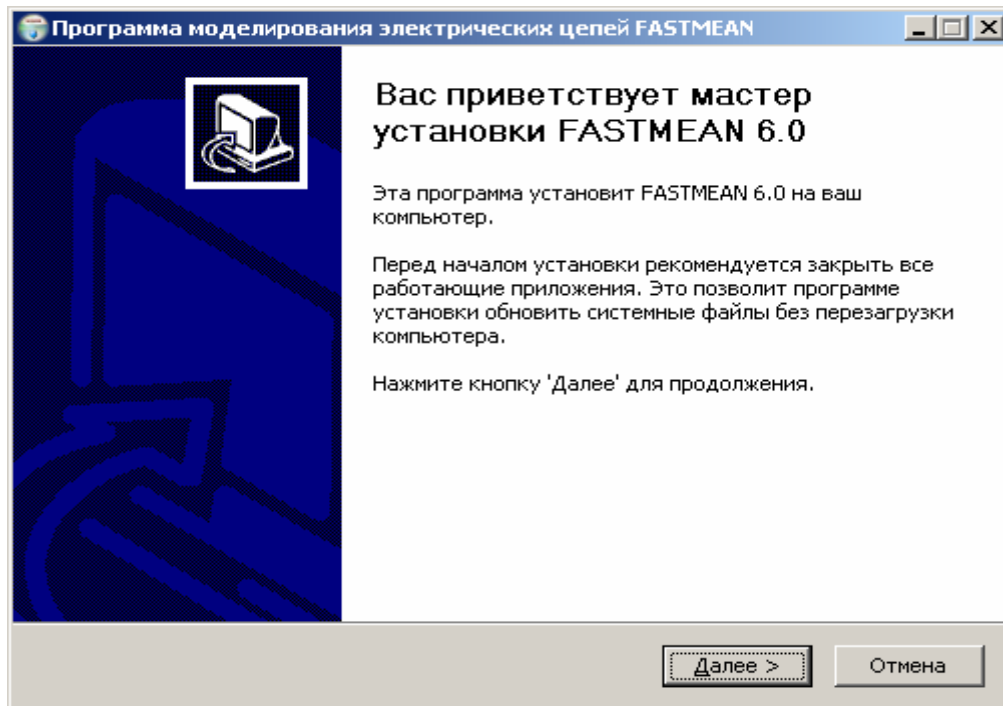


<b>1. УСТАНОВКА</b> .....	<b>2</b>
<b>2. СОВМЕСТИМОСТЬ</b> .....	<b>3</b>
<b>3. ОБЩИЕ ПРИНЦИПЫ РАБОТЫ С ПРОГРАММОЙ</b> .....	<b>4</b>
3.1. РЕДАКТОР СХЕМЫ .....	4
3.2. ЭЛЕМЕНТ «ПОРТ» .....	5
3.3. ИМЕНА ЭЛЕМЕНТОВ (ИДЕНТИФИКАТОРЫ).....	6
3.4. ВЫРАЖЕНИЯ В ПАРАМЕТРАХ ЭЛЕМЕНТОВ.....	7
3.5. МАТЕМАТИЧЕСКИЕ ВЫРАЖЕНИЯ В FASTMEAN.....	7
3.6. ТРЕБОВАНИЯ И РЕКОМЕНДАЦИИ К АНАЛИЗИРУЕМОЙ СХЕМЕ.....	11
<b>4. ИЗМЕНЕНИЯ ВЕРСИИ 6.0</b> .....	<b>16</b>
4.1. РАБОТА С ПОДСХЕМАМИ (НОВОЕ!).....	16
4.2. ОКНО НАСТРОЕК ПЕРЕХОДНОГО ПРОЦЕССА.....	19
4.3. ОКНО НАСТРОЕК ЧАСТОТНЫХ ХАРАКТЕРИСТИК.....	20
4.4. ОКНО ГРАФИКОВ.....	21
4.5. СИМВОЛЬНЫЙ АНАЛИЗ.....	22
4.6. ТЕКСТОВЫЙ ФОРМАТ ФАЙЛА СХЕМЫ (YAML).....	24
<b>5. ДИРЕКТИВЫ</b> .....	<b>27</b>
5.1. .DEFINE, .DEF .....	27
5.2. .PARAMETER, .PARAM.....	29
5.3. .GLOBAL.....	29
5.4. .SET .....	30
5.5. .WRITE.....	31
5.6. .WRITE AFTER.....	31
<b>6. АВТОМАТИЗАЦИЯ РАСЧЕТОВ</b> .....	<b>32</b>
6.1. ЗАПУСК РАСЧЕТА ИЗ КОМАНДНОЙ СТРОКИ .....	32
<b>7. РАСЧЕТ ЧАСТОТНЫХ ХАРАКТЕРИСТИК (ЧХ) ЧЕРЕЗ МОДЕЛИРОВАНИЕ ПРОЦЕССОВ</b> .....	<b>35</b>
7.1. ЧТО ЭТО ТАКОЕ.....	35
7.2. ЗАЧЕМ НУЖЕН ЭТОТ АНАЛИЗ .....	36
7.3. КАК ВЫПОЛНИТЬ ЭТОТ АНАЛИЗ В FASTMEAN .....	37
7.4. РЕКОМЕНДАЦИИ ПО НАСТРОЙКАМ .....	40
7.5. ФУНКЦИИ ДЛЯ РАСЧЕТА ЧХ ЧЕРЕЗ МОДЕЛИРОВАНИЕ ПРОЦЕССОВ.....	43

## 1. Установка

Начиная с версии 6.0, используется единый установочный файл для полной версии, DEMO версии и модуля символьного анализа.

Необходимо скачать установочный файл FASTMEAN\_6\_0.exe и запустить.



Начиная с версии 6.0, установка выполняется для текущего пользователя. Не требуются права Администратора системы.

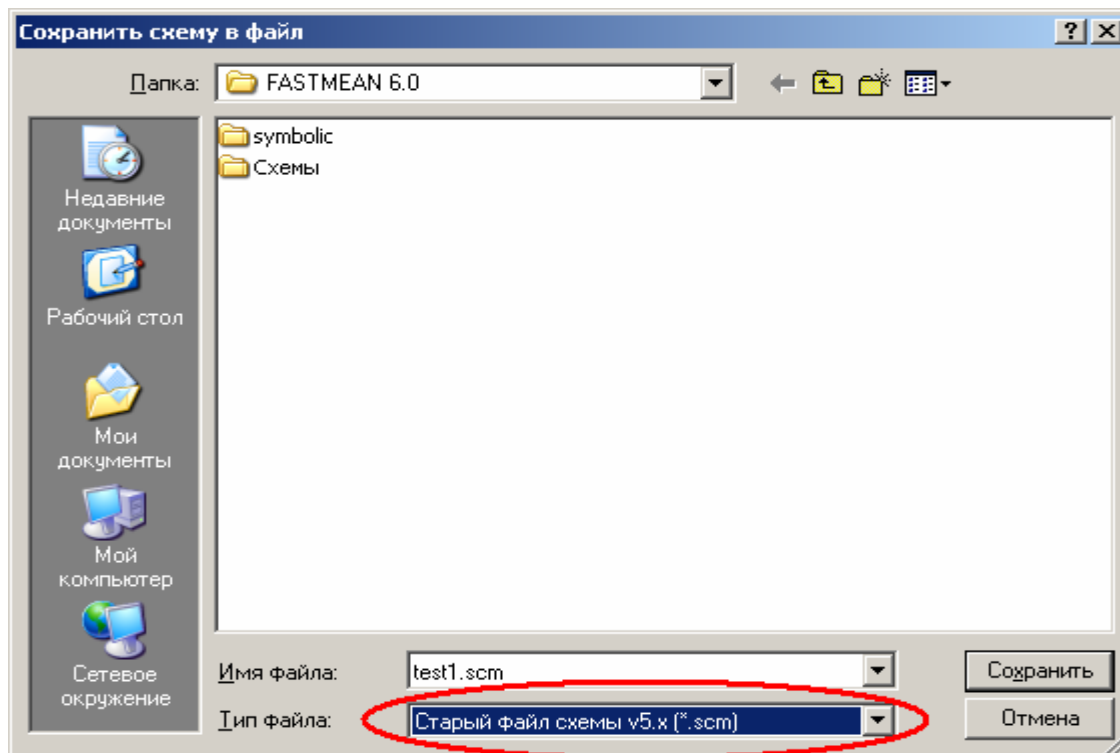
При первом запуске программа работает в режиме ограниченной функциональности (DEMO). Для перехода в режим полной функциональности необходимо указать файл лицензии и секретный ключ.

Файл лицензии (ключевой файл) и секретный ключ (серийный номер) можно взять от ранее приобретенной полной версии FASTMEAN 5.x, либо приобрести для версии 6.0.

## 2. Совместимость

Версия 6.0 открывает все схемы предыдущих версий FASTMEAN.

В версии 6.0 реализован режим частичной обратной совместимости. В тех случаях, когда схема не содержит новых элементов, она может быть сохранена в старом формате для версии 5.2.

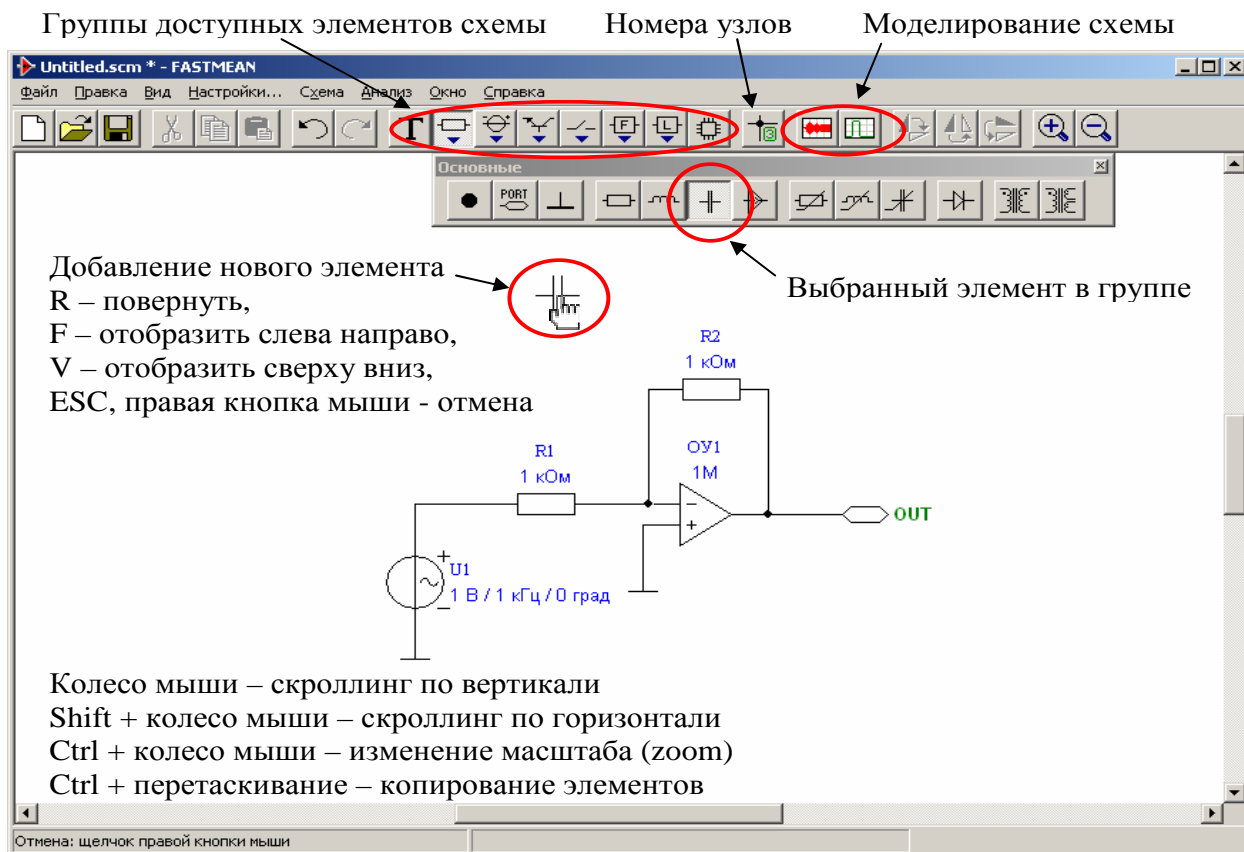


Схемы в новом формате версии 6.0 не могут быть открыты в FASTMEAN более ранних версий.

### 3. Общие принципы работы с программой

#### 3.1. Редактор схемы

Главное окно программы – редактор исследуемой схемы. Схема собирается из доступных элементов, затем выполняется ее моделирование. Внешний вид окна редактора схемы показан на рисунке:

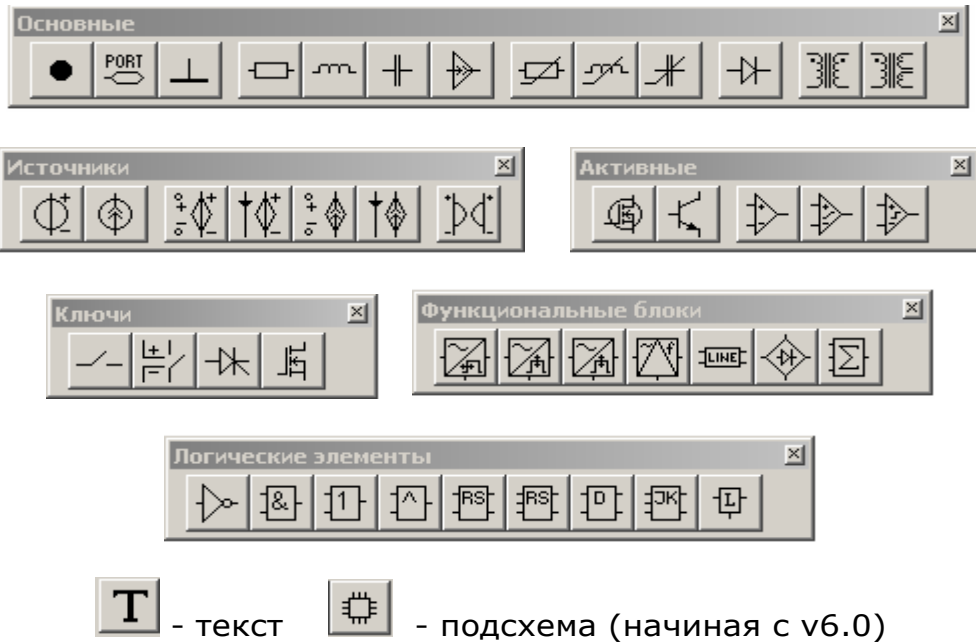


Для соединения выводов двух элементов проводом необходимо подвести мышь к выводу одного элемента (подсвечивается жирной точкой), нажать левую кнопку мыши, подвести мышь к выводу другого элемента (или к другому проводу) и отпустить кнопку мыши.

Для удаления провода необходимо подвести мышь к одному из концов провода (подсвечивается жирной точкой), нажать левую кнопку мыши, отвести провод на свободное место схемы и отпустить кнопку мыши.

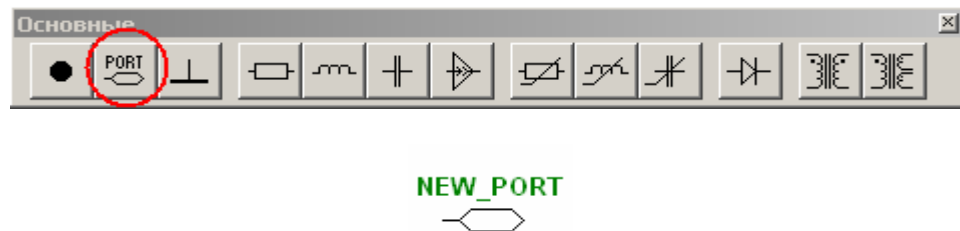
Для изменения параметров элемента необходимо сделать над ним двойной щелчок мышью – появится окно параметров.

## Доступные элементы схемы



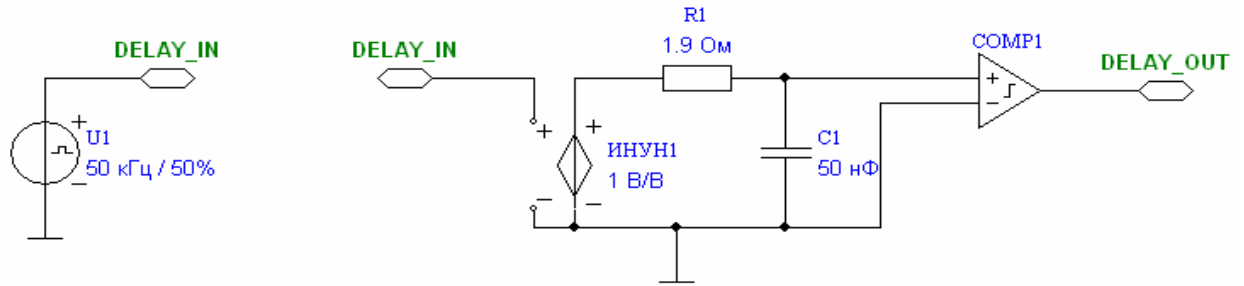
### 3.2. Элемент «Порт»

Элемент «Порт» позволяет соединять отдельные части схемы без протягивания проводов. Для этого различным элементам «Порт» нужно присвоить одинаковое имя.



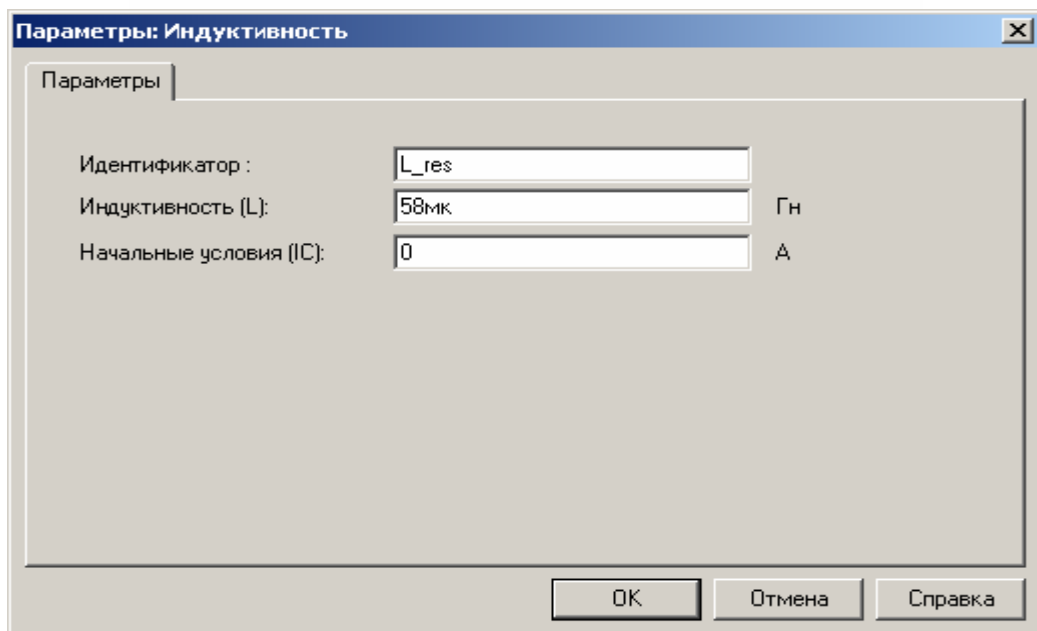
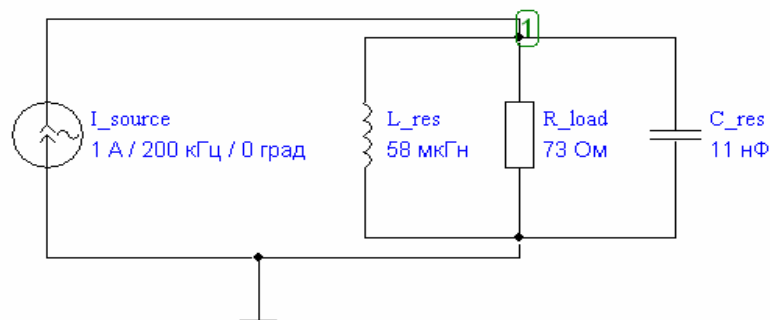
Этот элемент также удобно использовать для именования узлов и контрольных точек в схеме. В отличие от номеров узлов, которые могут меняться в FASTMEAN, имя порта не меняется и придает осмысленность выражениям в расчетах. Например: U(CLOCK\_OUT).

Пример:



### 3.3. Имена элементов (идентификаторы)

Имена элементов можно менять. Это удобно для обозначения роли элемента в схеме. Пример:

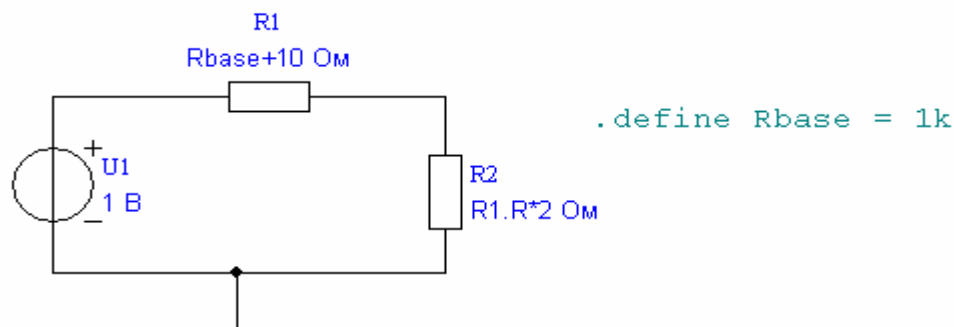


Начиная с версии 6.0, **введено ограничение** допустимых символов имени элемента (идентификатора): имя может содержать латинские и русские буквы, цифры, символ ``_'`. Другие символы в имени не допускаются. Имя не может начинаться с цифры. *Примеры:* PWM3, РусскиеБуквы2, \_Test\_1,

Конфликты имен устраняются автоматически. Если после задания нового имени оно не было принято или было присвоено другое имя, значит, был устранен конфликт с другим уже существующим в схеме именем.

### 3.4. Выражения в параметрах элементов

Параметры элементов, такие как сопротивление, коэффициент усиления и др. могут быть как числами, так и математическими выражениями, содержащими параметры других элементов схемы или функции, определенные пользователем в директиве `.define`. Пример:



Таким образом, возможна взаимосвязанная настройка группы элементов схемы путем изменения одного параметра. Можно вводить параметры в желаемой нотации, например  $10e-9$ ,  $10n$  или  $0.01u$  по выбору пользователя.

Данная возможность реализована для всех элементов, кроме логических (И, ИЛИ, RS-, D-триггер и т.д.). Подстановка значений и вычисление параметров элементов происходит один раз в начале каждого расчета. В процессе расчета параметры элементов не меняются. Выражения для параметров элементов **не могут** содержать переменные, зависящие от времени (токи, напряжения, время  $t$  и т.д.).

### 3.5. Математические выражения в FASTMEAN

Математические выражения в FASTMEAN используются в директивах пользователя (`.define`, `.set`, `.global` и др.), в параметрах элементов, в окнах расчета переходного процесса и частотных характеристик.

Математическое выражение может содержать **операции**:

+	сложение
-	вычитание
*	умножение
/	деление
^	возведение в степень
()	скобки для задания приоритета операций

Математическое выражение может содержать константы 'e' (2.71828182845905) и 'PI' (3.14159265358979). При расчете частотных характеристик можно вводить **комплексные числа**. Для обозначения мнимой части используется 'i' или 'j' (примеры: 17.1i, i3, 14.2j, j7.5)

Математическое выражение может содержать **числа**. Для разделения целой и дробной части используется '.' (точка). Для сдвига десятичной точки число записывается в научной нотации (пример: 1.2e-3) или используются постфиксы инженерной нотации (пример: 5.7n, 470k, 120M):

Показатель степени N (10^N)	Постфикс (латинский регистр)	Постфикс (кириллица)
-12	p	п
-9	n	н
-6	u	мк
-3	m	м
3	k	к
6	M	М
9	G	Г

Математическое выражение может содержать переменные 't' (время) и 'f' (частота).

Математическое выражение может содержать переменные вида: **<имя\_элемента>.<параметр>**. Начиная с версии 6.0, **нельзя** использовать параметр по умолчанию. Т.е. нельзя указать только <имя\_элемента>, необходимо писать явно R2.R вместо R2, L4.L вместо L4, I3.I0 вместо I3 и т.д.

Имена элементов и имена переменных вводятся с учетом регистра (т.е. R3 и r3 - это разные имена).

Математическое выражение может содержать **функции**:

▪ **напряжения, токи и др.**

U(<имя\_элемента>) - напряжение на элементе

U(LL<i>, <расстояние>) - напряжение в сечении длинной линии

I(<имя\_элемента>) - ток через элемент

P(<имя\_элемента>) - мгновенная мощность на элементе

W(LC) - суммарная энергия, запасенная в реактивных элементах

W(L) - суммарная энергия, запасенная в индуктивностях

W(C) - суммарная энергия, запасенная в емкостях

W(L<i>), W(C<i>) - энергия, запасенная в i-ом реактивном элементе

W(NLL<i>), W(NLC<i>) - энергия, запасенная в i-ом нелинейном реактивном элементе



*Примечания.*

Можно рассчитать токи и напряжения на источниках напряжения, источниках тока, резисторах, нелинейных резисторах, емкостях, нелинейных емкостях, индуктивностях, нелинейных индуктивностях, диодах, тиристорах, ключах; узловые напряжения.

Мощность можно рассчитать только для тех элементов, для которых можно рассчитать и ток, и напряжение.

Функция  $W(\dots)$  допустима только при расчете переходного процесса.

**тригонометрические (аргумент в радианах)**

$\cos(z)$  - косинус

$\sin(z)$  - синус

$\tan(z)$  - тангенс

$\cot(z)$  - котангенс

▪ **обратные тригонометрические (результат в радианах)**

$\text{acos}(x)$  - арккосинус

$\text{asin}(x)$  - арксинус

$\text{atan}(x)$  - арктангенс

▪ **гиперболические (аргумент в радианах)**

$\cosh(z)$  - гиперболический косинус

$\sinh(z)$  - гиперболический синус

$\tanh(z)$  - гиперболический тангенс

$\coth(z)$  - гиперболический котангенс

▪ **логарифмические**

$\ln(z)$  - натуральный логарифм (по основанию 'e')

$\lg(z)$  - десятичный логарифм (по основанию 10)

▪ **для работы с комплексными числами**

$\text{dB}(z)$  -  $20 \cdot \lg(|z|)$

$\text{RE}(z)$  - действительная часть  $z$

$\text{IM}(z)$  - мнимая часть  $z$

$\text{MAG}(z)$  - модуль  $z$

$\text{PH}(z)$  - аргумент  $z$  (результат в градусах)

$\text{PHS}(z)$  - аргумент  $z$  (результат в градусах)

*Примечания.*

Функция  $\text{PHS}(z)$  учитывает предыдущие значения функции и устраняет скачки фазы в 360 градусов. Ее предпочтительно использовать для расчета фазо-частотных характеристик (ФЧХ).

▪ **разные**

$\text{exp}(z)$  - 'e' в степени  $z$ , аналогично  $e^z$

$\text{abs}(z)$  - модуль  $z$ , аналогично  $\text{MAG}(z)$

$\text{sqrt}(z)$  - корень квадратный из  $z$ , аналогично  $z^{0.5}$

$\text{pow}(z, v)$  -  $z$  в степени  $v$ , аналогично  $z^v$

$\text{limit}(x, \text{min}, \text{max})$  - ограничение величины  $x$  в пределах  $\text{min} \dots \text{max}$

max(x) - максимальное значение x с начала анализа  
min(x) - минимальное значение x с начала анализа  
floor(x) - отсечение дробной части x в меньшую сторону,  
т.е. floor(5.5)=5, floor(-5.5)=-6  
round(x) - округление x (round(5.5)=6, round (-5.5)=-5)  
mod(x, y) - остаток от деления x на y (y<>0)  
wnd(type, x, start, end) - наложение окна (для x от start до end)  
type: 1 - прямоугольное  
2 - треугольное  
3 - Хеннинга  
4 - Хемминга

▪ **для получения сложных зависимостей**

last(<выражение>) - возвращает последнее значение выражения (в конечный момент времени при расчете переходных процессов).

*Примечание.*

С помощью этой функции и изменения параметра можно строить графики сложных зависимостей.

Пример: необходимо получить на графике зависимость средней мощности, потребляемой от источника U2, от сопротивления резистора R3. Выполняется расчет переходного процесса с изменением параметра R3.R. В окне переходного процесса по оси X указывается R3.R, по оси Y указывается last( avg( P(U2), 1/U2.f ) ). На графике выводится зависимость средней потребляемой мощности P(U2) за период, взятой в конце расчетного интервала, от сопротивления резистора R3.

▪ **усреднение**

AVG(<выражение>[, интервал усреднения]) - (от англ. average) среднее значение выражения за последние несколько секунд или герц. Если интервал усреднения не задан, то усреднение проводится с начала анализа.

RMS(<выражение>[, интервал усреднения]) - (от англ. root-mean-square) действующее значение выражения за последние несколько секунд или герц (буквально: корень квадратный из среднего значения квадрата выражения). Если интервал усреднения не задан, то усреднение проводится с начала анализа.

▪ **преобразования**

FFT(<выражение>) - (англ. Forward Fourier Transform) прямое преобразование Фурье. Интервал времени, на котором выполняется FFT, задается в отдельном окне.

▪ **Функции для расчета частотных характеристик через моделирование процессов во временной области:**

tmag(<выражение сигнала>, <частота>, <нач.время>, <кон.время>)  
tph(<выражение сигнала>, <частота>, <нач.время>, <кон.время>)  
tre(<выражение сигнала>, <частота>, <нач.время>, <кон.время>)  
tim(<выражение сигнала>, <частота>, <нач.время>, <кон.время>)

### Примечания.

Функции комплексного аргумента  $z$  возвращают комплексный результат. Функции действительного аргумента  $x$  возвращают действительный результат. Если такой функции передать комплексное значение, то функция будет вычислена для действительной части аргумента. Если функция возвращает комплексный результат, необходимо указывать, что именно выводится на график (MAG, PH, RE или IM). Если этого не сделать, то будет выведена действительная часть результата (RE).

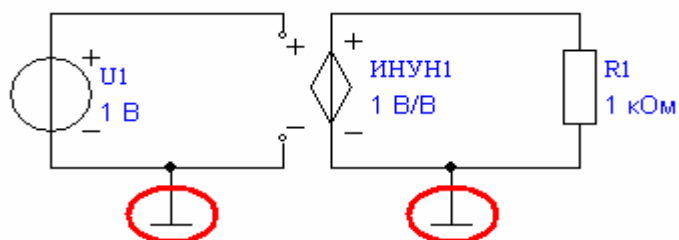
Функция FFT возвращает комплексные составляющие спектра (гармоники) периодического сигнала. MAG(FFT()) дает спектр амплитуд, PH(FFT()) дает спектр фаз.

Имена функций можно вводить без учета регистра (т.е. AVG и avg – одно и то же).

### 3.6. Требования и рекомендации к анализируемой схеме

Для успешного моделирования электрической схемы она должна удовлетворять ряду требований. Это необходимо, чтобы математическая модель (система уравнений), соответствующая схеме, была корректна и разрешима.

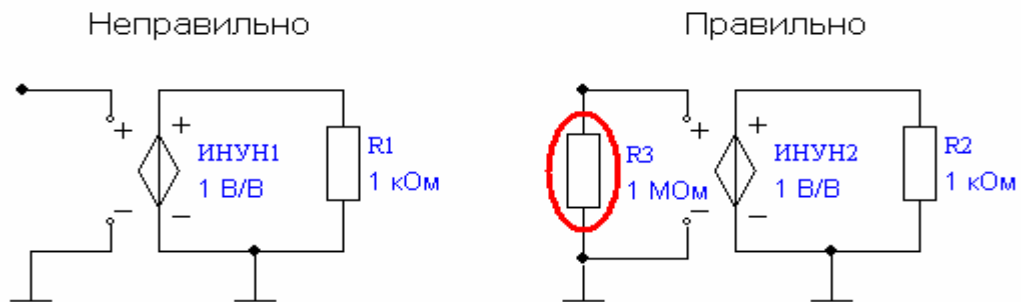
1. В схеме должен быть хотя бы один элемент **«Земля»**, обозначающий узел с нулевым потенциалом (напряжением). При наличии в схеме элементов с гальванической изоляцией (ИНУН, ИТУН, трансформатор и др.) в каждом гальванически изолированном «островке» схемы должен быть хотя бы один элемент «Земля».



2. Рекомендуется избегать неподключенных выводов элементов. Для них используется элемент «Соединитель».

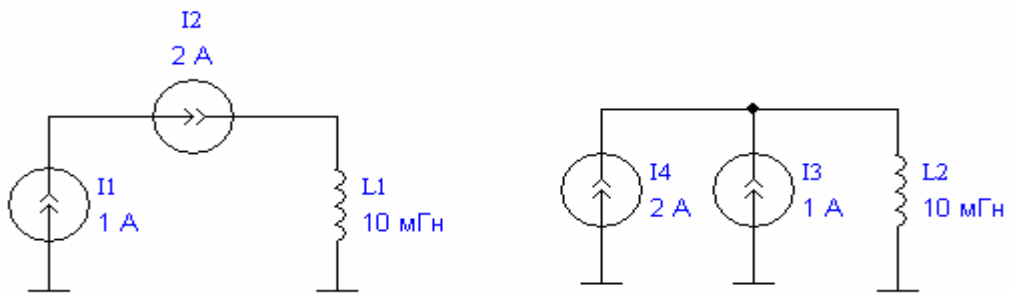


3. Управляющие входы зависимых источников (ИНУН, ИТУН и др.) должны быть соединены с одной из частей схемы. Если необходимо оставить «плавающий» вход, он подключается к «Земле» через большой резистор:

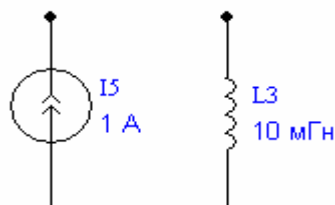


4. Математически **некорректными** являются:

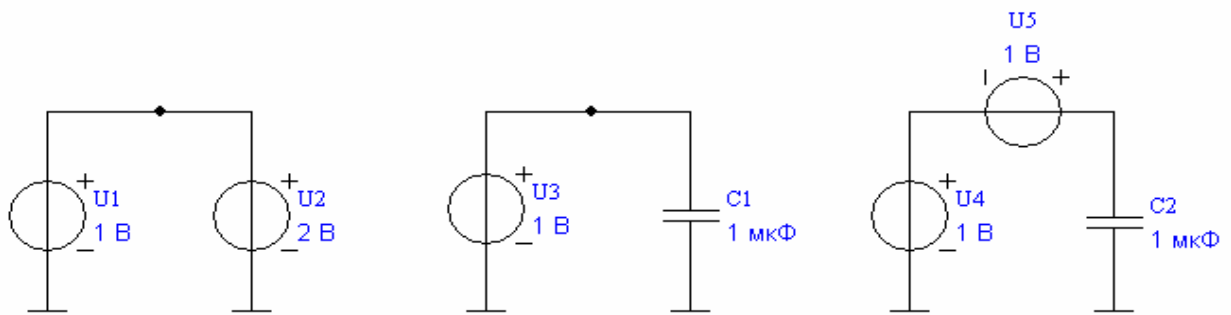
последовательное соединение источников тока между собой и с индуктивностью, либо, в общем случае, образование сечения из этих элементов:



неподключенный источник тока или индуктивность (току некуда течь):



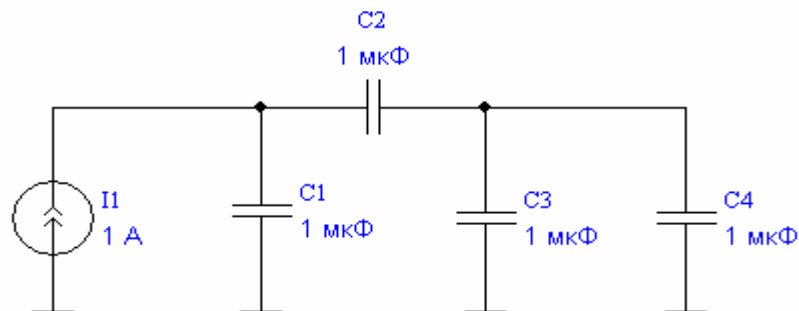
параллельное соединение источников напряжения между собой и с конденсатором, либо, в общем случае, образование контура из этих элементов:



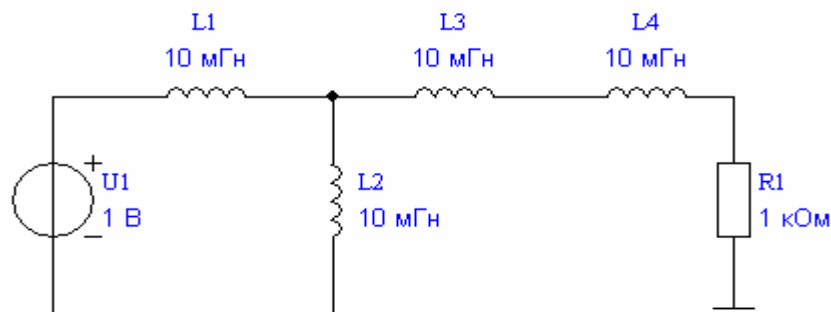
И другие аналогичные схемы, включающие ИТУН, ИНУН и т.д.

5. Математически **корректными** являются:

схемы с емкостными контурами (параллельными конденсаторами):

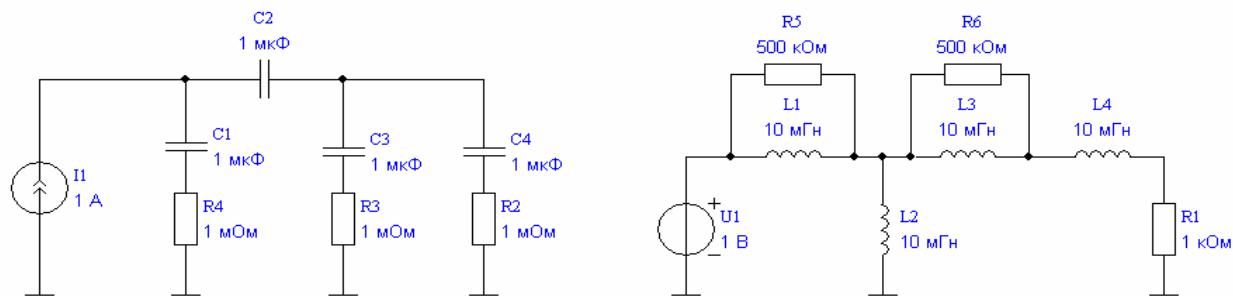


схемы с индуктивными сечениями (последовательными индуктивностями):



Расчет таких схем в FASTMEAN требует больше вычислительных затрат, чем расчет схем без особенностей. Для ускорения расчета **рекомендуется** устранить особый случай путем подключения небольшого резистора последовательно с емкостью, и большого резистора параллельно с индуктивностью.

Модификация схем для устранения математических особенностей:



## 6. Направления отсчета токов и напряжений

В программе FASTMEAN, когда используется выражение вида  $U(R1)$ ,  $I(L3)$  и т.п., подразумевается некоторое положительное направление отсчета. Это направление определяется номерами узлов, к которым подключен элемент. Правила выбора положительного направления отсчета такие:

### **Напряжение**

Для всех элементов за положительное направление отсчета напряжения принимается направление от узла с большим номером к узлу с меньшим номером.

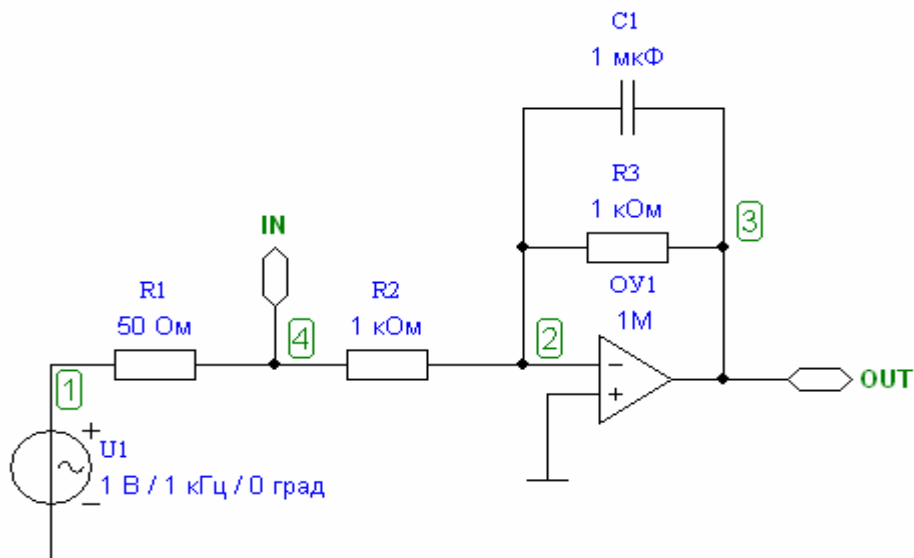
### **Ток**

Для резистора, нелинейного резистора и ключей за положительное направление отсчета тока принимается направление от узла с большим номером к узлу с меньшим номером, т.е. для этих элементов направления отсчета напряжений и токов выбраны согласно.

Для остальных элементов за положительное направление отсчета тока принимается направление от узла с меньшим номером к узлу с большим номером, т.е. направления отсчета напряжений и токов выбраны встречно.

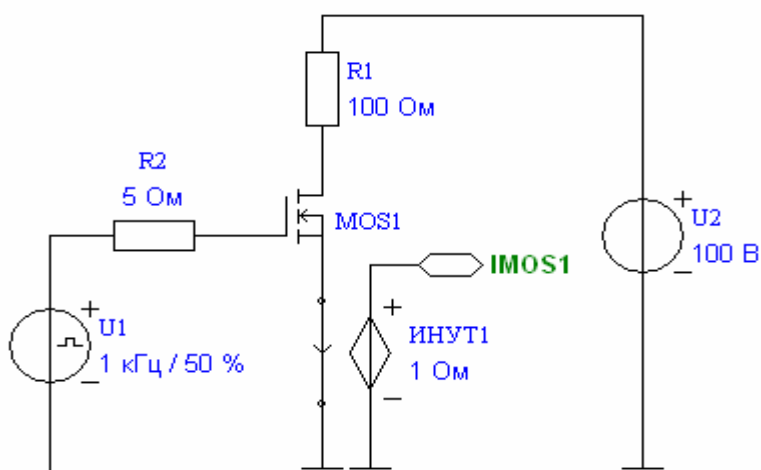
Эти правила существуют в программе по историческим причинам. Существенное **неудобство** возникает из-за того, что изменение проводов в схеме (удаление, подключение) может приводить к изменению номеров узлов и, как следствие, к изменению направлений отсчета токов, напряжений и номеров контрольных точек в схеме.

Чтобы обойти это неудобство **рекомендуется** именовать узлы явно с помощью элемента «Порт»:



Пример: используется U(OUT) вместо U(3), U(IN) вместо U(4)

Также **рекомендуется** направление тока указывать явно с помощью измерительного ИНУТ:



Пример: используется U(IMOS1) вместо I(MOS1) или I(R1).

Такой подход позволяет избежать недоразумений, связанных с выбором положительного направления токов и напряжений в программе FASTMEAN по номерам узлов.

Такой подход рекомендуется для лучшей совместимости файлов схем с будущими версиями программы.

## 4. Изменения версии 6.0

### 4.1. Работа с подсхемами (новое!)

Подсхемы – это обычные схемы FASTMEAN, которые могут быть использованы как составные части в других схемах.

Подсхема создается по определенным правилам:

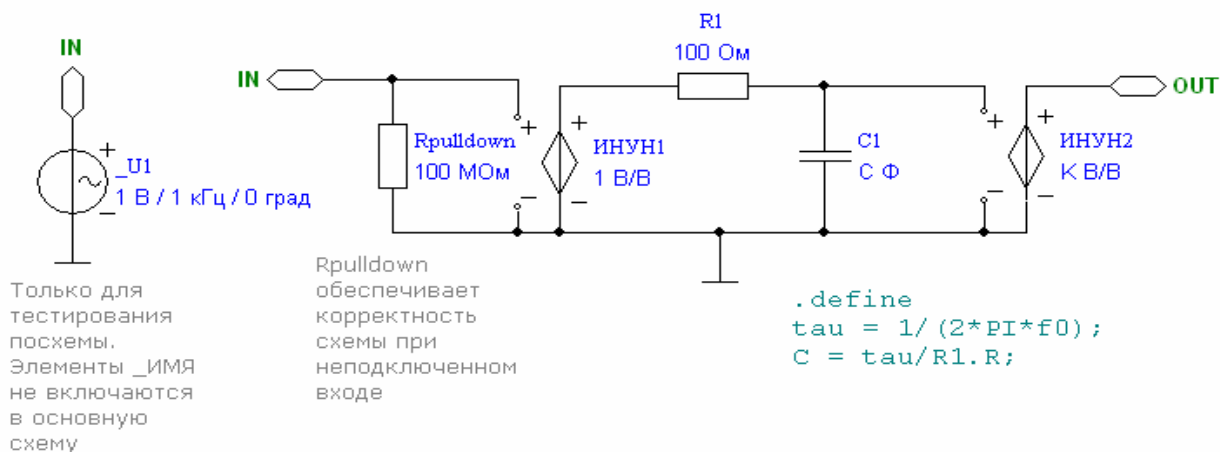
1. Внешние выводы обозначаются с помощью элемента «Порт». Направление порта определяет направление вывода. Порты, имя которых начинается с символа `'\_', не являются внешними.
2. Параметры подсхемы описываются с помощью директивы `.parameter`. Они будут доступны в основной схеме через окно параметров. Подробнее о `.parameter` см. п.5.2.
3. Вспомогательные элементы для тестирования подсхемы должны иметь имя, которое начинается с символа `'\_'. Такие элементы не включаются в основную схему.

Пример подсхемы:

#### Фильтр нижних частот (ФНЧ) 1-го порядка

```
.parameter  
f0 = 1k "Частота среза [Гц]";  
K = 1.0 "Плоское усиление";
```

Эти параметры будут доступны в основной схеме через окно параметров

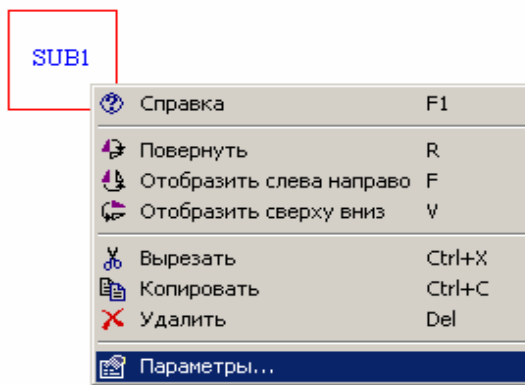


Для использования подсхемы в основной схеме добавляется элемент «Посхема»:

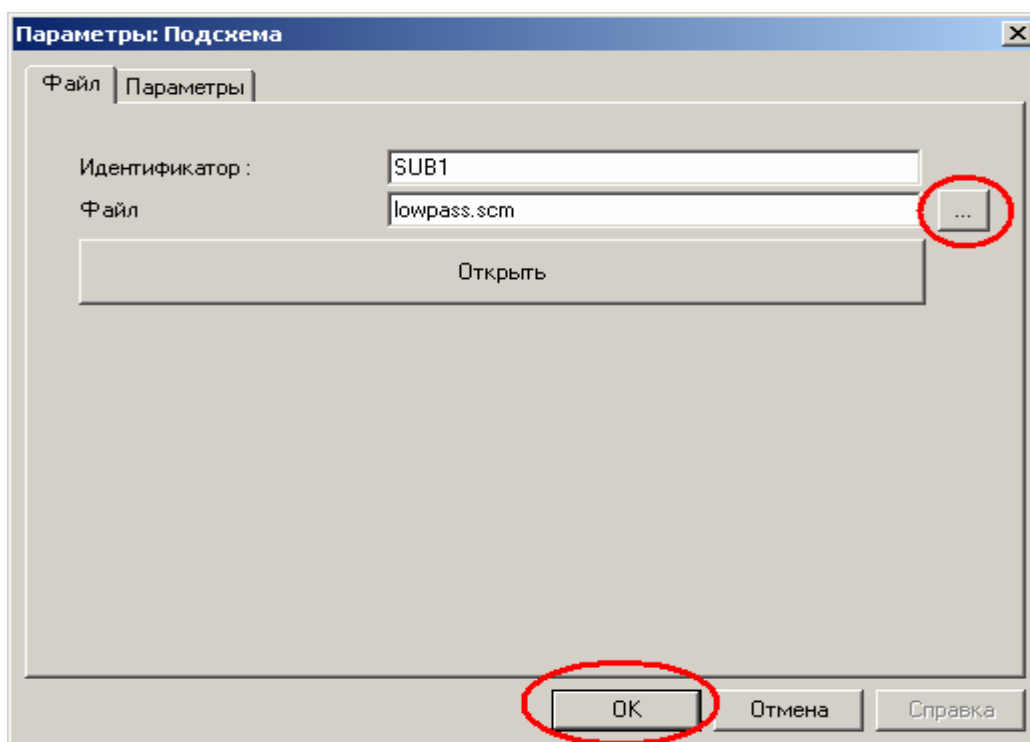




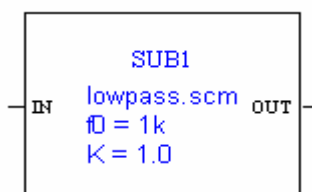
В основной схеме в окне параметров подсхемы:



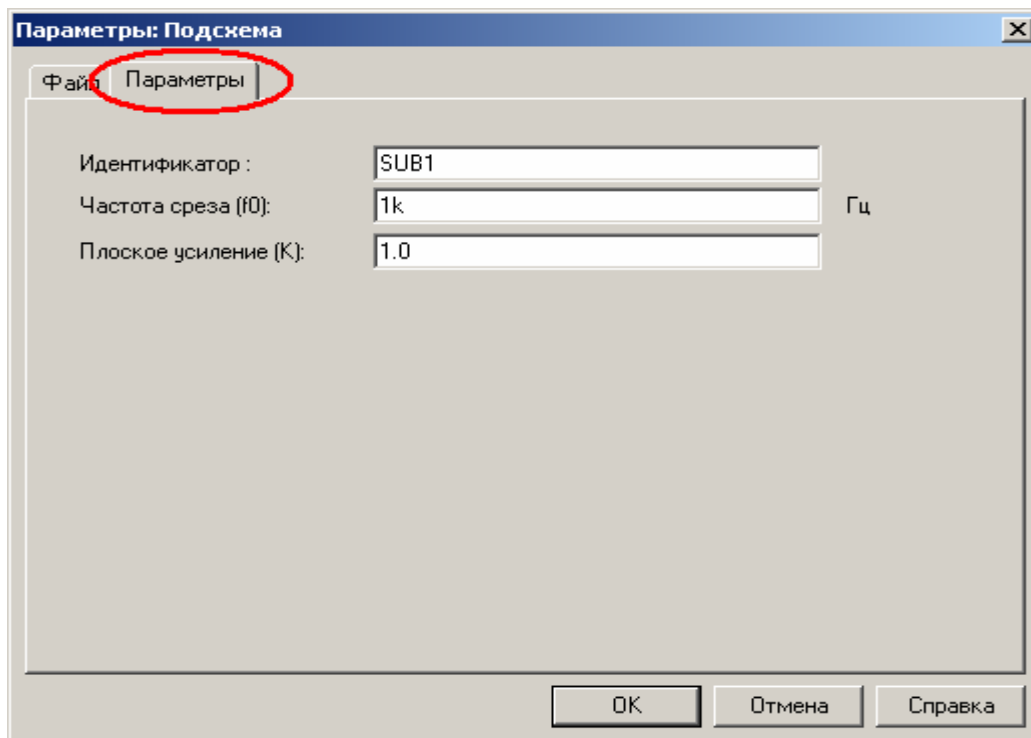
Выбирается файл подсхемы:



После этого изображение элемента на основной схеме изменяется:



Могут быть заданы параметры подсхемы:



Если в файле подсхемы были изменены внешние выводы или список параметров, для обновления в основной схеме необходимо нажать F5 (или выбрать Обновить в меню Схема).

Для доступа к элементам, сигналам, переменным и функциям подсхемы из основной схемы используется запись вида:

<имя\_подсхемы>.<имя\_элемента>  
<имя\_подсхемы>.<имя\_элемента>.<имя\_параметра>  
<имя\_подсхемы>.<имя\_функции>  
I(<имя\_подсхемы>.<имя\_элемента>)  
U(<имя\_подсхемы>.<имя\_элемента>)

*Примеры:*

SUB1.C3.C – емкость C конденсатора C3 в подсхеме SUB1  
I(SUB1.R4) – ток через резистор R4 в подсхеме SUB1  
U(SUB1.OUT) – напряжение на узле OUT подсхемы SUB1  
SUB1.function1(x) – вызов функции function1, определенной в SUB1  
SUB1.SUB2.U3 – источник U3 в подсхеме SUB2 в подсхеме SUB1  
и т.д.

Другие примеры подсхем находятся в:

<папка установки FASTMEAN>\Схемы\Подсхемы\.

## 4.2. Окно настроек переходного процесса

Удалена возможность разбивать интервал расчета на части.

Удалена возможность явного выбора между числом точек и шагом расчета. В явном виде указывается только шаг расчета. Для указания числа точек расчета используется выражение для шага  $t_{\max}/\langle \text{число\_точек} \rangle$ .

№	График	Выражение по оси X	Выражение по оси Y
1	1	t	U(IN)
2	1	t	U(OUT)
3		t	
4		t	
5		t	
6		t	
7		t	

Общее число сигналов увеличено до 30. Число одновременно видимых сигналов увеличено до 7. Снято ограничение не более 5 сигналов на одном графике.

### 4.3. Окно настроек частотных характеристик

Начальная частота, конечная частота, число точек теперь могут быть математическими выражениями.

Частотные характеристики

Параметры расчета

Начальная частота (fmin): 1 Гц

Конечная частота (fmax): 1M Гц

Число точек: 100

Развертка по частоте: Логарифмическая

№	График	Выражение по оси X	Выражение по оси Y
1	1	f	db(U(OUT)/U(IN))
2	2	f	phs(U(OUT)/U(IN))
3		f	
4		f	
5		f	
6		f	
7		f	

Параметр

Установки IFT

OK

Отмена

Параметр ...

Установки IFT ...

Справка

Общее число кривых на графиках увеличено до 30. Число одновременно видимых кривых увеличено до 7. Снято ограничение не более 5 кривых на одном графике.

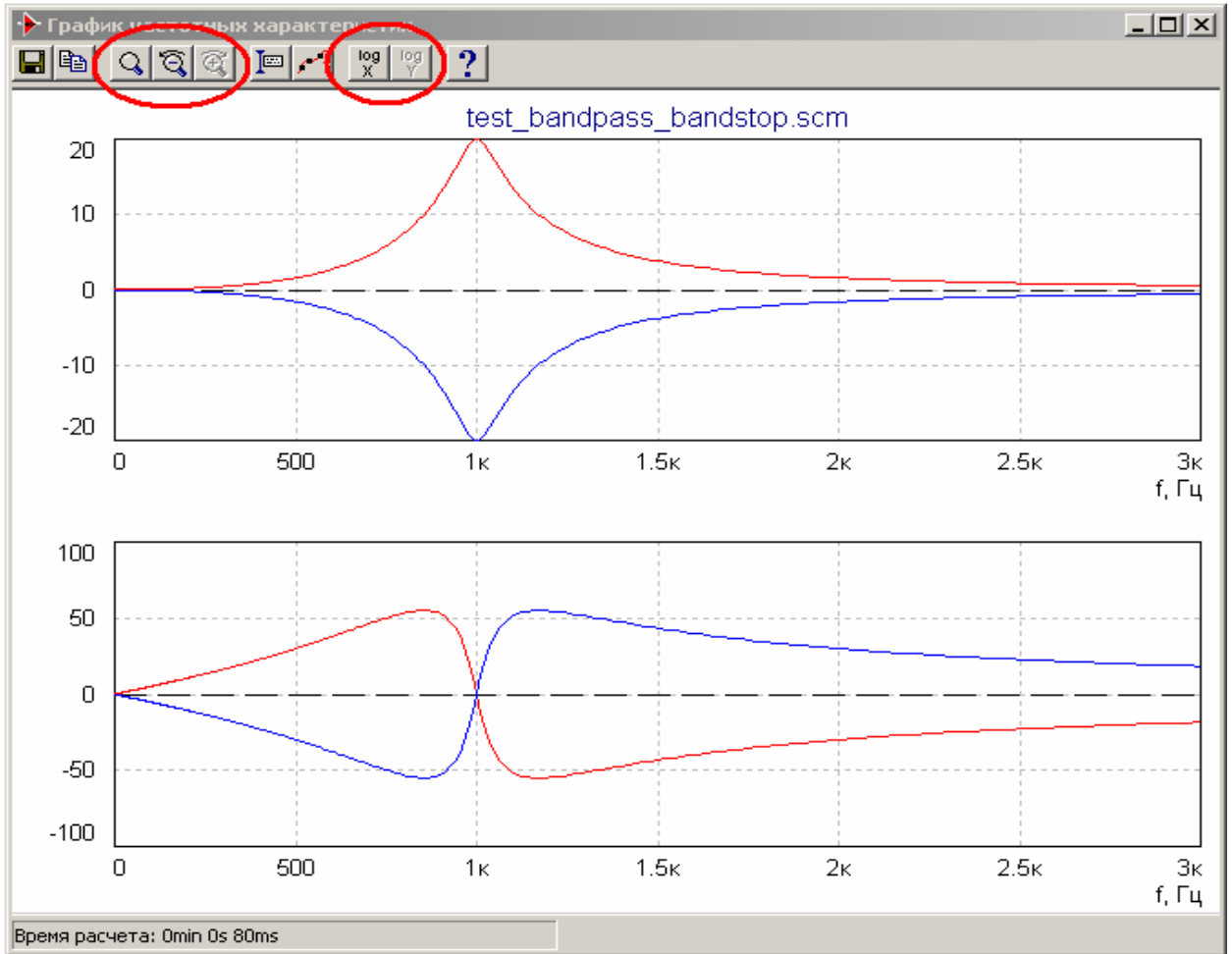
#### 4.4. Окно графиков

В окне графиков улучшена работа с масштабом:

- синхронное изменение масштаба у графиков, связанных по оси X
- память масштаба и возможность поэтапного возврата

Добавлены кнопки быстрого переключения линейного/логарифмического масштаба по осям X и Y.

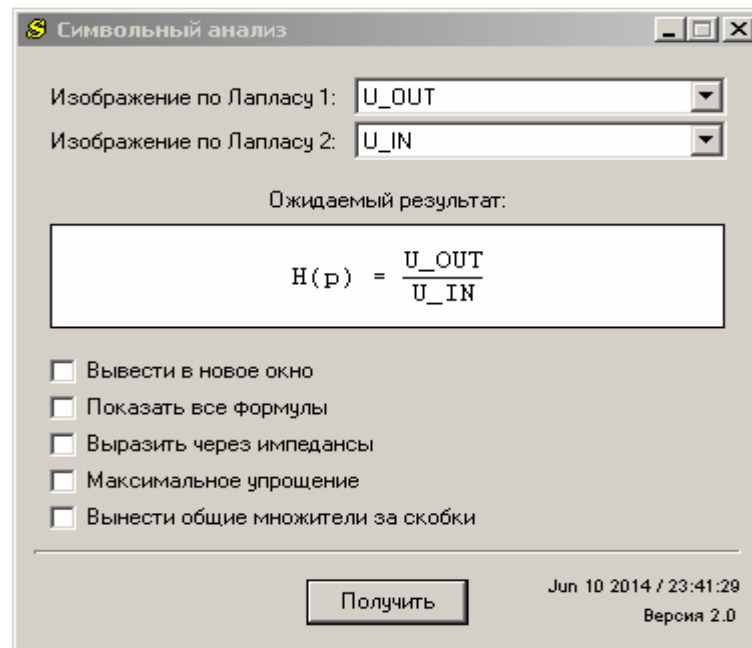
Убрано ограничение не более 5 кривых на одном графике.



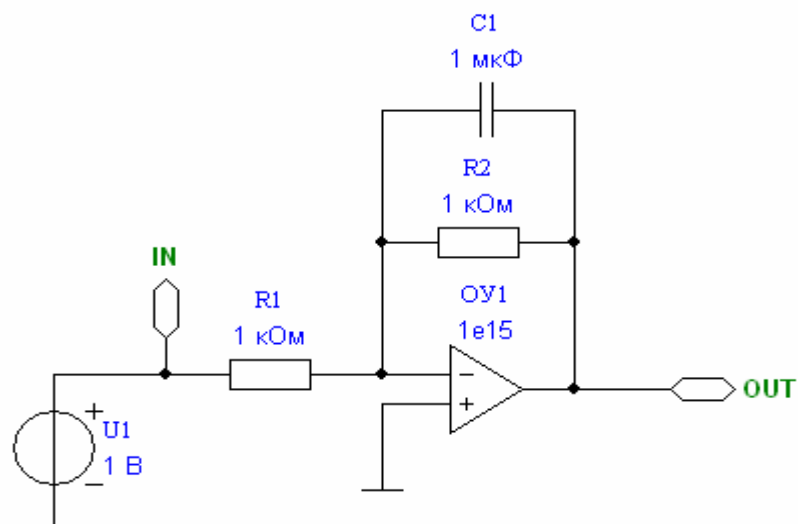
#### 4.5. Символьный анализ

Символьный анализ позволяет получить в аналитическом виде изображения по Лапласу (р-область, синоним s-область) для токов и напряжений схемы.

Начиная с версии 6.0, возможно получение операторных передаточных функций, сопротивлений и проводимостей, как отношение двух напряжений и/или токов. Возможен выбор именованных портов в качестве тестовых точек для анализа:

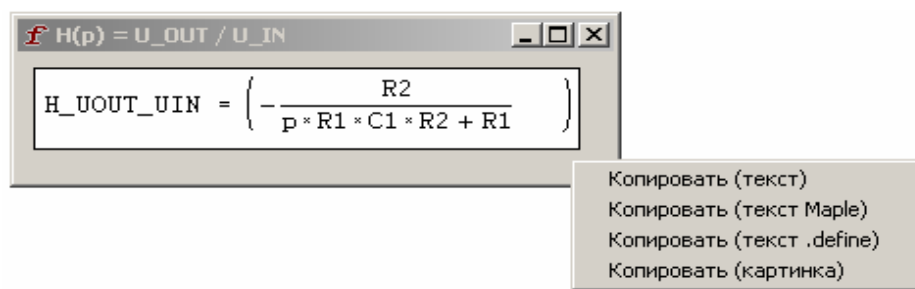


Пример:



(идеальный ОУ при  $K \geq 1e15$ )

Результат символьного анализа:



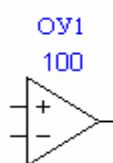
$H(p) = U_{OUT} / U_{IN}$

$$H_{UOUT\_UIN} = \left( -\frac{R2}{p \cdot R1 \cdot C1 \cdot R2 + R1} \right)$$

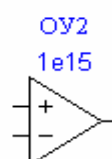
Копировать (текст)  
Копировать (текст Maple)  
Копировать (текст .define)  
Копировать (картинка)

Примечание:

**ОУ с конечным коэффициентом усиления**



**Идеальный ОУ с бесконечным коэффициентом усиления**



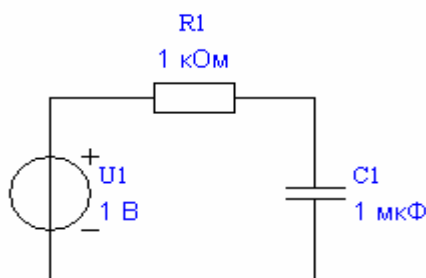
#### 4.6. Текстовый формат файла схемы (yaml)

До версии 6.0 формат файла схемы был двоичным и недокументированным. Начиная с версии 6.0, для описания схемы используется стандартный текстовый формат yaml.

Хотя это изменение является внутренним, оно косвенно имеет положительные последствия для пользователя:

- работает поиск по ключевым словам в файле;
- удобнее работать с системой контроля версий (например, git);
- файл схемы можно прочитать с помощью стандартных инструментов (например, python), изменить и записать обратно;
- могут быть разработаны инструменты для автоматической генерации схем по заданным параметрам (пример: программа синтеза фильтров).

*Пример:*



#### rc.scm (фрагменты)

```
scm:
  MajorVersion: 5
  MinorVersion: 1
  General:
    Scale: 100
    HPos: 1500
    VPos: 1500
    SWidth: 3000
    SHeight: 3000
  Special:
    LangVersion: 2
  Scheme:
    MajorSchVer: 3
    MinorSchVer: 0
  Objects:
    - ID: 17
      Type: 2010 # U
      Name: U1
      X: 1180
      Y: 1456
      Rot: 3
      FlipV: 0
```



```

FlipH: 1
active_property_page: 0
Params:
  ST: 1
  DC:
    U0: 1
    Delay: 0
    TFR: 1n
  AC:
    ...
- ID: 21
  Type: 200 # GND
  Name: GND2
  X: 1180
  Y: 1500
  Rot: 0
  FlipV: 0
  FlipH: 0
  active_property_page: 0
- ID: 22
  Type: 1000 # R
  Name: R1
  X: 1252
  Y: 1404
  Rot: 0
  FlipV: 0
  FlipH: 0
  active_property_page: 0
  Params:
    R: 1k
- ID: 24
  Type: 1001 # C
  Name: C1
  X: 1312
  Y: 1456
  Rot: 1
  FlipV: 0
  FlipH: 0
  active_property_page: 0
  Params:
    C: 1u
    IC: 0
- ID: 26
  Type: 200 # GND
  Name: GND1
  X: 1312
  Y: 1500
  Rot: 0
  FlipV: 0
  FlipH: 0
  active_property_page: 0
Wires:
- obj1:

```

```

        ID: 21
        pin: 1
    obj2:
        ID: 17
        pin: 1
- obj1:
    ID: 22
    pin: 1
    obj2:
        ID: 17
        pin: 2
- obj1:
    ID: 24
    pin: 1
    obj2:
        ID: 22
        pin: 2
- obj1:
    ID: 26
    pin: 1
    obj2:
        ID: 24
        pin: 2
TD:
Expressions:
- plot: 1
  x: t
  y: U(1)
- plot: 1
  x: t
  y: U(2)
  ...
ExprRowOffset: 0
IPNum: 1
IP:
- StartTime: 0
  EndTime: 1m
  TimeStep: tmax/100
  NumPts: 100
  PtsStepFl: 2
  EPS: 0.001
  GraphType: 1
IC: 0
SVD:
  IC: 0
  ICReadFileName: data.sv
  SVWriteFlag: 0
  SVWriteFileName: data.sv
  SV2SchemeFlag: 0
  SVPredictFlag: 0
ParamSweep: 0
  ...

```

## 5. Директивы

Директивы – это специальные указания программе, которые задаются в текстовом виде. Для создания директивы в произвольном месте схемы размещается соответствующий ей элемент «Текст». Директивы начинаются с символа `.` (точка) и ключевого слова.

### 5.1. `.define`, `.def`

Директива `.define` позволяет задавать функции и переменные пользователя. Эти функции и переменные могут затем быть использованы в выражениях параметров элементов, в окне расчета переходного процесса, в окне расчета частотных характеристик и др.

`.def` является коротким синонимом `.define`

Синтаксис:

`.define`

`имя_функции(аргумент_1, аргумент_2, ..., аргумент_N) = выражение;`

`имя_функции() = выражение;`

`имя_переменной = выражение;`

Несколько определений в пределах одного элемента «Текст» отделяются друг от друга символом `;`. Символ `%` используется для обозначения комментариев до конца текущей строки. Длинные выражения могут занимать несколько строк.

Функции без аргументов можно рассматривать как переменные, определенные пользователем.

При наличии разных определений одной и той же функции (переменной) выводится предупреждение.

*Пример:* коэффициент мощности (косинус фи)

```
.define
T = 1/U1.f; % период усреднения
P_active(U,I) = avg(U*I, T); % активная составляющая мощности
P_full(U,I) = rms(U, T) * rms(I, T); % полная мощность
cos_phi(U,I) = limit(P_active(U,I) / P_full(U,I), 0, 1);
```

Использование: cos\_phi(U(U1), I(U1)), cos\_phi(U(7), I(L1)), ...

*Пример:* результат экспорта из модуля символьного анализа

```
.define
p=i*2*PI*f;
C1 = 1.759;
y1 = p*C1;
C2 = 1.202;
y2 = p*C2;
C3 = 0.2588;
y3 = p*C3;
L1 = 1.553;
Z1 = p*L1;
L2 = 1.553;
Z2 = p*L2;
L3 = 0.7579;
Z3 = p*L3;
R1 = 1;
U1 = 1;
U_R1 = U1*R1/((Z1*Z2*y1+Z1+Z2)*(R1*y3+1)+
(y2*(Z1*Z2*y1+Z1+Z2)+Z1*y1+1)*(R1*Z3*y3+R1+
Z3));
```

Использование: АЧХ - mag(U\_R1/U1) или dB(U\_R1/U1), ФЧХ - phs(U\_R1/U1)

## 5.2. .parameter, .param

Директива `.parameter` позволяет задавать параметры подсхемы, которые будут доступны для изменения в основной схеме через окно параметров элемента. Внутри подсхемы определение `.parameter` по своему поведению аналогично определению `.define`.

`.param` является коротким синонимом `.parameter`

Синтаксис:

```
.parameter  
имя_параметра = выражение;  
имя_параметра = выражение "описание параметра";  
имя_параметра = выражение "описание параметра [размерность]";
```

Несколько определений в пределах одного элемента «Текст» отделяются друг от друга символом `';`. Символ `'%'` используется для обозначения комментариев до конца текущей строки. Длинные выражения могут занимать несколько строк.

*Пример:*

```
.parameter  
f0 = 1k "Частота среза [Гц]";  
K = 1.0 "Плоское усиление";
```

## 5.3. .global

Директива `.global` полностью аналогична `.define`, но при использовании в подсхеме имеет глобальную область видимости.

*Пример:*

Если в подсхеме SUB1.SUB2 определить `.global x=3`, то в основной схеме появится переменная `x=3`.

Если в подсхеме SUB1.SUB2 определить `.define x=3`, то в основной схеме появится переменная `SUB1.SUB2.x=3`.

## 5.4. .set

Директива .set используется для изменения значений:

- параметров элементов (R1.R, L3.L и т.д.)
- переменных, определенных через .define и .global
- параметров, определенных через .parameter

Синтаксис:

```
.set  
имя_элемента.имя_параметра = выражение;  
имя_переменной = выражение;  
имя_параметра = выражение;
```

Несколько определений в пределах одного элемента «Текст» отделяются друг от друга символом ';'. Символ '%' используется для обозначения комментариев до конца текущей строки. Длинные выражения могут занимать несколько строк.

При наличии нескольких директив .set для одной и той же переменной применяется последняя директива без вывода предупреждения.

Директивы .set применяются сначала внутри подсхемы, затем - в основной схеме. Таким образом, в основной схеме можно переопределять директивы .set подсхемы.

*Пример:*

```
.set  
U1.f0 = 1k; % Частота  
U1.U0 = 10.0; % Амплитуда  
x = R3.R*2; % Вспомогательная переменная  
SUB1.SUB2.Rload.R = x; % Сопротивление нагрузки в модуле SUB1.SUB2
```

## 5.5. .write

Директива `.write` используется для записи результатов расчета в файл.

Синтаксис:

```
.write имя_файла выражение_1 выражение_2 ... выражение_N;
```

имя\_файла – выходной текстовый файл

выражение\_i – математическое выражение FASTMEAN

Формат файла: текстовый, таблица чисел с плавающей точкой, каждая строка – одна точка расчета.

Если имя файла или выражение содержит пробелы, необходимо заключить его в кавычки. Задача пользователя избегать одинаковых имен файлов в разных директивах.

*Примеры:*

```
.write      "main signals.txt"  
            t                               % время  
            U(R_load)  
            "1 - U(CLOCK_OUT)"; % not CLOCK_OUT
```

```
.write pwm_signals.txt t U(MOS_PWM) I(MOS_PWM) U(MOS_GATE);
```

## 5.6. .write after

Директива `.write after` позволяет записывать данные в файл после того, как время достигнет заданного момента. Это удобно когда нужно отбросить начальный переходный процесс.

Директива `.write after` является общей для всех директив `.write`. В схеме должна быть только одна директива `.write after`.

Синтаксис:

```
.write after выражение
```

выражение – момент времени, после которого нужно записывать сигналы

*Пример:*

```
.write after tmax-10u % записывать в файл последние 10мкс процесса
```

## 6. Автоматизация расчетов

Данный раздел написан для инженеров и исследователей, которым требуется многократное автоматическое моделирование процессов и последующий анализ результатов в специализированных математических системах (MATLAB, python+numpy+scipy и др.)

### 6.1. Запуск расчета из командной строки

Для автоматизации расчетов FASTMEAN запускается из командной строки с указанием имени файла скрипта:

Синтаксис:

```
Fastmean.exe -s файл_скрипта  
Fastmean.exe -dir рабочая_папка -s файл_скрипта  
Fastmean.exe -o файл_схемы.scm -s файл_скрипта  
Fastmean.exe -o файл_схемы.scm -sim [-exit]  
Fastmean.exe -o файл_схемы.scm -ac [-exit]
```

После запуска программы рабочей папкой (директорией) является папка с файлом Fastmean.exe.

Если указана опция `-dir`, то рабочая папка (директория) устанавливается соответственно.

Если указана опция `-o`, то после открытия схемы рабочая папка устанавливается на папку с файлом схемы.

Если указана опция `-s`, то при начале исполнения скрипта рабочая папка устанавливается на папку с файлом скрипта. Таким образом, относительные пути, указанные в скрипте, отсчитываются от папки скрипта.

При указании абсолютных путей текущая рабочая папка не имеет значения.

Опция `-sim` запускает расчет переходного процесса.

Опция `-ac` запускает расчет частотных характеристик.

Опция `-exit` закрывает программу после окончания расчета.



Файл скрипта может содержать:

**cd <путь>;**

Изменить текущую рабочую папку.  
Рекомендуется указывать полный путь.

**open <имя схемы>;**

Открыть схему.  
Если указан относительный путь, то отсчет выполняется от папки скрипта, либо от текущей рабочей папки, если была команда cd <путь>.

**save;**

Сохранить схему в тот же файл.

**save\_as <имя файла>;**

Сохранить схему в другой файл.

**save\_netlist <имя файла>;**

Сохранить список элементов в файл.

**write <имя файла> <сигнал1> <сигнал2> ... <сигналN>;**

Записывать результаты расчета в файл.  
Аналогично добавлению директивы .write на схему.

**write after <время>;**

Записывать результаты расчета в файл после момента <время>.  
Аналогично добавлению директивы .write after на схему.

**define <...>;**

Определить функцию или переменную.  
Аналогично добавлению директивы .define на схему.

**set <...>;**

Изменить значение параметра элемента или переменной.  
Аналогично добавлению директивы .set на схему.

**<элемент>.<параметр> = <значение>;**

Изменить значение параметра элемента.  
Аналогично set <элемент>.<параметр> = <значение>;

**sim [end\_time | tmax | tend = <выражение>]  
[time\_step | step = <выражение>]  
[eps = <значение>]  
[sv2scheme = yes|no];**

Выполнить расчет переходного процесса до момента времени end\_time (tmax и tend - синонимы) с шагом time\_step (step - синоним) с точностью eps.

Если некоторые аргументы sim не указаны, то используются настройки схемы.

Если sv2scheme=yes, то в конце расчета значения переменных состояния переносятся на схему как начальные условия.

**ac [fmin | fstart = <выражение>]  
[fmax | fend = <выражение>]  
[numpts | pts = <выражение>];**

Выполнить расчет частотных характеристик от частоты fmin до частоты fmax с числом точек numpts.

Если некоторые аргументы ac не указаны, то используются настройки схемы.

**exit;**

Выйти из программы.

Команды отделяются между собой символом «;». Символ «%» означает комментарий до конца текущей строки.

*Пример:* файл скрипта run1.txt

```
open "D:\мои схемы\самая лучшая схема.scm";  
write "самая лучшая схема.txt" t U(U_in) U(R_load);  
sim;  
exit;
```

*Пример:* файл скрипта run2.txt

```
cd "D:\мои схемы";  
open "самая лучшая схема.scm";  
U_in.U0 = 540;  
R_load.R = 73;  
U_K_pwm.U0 = 0.04;  
set overload=3.0;  
save_as "самая лучшая схема2.scm"  
write "самая лучшая схема2.txt" t U(U_in) U(R_load);  
write after 1m;  
sim tmax=10m step=1u eps=10u;  
exit;
```

Использование из командной строки:

```
Fastmean.exe -s <полный_путь>\run1.txt  
Fastmean.exe -s <полный_путь>\run2.txt
```

## 7. Расчет частотных характеристик (ЧХ) через моделирование процессов

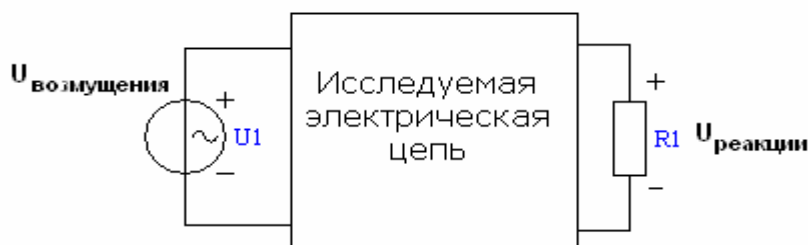
Данный раздел написан для инженеров и исследователей, работающих с импульсными (в частности, ШИМ) и нелинейными системами.

Подобные расчеты ЧХ могут быть выполнены либо с помощью внешних инструментов (MATLAB, python) путем автоматизации расчетов (см. раздел 6 данного документа), либо с помощью встроенных средств программы FASTMEAN. Последний вариант описывается в данном разделе.

Описываемый способ достаточно сложен в настройке, однако позволяет полностью управлять всеми параметрами расчета. В будущем планируется реализация более простого способа.

### 7.1. Что это такое

Пусть есть некоторая электрическая цепь (система), и необходимо получить ее частотные характеристики (ЧХ).

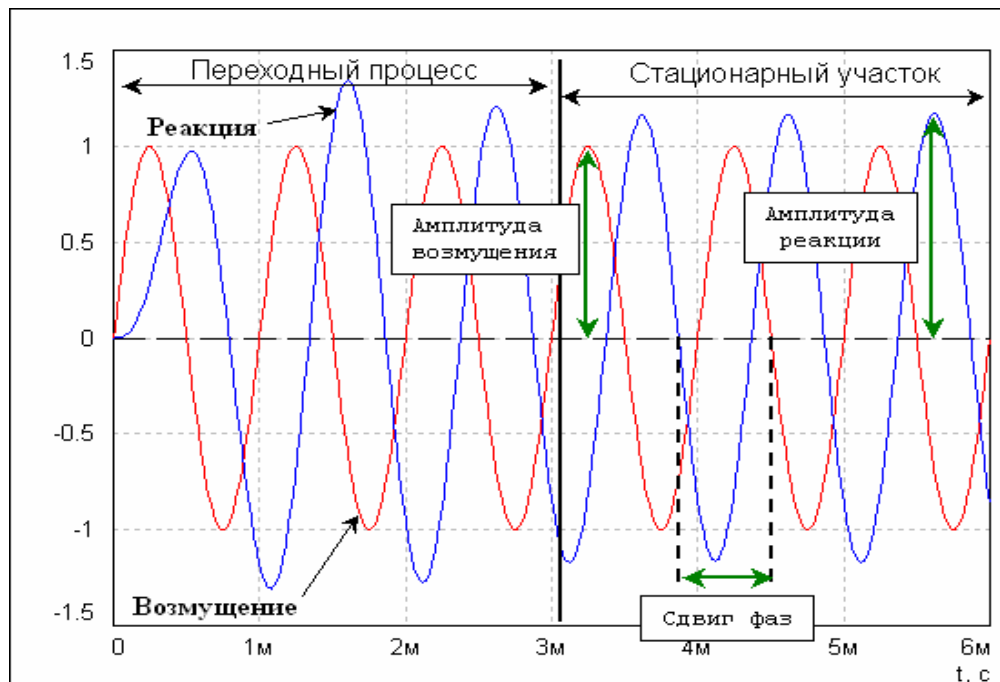


Для этого:

1. Подаётся гармоническое возмущение фиксированной частоты;
2. Определяется реакция системы во временной области;
3. Отбрасываются переходные процессы, на стационарном участке определяются амплитуда и фаза реакции системы на частоте возмущения;
4. Определяется отношение амплитуд и сдвиг фаз между реакцией системы и возмущением и, таким образом, получается одна точка ЧХ;
5. Изменяется частота возмущения и пп.1-4 повторяются;
6. В итоге получают ЧХ исследуемой системы, вычисленные в дискретных точках частотного диапазона.

В этом состоит суть расчета частотных характеристик через моделирование процессов.

К пояснению анализа во временной области:



## 7.2. Зачем нужен этот анализ

В линейных цепях данный анализ не имеет практической ценности. При корректном выполнении результат полностью совпадает с результатом метода комплексных амплитуд, но требуется больше вычислительных затрат и аккуратность при настройке процедуры анализа.

Все достоинства этого подхода проявляются при анализе нелинейных систем, в том числе импульсных, для которых метод комплексных амплитуд в явном виде неприменим (требует предварительной линеаризации системы).

Расчет ЧХ через моделирование процессов, напротив, не требует никаких изменений исследуемой системы и учитывает реальные протекающие процессы. Процедура расчета ЧХ для таких систем в целом оказывается проще.

Точность расчета ЧХ через моделирование процессов была проверена автором на целом ряде систем с импульсными преобразователями электроэнергии. Запасы устойчивости, определенные по ЧХ, очень точно совпадали с результатами моделирования во временной области.

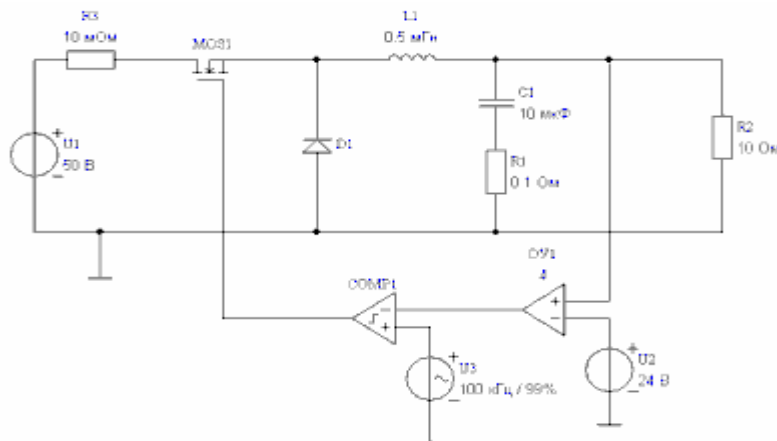
Недостатком данного способа расчета ЧХ являются большие затраты времени, необходимые для многократного моделирования процессов во временной области.

Для получения достоверных результатов с помощью данного метода необходимо следовать рекомендациям, изложенным в следующих параграфах.

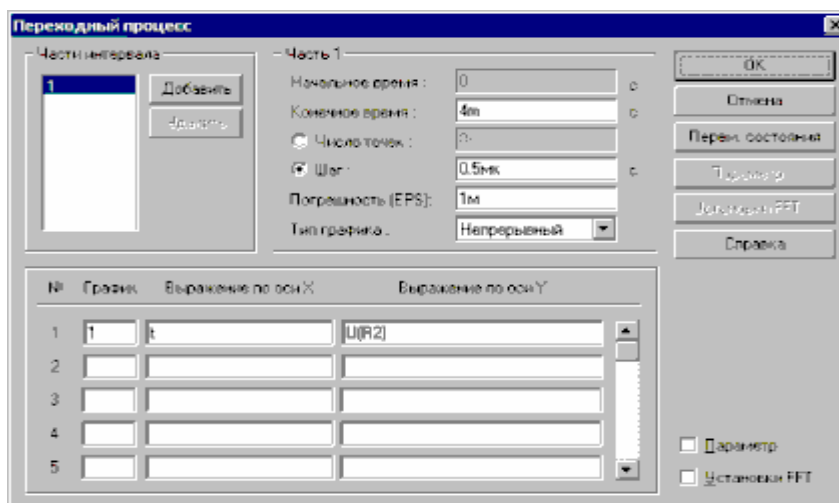
### 7.3. Как выполнить этот анализ в FASTMEAN

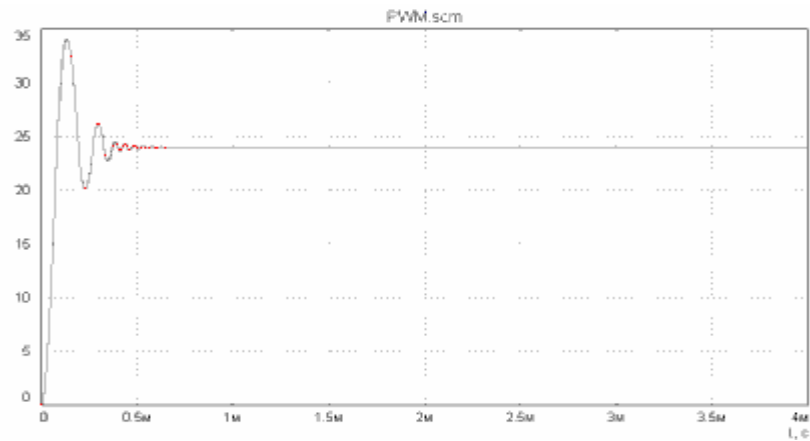
Процесс расчета ЧХ через моделирование процессов поясняется на примере.

Требуется получить ЧХ коэффициента передачи вход-выход для ШИМ-преобразователя понижающего типа:



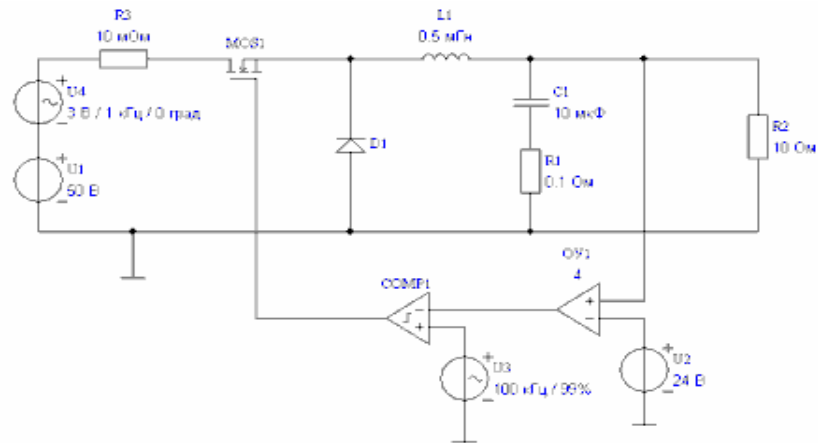
Моделируется переходный процесс для выходного напряжения:



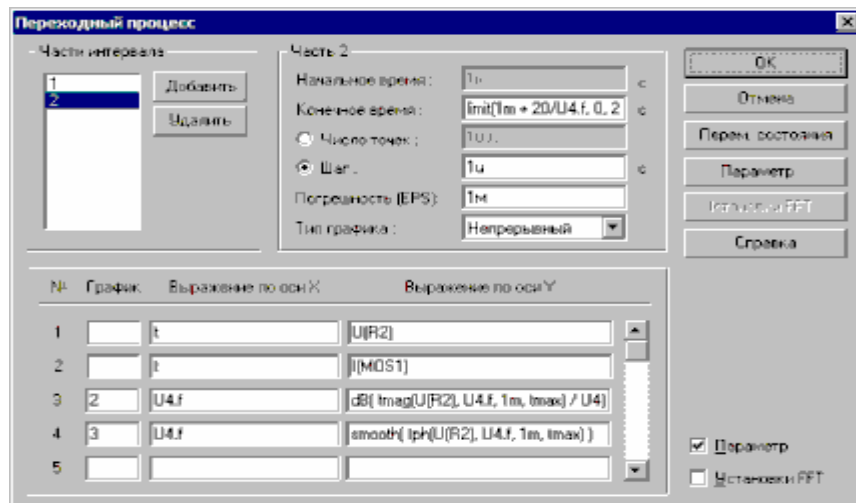


Отмечается, что примерно через 1 мс наступает установившийся режим.

На вход подаётся гармоническое возмущение (источник напряжения U4) с амплитудой 3В.



Заполняется окно расчета переходного процесса:



Расчет выполняется до момента времени **limit(1m + 20/U4.f, 0, 21m)** с шагом 1мксек (10 точек на период тактовой частоты преобразователя).

Пояснения к выражению **limit(1m + 20/U4.f, 0, 21m)**:

- **1m + 20/U4.f** означает, что моделируются процессы на интервале 20 периодов возмущения после 1 мсек (время переходных процессов).
- Ограничение с помощью функции **limit** в пределах от **0** до **21m** сек сделано для того, чтобы на низких частотах возмущения уменьшить интервал анализа и сократить время расчета. Например, при частоте 100 Гц по формуле получается:  $1m + 20/100 = 201$  мсек. Моделирование процессов на интервале такой длительности в импульсных системах с высокой частотой коммутации (десятки-сотни кГц) требует вычислительных затрат, необоснованно высоких для данного вида анализа в области низких частот.
- Для 100 Гц в интервал 21 мсек укладываются два целых периода возмущения – этого оказывается достаточно. По мере увеличения частоты возмущения увеличивается число периодов, укладываемых на интервале анализа, вплоть до заданных 20 периодов. Это необходимо для повышения точности расчета на частотах, соизмеримых с половиной тактовой частоты импульсной системы.

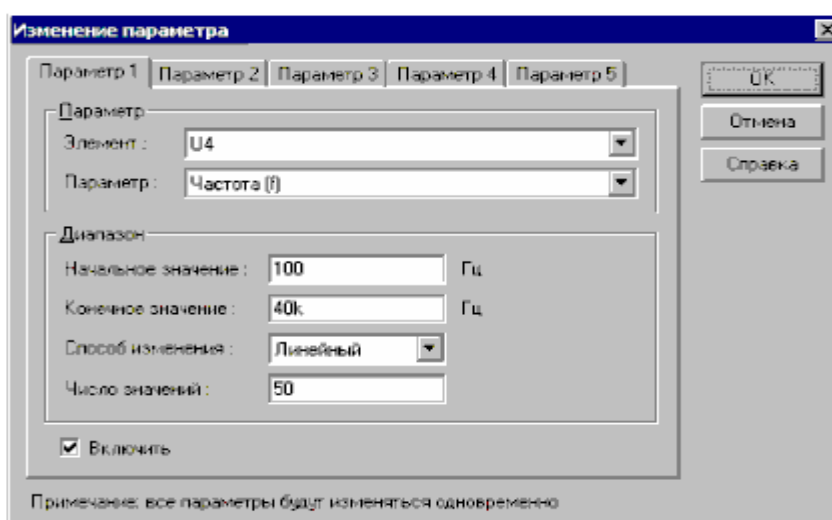
Используются функции **tmag** и **tph**.

На графиках по оси X откладывается частота возмущения **U4.f**, по оси Y: **dB( tmag(U(R2), U4.f, 1m, tmax) / U4.U0 )** и **smooth( tph(U(R2), U4.f, 1m, tmax) )**

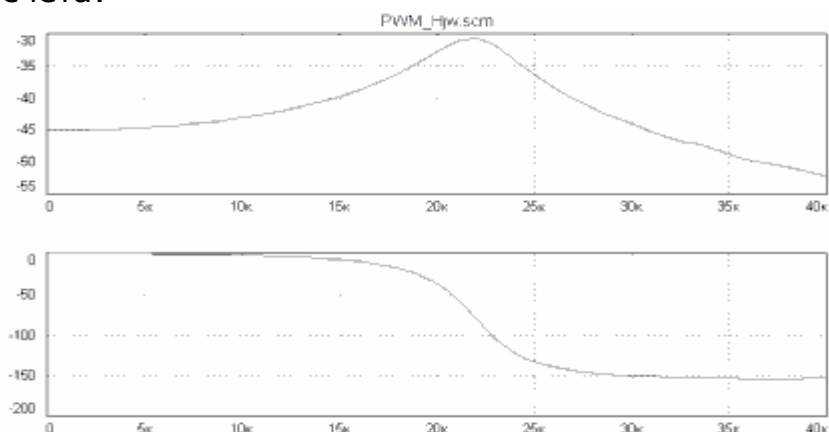
*Примечания:*

- Начальная фаза возмущения не вычитается, т.к. равна 0.
- Анализ ЧХ выполняется по сигналам после момента времени 1 мс.

Настраивается изменение параметра (частоты источника U4) от 100Гц до 40кГц:



Результат расчета:

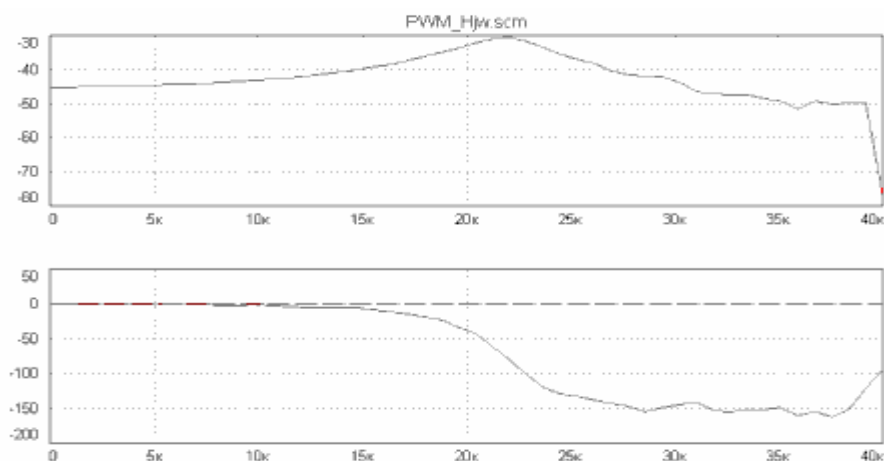


#### 7.4. Рекомендации по настройкам

##### **Выбор длительности интервала анализа**

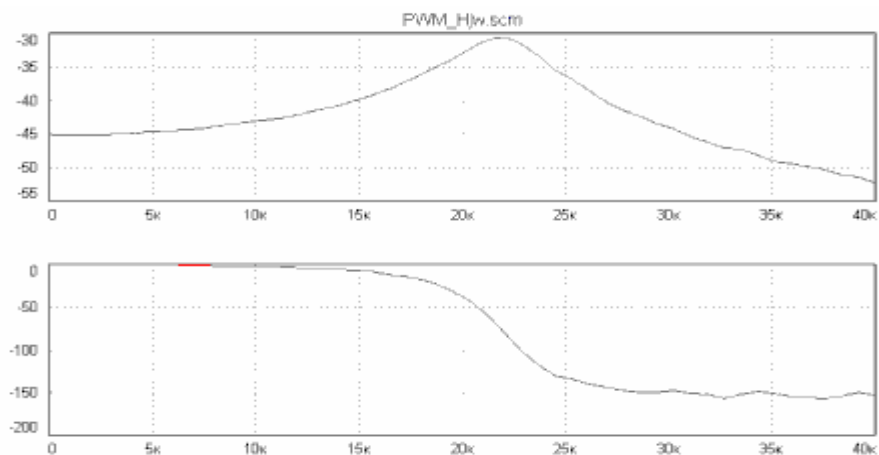
- Чем больше длительность анализируемого фрагмента сигнала, тем точнее результат. Точность повышается как за счет усреднения по нескольким периодам, так и за счет увеличения разрешающей способности по частоте (больше время наблюдения => точнее выделяется гармоника из спектра => меньше влияют шумы и др). Но при этом увеличивается время расчета.
- Корректность результата можно косвенно проверять по графику - если он плавный, без выпадающих точек, то все в порядке. Если нет, то следует увеличить интервал анализа.

Такой результат получается, если в предыдущем примере сигнал обрабатывается только по двум периодам возмущения:



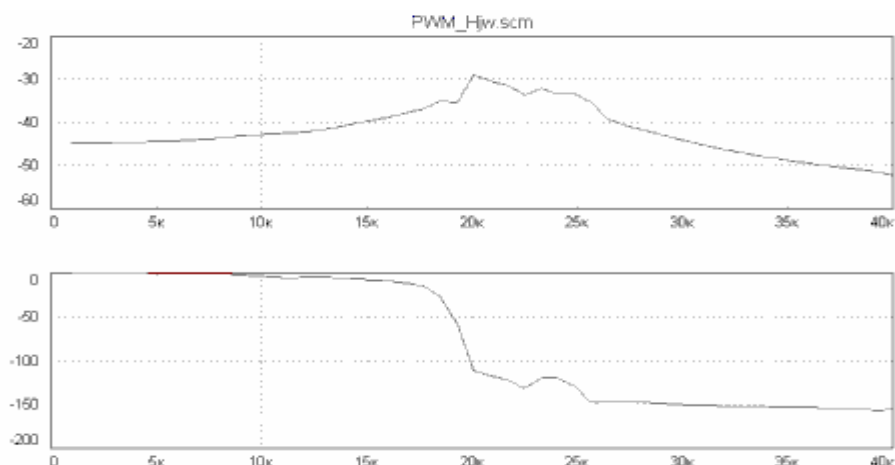


Сигнал обрабатывается по пяти периодам возмущения:

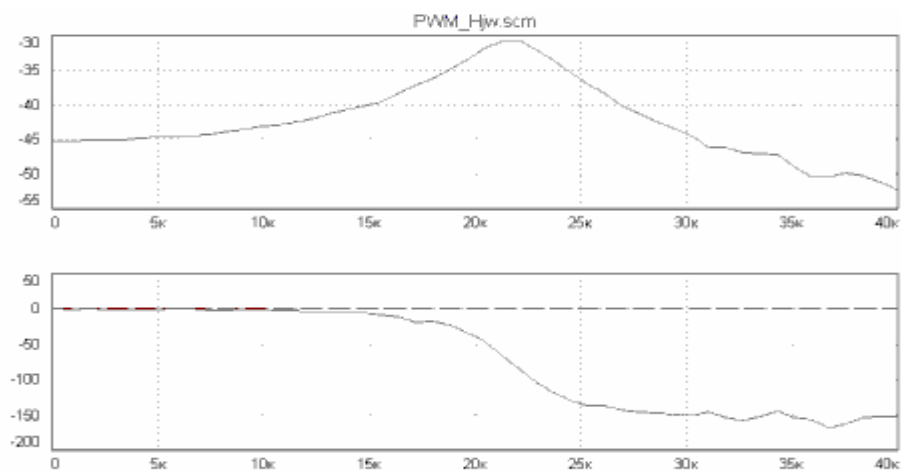


### Выбор амплитуды возмущения

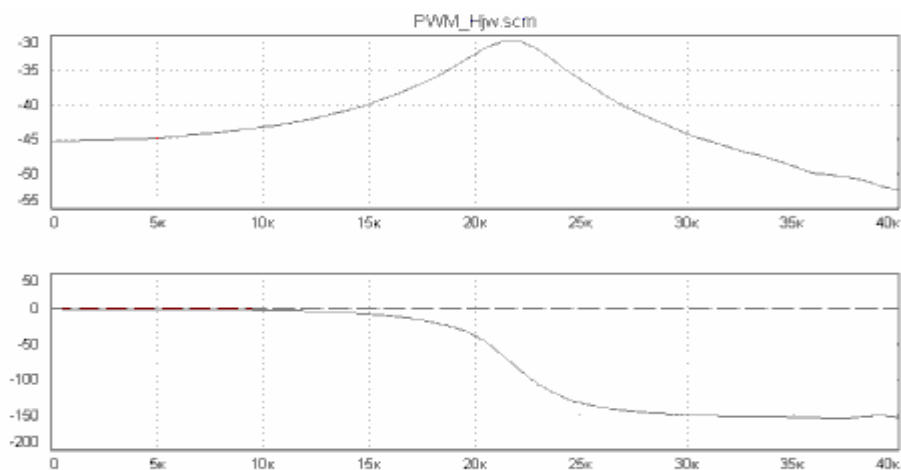
- Амплитуда возмущения не может быть выбрана слишком большой, чтобы не нарушать режим работы исследуемой системы. Косвенный контроль выполняется по наличию импульсов ШИМ - если импульсы начинают пропадать, то режим работы системы нарушен (слишком большая глубина модуляции). Результаты расчета при подаче на вход системы возмущения 9В вместо 3В:



- Когда амплитуда возмущения недостаточно высокая, результат также ухудшается. Результаты расчета при подаче на вход системы возмущения 0.3В вместо 3В:



- Для компенсации низкой амплитуды возмущения может быть увеличен интервал анализа. Результаты расчета при увеличении времени наблюдения с 20 до 100 периодов возмущения (амплитуда 0.3 В):



### Общий подход

- Для настройки расчета ЧХ через моделирование процессов необходимо провести несколько предварительных расчетов, подбирая длительность интервала анализа и амплитуду возмущения.
- Косвенный контроль корректности результатов выполняется по гладкости графиков: кривые должны быть плавными и без выпадающих точек.

## 7.5. Функции для расчета ЧХ через моделирование процессов

**tmag**(<выражение сигнала>, <частота>, <нач.время>, <кон.время>)  
возвращает модуль спектральной составляющей

**tph**(<выражение сигнала>, <частота>, <нач.время>, <кон.время>)  
возвращает начальную фазу спектральной составляющей

**tre**(<выражение сигнала>, <частота>, <нач.время>, <кон.время>)  
возвращает действительную часть спектральной составляющей

**tim**(<выражение сигнала>, <частота>, <нач.время>, <кон.время>)  
возвращает мнимую часть спектральной составляющей

Данные функции выполняют анализ <сигнала> на отрезке времени от <нач.время> до <кон.время> и возвращают параметры гармоника сигнала на заданной <частоте>:

*Параметры:*

**<выражение сигнала>** - напряжение, ток и др., для которого выполняется анализ. Примеры: U(C1); I(R2)\*2; I(VCK4)-3.

**<частота>** - частота, на которой выполняется анализ. Почти всегда совпадает с частотой возмущения. Пример: 'U4.f'.

**<нач. время>** - начало интервала, на котором обрабатывается сигнал. Обычно фиксированное число. Важно, чтобы интервал обработки содержал сигнал в установившемся режиме и не включал в себя переходные процессы.

**<кон. время>** - конец интервала, на котором обрабатывается сигнал. Можно использовать переменную '**tmax**' – конечное время расчета переходного процесса.

*Примечания.*

- <выражение сигнала>, <частота>, <нач.время>, <кон.время> могут быть как числами, так и математическими выражениями.
- эти функции самостоятельно выделяют фрагмент сигнала, состоящий из целого числа периодов. От пользователя требуется определить начало установившегося режима и выделить достаточно длинный фрагмент сигнала (несколько периодов и более).
- эти функции имеет смысл использовать только вместе с изменением параметра – частоты какого-нибудь источника.
- чтобы фаза была без скачков 360 градусов, применяется функция **smooth**. Пример использования: **smooth( tph( ... ) )**
- **smooth(ph(...)) = phs(...)**

*Примеры:*

`tmag(U(R1)/2, U2.f, 2m, tmax)`

возвращает модуль спектральной составляющей сигнала  $U(R1)/2$  на частоте  $U2.f$ , вычисленный по фрагменту сигнала на интервале от 2 мсек до конца анализа.

`smooth(tph(U(OUT), U1.f, 2.4m, tmax) - tph(U(IN), U1.f, 2.4m, tmax))`

возвращает разность фаз между напряжениями IN и OUT на частоте  $U1.f$ ; анализ выполняется на интервале от 2.4 мсек до конца анализа; устраняются скачки фазы в 360 град.