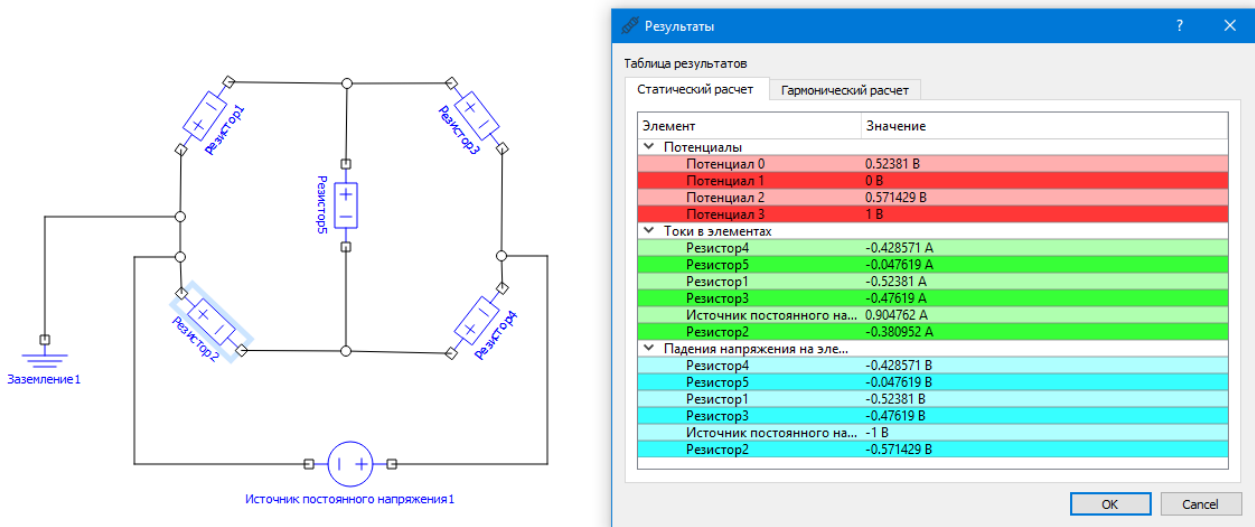


Рисунок 13 – Представление результатов статического расчета в табличной форме

нением расчета пользователю предлагается выбрать диапазон частот, для которых требуется вычислить амплитудные и частотные характеристики.

Результаты гармонического расчета доступны в табличной форме через меню «Результаты — Показать диалог результатов», вкладка «Гармонический расчет» (рисунок 16).

Результаты также могут быть представлены в графическом виде в форме графиков АЧХ и ФЧХ токов и падений напряжений. Для активации графического отображения результатов необходимо выбрать соответствующий пункт в меню «Гармонический расчет» на панели инструментов документа. При активации опции отображения графических результатов при наведении на элемент схемы отображается график с запрашиваемой характеристикой (рисунок 17)

5.3 Динамический расчет

Выполнение динамического расчета доступно через меню «Расчет - Динамический расчет». В диалоговом окне «Параметры динамического расчета» имеются две вкладки: «Основные параметры» (рис. 18a) и «Дополнительные параметры» (рис. 18b).

В основных параметрах предлагается установить время моделирования и величину шага расчета. Параметр «Шаг по времени» определяет расстояние между точками расчетной сетки, в которых значение искомого сигнала должно быть вычислено. В случае необходимости программа автоматически уменьшает величину шага дискретизации для достижения необходимой точности, однако *любая рассчитываемая траектория гарантированно проходит через все точки расчетной сетки*. Включение опции «Использовать результаты статического расчета в качестве начальных условий» приводит к решению задачи статики (нахождения равновесного состояния системы без учета динамических элементов типа дифференциаторов и интеграторов) в момент времени $t = 0$. В случае отключения данной опции в начальный момент времени значения во всех связях модели принимаются равными нулю.

При проведении динамического расчета предлагается выбрать один из трех вариантов сохранения результатов:

- запомнить все результаты. При активации данного режима по завершении расчета будут доступны сигналы во всех связях модели;
- запоминать только результаты на датчиках. В данном режиме будут доступны только результаты в связях, примыкающих к элементам типа «Приемник»;
- не запоминать ничего. Активация данного режима предполагает, что по завершении расчета ни один сигнал не может быть показан на рабочей области. Данную опцию следует использовать только для проверки корректности больших моделей с целью экономии памяти вычислительной машины, либо когда требуются только результаты, выводимые непосредственно в процессе расчета, например посредством элемента «Приемник с выводом в файл».

Во вкладке «Дополнительные параметры расчета» доступны более тонкие настройки процедуры расчета, включающие выбор метода дискретизации дифференциально-интегральных уравнений, метода решения возникающих в процессе моделирования систем нелинейных алгебраических уравнений.

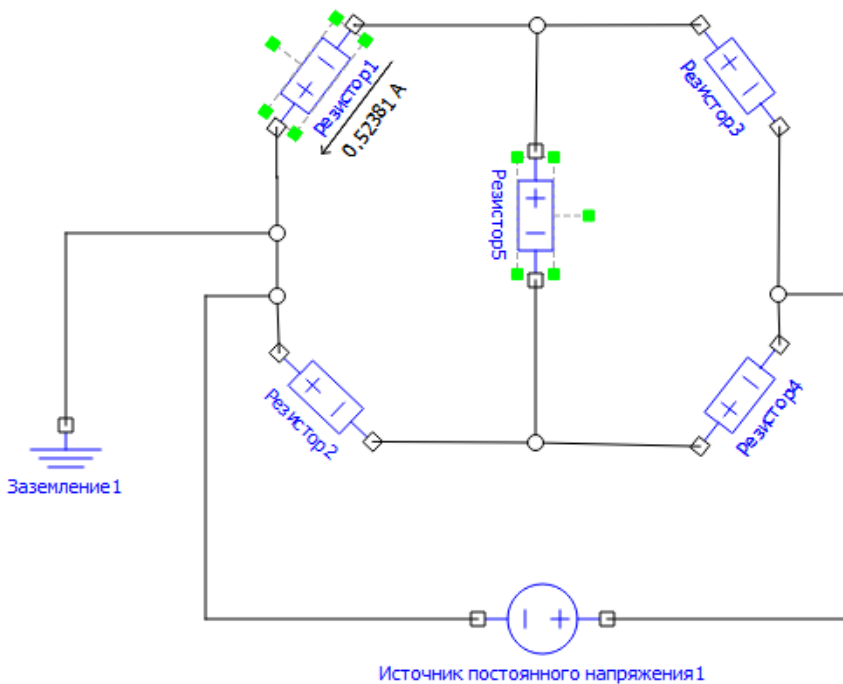


Рисунок 14 – Представление результатов статического расчета графической форме

В программе APM ECA доступны следующие методы дискретизации дифференциально-интегральных операторов:

- неявный метод Эйлера.

Для методов решения систем нелинейных алгебраических уравнений можно указать следующие параметры:

- максимальное количество выполняемых итераций, по достижению которого расчет считается неудачным. Данную величину, как правило, не следует делать слишком большой, поскольку большое количество выполняемых итераций часто свидетельствует о необходимости уменьшить шаг по времени. Исключение составляют модели, в которых наблюдаются резкие скачки или разрывы производных;

- метод аппроксимации матрицы Якоби. Определяет способ приближения неизвестной части матрицы $\frac{\partial G}{\partial X}$ уравнения $G(X) = 0$, описывающего состояние системы в определенный момент времени.

Доступны следующие варианты аппроксимации:

- 2-х точный метод секущих. Является наиболее точной численной аппроксимацией, однако требует наибольшего количества вспомогательных вычислений. Рекомендуется только в случае, когда использование метода аппроксимации, установленного по умолчанию приводит к невозможности получения результата установленной точности.
- конечно-разностная аппроксимация в начальной точке. Наименее затратна в вычислительном плане, однако наименее точна и может приводить к расхождению расчета при большом количестве требуемых итераций;
- метод Бroyдена (первого порядка). Требуется незначительно больше вычислительных ресурсов по сравнению с конечно-разностной аппроксимацией в начальной точке, однако обладает большей скоростью сходимости и устойчивостью. Используется по умолчанию.

- уровень сходимости ϵ . Определяет максимальное значение приращения между итерациями, при котором процесс решения системы нелинейных уравнений может считаться сошедшимся. Решение X_{i+1} считается допустимым приближением истинного решения только при выполнении условия $\|X_{i+1} - X_i\| < \epsilon$. Рекомендуется устанавливать значение данного параметра большим, чем величину минимально допустимой невязки, однако чрезмерно большое значение данного параметра может приводить к появлению ложных скачков в решении в случае наличия широких областей малого изменения функции $G(X)$. При появлении неоправданно резких изменений в решении следует уменьшить величину данного параметра;

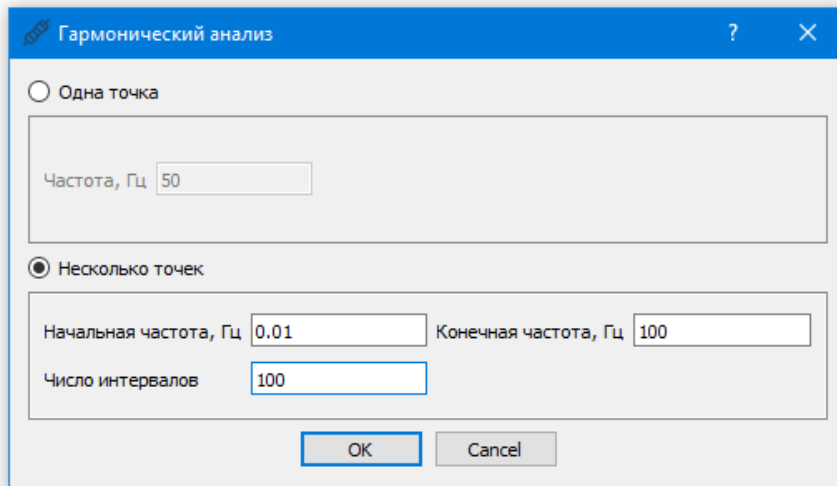


Рисунок 15 – Настройки гармонического расчета

– минимальная невязка δ . Определяет наибольшее допустимое отклонение функции $G(X)$ от нуля. Иначе, X_i принимается в качестве приближенного решения уравнения $G(X) = 0$ только если $G(X_i) < \delta$. Значение данной величины не должна быть меньше погрешности вычисления $G(X)$, что следует учитывать при использовании программируемых элементов. Не рекомендуется устанавливать значение минимальной невязки меньше $1e - 9$.

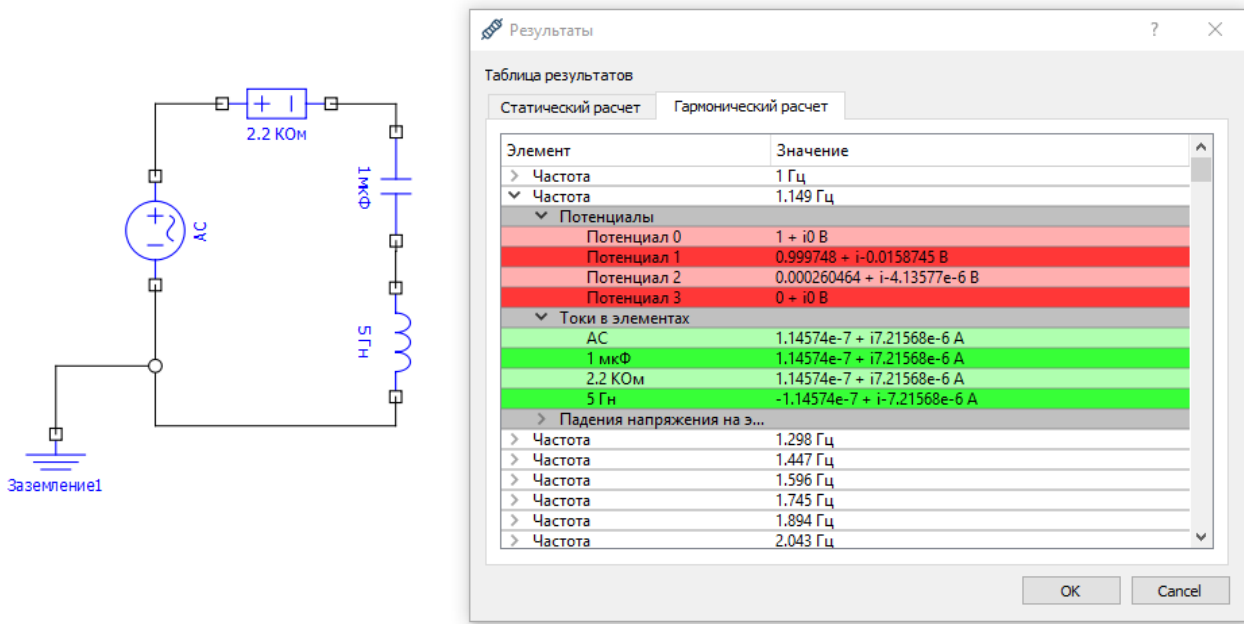


Рисунок 16 – Табличное представление результатов гармонического расчета

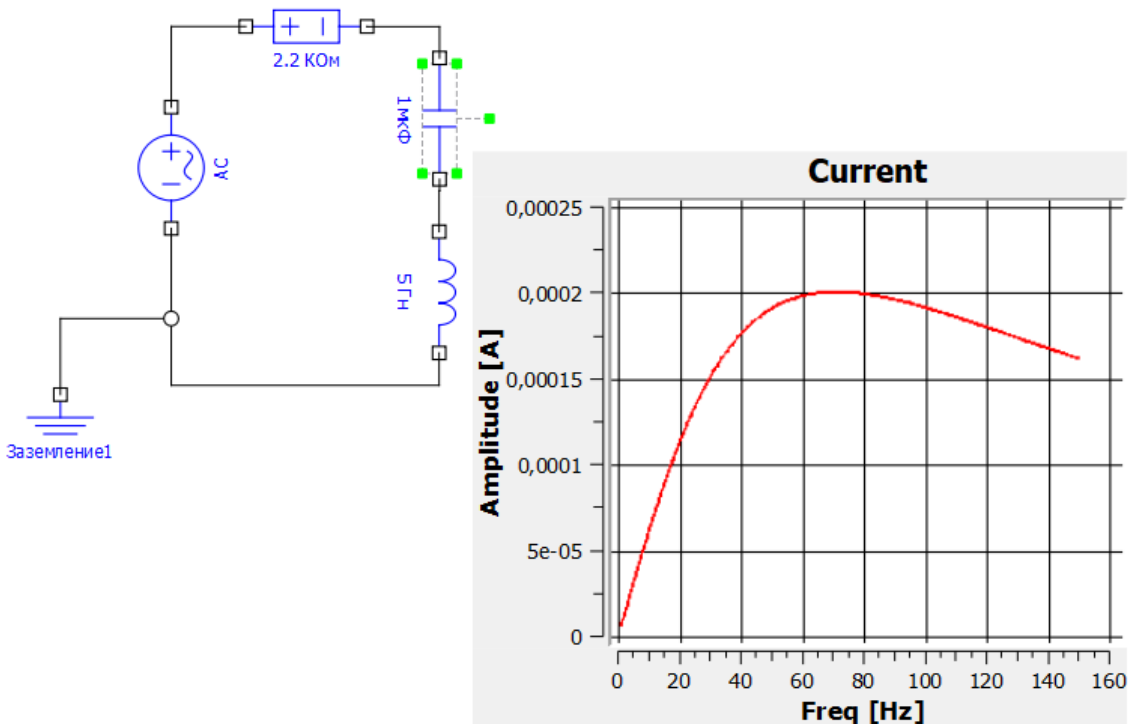
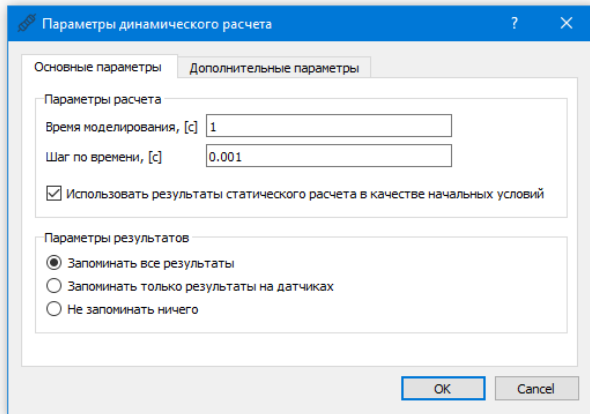
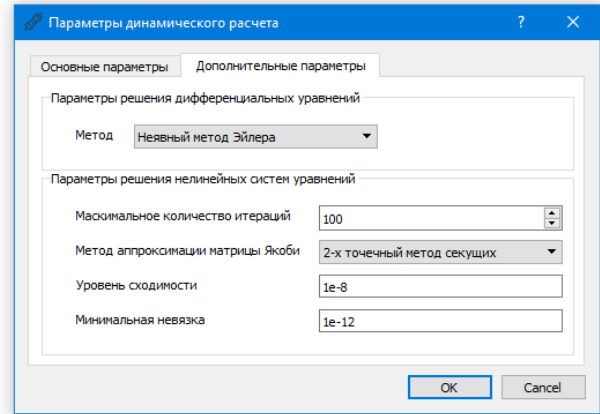


Рисунок 17 – Графическое представление АЧХ тока через элемент схемы



(a) основные параметры



(b) дополнительные параметры

Рисунок 18 – Параметры динамического расчета имитационных моделей

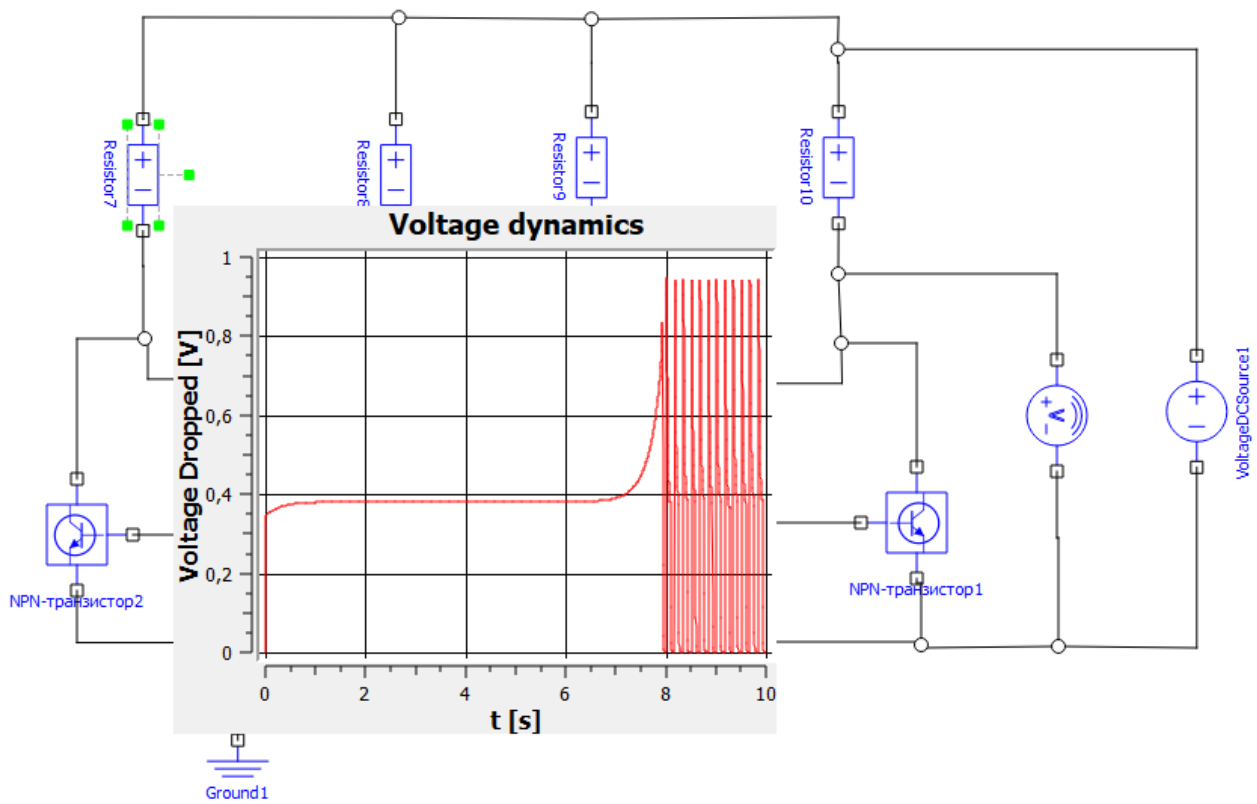


Рисунок 19 – Результаты динамического расчета мультивибратора