

TALLINNA TEHNIKAÜLIKOOL  
Thomas Johann Seebecki Elektroonikainstituut

IEE40LT

Gleb Bogomol 120483 IAEB

**TÜÜRITAV KOLMEFAASILINE  
SAGEDUSMUUNDUR MODELLEERIMIS  
KESKKONNAS LT-SPICE IV**

Bakalaureusetöö

Juhendaja: Mihhail Pikkov

PhD

Dotsent

Tallinn 2016



TALLINN UNIVERSITY OF TECHNOLOGY  
Thomas Johann Seebeck Department of Electronics

IEE40LT

Gleb Bogomol 120483 IAEB

**A CONTROLLED THREE-PHASE  
FREQUENCY CONVERTER IN A  
MODELING ENVIRONMENT LT-SPICE IV**

Bachelor`s thesis

Supervisor: Mihhail Pikkov  
PhD  
Associate Professor

Tallinn 2016



ТАЛЛИНСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
Институт Электроники Томаса Иогана Зеебека

ИЕЕ40LT

Глеб Богомол 120483 ИАЕВ

# **УПРАВЛЯЕМЫЙ ТРЕХФАЗНЫЙ ПРЕОБРАЗОВАТЕЛЬ ЧАСТОТЫ В СРЕДЕ МОДЕЛИРОВАНИЯ LT-SPICE IV**

Бакалаврская работа

Руководитель: Михаил Пикков  
Доктор Философии  
Доцент

Таллинн 2016

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Gleb Bogomol

23.05.2016

## **Annotatsioon**

Selle diplomitöö autorina esitlen ja selgitan adaptiivse kolmefaasilise tüüritava sagedusmuunduri mudelit koos alalisvooluosaga, mis on modelleeritud LT-SPICE IV keskkonnas. See mudel annab võimaluse paljudele raadiohuvilistele, inseneridele ja ettevõtetele säästa aega ja raha häälestamisel, maketeerimisel ja alaldite, inverterite ja muundurite modelleerimisel. See on väga aktuaalne tänapäeval, sest nõudlus jõumuundurite järele kasvab hoogsalt näiteks elektriautode ja igapäevase elektrikasutamise arendamisega seoses.

Väljatöötatud mudel on esitatud erinevate moodulite kujul, nagu kolmefaasiline alaldi, inverter, mitmesugused filtrid, pingetrafo jms. Samuti on välja töötatud mugav ning efektiivne juhtimissüsteem, mis on esitatud erinevate algoritmidenä, nagu proportsionaal-diferentsiaalne reguleerimine, muunduri sujuv käivitamine, laiusimpulssmodulatsioon, faasiline sünkroniseerimine jpt. Mudeli liides on projekteeritud nii, et kasutaja saaks lihtsalt süsteemi ümber korraldada oma vajaduste järgi, alustades konkreetsete päikeseinverterite modelleerimisest ja lõpetades proportsionaal-integraalse regulaatori häälestusega uue sagedusmuunduri jaoks.

Sellise funktsionaalsusega mudelid juba eksisteerivad, kuid peaaegu kõik neist on realiseeritud MATLABi keskkonnas SIMULINKi paketiga, mille umbkaudne hind on 5000 eurot (mai 2016) ja see on noortele firmadele, eriti raadiotele keskenduvatele, üle jõu käiv. Seda enam, et nende programmide omandamine on keerukas ning selleks vajatakse kvalifitseeritud asjatundjaid. Sellele vastandudes, käesolevas projektis kirjeldatav muunduri mudel on töötatud välja LT-SPICE IV keskkonnas, mida on kerge kasutada, mis on populaarne ning mis kõige tähtsam, see on tasuta. Seda programmi kasutatakse integraalskeemide modelleerimises ja seetõttu oli programmi rakendamiseks jõuelektroonika valdkonnas tarvis viimistleda olemasolevaid modelleerimismeetodeid ning töötada välja ka uusi. Kõik see teeb väljatöötatud mudeli unikaalseks.

Diplomitöö koosneb paljudest detailsetest tehniliste lahenduste kirjeldustest, mida saab kasutada ka teistes ülesannetes. Töö maht on umbes 50 lehekülge.

## **Аннотация**

Автором данной дипломной работы была разработана и представлена адаптивная модель трехфазного управляемого преобразователя частоты со звеном постоянного тока в среде моделирования LT-SPICE IV. Данная модель позволит многим радиолюбителям, инженерам и компаниям сэкономить время и деньги на настройке, макетировании и натурном моделировании выпрямителей, инверторов и преобразователей частоты. Это особенно актуально сегодня, когда спрос на силовые преобразователи стремительно растет в связи с развитием индустрии электромобилей и повсеместной электрификацией.

Разработанная модель представлена в виде различных модулей, таких как трехфазный выпрямитель, инвертор, различные фильтры, трансформатор напряжения и другие. Так же была разработана удобная и эффективная система управления, представленная различными алгоритмами, такими как пропорционально-дифференциальное регулирование, плавный пуск преобразователя, широтно-импульсная модуляция, фазовая синхронизация и многие другие. Интерфейс модели был разработан таким образом, чтобы пользователь мог с легкостью перестроить систему под свою конкретную задачу, начиная от моделирования конкретных солнечных инверторов, заканчивая подбором коэффициентов пропорционально-интегрального регулятора для нового преобразователя частоты.

Модели с подобной функциональностью уже существовали, однако практически все они реализованы в среде MATLAB с пакетом Simulink, чья нормальная суммарная стоимость на май 2016 года равняется 5000 евро, что достаточно много для молодых компаний и тем более для радиолюбителей. Более того, эти программы достаточно сложные в освоении, поэтому требуют специалистов с дополнительной квалификацией. В связи с этим, данная модель преобразователя была разработана в среде LT-SPICE IV, которая очень проста в освоении, популярна и, главное, абсолютно бесплатна. Эта программа используется, прежде всего, в моделировании интегральных схем, поэтому для применения в области силовой



электроники потребовалось произвести доработку существующих методов моделирования и разработку новых. Все это делает разработанную модель абсолютно уникальной.

Сама дипломная работа в основе своей содержит детальное описание множества разработанных технических решений, которые могут быть применены отдельно и в совершенно иных задачах. Объем работы составляет около 50ти страниц.

Введение .....	7
1. Обзор дипломной работы .....	9
1.1 Актуальность работы .....	9
1.2 Объект исследования .....	10
1.3 Уточнение цели и задач .....	11
1.4 Практическая значимость и новизна данной работы.....	12
1.5 Структура работы .....	13
2. Обзор разрабатываемого преобразователя частоты.....	15
2.1 Принцип работы силового преобразователя частоты.....	15
2.2 Выбор характеристик преобразователя частоты .....	17
3. Программное обеспечение LT-SPICE IV .....	18
3.1 Исходная среда разработки .....	18
3.2 Компоненты стандартной библиотеки .....	19
3.3 Разработка методов программирования в LT-SPICE IV.....	19
4. Разработка модели преобразователя частоты.....	23
4.1 Разработка модели сетевого источника напряжения.....	24
4.2 Разработка модели выпрямителя .....	27
4.3 Разработка модели тиристоров .....	30
4.4 Сглаживающий фильтр выпрямителя .....	34
4.5 Разработка модели мостового инвертора напряжения .....	35
4.6 Разработка модели трансформатора напряжения .....	37
4.7 Модель фильтра на выходе трансформатора.....	39
4.8 Разработка модели нагрузки преобразователя частоты.....	40
4.9 Разработка систем регуляции и стабилизации .....	43
4.9.1 Разработка алгоритмов обратной связи и расчета RMS .....	44
4.9.2 Разработка пропорционально-интегрального регулятора (ПИР) .....	46
4.9.3 Разработка алгоритма широтно-импульсной модуляции (ШИМ).....	48
4.10 Разработка средств тестирования системы.....	52
5. Подведение итогов .....	53
5.1 Организация удобного интерфейса .....	53
5.2 Анализ проблем моделирования .....	54
5.3 Анализ возможностей дальнейшего развития данной системы .....	55
Заключение.....	57

Использованная литература .....	59
Приложения .....	61
Приложение 1.....	61
Приложение 2.....	69

## Введение

Рынок силовых преобразователей напряжения стремительно развивается в связи с повсеместной электрификацией. К ним относятся инверторы, выпрямители и преобразователи частоты. Они используются, прежде всего, для преобразования электрической энергии с одними параметрами в электрическую энергию с другими параметрами. Область их применения невероятно широка - это и управление электродвигателем, и зарядка аккумуляторов, и питание электроники. Производство таких устройств требует денег, времени и квалифицированных специалистов. С распространением компьютерных технологий у инженеров появилась эффективная возможность уменьшить нагрузку и расходы благодаря компьютерному моделированию.

К сожалению, несмотря на многолетние разработки в области моделирования и симуляции, выбор программ для такого моделирования очень мал. Монополия на данного рода симуляции принадлежит программе MATLAB с пакетом Simulink. Полная стоимость MATLAB в MathWorks Store составляет 2000 евро, стоимость пакета Simulink - 3000 евро, т.е. стоимость работы в данной среде составляет 5000 евро, что может быть весьма значительной суммой для малых предприятий или, тем более, для радиолюбителей. Помимо высокой стоимости, пакет MATLAB и Simulink это достаточно сложная система, поэтому требует достаточно высокого уровня квалификации специалиста для эффективного использования системы.

Альтернативой MATLAB может являться бесплатная программа Scicos, однако она не очень популярна, поэтому у нее слабая интеграция с другими сервисами и крупными производителями; к тому же, большинство специалистов с ней не знакома, при этом, как и для использования MATLAB, для Scicos требуется высокая квалификация специалиста.

Решение данных проблем автор находит в программах на ядре SPICE, а именно в бесплатной программе LT SPICE IV от компании Linear Technology. Данная программа имеет ряд преимуществ: простой и дружелюбный интерфейс, приличная скорость симуляции, распространенность и популярность, широкие возможности моделирования и, конечно же, бесплатность. В основном, программа давно и широко применяется в области аналогово-цифровых схем, поэтому для применения в области силовой электроники, методы моделирования требовали доработки. Именно эта задача была решена в данной работе. Конкретная же цель - это разработка полноценной модели силового преобразователя частоты в среде LT-SPICE IV.

# 1. Обзор дипломной работы

## 1.1 Актуальность работы

С каждым годом системы автоматического проектирования и симуляции все больше и больше распространяются как в производственной, так и в научной среде. Эти системы перекладывают большую часть вычислительной работы с инженера или ученого на компьютер. Использование такого программного обеспечения значительно может снизить финансовые и временные расходы на производство продукта. Это особенно важно в наше время, когда нестабильность рынка очень высока. Предприятия стремятся не только уменьшить прямые расходы, но и минимизировать риски ошибок, чтобы избежать дальнейших убытков. Такое положение дел делает очень актуальными задачи, поставленные и решенные в данной дипломной работе.

Множество предприятий занимаются разработкой и производством силовых преобразователей. В Эстонии таковыми являются компании Ericsson, ABB, ElectroAir, TET Estel и другие. Спрос на данного рода изделия, а также конкуренция предложений являются высокими в связи с глобальной электрификацией. Силовые преобразователи - это устройства преобразующие один вид энергии в другой. Обычно это преобразование энергии сети в энергию, необходимую для конкретного устройства. Несмотря на существование международных стандартов, ко многим устройствам-потребителям нужны уникальные преобразователи. На разработку каждого из них требуется время и ресурсы, которые можно было бы значительно сократить, если существовала бы гибкая система моделирования, которая легко подстраивалась бы под каждую задачу конкретно.

Очевидно, что введение подобной системы в эксплуатацию так же требует определенных затрат, таких как стоимость программного обеспечения и время освоения данной системы специалистом. Актуальность данной работы проявляется и в

том, что данная проблема сводится к минимуму, благодаря использованию в качестве основы системы - условно бесплатное и весьма распространенное программное обеспечение LT-SPICE.

## **1.2 Объект исследования**

Для разработки представленной здесь системы потребовалось провести тщательное исследование, которое можно разделить на три принципиальные категории:

1) Программная среда схемотехнического моделирования LT-SPICE. Основана на ядре SPICE (Simulation Program with Integrated Circuit Emphasis) и разработана компанией Linear Technology. Эта программа бесплатна и способна успешно работать как в среде Windows, Mac OS, так и в среде Linux (под Wine). Вся система реализована внутри данной программы.

2) Силовой преобразователь напряжения - это собирательное название для многих типов устройств, таких как выпрямитель, инвертор и преобразователь частоты. Разработанная модель содержит возможность симуляции множества конфигураций: неуправляемый, полу управляемый или управляемый мостовой выпрямитель или инвертор, трехфазная система входа/выхода или однофазная, различные подключения трансформатора и т.д.. Все эти параметры сильно влияют на качество преобразованной энергии, габариты и стоимость устройства.

3) Система управления преобразователем и пользовательский интерфейс. Все это составляет примерно половину представленной здесь системы, чья организация позволяет настраивать необходимые пользователю конфигурации максимально ясно и с минимальным количеством “нажатий”. Под системой управления подразумеваются: генераторы сигналов, источники напряжения, сенсоры и алгоритмы управления выпрямителем или инвертором. В связи с очень ограниченными возможностями классического программирования в среде LT-SPICE, возникла необходимость разработки новых алгоритмов управления и анализа.

### 1.3 Уточнение цели и задач

Главной целью данной работы является разработка реалистичной модели, а так же гибкой системы симуляции силового преобразователя в среде LT-SPICE IV с эффективной системой управления. Данную задачу возможно разделить на несколько подзадач:

- 1) Разработка модели источника сетевого напряжения;
  - а) трехфазный источник 230/400В, 50Гц;
  - б) имитация искажений сетевого напряжения;
  
- 2) Разработка модели трехфазного управляемого тиристорного выпрямителя;
  - а) мост Ларионова;
  - б) разработка модели тиристора;
  
- 3) Разработка системы управления выпрямителем;
  - а) управление углом отпирания тиристорov;
  - б) мягкий старт выпрямителя;
  
- 4) Расчет модели дросселей (трехфазных и однофазных), емкостей, LC-фильтров, а так же имитация индуктивности проводов;
  
- 5) Разработка системы мостового инвертора напряжения (400 Гц);
  
- 6) Разработка системы управления инвертором;
  - а) управление широтно-импульсной модуляцией;
  
- 7) Разработка системы обратной связи;
  - а) разработка модели сенсоров (RMS);
  - б) разработка алгоритма и подбор коэффициентов пропорционально-дифференциального регулятора;
  
- 8) Разработка модели трехфазного трансформатора (треугольник-звезда);
  - а) имитация нелинейных искажений на обмотках трансформатора;
  - б) имитация индуктивного рассеяния трансформатора;



9) Разработка удобного интерфейса пользователя;

а) вывод и указание используемых переменных (частота сети, амплитуда сети, время плавного пуска, требуемое напряжение на выходе и др.);

10) Разработка возможности упрощения модели для проектирования и симуляции иных типов преобразователя;

а) Разработка систем частичной симуляции и тестирования;

Помимо основных вышеописанных задач можно выделить задачи исследовательского и аналитического характеров:

11) Изучить существующий материал по данной тематике;

12) Проанализировать работоспособность системы, выявить проблемы;

13) Найти и описать потенциальные возможности выполнить поставленную цель более эффективным и удобным для пользователя способом.

#### **1.4 Практическая значимость и новизна данной работы**

Очевидно, что удобная и бесплатная система симуляции и моделирования электрических систем преобразователей может значительно сэкономить ресурсы предприятия, занимающегося разработкой и производством “классических” преобразователей, однако область применения данной работы значительно шире.

Данную работу можно рассматривать как собрание технологических решений в области симуляции и моделирования в среде LT-SPICE. Каждое из этих решений может быть применено самостоятельно в решении задач, отличных от описанных выше. Так же система открыта и любой инженер сможет ее дополнять и совершенствовать.

Данная дипломная работа затрагивает множество разных возможностей LT-SPICE IV, которые подробно описаны вместе с техническими решениями, в которых они использованы. Это создает высокую образовательную ценность данной работы.

Изучение данной работы позволит многим специалистам, как знакомых с LT-SPICE, так и не знакомым, освоить новые приемы моделирования и симуляции и, тем самым, подняться на новый уровень владения данной средой моделирования.

Новизна данной работы заключается в трех вещах:

Во-первых, существующие системы симуляции обычно либо требуют покупки дорогостоящего программного обеспечения (Proteus, MATLAB и др.), либо имеют крайне ограниченный сервис (например SemiSel Simulation от Semikron). Более того, алгоритмы моделирования часто скрыты от пользователя. Все это делает систему, разработанную и представленную в данной работе, уникальной и не имеющей настоящих аналогов.

Во-вторых, практически вся информация об LT-SPICE IV содержится либо в справочниках, описывающих конкретные элементы отдельно, либо разбросана по форумам. В данной же работе рассматривается комплексное взаимодействие разных приемов и решений, объединенную в полноценную систему.

И, в-третьих, большая часть решений, в особенности алгоритмы генерации и управления, являются новыми и уникальными.

## **1.5 Структура работы**

Первая глава раскрывает актуальность задачи, определяет объекты исследования, цели и задачи; показывает практическую и теоретическую значимость работы.

Во второй и третьей главах рассматриваются базовые вопросы о системе, раскрывается понятие силового преобразователя и программной среды LT-SPICE IV.

Четвертая глава самая крупная - в ней содержится вся техническая информация, с помощью которой даже начинающий пользователь сможет воссоздать данную систему и успешно ее применять.

Пятая глава об интерфейсе, о том, какие проблемы осталось решить и какие у программы есть перспективы.

В заключении подводятся итоги и формулируются выводы.

После этого предоставляется список использованной литературы и несколько справочных приложений, которые могут быть полезны пользователю, в том числе и схема разработки в большом формате.

## **2. Обзор разрабатываемого преобразователя частоты**

### **2.1 Принцип работы силового преобразователя частоты**

Формально, преобразователь - это электротехническое устройство, преобразовывающее электроэнергию с одними параметрами в электроэнергию с другими параметрами. Такими параметрами могут быть частота, напряжение, сила тока, число фаз и сами фазы напряжения. Если предусмотрена возможность контролируемого изменения каких-либо параметров, то преобразователь называется управляемым. Так же преобразованию может поддаваться и качество электроэнергии, которое обычно выражается в присутствии высших гармоник, и в отклонениях от заданных параметров, названных выше. Механизмов преобразования энергии существует множество, в данной работе разрабатывается, прежде всего, статический преобразователь на полупроводниковых элементах. По характеру преобразования электроэнергии силовые преобразователи можно разделить на:

- 1) Выпрямители - преобразование переменного напряжения в постоянное. Причем под постоянным может подразумеваться любой ток, у которого среднее напряжение отличается от нуля.
- 2) Инверторы - преобразование постоянного напряжения в переменное.
- 3) Преобразователи частоты - преобразование одной частоты в другую. Обычно они являются самыми сложными типами преобразователей и могут содержать в себе как выпрямитель, так и инвертор. Поэтому в основе данной системы лежит полная модель преобразователя частоты со промежуточным звеном постоянного напряжения.

Рассмотрим блок-схему преобразователя частоты, на котором основана система:

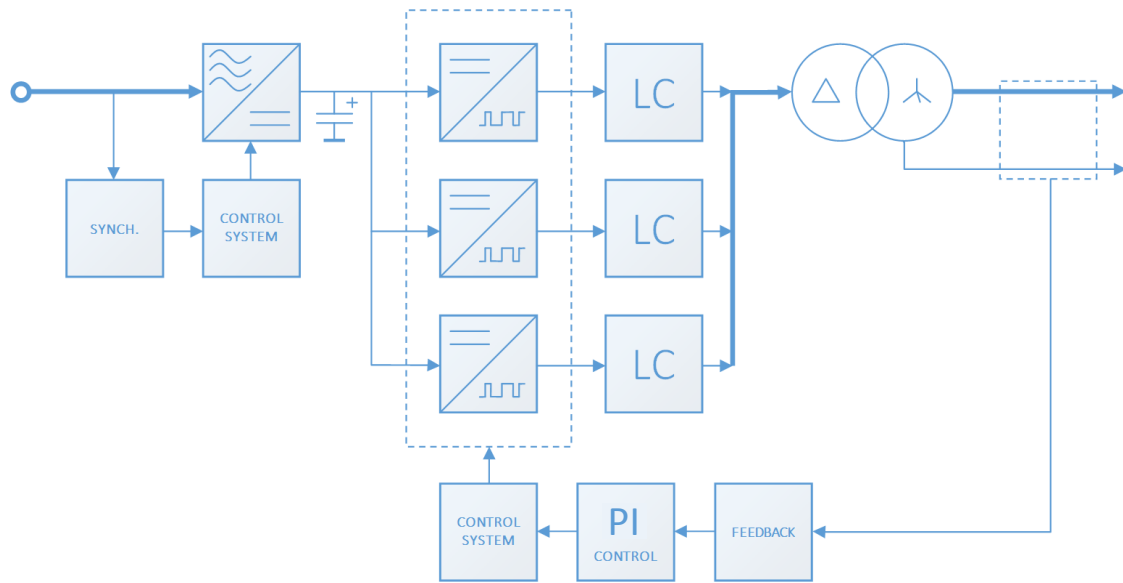


Рис.2.1.1. Блок-схема преобразователя частоты со звеном постоянного напряжения. Жирные линии представляют собой силовую шину из трех фаз.

Энергия поступает от трехфазного источника переменного напряжения - обычно это энергия сети 230/400В, 50Гц. Напряжение проходит через дроссель и подается на трехфазный управляемый тиристорный выпрямитель, соединенный в виде моста Ларионова. Это напряжение считывается датчиками для осуществления синхронизации в управлении выпрямителем. Выпрямленное напряжение проходит через сглаживающий конденсатор, после чего идет на три независимых мостовых инвертора. К выходам инверторов подключены LC-фильтры из дросселей и конденсаторов. Теперь напряжение проходит через трехфазный трансформатор в соединении треугольник-звезда. Выходное напряжение считывается датчиками, создавая обратную связь для регулирования инверторов. Автоматическая регуляция осуществляется методом пропорционально-интегрального регулирования (PI control) и широтно-импульсной модуляцией. На выходе имеем три фазы с нейтралью. Система из трех однофазных инверторов обеспечивает независимость фаз, т.е. система сохраняет сдвиг фаз в 120 градусов, однако любая фаза может работать независимо от двух других. Это повышает надежность системы и повышает работоспособность в случае асимметричной нагрузки. Если одна фаза выйдет из строя, то это не повлияет на остальные две. Также фазы можно подключать к разным устройствам.

## 2.2 Выбор характеристик преобразователя частоты

Разработка данной системы соответствует разработке реального преобразователя, поэтому здесь рассматривается решение конкретного технического задания.

На входе 230/400В, 50Гц.

На выходе три независимые фазы с нейтралью в 127/220В и 400Гц.

Мощность преобразователя на выход 250кВА (200кВт без реактивной составляющей).

Входной дроссель подавляет гармоники, наведенные на линии электропередач, и защищает преобразователь от резких скачков. Падение напряжения на нем должно составить примерно 4%. Для этого при мощности 250кВА и линейном напряжении 400В индуктивность должна составить примерно 70мкГн.

На выпрямленном участке напряжение в номинальном режиме работы равно 540В, следовательно, на выходе инвертора с коэффициентом широтно-импульсной модуляции 0.95, эффективное (среднеквадратичное) напряжение составит примерно 362В. С запасом на падение напряжения на 10% примем минимальными 326В - т.е. трансформатор должен иметь коэффициент трансформации  $326/127 = 2.57$ .

Для сглаживания выпрямленного тока используем емкость в 10мФ, которую на практике легко реализовать в виде трех параллельных конденсаторов по 3.3мФ.

Частота широтно-импульсной модуляции составит 10кГц, несущий сигнал модуляции составит требуемые 400Гц. Это значит, что частота среза сглаживающего фильтра должна быть в диапазоне от 400Гц до 10кГц.

В качестве LC-фильтра, превращающего поток пульсаций в синусоиду, используем конденсатор в 100мкФ и индуктивность рассеяния трансформатора - это нестандартное решение, которое может позволить сэкономить на выходном дросселе.

## 3. Программное обеспечение LT-SPICE IV

### 3.1 Исходная среда разработки

Создание модели реализовано в программной среде LT-SPICE IV от компании Linear Technologies. Простым языком - это программа для симуляции электрических схем. В ней есть встроенные библиотеки с большим ассортиментом электрических компонентов, такие как сопротивления, индуктивности, емкости, диоды, транзисторы, источники напряжения, переключатели и многие другие. Названные компоненты будут кирпичиками представленной здесь системы. Все эти компоненты имеют множество внутренних параметров, часть из которых влияет на симуляцию, однако некоторые параметры нужны лишь для информации о реальном аналоге. Обычно это параметр номинальной мощности, который есть почти у каждого компонента и ни на что не влияет.

Библиотеку можно пополнять либо устанавливая пакеты компонентов, либо создавая их самим. Многие производители компонентов имеют SPICE-модели своих компонентов, совместимые с программой LT-SPICE IV. Это позволяет пользователям тестировать конкретные модели, которые в последствии могут быть куплены для установки на устройство. Недостатком таких библиотек зачастую является отсутствие информации о внутреннем строении компонента. Другими словами пользователь не всегда знает, какие явления компонент учитывает, а какие опускает. Создание собственной библиотеки решает эту проблему. “Создать компонент” означает создание под-схемы из базовых компонентов. Каждый выход которой точно определен. Эти под-схемы с контактами заменяются графическим символом с фиксированными контактами, соответствующими контактам под-схемы. В дальнейшем этот символ можно использовать как обычный компонент в любой схеме. Во встроенных библиотеках отсутствует подходящая модель тиристора, поэтому возникла необходимость разработать таковую.

## 3.2 Компоненты стандартной библиотеки

Поскольку SPICE разрабатывался для моделирования аналогово-цифровых схем, а в особенности интегральных схем, модель каждого компонента учитывает кроме основных, множество дополнительных и паразитных параметров. Практически в каждом компоненте учтены паразитные емкости, индуктивности, а также последовательное и шунтирующее (параллельное) сопротивление. На рис.3.2.1 изображена эквивалентная схема замещения конденсатора.

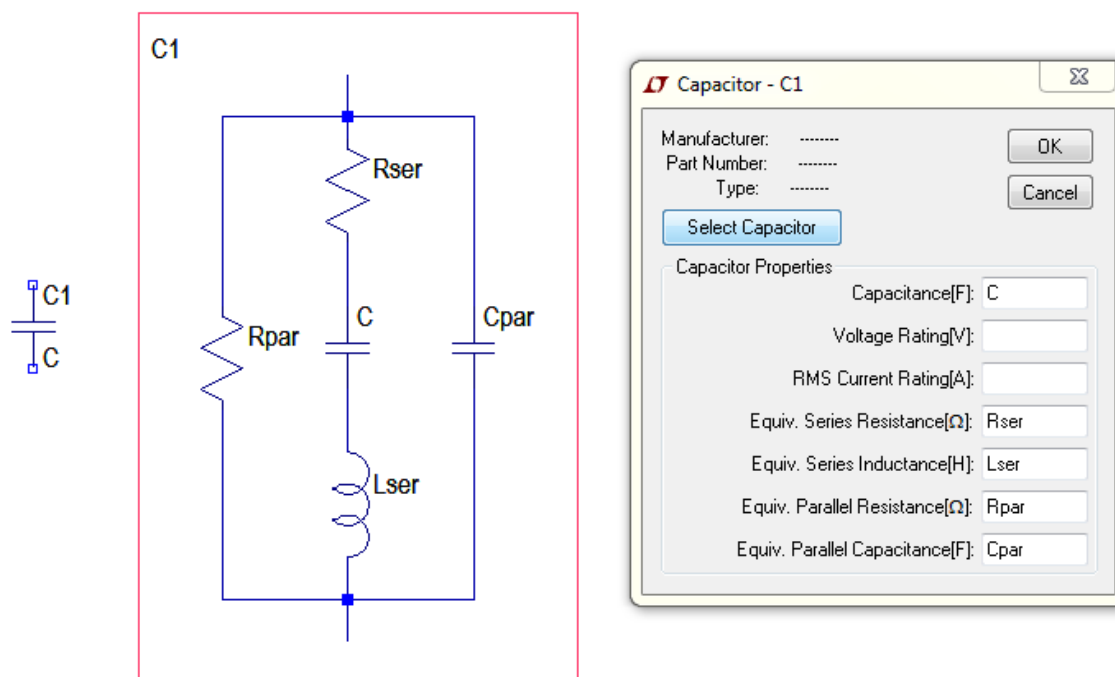


Рис.3.2.1 Символ конденсатора, эквивалентная схема, меню конденсатора.

Такие компоненты как транзисторы или диоды обладают намного большим набором параметров, о них речь пойдет в четвертой главе.

## 3.3 Разработка методов программирования в LT-SPICE IV

К сожалению, в программной среде LT-SPICE IV отсутствует возможность классического программирования, это значит, что в программе отсутствует возможность программировать компоненты основываясь на классических методах языков C и Java. В программе, к примеру, нельзя произвольно инициализировать переменные, создавать полноценные циклы for или запоминать некоторые значения



во время процесса симуляции. Последняя проблема оказалась самой трудно-разрешимой. Несмотря на все это, многие недостатки можно преодолеть.

Самым важным компонентом в данной работе является функциональный источник напряжения. Суть его заключается в том, что разность потенциалов на этом источнике задается математической функцией от времени. Причем переменными так же могут выступать значение потенциалов или токов на любых участках цепи. Помимо этого в программе есть аналогичные функциональные источники тока. Все это позволяет создавать алгоритмы и программы в среде LT-SPICE IV. Таким образом, каждый компонент и каждая связь представляют собой переменные (сила тока и потенциал соответственно), которые можно использовать в функциональных источниках. Нужно понимать, что эти переменные, в свою очередь, являются функциями от времени, так что глобальная переменная тут одна - время. Это является крупным недостатком, однако об этом позднее. Константы же можно инициализировать отдельно, что очень удобно для пользователя. Математические функции для программирования функциональных источников представляют собой сочетания элементарных алгебраических и логических функций, заданных через особый синтаксис SPICE. Однако в большинстве своем этот синтаксис совпадает с общепринятым в математике, что делает его интуитивно понятным.

Для того, чтобы представить, как это работает, разберем источник, генерирующий нужный нам гармонический сигнал, т.е. синусоиду с заданной частотой, амплитудой и фазой. Конечно, обычную синусоиду можно было бы задать и обычным источником напряжения, однако только функциональный источник дает такое количество контроля над сигналом. Рассмотрим рис.3.3.1.

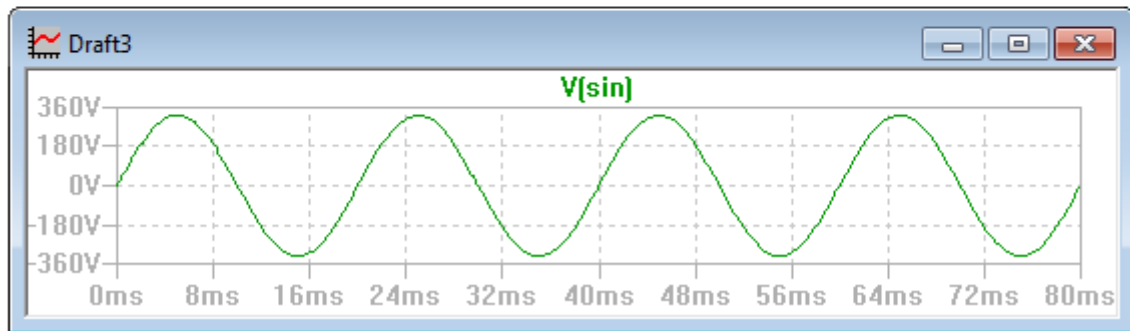
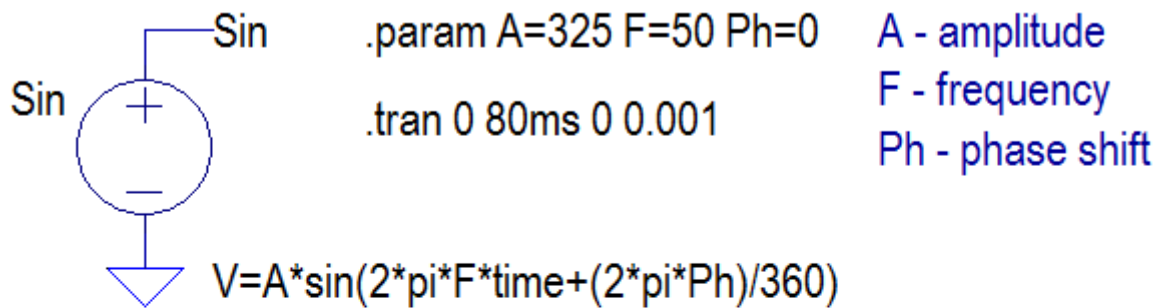


Рис.3.3.1 Пример функционального источника с заданными директивами, комментариями и симуляцией.

Верхний потенциал назван Sin, нижний, в виде треугольника - это “земля”, т.е. нулевой потенциал. Сам источник тоже назван Sin - это очень удобно. Напряжение на этом источнике, а значит и величина потенциала Sin, равняется следующей функции:  $V=A*\sin(2*\pi*F*time+(2*\pi*Ph)/360)$ . Рассмотрим подробнее:

- sin() - обычная функция синуса,
- «\*» - оператор умножения
- «/» - оператор деления
- pi - константа равная приближенному числу пи,
- time - переменная времени,
- A - амплитуда в вольтах,
- F - частота в герцах,
- Ph - фазовый сдвиг в градусах.

Величины pi и time - определены самой программой, A, F и Ph - определяются пользователем в строке “.param” (рис.3.3.1).

Использование такой строки очень удобно для пользователя, т.к. установив значение параметра, оно устанавливается во все функции, где использовано. После этого, если конфликтов в системе нет, при запуске симуляции вычисляются значения напряжения и тока для данного источника. Для того, чтобы другой функциональный источник мог использовать эти значения, нужно в его функции использовать переменные  $V(\text{Sin})$  или  $I(\text{Sin})$ . В первом случае Sin - название потенциала, во втором случае - это название источника. Очевидно, что, в данном примере, для расчетов от функции источника напряжения следует использовать переменную  $V(\text{Sin})$ , а будь на ее месте функциональный источник тока, то больше подошла бы  $I(\text{Sin})$ . Для того, чтобы не возникало путаницы, в данной работе все вычисления и функции производятся в функциональных источниках напряжения, а в качестве переменных используются  $V(\text{название\_потенциала})$ , обозначающее напряжение на участке. Если необходимо указать напряжение между двумя ненулевыми потенциалами, то можно использовать переменную в виде  $V(\text{название\_одного\_потенциала,название\_другого\_потенциала})$ . Всю функцию от начала до конца можно записать и через один источник, но для наглядности ее удобно разделить на несколько источников, тогда можно будет после симуляции пронаблюдать промежуточные вычисления. В своих моделях я использовал оба варианта.

Подробнее о работе программы LT-SPICE IV будет рассказано в следующих главах. Перечень элементарных функций функциональных источников представлен в Приложении 2.

## 4. Разработка модели преобразователя частоты

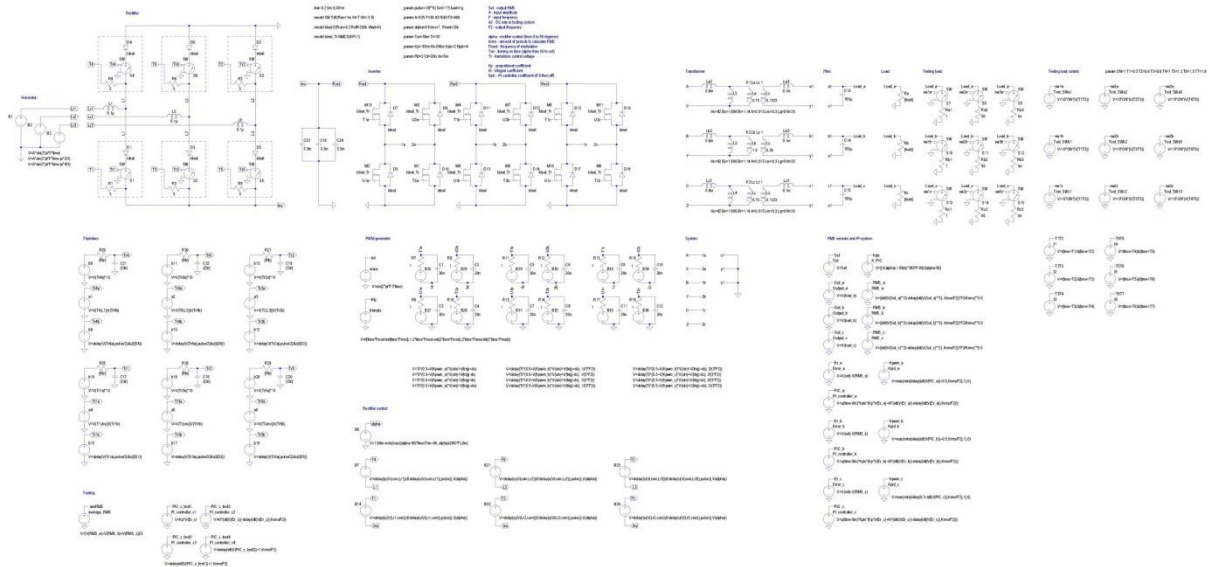


Рис.4.0.1 Общая схема разработанных моделей преобразователя, системы управления и тестирования. Эта же схема находится в приложении 1.

На рис.4.0.1 представлена заключительная схема системы, она же представлена в Приложении 1. Более детально каждый модуль показан в следующих главах. Как видно, крупные под-системы, такие как выпрямитель, инвертор, трансформатор и другие выделены отдельно. Связаны они потенциалами - если задать потенциалам двух разных участков цепи одинаковое название, то программа будет воспринимать их как единый потенциал, т.е. как соединенные проводом без какого либо сопротивления. Это позволяет очень легко манипулировать соединениями модулей. Так же можно заметить, что в системе каждая схема имеет соединение с нулевым потенциалом - это требование программы, “висящие в воздухе” схемы при симуляции будут выдавать ошибку. В верхней части системы находятся “директивы”. Это строки .param, определяющие все константы в системе и краткие текстовые пояснения к названиям констант. Так же там есть строки .model - это такие директивы, которые определяют модели сложных компонентов вроде транзисторов или диодов. Изменив эту строку можно легко менять модели аналогичных

компонентов сразу во всей системе. Строка `.tran` включает в себе параметры симуляции, такие как время начала записи, время конца, разрешение симуляции (т.е. шаг вычислений) и другие. Для разных целей эти строки могут быть как дополнены, так и заменены. Каждому модулю посвящена отдельная под-глава.

## 4.1 Разработка модели сетевого источника напряжения

Задача заключалась в разработке модели трехфазного источника сетевого напряжения. Пользователь может управлять такими величинами как амплитуда и частота. Так же, опционально можно добавить возможность управления помехами и фазовым сдвигом, однако первое сильно снижает скорость симуляции, а второе обычно не востребовано.

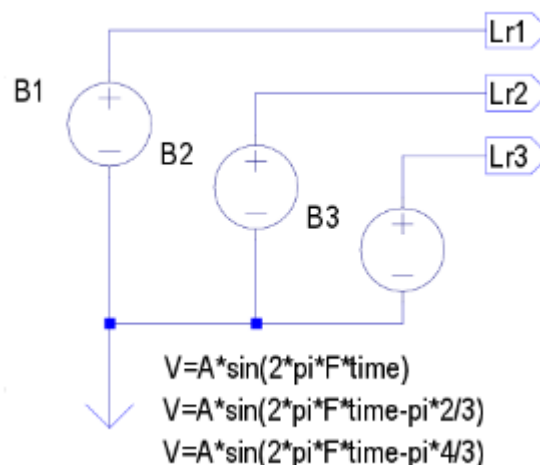


Рис.4.1.1 Источники, имитирующие трехфазное сетевое напряжение.

Для реализации возьмем три источника напряжения и соединим их “звездой”, как изображено на рис. 4.1.1. Треугольничек в соединении представляет собой нейтраль. Источники имеют следующие функции:

$$B1: V(Lr1)=A*\sin(2*\pi*F*time)$$

$$B2: V(Lr2)=A*\sin(2*\pi*F*time-\pi*2/3)$$

$$B3: V(Lr3)=A*\sin(2*\pi*F*time-\pi*4/3)$$

Как видно “ $\pi*2/3$ ” и “ $\pi*4/3$ ” - просто реализуют фазовый сдвиг на 120 градусов, причем программа воспринимает только радианы. А и F - это амплитуда и частота

соответственно. Они определены в строке `.param` и имеют значения  $A=325$  и  $F=50$ , т.е. амплитуда в корень из двух раз больше действующего напряжения между фазой и нейтралью, т.е.  $230 \cdot 1,41 = 325\text{В}$ ; частота равна 50Гц, что является обычной частотой в сети. Если необходимо усложнить функцию помехами и управляемым фазовым сдвигом, то функции примут следующий вид (рис.4.1.2 и рис.4.1.3):

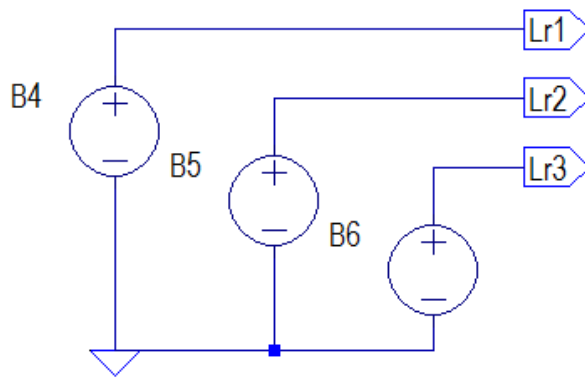
$$B1: V(Lr1)=(1+Noise*white(10000*time))*A*\sin(2*pi*F*time+(2*pi*Ph)/360)$$

$$B2: V(Lr2)=(1+Noise*white(9000*time+12341))*A*\sin(2*pi*F*time- pi*2/3+(2*pi*Ph)/360)$$

$$B3: V(Lr3)=(1+Noise*white(11000*time+4321))*A*\sin(2*pi*F*time- pi*4/3+(2*pi*Ph)/360)$$

На счет фазового сдвига все просто:  $(2*pi*Ph)/360$  это перевод Ph, величины фазового сдвига, из градусов в радианы.

Множитель  $(1+Noise*white(10000*time))$  вносит случайные помехи в синусоиду. Функция `white()` возвращает произвольное число от -0.5 до 0.5 зависящее от целого значения переменной. Поэтому переменная представлена как большое число умноженное на время - это позволяет функции примерно каждые 100мкс выдавать произвольное значение, увеличив это число, увеличим и частоту изменения функции. Для того что создать видимость независимости шума на фазах этот коэффициент слегка различается и, к тому же, добавляется произвольный сдвиг в виде произвольного целого числа. Коэффициент `Noise` задается в параметрах и определяет влияние шума на синусоиду. При `Noise=1` амплитуда множителя шума будет колебаться от 0.5 до 1.5, значит, при `Noise=0.1` шум будет составлять не более 5% синусоиды. При `Noise=0` эта дополнительная функция обратится в единицу и никак не будет влиять на синусоиду. Конечно, вносить помехи можно совершенно по-разному.



$$V=(1+\text{Noise}*\text{white}(10000*\text{time})) * A * \sin(2*\pi * F * \text{time} + (2*\pi * Ph)/360)$$

$$V=(1+\text{Noise}*\text{white}(9000*\text{time}+12341)) * A * \sin(2*\pi * F * \text{time} - \pi * 2/3 + (2*\pi * Ph)/360)$$

$$V=(1+\text{Noise}*\text{white}(11000*\text{time}+4321)) * A * \sin(2*\pi * F * \text{time} - \pi * 4/3 + (2*\pi * Ph)/360)$$

```
.param Noise=1 F=50 A=325 Ph=20
.tran 0 100m 0 0.001m
```

Рис.4.1.2 Схема источников с учетом помех.

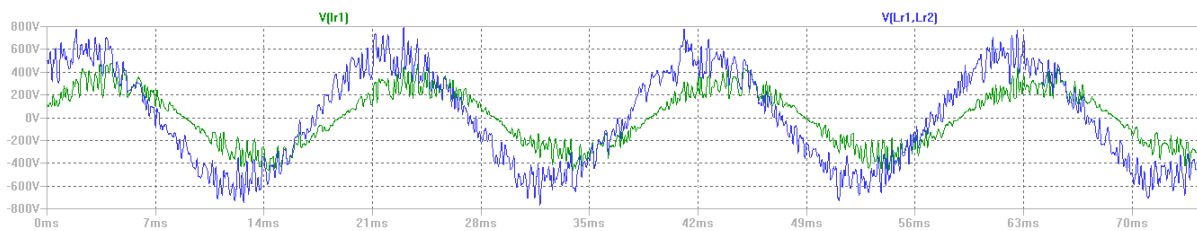


Рис.4.1.3 Симуляция помех: зеленое – фазное напряжение, синее – линейное.

Можно добавлять конкретные гармоники, можно манипулировать ими, да и чем угодно благодаря функциям `rand()` и `white()`, первая, кстати, отличается от второй тем что принимает значение от нуля до единицы. Полный перечень доступных функций представлен в Приложении 2. Если необходимо “сгладить” шумы, то можно усложнить схему конденсаторами, однако все это значительно повышает время симуляции, так что в окончательной модели было принято решение отказаться от шумов в источнике.

## 4.2 Разработка модели выпрямителя

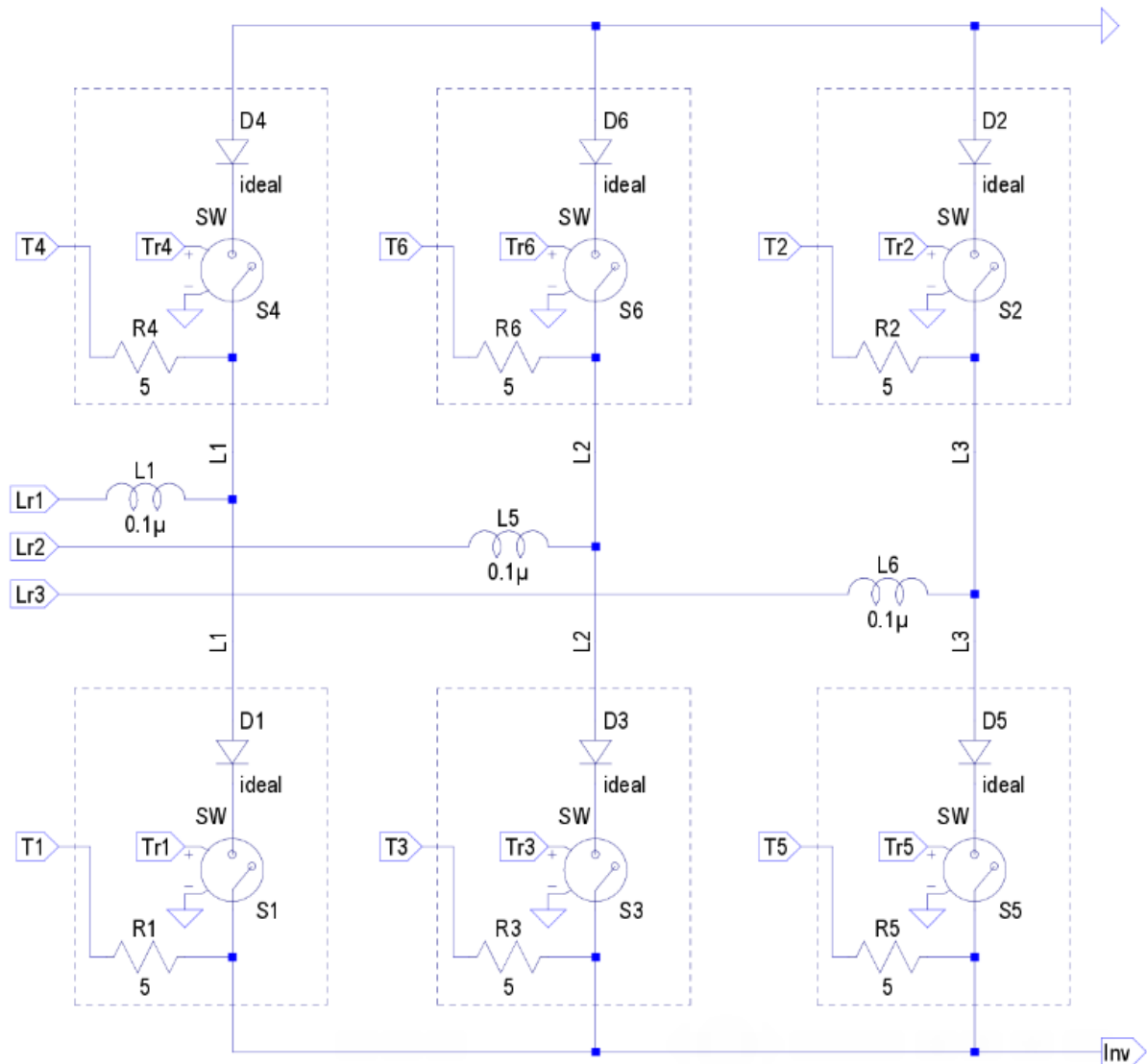


Рис.4.2.1 Модель трехфазного выпрямителя с дросселями на входе.

Сам выпрямитель представляет собой простую трехфазную тиристорную схему моста Ларионова (рис. 4.2.1). Каждый тиристор отпирается подачей определенного напряжения между электродом и катодом. Управление сводится к двум вещам: синхронизация с фазой сетевого напряжения и возможность управления углом отпирания тиристора. В данном случае представлен “мягкий старт”, когда при включении угол отпирания за выбранный пользователем промежуток времени проходит значения от девяноста до нуля градусов (рис. 4.2.2).



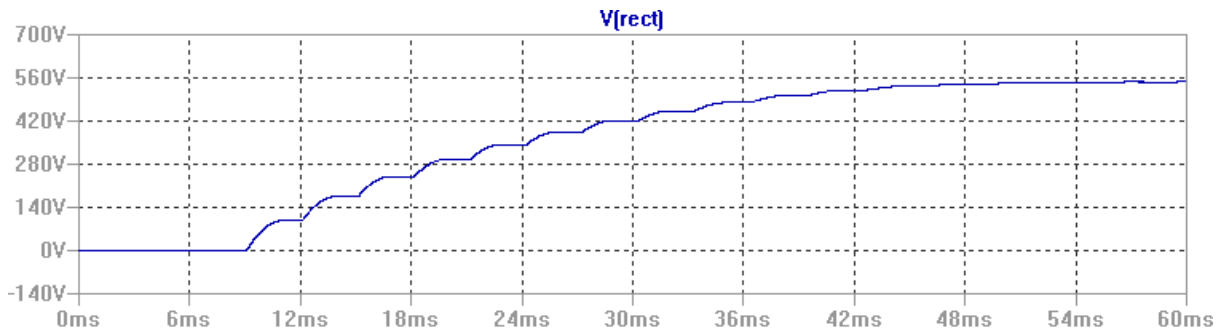


Рис. 4.2.2 Симуляция плавного пуска выпрямителя.

Синхронизация реализована с помощью следующих элементарных функций:

Функция  $u(x)$  - принимает значение 1 если  $x$  больше нуля, в противном случае принимает значение 0.

Функция  $\text{delay}(x,y)$ , принимает значение  $x$  с задержкой  $y$ .

А также операторов И и НЕ:  $\&$  и  $!$  соответственно.



Рис. 4.2.3 Синхронизация и управление тиристорами.

Генератором сигналов управления тиристорами служит функциональный источник (рис. 4.2.3) со следующей функцией:

$$V = \text{delay}((u(V(Lr1,com)) \& !\text{delay}(u(V(Lr1,com))), \text{pulse})), V(\alpha)).$$

Переменных тут две -  $V(Lr1, com)$  - это напряжение между первой фазой сети и нейтралью, и  $V(\alpha)$  - функция, которая изменяется от 90 градусов до установленного значения  $\alpha$  за установленное время плавного пуска  $T_{on}$ . Если необходимость в плавном включении отсутствует, то  $T_{on}$  приравнивается к нулю. Так же тут одна константа, определяемая пользователем:  $\text{pulse}$  - это длительность управляющего импульса. По умолчанию в строке параметра указано, что  $\text{pulse} = 1/(F*6)$ , т.е.  $1/6$  периода. Это значит, что пользователь может выставить лишь значение частоты сети и не беспокоиться о длительности импульса.

Рассмотрим работу данной функции по шагам:

- 1) Определяется значение напряжение фазы  $V(Lr1, com)$ .
- 2) Если оно положительное, то  $u(V(Lr1,com))$  принимает значение один, в противном случае ноль. Это создает из синусоидальной  $V(Lr1,com)$  импульсную функцию с тем же периодом.
- 3) Функция  $delay(u(V(Lr1,com)),pulse)$  сдвигает предыдущую на величину  $pulse$ , которая по умолчанию равна шестой части периода сетевого напряжения. Это как раз время работы каждого тиристора.
- 4) Берется отрицание функции  $delay()$ , что создает функцию  $!delay(u(V(Lr1,com)),pulse)$ . Т.е. просто значения импульсов инвертируются - там где был ноль будет единица и наоборот.
- 5) Импульсы пункта 2 и 4 накладываются с пересечением (оператор И (&)) принимает значение 1 только когда обе функции логически равны единице, в противном случае будет 0) и образуют функцию  $u(V(Lr1,com))\&!delay(u(V(Lr1,com)),pulse)$ . Она представляет собой импульсы которые начинаются в момент перехода напряжения фазы в положительную сторону и заканчиваются через время  $pulse$ .
- 6) Теперь к этим импульсам применяется функция  $delay()$ , которая сдвигает импульсы на время  $V(alpha)$ , которое задается другим источником, управляющим плавным запуском.

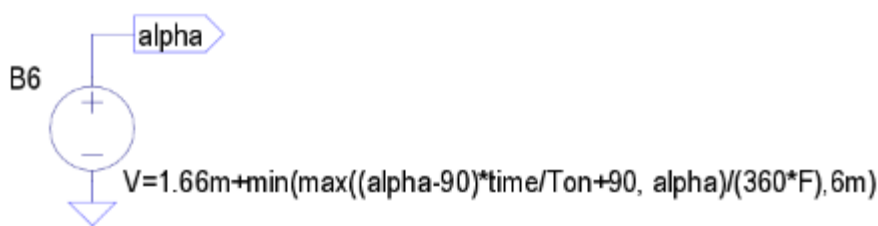


Рис.4.2.4 Функция управления углом отпирания тиристорюв.

Источник, определяющий потенциал  $V(alpha)$  имеет следующую функцию (рис.4.2.4):  $V=1.66m+min(max((alpha-90)*time/Ton+90, alpha)/(360*F),6m)$ . Значения этой функции определяют угол открытия тиристорюв. Тут использованы две элементарные функции:

Функция  $\min(x,y)$  принимает то значение  $x$  или  $y$ , которое в данный момент меньше.

Функция  $\max(x,y)$  принимает то значение  $x$  или  $y$ , которое в данный момент больше.

Эти функции не позволят значению угла выйти за пределы  $0...90$  градусов. В остальном это обычная линейная функция, которая достигает значения  $\alpha$  за время  $T_{on}$ . Оба этих значений определяются в строке `.param`.

В результате эти два источника создают управляемые импульсы для отпирания тиристорov, полностью синхронные с напряжением сети.

### **4.3 Разработка модели тиристорov**

К сожалению, во встроенной библиотеке отсутствует рабочая модель тиристора. Модели из сторонних библиотек, которых, к слову, очень мало, значительно повышали время симуляции, а так же в большинстве алгоритмы были скрыты. Поэтому было необходимо разработать собственную модель тиристора. Для начала нужно определить основные свойства тиристора. В нашем случае тиристор будет выступать в роли ключа. Подача слабого напряжения будет его отпирать и функционировать он будет как обычный диод. Когда напряжение на нем упадет до нуля, тиристор автоматически закрывается.

Реализовать такую модель можно просто последовательно соединив диод и переключатель, который будет отпираться при подаче напряжения между контактом-электродом и катодом. Вся сложность заключается в том, как реализовать этот переключатель (рис.4.3.1).

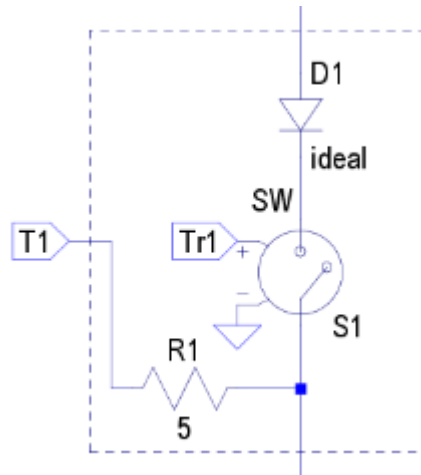


Рис.4.3.1 Разработанная схема тиристора.

В стандартной библиотеке есть ключ, управляемый напряжением. Можно задать для него множество параметров: пороговое напряжение ( $V_t$ ), напряжение гистерезиса ( $V_h$ ), сопротивление ключа в замкнутом состоянии ( $R_{on}$ ), сопротивление ключа в разомкнутом состоянии ( $R_{off}$ ), последовательная индуктивность ( $L_{ser}$ ), последовательное напряжение ( $V_{ser}$ ) и предельный ток ключа ( $I_{limit}$ ). Все эти параметры можно задать в SPICE-директиве. В данной работе эта строчка имеет следующий вид: `.model SW SW(Ron=1m Vt=7 Vh=-3.5)`. Разберем это подробнее:

- 1) `.model` - директива указывающая на описание компонента (не все компоненты требуют отдельной директивы)
- 2) `SW` - модель компонента - это название указывается в поле `value` (значение) в параметрах ключа. Название может быть любым. Но оно должно совпадать с тем, что написано в директиве.
- 3) `SW()` - тип компонента для данной директивы - `SW` от `switch` - это ключ, управляемый напряжением. В скобках указываются параметры ключа.
- 4) `Ron=1m` - сопротивление в открытом состоянии равно одному миллиому.
- 5) `Vt=7` - напряжение открытия ключа равно семи вольтам.
- 6) `Vh=-3.5` - напряжение гистерезиса равно минус трем с половиной вольтам.

Это значит, что триггер будет плавно изменять сопротивление между  $V_t - V_h$  и  $V_t + V_h$ , т.е. от 3.5В до 10.5В. Если бы значение равнялось бы нулю, то переключение было бы мгновенным, а если бы было положительным, то ключ управлялся бы с гистерезисом, т.к. если бы он управлялся от триггера Шмитта с порогами переключения  $V_t - V_h$  и  $V_t + V_h$ .

Это значит, что управление выключателем может происходить функциональным источником напряжения, который будет выдавать сигналы следующим образом: когда сигнал генератора управления тиристорами принимает значение 1, то данный источник внутри модели тиристора примет значение открытия ключа (в данном случае это 10В). Если управляющий импульс примет значение ноль до того как напряжение на тиристоре не опустится до нуля, то напряжение источника управления ключом останется на прежней величине. Когда напряжение на тиристоре достигает нуля, то напряжение источника станет равной нулю. При отрицательном напряжении на тиристоре источник управления ключом сохраняет значение ноль независимо от управляющего тиристором источника. Если абстрагироваться от источников, то нам нужно получить такой сигнал, которым можно управлять другими сигналами, а именно их передними фронтами. В обычной электронике и программировании это триггеры, умеющие сохранять свои значения. В LT-SPICE IV простых программных способов сохранять значения нету, т.к. все математические функции зависят, по сути, только от времени.

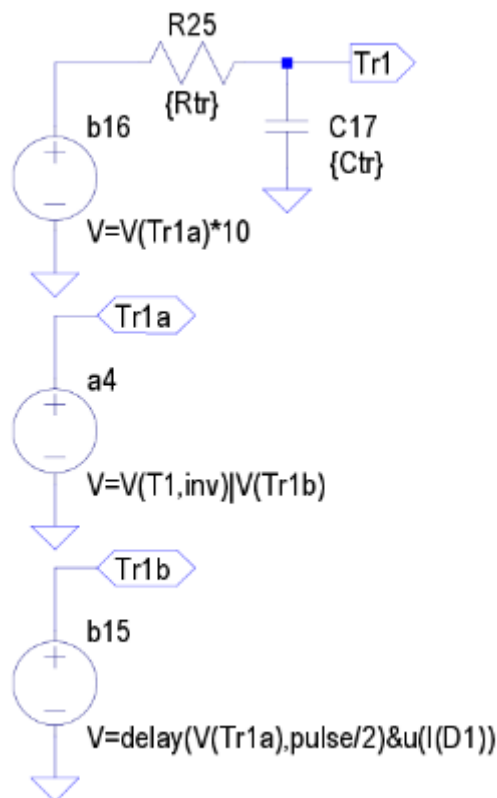


Рис.4.3.2 Функции управления ключом в модели тиристора.

Для решения этой проблемы автором был разработан алгоритм, основанный на обратной связи. Создадим два источника (рис.4.3.2).

Функция первого  $V(Tr1a) : V=V(T1,inv)/V(Tr1b)$ , функция второго  $V(Tr1b): V=delay(V(Tr1a),pulse/2)\&u(I(D1))$ .

$V(T1,inv)$  - импульсы управления первым тиристором, другими словами напряжение между электродом и катодом первого тиристора. Оператор  $|$  означает ИЛИ, т.е. если хотя бы одна из функций примет значение 1, то оператор тоже даст 1. И так.

- 1) Когда импульс поступает на электрод тиристора, первый источник принимает значение 1.
- 2) Второй источник принимает значение 1 вслед за первым со сдвигом величиной  $pulse/2$ . Однако из-за присутствия оператора И (&) чтобы источник принял значение 1, ток на диоде должен быть положительным.
- 3) Если управляющий импульс пропадает, то первый источник все еще сохранит значение один, т.к. второй источник все еще будет передавать 1, ибо сдвинут на половину величины самого импульса управления.
- 4) Если ток на диоде прекратится, то второй источник примет значение ноль. В таком случае без управляющего сигнала первый источник так же примет значение 0.

Как раз функция первого источника и будет определять состояние ключа. Но этого недостаточно. В таком виде все переключения будут носить мгновенных характер, что во-первых нереалистично, и, во-вторых, из-за паразитных параметров может вызвать значительные скачки напряжения или тока. Для решения этой проблемы можно создать третий источник (рис.4.3.2 – источник b16) подключенный к простому LC-фильтру, который немного сгладит импульсы. Функция такого источника это просто умноженное в десять раз значение первого источника, т.е. выдавать он будет 10В на ключ, управляемый напряжением.

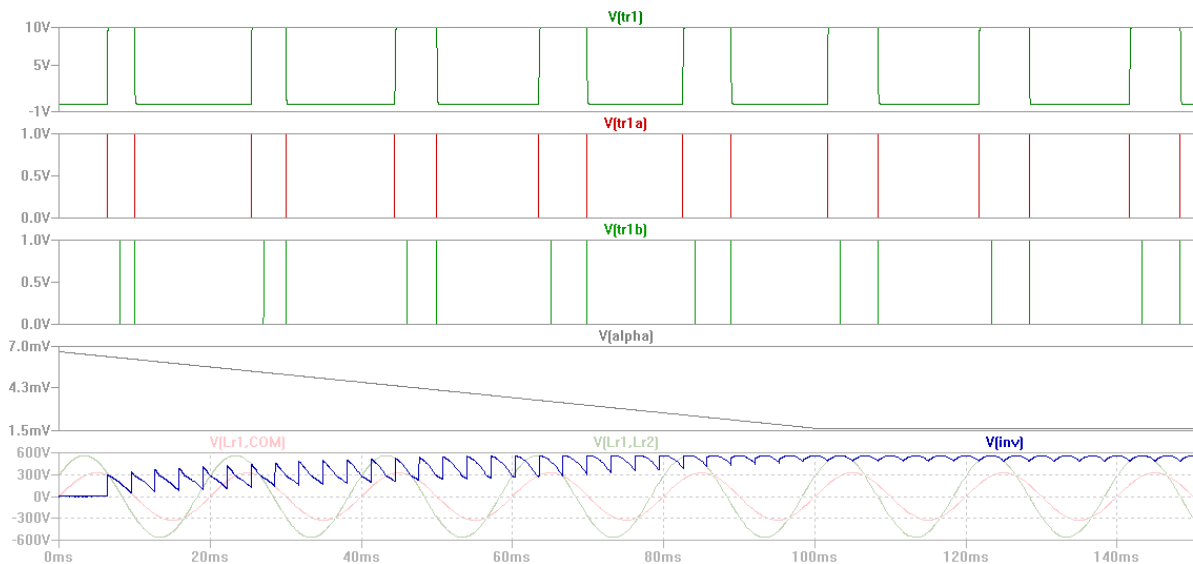


Рис.4.3.3 Снизу (синим) изображено выпрямленное напряжение без сглаживающего диода. Выше (серым) - функция угла отпирания тиристоров. Верхние три графика соответствуют импульсам источников на рис.4.3.2.

Таким образом, разработанная модель тиристора представляет собой три источника, ключ, диод, емкость и два резистора. А для управления тиристорами нужно по одному источнику на каждого и один источник, управляющий углом отпирания тиристоров, на всех. Конечно же, пользователь может выбирать, какую модель тиристора ему использовать.

#### 4.4 Сглаживающий фильтр выпрямителя

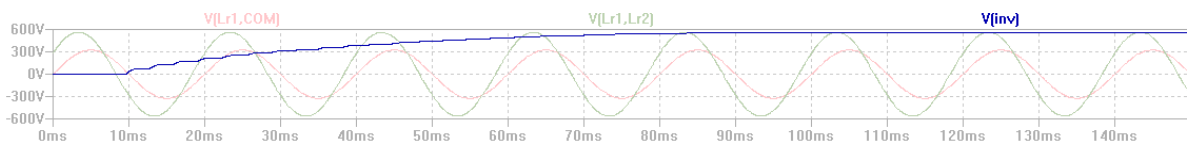


Рис.4.4.1 В отличие от рис.4.3.3 тут установлен сглаживающий фильтр.

Фильтр выпрямителя достаточно прост - это всего лишь один или несколько параллельных конденсаторов. Одна из его задач - это сгладить импульсы после выпрямителя (рис.4.4.1), однако т.к. выпрямитель собран в виде моста Ларионова - относительная амплитуда пульсаций не превышает 14%. Вторая задача емкости - это не допустить избытка зарядов с инверторов на выпрямитель, т.е. защитная функция. Так же для некоторых задач бывает важно учитывать паразитные параметры конденсаторов. Для этого достаточно указать эти параметры в опциях конденсатора:

$C$  - емкость,  $R_{ser}$  - эквивалентное последовательное сопротивление,  $L_{ser}$  - эквивалентная последовательная индуктивность,  $R_{par}$  - эквивалентное параллельное сопротивление,  $C_{par}$  - эквивалентная параллельная емкость и некоторые другие. В данном случае указана только емкость в 10 мОм

## 4.5 Разработка модели мостового инвертора напряжения

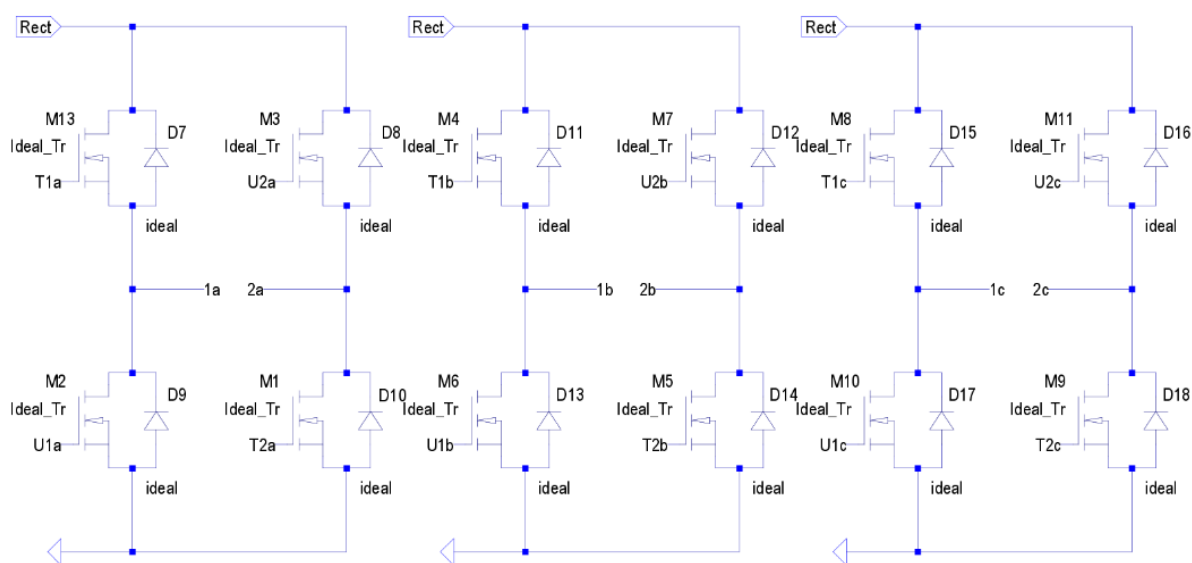


Рис.4.5.1 Три независимых однофазных мостовых инвертора напряжения.

В данной работе в угоду стабильности системы и определенной независимости фаз использованы три однофазных инвертора, подключенных параллельно (рис.4.5.1). Каждый из них представлен в виде простой мостовой схемы с ключами в виде транзисторов и обратно направленными диодами. Очевидно, что каждый транзистор работает лишь в одном направлении, поэтому чтобы защитить систему, установлены диоды, выпрямляющие возможное обратное напряжение. Если бы не было транзисторов, то диоды образовали бы обычный диодный мост. И так, в библиотеке программы огромное количество диодов и различных транзисторов. Присваиваются они через директиву `.model`, как и ключи. Конечно, модель можно выбрать и из списка предложенных, но, к сожалению, использование сложных моделей значительно увеличивает время симуляции, поэтому в данной работе используется достаточно простая модель диода.



В программе можно выбирать из нескольких типов диодов: DIODE - обычный диод, ZENER - стабилитрон, SHOTTKY - диод Шоттки и VARACTOR - варикап. Разница между ними лишь во внешнем виде элемента, все они основаны на одной модели. Параметров у диода достаточно много. Директива, описывающая модель используемого диода - *.model Ideal D(Ron=0.2 Roff=250k Vfwd=0)*.

- 1) Ideal - название модели, его может выбирать пользователь.
- 2) D() - тип элемента: диод.
- 3) Ron - сопротивление прямой проводимости, в данном случае 0.2Ом, по умолчанию, если не указывать значение, программа будет считать 1Ом.
- 4) Roff - сопротивление в запертом состоянии, в данном случае 250кОм; по умолчанию число обратное самому малому в программе.
- 5) Vfwd - пороговое напряжение прямой проводимости. Тут как и по умолчанию 0В.

С транзисторами ситуация немного отличается - действующие модели IGBT транзисторов, которые часто применяются в таких инверторах, не позволяют закончить симуляцию в приемлемые сроки. Однако настраиваемые параметры IGBT и MOSFET транзисторов очень похожи, так что в первом приближении возможно обойтись удобными моделями MOSFET. Параметров транзисторов в программе огромное количество. Директива, описывающая модель используемого транзистора: *.model Ideal\_Tr NMOS(KP=1)*

- 1) Ideal\_Tr - название модели.
- 2) NMOS - тип элемента: монолитный n-канальный MOSFET.
- 3) Kp - параметр межэлектродной проводимости.

Для реалистичной симуляции можно подобрать модели транзисторов и диодов соответствующие действительным компонентам, однако для этого потребуется действительно мощный компьютер, т.к. время симуляции значительно увеличится.

О том, как управлять транзисторами будет рассказано позднее в разделе управление инвертором.

## 4.6 Разработка модели трансформатора напряжения

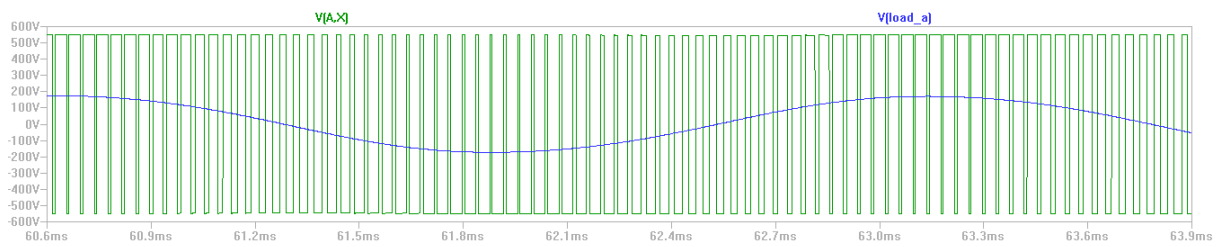


Рис.4.6.1 Импульсы на входе трансформатора (зеленые) и синусоида на выходном конденсаторе (синяя).

На выходе импульсного инвертора мы получаем импульсы примерно с одним верхним напряжением и разной шириной (рис.4.6.1). В таком виде закодирован тот вид кривой напряжения, который требуется, обычно это синусоида. Для того чтобы воссоздать закодированную синусоиду нужно пропустить импульсы через LC-фильтр. Но в данной работе рассматривалось иное возможное техническое решение, а именно подсоединение трансформатора на выход инвертора для того чтобы использовать индуктивность рассеяния в качестве дросселя фильтра. Это нестандартное решение, которое помогло бы сэкономить на дорогостоящих дросселях. Именно для разработки подобных нестандартных решений возникает необходимость в своевременном моделировании системы.

Разработка модели трансформатора в данной системе является самой трудной. В процессе исследования было найдено несколько возможных решений, учитывающих нелинейность трансформаторов, однако одно из них попросту не смогло запуститься, похоже, в виду ограниченности ресурсов компьютера. Другим препятствием в решении данного вопроса является отсутствие достаточных данных о трансформаторе. Такие данные как индуктивность рассеяния редко предоставляются производителем, не говоря уже о таких параметрах сердечника, как коэрцитивная сила и остаточная индукция.

Реализация данной модели основана на трех однофазных нелинейных моделях трансформаторов с параметрами реального трехфазного Ш-образного с соединениями треугольник-звезда. Рассмотрим один из них (рис.4.6.2):

## Transformer:

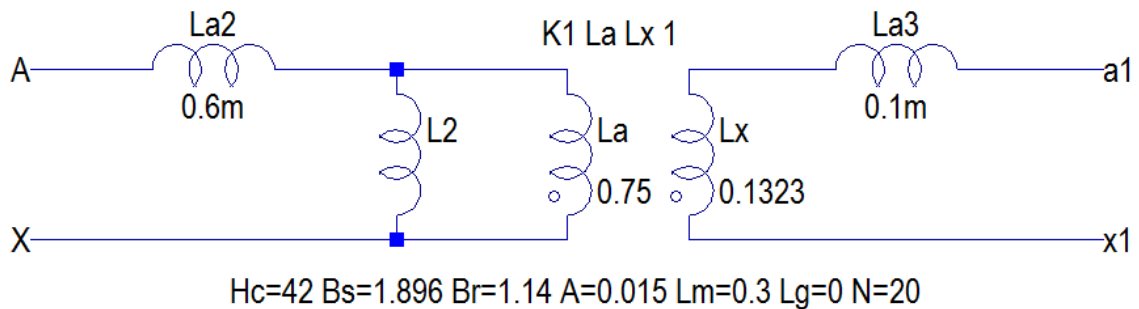


Рис.4.6.2. Модель однофазного нелинейного трансформатора.

- 1)  $La_2$  и  $La_3$  - представляют собой индуктивности рассеяния.
- 2)  $L_2$  - нелинейный элемент, имитирующий нелинейное намагничивание первичной обмотки. Его параметры описаны в строчке под моделью, и вводятся в поле inductance:
  - a)  $H_c$  - коэрцитивная сила - это значение напряжённости магнитного поля, необходимое для полного размагничивания ферро- или ферромагнитного вещества.
  - b)  $B_s$  - индукция насыщения - это максимальная индукция, которую можно получить в данном магнитном материале при данной температуре.
  - c)  $B_r$  - остаточная индукция - это намагниченность, которую имеет ферромагнитный материал при напряжённости внешнего магнитного поля, равного нулю.
  - d)  $L_m$  - средняя длина магнитной линии за вычетом немагнитного зазора.
  - e)  $L_g$  - величина немагнитного зазора.
  - f)  $A$  - сечение сердечника.
  - g)  $N$  - количество витков.
- 3)  $La$  и  $Lx$  - индуктивности первичной и вторичной обмоток.
- 4) Директива над моделью -  $K_1 La Lx 1$  означает, что связь  $K_1$  между  $La$  и  $Lx$  стопроцентная. Для корректной работы нелинейного трансформатора  $K$  должен быть равным одному.

Как видно, данная модель основана на распространенной Т-образной линейной модели. Определение параметров может производиться с помощью опытов короткого замыкания КЗ и холостого хода ХХ. Например, суммарную индуктивность рассеяния, приведенную к первичной обмотке, можно определить, разделив напряжение КЗ на первичной обмотке на произведение тока КЗ на первичной обмотке и угловой частоты переменного напряжения. В среднем можно считать, что индуктивность рассеяния приблизительно поровну распределена между обмотками, поэтому индуктивность рассеяния первичной обмотки будет в два раза меньше общей индуктивности рассеяния приведенной к первичной обмотке. Значит индуктивность рассеяния вторичной обмотки трансформатора будет в квадратный корень из коэффициента трансформации раз меньше. Конечно, это действительно при достаточно малом активном сопротивлении в опыте КЗ, в противном случае необходимо использовать ваттметр и по теореме Пифагора находить реактивную составляющую. Индуктивности обмоток находятся аналогично из опыта ХХ.

Такие параметры как  $L_g$ ,  $L_m$ ,  $A$  и  $N$  находятся либо по паспорту трансформатора, либо простыми замерами. С магнитными свойствами сложнее, т.к. опыты по их измерению сложны, и в паспорте они редко указываются. Однако зная тип сердечника можно найти свойства материала в справочнике. Именно таким образом были выбраны параметры, используемые в данной работе.

#### 4.7 Модель фильтра на выходе трансформатора

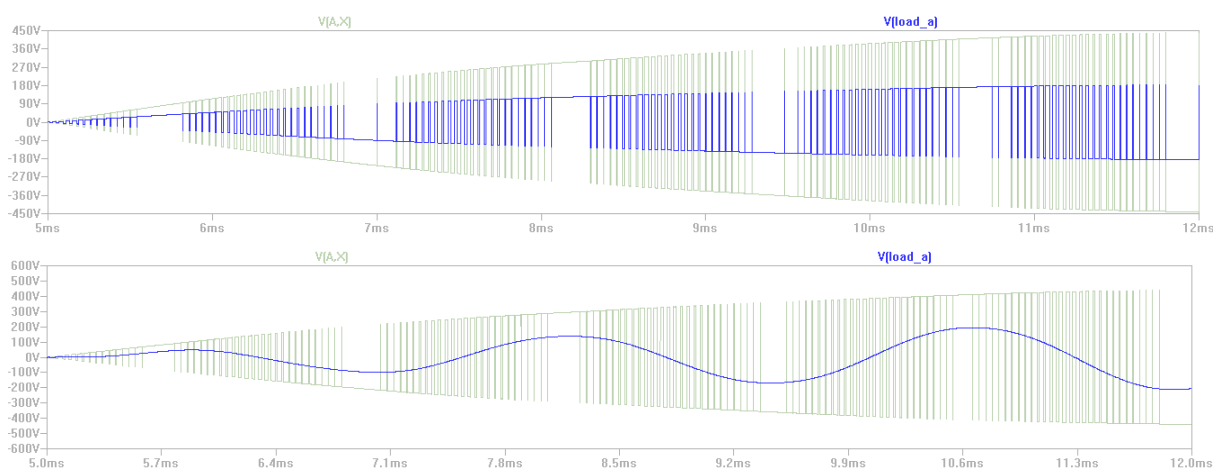


Рис.4.7.1 Выход трансформатора (синий) без фильтра сверху, с фильтром снизу.

Этот фильтр предназначен для сглаживания импульсов до состояния, близким к синусоиде (рис.4.7.1). Для частоты модуляции в 25кГц конденсаторы на 100мкФ вполне справляются. Соединение этих конденсаторов имеет вид звезды, как и параллельная им нагрузка. В стандартном случае фильтр представляет собой LC-фильтр, и стоит на выходе инвертора, т.е. на входе трансформатора.

#### 4.8 Разработка модели нагрузки преобразователя частоты

В простом случае нагрузка преобразователя - это сопротивление (рис.4.8.1). При мощности 250кВА и напряжении на выходе 115В сопротивление должно составить  $115^2/(250000/3^{1/2})=0.0916\text{Ом}$  или примерно 92мОм. Если учесть активно-индуктивный характер нагрузки, то можно рассчитать величину индуктивности нагрузки, последовательной резисторам.  $200\text{кВТ}/250\text{кВА}=\sin(a)=0.8$ , следовательно,  $\cos(a)=(1-0.8^2)^{1/2} = 0.6$ . В таком случае активное сопротивление на каждой фазе составит  $115^2/(200000/3^{1/2})=0.115\text{Ом}$  или 115мОм. А реактивное сопротивление составит  $115^2/(250000\cdot 0.6/3^{1/2})=0.153\text{Ом}$ , значит, индуктивность составит  $0.153/(2\cdot\pi\cdot 400)=60\text{мкГн}$ .

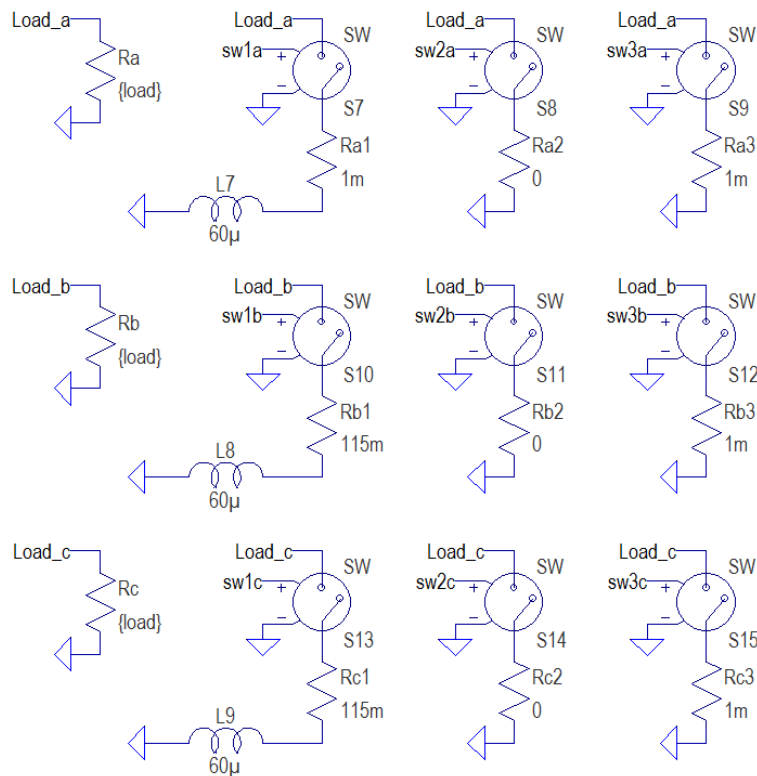


Рис.4.8.1 Слева - базовая нагрузка, три колонки справа - это нагрузки, подключаемые во время симуляции для проведения испытаний.

На рис.4.8.1. показана нагрузка в разработанной модели преобразователя частоты. В данной модели учтена возможность переключения нагрузки во время симуляции, причем независимо для каждой фазы. Как видно, такие нагрузки управляются ключом, управляемым напряжением. Далее на рис.4.8.2 и рис.4.8.3 показаны источники, генерирующие сигналы управления, в зависимости от настроек.

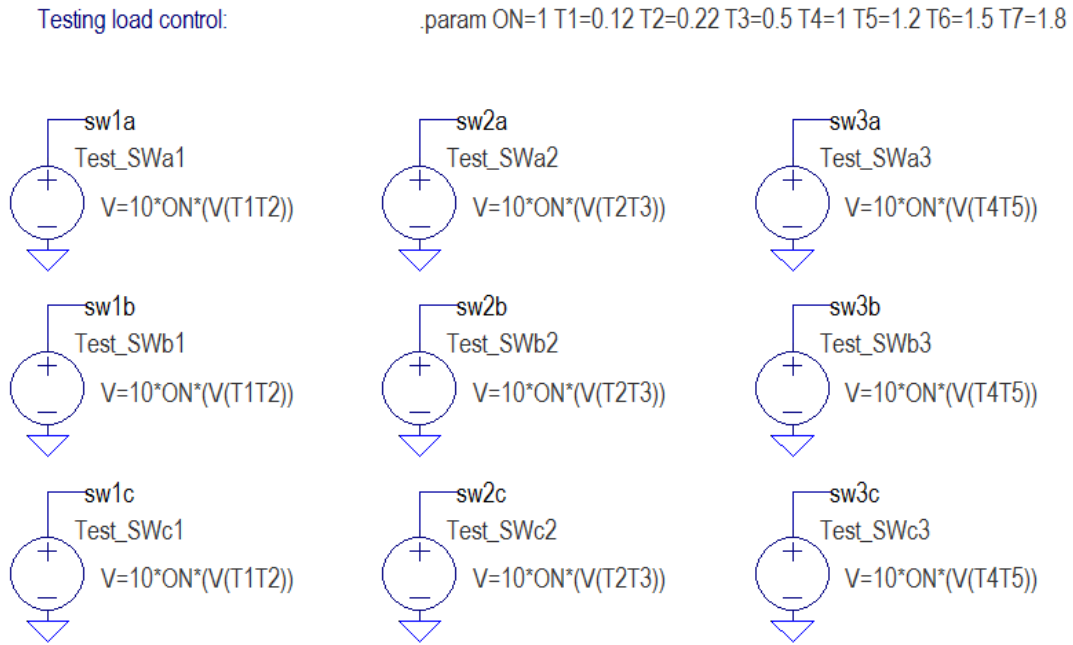


Рис.4.8.2 Источники, управляющие подключением и отключением нагрузки, сверху особая строка `.param`, в которой занесены моменты переключения нагрузки.

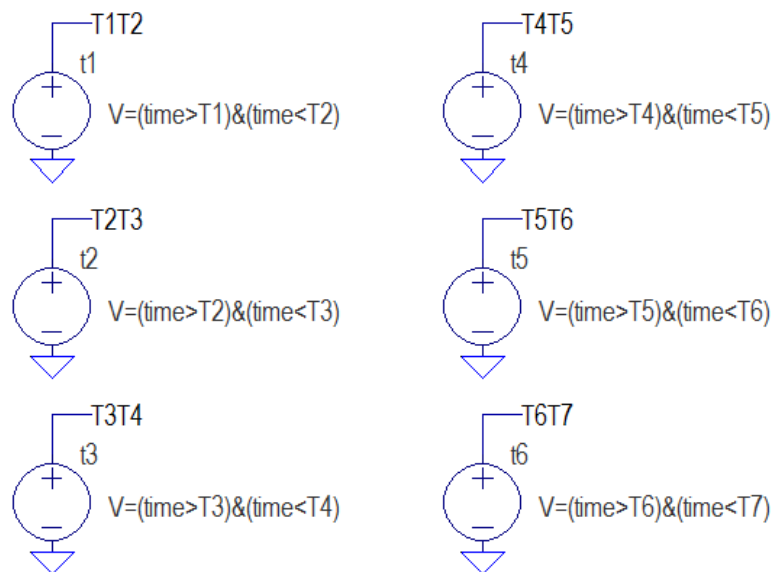


Рис.4.8.3 Промежуточные функции для системы управления нагрузкой.

Автором был разработан несложный алгоритм подключения и отключения нагрузки. Пользователь задает в директиве .param значение для следующих констант:

ON – если параметр равен единице, то алгоритм переключения нагрузки включается, если параметр равен нулю, то последующие параметры влияния на симуляцию не оказывают.

$T_1, T_2, \dots, T_7$  – время переключения нагрузки. Источники, изображенные на рис.4.8.3 генерируют импульсы между соседними значениями. Например если  $T_1 = 2\text{с}$ , а  $T_2 = 2.5\text{с}$ , тогда источник  $t_1$  выдаст потенциал  $V(T_1T_2)$  равный единице в промежутке с  $2\text{с}$  до  $2.5\text{с}$ . В остальное время  $V(T_1T_2) = 0$ .

Таким образом, 6 источников ( $t_1 - t_6$  рис.4.8.3) могут сгенерировать 6 разных промежутков времени, на каждый из которых можно назначить свою нагрузку. Функция этих источников простая, например, для потенциала  $V(T_1T_2)$  функция такова:  $V=(\text{time}>T_1)\&(\text{time}<T_2)$ . Т.е. чтобы функция выдавала единицу, должно быть выполнено одновременно два условия: время больше первого заданного и меньше второго заданного. Потенциал подходящего промежутка времени можно записать в источники, прямо управляющие ключами подключения нагрузки. Они представлены на рис.4.8.2. Вот как могут выглядеть импульсы управления нагрузкой (рис.4.8.4):

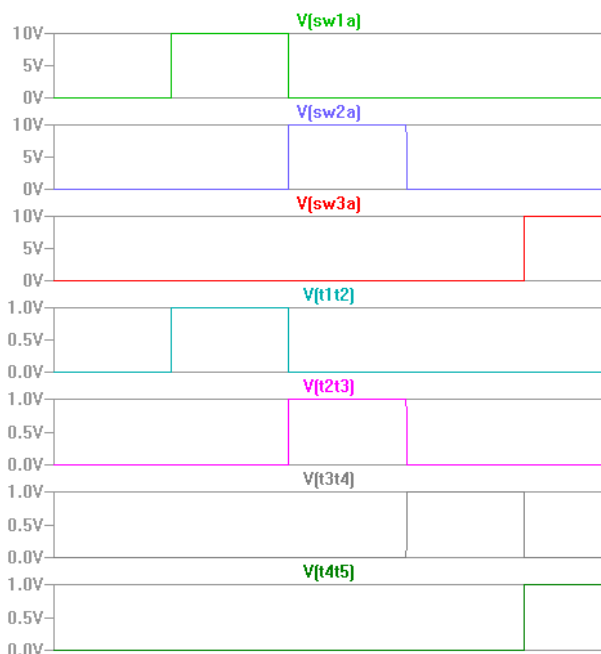


Рис.4.8.4 Импульсы управления нагрузкой первой фазы и импульсы источников  $t_1 - t_4$ .

## 4.9 Разработка систем регуляции и стабилизации

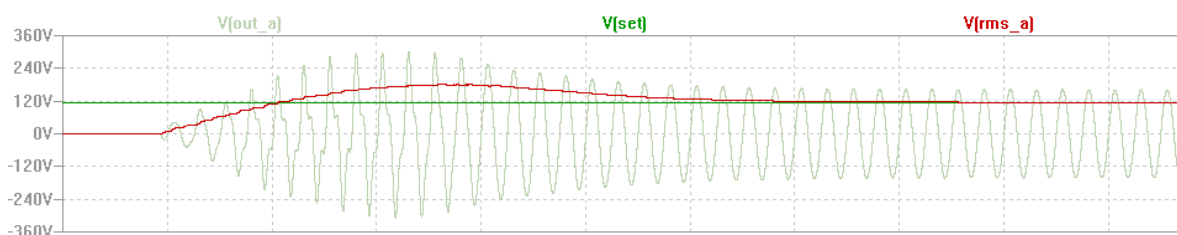


Рис.4.9.1 Серое - напряжение на выходе, зеленое - заданное, красное - эффективное.

Как известно, изменение нагрузки сразу может повлечь изменение напряжения, однако задача преобразователя - поддерживать заданный уровень напряжения в широком диапазоне нагрузок. Это проблему решает пропорционально-интегральное регулирование широтно-импульсной модуляцией. Осуществляется оно за счет обратной связи с выходом и измерением среднеквадратичного значения выходного напряжения. Суть заключается в следующем:

- 1) Датчики фиксируют напряжение на выходе.
- 2) Происходит вычисление эффективного значения (RMS) этого напряжения.
- 3) Сравнивается с заданным значением.
- 4) Разница между этими значениями отправляется в ПИ-регулятор.
- 5) Чем больше разница, тем значительнее сигнал поправки для управления шириной импульсов управления транзисторами в инверторе.

Для большей стабильности процесс корректировки происходит с небольшой задержкой. На рисунке 4.9.1 видно, как система самостоятельно подстраивается под требуемое значение. Искажения выходного напряжения на первых миллисекундах симуляции обусловлено тем, что в процессе плавного пуска напряжение на выходе выпрямителя было недостаточным, чтобы поддерживать соответствующий уровень синусоидального напряжения на выходе. Система может поддерживать требуемый уровень эффективного напряжения на выходе даже при пониженном выпрямленном напряжении, немного жертвуя при этом линейностью. Пользователь, ограничив функцию регулирования широтно-импульсной модуляцией, может запретить появление таких искажений.



Данная глава является важнейшей частью данной работы. Алгоритмы, разработанные здесь, и их общая реализация в среде моделирования LT-SPIICE IV, являются достаточно уникальными.

#### 4.9.1 Разработка алгоритмов обратной связи и расчета RMS

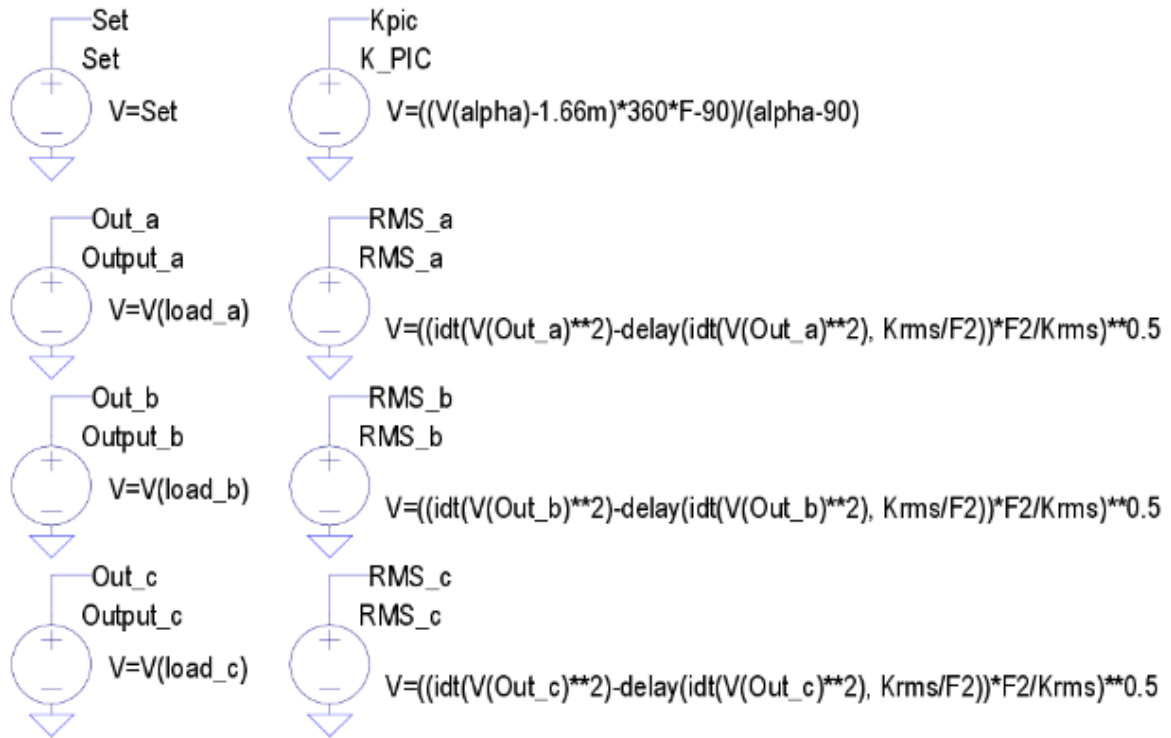


Рис.4.9.1.1. Источники, имитирующие датчики.

Для начала, необходим источник, который будет выдавать требуемое выходное напряжение (рис.4.9.1.1 – источник Set). Лучше использовать источник, а не константу, т.к. может возникнуть необходимость симуляции переходных процессов, при которых требуемое значение будет изменяться во времени. Конечно, в простом случае, функция этого источника  $V=Set$ , где Set - константа, определяемая в директиве .param; в нашем случае Set=115. Потенциал же, который будут принимать другие функции, будет  $V(Set)$ .

Затем, для удобства создадим отдельные источники, имитирующие датчики. Каждый из них обладает следующей простой функцией  $V=V(load\_a)$  и потенциал их  $V(Out\_a)$ . Таких источников три (рис.4.9.1.1 – источники Output\_a, Output\_b и Output\_c) - каждый на свою фазу. Разница у них только в последней букве (она

будет принимать значение  $a$ , как в примере,  $b$  и  $c$ ). Это удобно и наглядно. В случае если пользователь захочет изменить объект регулирования, ему не придется копаться в других функциях. Так же, систему можно усложнить погрешностями датчиков.

Теперь необходимо рассчитать эффективное значение напряжения на нагрузке, известное как среднеквадратичное или RMS (Root Mean Square). Эта величина равняется корню квадратному из частного определенного интеграла квадрата функции и периода интегрирования. Причем необходимо учитывать, что эффективное значение напряжения может изменяться со временем. Это значит, что интегрирование должно захватывать область в один или несколько периодов до текущего момента. Один источник будет измерять RMS своей фазы, т.е. источников три. Функция первого из них имеет следующий вид:

$$V = ((idt(V(Out\_a)**2) - delay(idt(V(Out\_a)**2), Krms/F2)) * F2 / Krms) ** 0.5$$

- 1) Функция  $idt(x)$  - есть ни что иное как интеграл. Он возвращает значение как определенный интеграл от начала симуляции до момента времени  $time$ .
- 2) «\*\*» - оператор возведения в степень.
- 3)  $Krms$  - это количество периодов, которые должны быть проинтегрированы. Чем больше периодов интегрируются тем, во-первых, точнее будет измерение и, во-вторых, медленнее будет происходить подстройка ШИМ.
- 4)  $F2$  - частота модулированной синусоиды, т.е. заданная частота на выходе преобразователя.

Для начала функция берет значение  $V(Out\_a)$  и возводит его в квадрат. Затем интегрирует полученную функцию от начала симуляции до текущего момента. Далее применяется функция  $delay()$ , которая “замораживает” полученное значение интеграла на время  $Krms/F2$ , что есть ничто иное как период умноженное на количество интегрируемых периодов  $Krms$ , другими словами это период интегрирования. После этого от текущего  $idt(V(Out\_a)**2)$  отнимается отложенная функция  $delay(idt(V(Out\_a)**2), Krms/F2)$ . Таким образом мы получаем интеграл лишь на небольшом участке длиною в  $Krms/F2$ . Полученное значение мы делим на участок интегрирования (или умножаем на  $F2/Krms$ ) и в конце возводим в степень

0.5, т.е. извлекаем квадратный корень. Такие алгоритмы вычисления RMS аналогичны для каждой фазы. Потенциал такого источника имеет вид  $V(RMS\_a)$ .

#### 4.9.2 Разработка пропорционально-интегрального регулятора (ПИР)

После того, как система обратной связи рассчитывает эффективное напряжение, для каждого источника рассчитывается величина ошибки, т.е. отклонения от  $V(Set)$ . Такие источники имеют простую функцию  $V=V(set)-V(RMS\_a)$ . Потенциал имеет вид  $V(Er\_a)$ .



Рис.4.9.2.1 Наборы источников-регуляторов на каждую фазу.

Теперь это значение идет в пропорционально-интегральный регулятор (источники  $PI\_controller\_a$ ,  $PI\_controller\_b$  и  $PI\_controller\_c$  на рис. 4.9.2.1). Очевидно из

названия, что пропорционально-интегральный регулятор сочетает в себе полезные качества пропорциональных и интегральных регуляторов, а так же в некоторой степени компенсирует их недостатки. Пропорциональный регулятор хорошо справляется при небольших отклонениях, но может легко стать нестабильным при более резких возмущениях. Интегральный же регулятор хорошо справляется со статическими ошибками, однако это в свою очередь удлиняет время регулирования. В ПИ-регуляторах обратная связь называется упругой. Так же несложно модифицировать функцию, добавив составляющую дифференциального регулятора, который может пытаться устранять предсказуемые отклонения, однако в данной задаче это не являлось необходимым. И так, сигнал ПИ-регулятора является суммой пропорциональной и интегральной составляющих. Пропорциональная составляющая - это функция ошибки  $V(Er\_a)$  умноженная на подобранный коэффициент  $Kp$ , определяемый пользователем. Интегральная составляющая это интеграл ошибки  $V(Er\_a)$  умноженный на коэффициент  $Ki$ . Из-за того, что требования системы могут меняться во время симуляции интеграл будет подобен тому, что был в RMS, т.е. интегрируется функция не с начала запуска, а лишь по нескольким периодам. В итоге функция источника ПИ-регулятора имеет следующий вид:

$$V=u(time-8m)*Kpic*(Kp*V(Er\_a)+Ki*(idt(V(Er\_a))-delay(idt(V(Er\_a)),Krms/F2)))$$

- 1) Сумма составляющих описанная выше:  $Kp*V(Er\_a)+Ki*(idt(V(Er\_a))-delay(idt(V(Er\_a)),Krms/F2))$
- 2) Эта величина умножается еще на коэффициент  $Kpic$  - он нужен только для удобства: что бы изменить общее влияние регулятора, а так же отключить его достаточно установить соответствующее значение  $Kpic$ .
- 3) Все это умножается на  $u(time-8m)$  – это делает влияние регулятора до 8мс нулевым, т.к. в этот период система находится в режиме включения и является нестабильной. Пользователь может, при желании, дополнить эту функцию выбирая, когда и как регулятор будет влиять на систему. Возможно, это будет функция от  $\alpha$ .

Потенциал регулятора имеет вид  $V(PIC\_a)$ . Самое сложное в ПИ-регуляторе - это подобрать коэффициенты. Симуляция на компьютере значительно упрощает эту задачу.

После этого значение  $V(PIC\_a)$  поступает на источник управляющий коэффициентом широтно-импульсной модуляцией. Этот источник имеет функцию  $V=delay(idt(V(PIC\_a))+1,Krms/F2)$ , которая хорошо себя показала в симуляции. Как видно функция интегрирует и откладывает значение  $V(PIC\_a)$ , что повышает стабильность. Потенциал этого источника -  $V(Kpwm\_a)$ . Опять-таки каждой фазе свой источник.

### 4.9.3 Разработка алгоритма широтно-импульсной модуляции (ШИМ)

Теперь рассмотрим заключительный этап управления тиристорами: формирование широтно-импульсной модуляции. Сама идея модуляции очень проста - необходимо сравнить пилообразный сигнал с синусоидальным. Тогда, когда, например, синус больше треугольников - ШИМ принимает значение 1, в противном случае - 0. Трудности заключаются в двух вещах: как создать функцию пилообразного симметричного сигнала (ибо в ней меньше паразитных гармоник по сравнению с ассиметричным) и как разделить эти сигналы по конкретным транзисторам? В данной работе представлено решение данных вопросов.

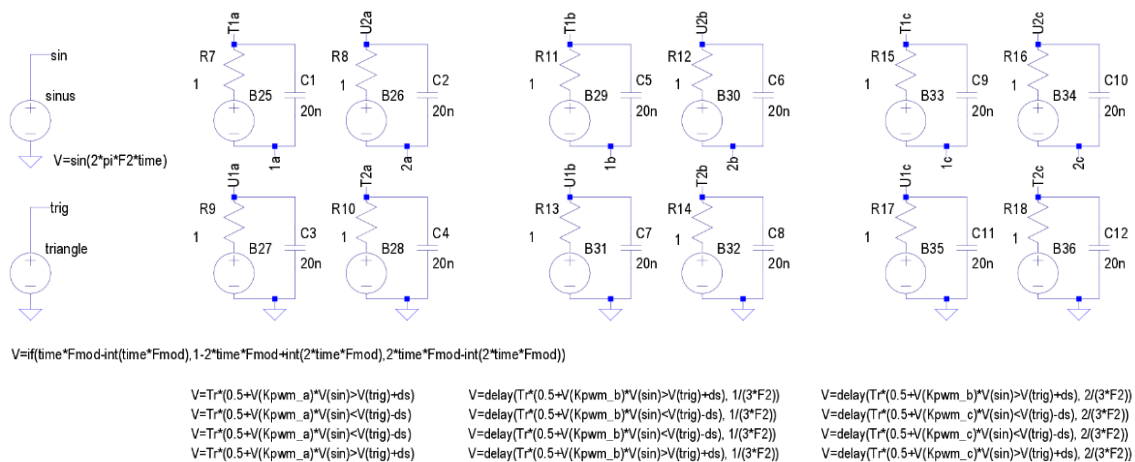


Рис.4.9.3.1 Слева генераторы синуса и треугольников (источники sinus и triangle соответственно), остальные источники (B25-B36) управляют конкретными транзисторами. Под источниками написаны их функции.

Для начала, можно создать функциональный источник напряжения, формирующий синусоиду необходимой частоты (рис.4.9.3.1 – источник «sinus» и рис.4.9.3.7 – зеленая функция  $V(\sin)$ ). Его функция  $V=sin(2\pi F_2 \cdot time)$ , а потенциал -  $V(\sin)$ .

Так же можно создать источник симметричного пилообразного сигнала. Его функция получилась следующей (рис. 4.9.3.2):  $V=if(time*Fmod-int(time*Fmod),1-2*time*Fmod+int(2*time*Fmod),2*time*Fmod-int(2*time*Fmod))$

- 1) Fmod - частота модуляции, задаваемая пользователем. Чем выше частота модуляции, тем меньше будет в итоге влияния высших гармоник. Однако не все компоненты смогут справиться с очень большой частотой. Обычно она должна быть в несколько раз больше требуемой частоты. Для 400Гц обычно используются частоты модуляции от 9600Гц до 24000Гц. Однако чем меньше частота, тем дешевле. Компьютерная симуляция позволит эффективно подобрать и проверить частоту модуляции.
- 2) Функция if(a, x, y) принимает значение x если a>0.5, в противном случае y.
- 3) Функция int(x) преобразует x в целое.

Данная функция - это хитрая комбинация линейных функций. Проще всего разобраться в этой функции с помощью рисунков (рис.4.9.3.2 - рис.4.9.3.6)

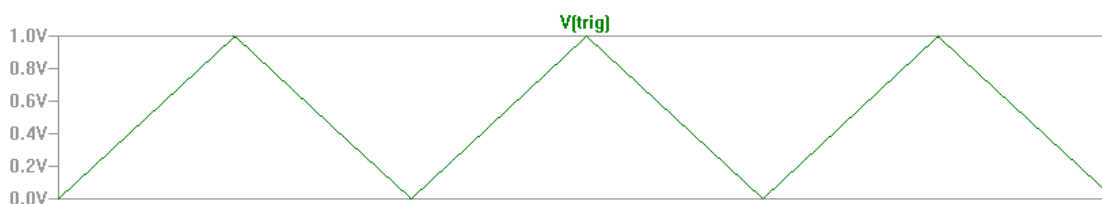


Рис.4.9.3.2 Симметричный пилообразный сигнал.

Симметричный пилообразный сигнал состоит из функции if(). Переменными этой функции служат три функции ассиметричных пилообразных сигналов. Первая функция – это условие, ее отображает функция  $V=time*Fmod-int(time*Fmod)$ . Она изображена на рис.4.9.3.3 синим.

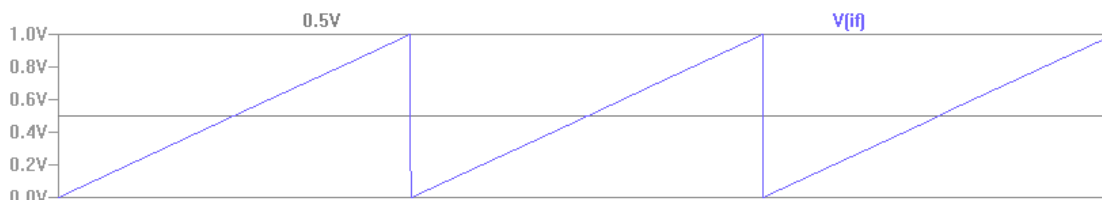


Рис.4.9.3.3 Условие – ассиметричный пилообразный сигнал.

Как видно, ее пересекает прямая, равная 0.5. Свойство функции if(a, x, y) таково, что если «a» (a это как раз синий график на рис.4.9.3.3) меньше 0.5, то функция

принимает значения функции «x»:  $V=1-2*time*Fmod+int(2*time*Fmod)$ , изображенной на рис.4.9.3.4:

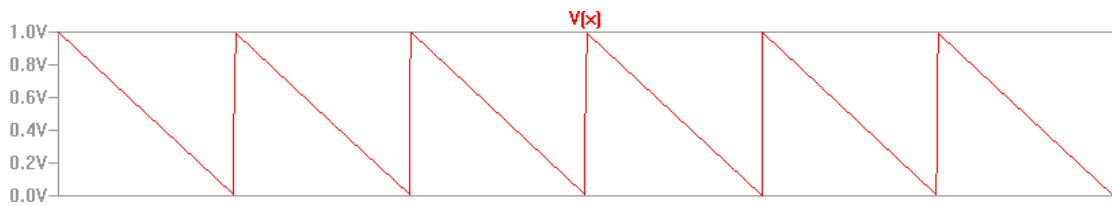


Рис.4.9.3.4 Состояние 1 – ассиметричный пилообразный сигнал «x».

А если «a» больше 0.5, то функция принимает значение функции «у»:

$V=2*time*Fmod-int(2*time*Fmod)$ , изображенной на рис.4.9.3.5г:

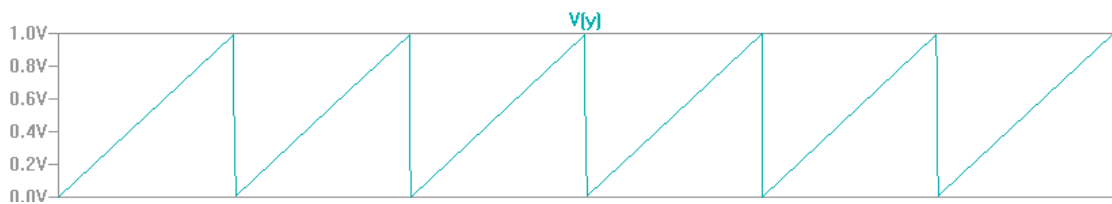


Рис.4.9.3.5 Состояние 2 – ассиметричный пилообразный сигнал «у».

Отсюда и получается что треугольники «x» и «у» чередуются друг с другом, образуя функцию, изображенную на рис.4.9.3.2. А теперь разберем сам ассиметричный пилообразный сигнал. Сам сигнал основан на реализации функции «мантисса», это такая функция, которая принимает только дробную часть числа. Поскольку SPICE такой функции не имеет, ее необходимо вывести. Мантиссу можно получить, если от функции прямой отнять функцию «антье» - это функция, принимающая только целые числа. Как раз аналог последней в SPICE присутствует – это функция `int()`. Функция ассиметричного сигнала (рис.4.9.3.3) состоит из двух частей:  $time*Fmod$  – прямая, и  $int(time*Fmod)$  – ее целочисленное значение. Графики этих функций представлены на рис.4.9.3.6:

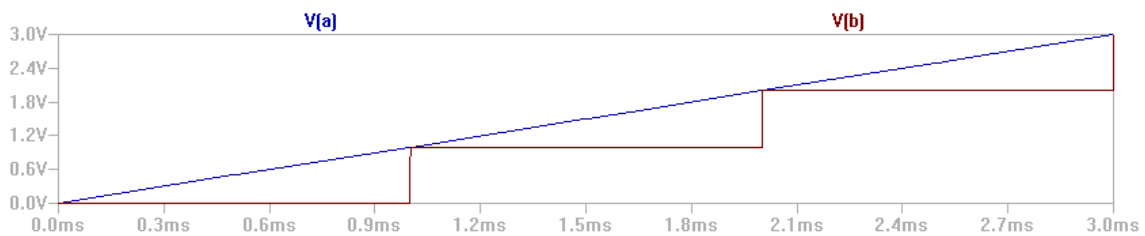


Рис.4.9.3.6 Функции, чья разность дает ассиметричный пилообразный сигнал.

Функция состояния «у» получается умножением частоты модуляции на два, и функция состояния «x» образуется вычитанием функции «у» из единицы.

Теперь эти две функции (синус и треугольники) будут использованы для создания сигналов, управляющих транзисторами. Разберем один из таких источников. Как видно они подключены через RC-фильтр. Причина такая же, как и в случае с управлением тиристорами – бесконечно быстро возрастающие импульсы недопустимы. Источники обладают функциями следующего вида:

$$V = Tr * (0.5 + V(Kpwm\_a) * V(sin) > V(trig) + ds)$$

- 1) Tr - константа, определяемая пользователем, больше чем напряжение открытия транзистора. Т.к. второй множитель не превысит единицу, то сигнал управления в целом будет управлять именно значением Tr.
- 2) ds - маленькая величина равная 5m служащая минимизировать ошибки программы, вызванные неопределенностью при сравнении.
- 3) При  $V(Kpwm\_a) = 0.5$  мы увидим сравнение синусоиды “лежащей” на оси и пилообразного сигнала. Если значение на выходе будет слишком большим, то  $V(Kpwm\_a)$  уменьшится, за счет чего импульсы станут уже, а, значит, амплитуда синусоиды на нагрузке также уменьшится. И наоборот - если значение напряжения ниже заданного, тогда  $V(Kpwm\_a)$  увеличится, и ширина импульсов станет шире. Чем ближе к заданному значению будет напряжение на нагрузке, тем медленнее будет меняться  $V(Kpwm\_a)$  - это и есть упругая отрицательная обратная связь.

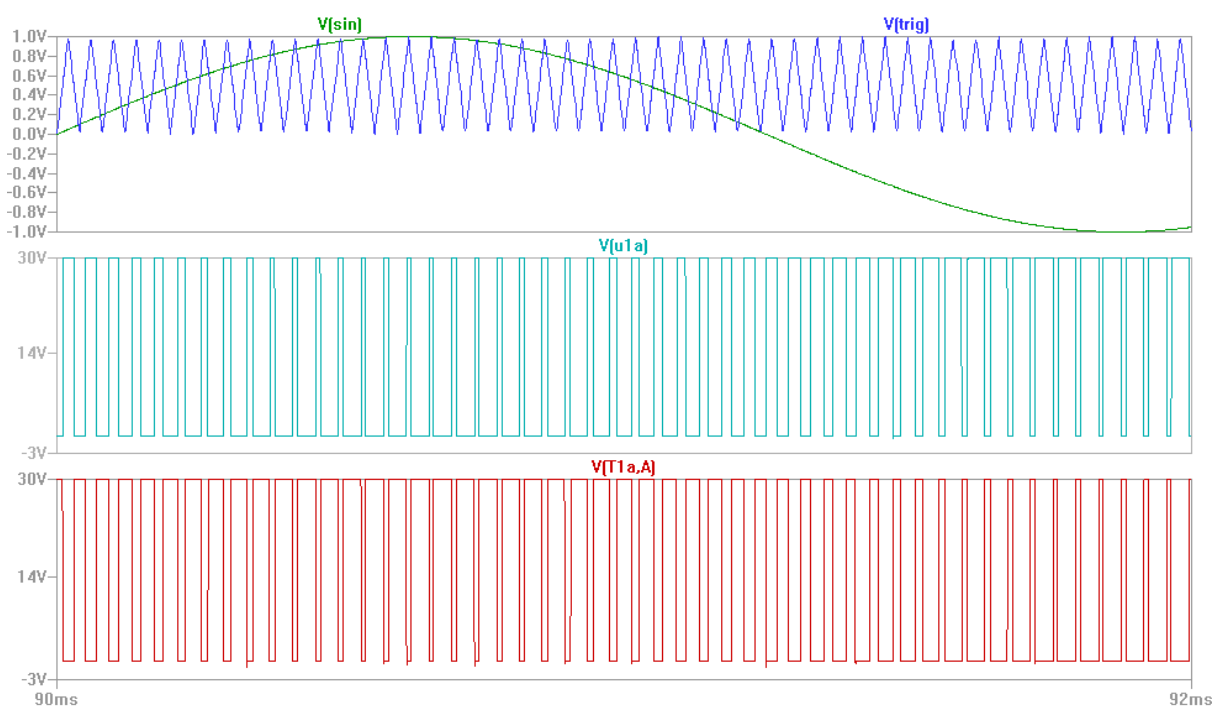


Рис.4.9.3.7 Составляющие ШИМ и ШИМ для пары противоположных транзисторов.



Все остальные транзисторы управляются такими же функциями, только скорректированными под их фазу и ориентацию.

## 4.10 Разработка средств тестирования системы

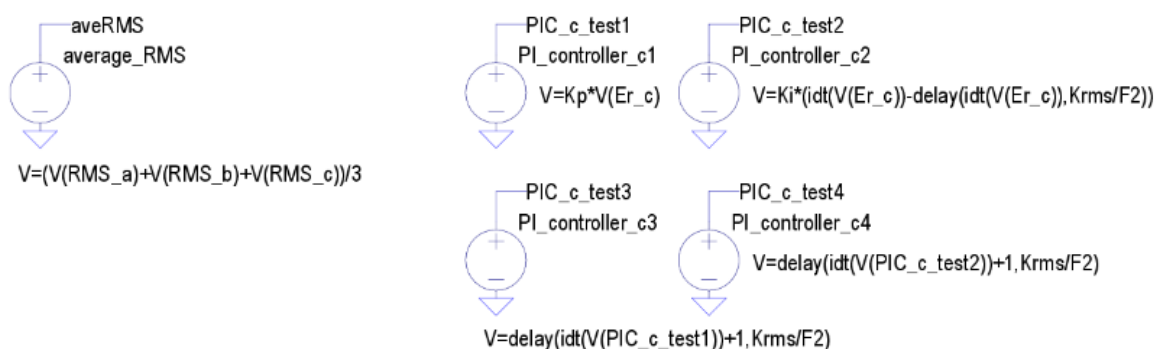


Рис.4.10.1 Разные источники для специфических симуляций, измерений.

В данной работе внедрено несколько блоков для тестирования и симуляции отдельных участков. В системе есть группа источников, умеющих переключать нагрузку во время симуляции. Они были подробно представлены в главе 4.8. Так же тут есть различные источники для задания произвольных функций. Например, есть источник для измерения среднего RMS (рис.4.10.1 – источник average\_RMS). Так же в модуле Testing присутствуют и другие функциональные источники (рис.4.10.1 – тестирование ПИ-регулирования). Пользователь, при желании может самостоятельно записывать в них функции. Например, чтобы подобрать оптимальный параметр для какой-нибудь константы, можно задать этот параметр в виде медленно меняющейся линейной функции и смотреть как будет изменяться симуляция. Все эти дополнительные системы состоят из простых функций и позволяют свободно работать с системой, а так же совершенствовать различные методы моделирования и симуляции.

## 5. Подведение итогов

### 5.1 Организация удобного интерфейса

Вся система четко разбита на модули. Это позволяет пользователю легко менять модули местами, дополнять их или вовсе отказываться. Система, в этом смысле, похожа на конструктор.

<code>.tran 0 2 0m 0.001m</code>	<code>.param pulse=1/(F*6) Set=115 load=1g</code>	Set - output RMS
<code>.model SW SW(Ron=1m Vt=7 Vh=3.5)</code>	<code>.param A=325 F=50 A2=540 F2=400</code>	A - input amplitude
<code>.model Ideal D(Ron=0.2 Roff=250k Vfwd=0)</code>	<code>.param alpha=0 Krms=1 Fmod=25k</code>	F - input frequency
<code>.model Ideal_Tr NMOS(KP=1)</code>	<code>.param Ton=50m Tr=30</code>	A2 - DC rate in testing system
	<code>.param Kp=100m Ki=200m Kpic=2 Npic=4</code>	F2 - output frequency
	<code>.param Rtr=2 Ctr=20n ds=5m</code>	alpha - rectifier control (from 0 to 90 degrees)
		Krms - amount of periods to calculate RMS
		Fmod - frequency of modulation
		Ton - turning on time (alpha from 90 to set)
		Tr - transistors control voltage
		Kp - proportional coefficient
		Ki - integral coefficient
		Kpic - PI-controller coefficient (if 0 then off)

Рис.5.1.1 Первые два столбика - директивы, синие справа - комментарии.

Каждый компонент можно менять, подстраивая модель под конкретную задачу, однако самые основные параметры можно менять через строки директив (рис.5.1.1). SPICE-директива - это текст, написанный на языке программы. Он позволяет определять введенные константы, модели компонентов и их параметры. В строках `.param` пользователь может четко определить параметры системы, такие как частота, амплитуда, время плавного пуска и многие другие. Обнулив определенные параметры, такие как время плавного пуска или  $K_{pic}$  можно убрать из симуляции некоторые явления. Пользователю не нужно копаться во всех уравнениях, чтобы настроить систему под себя, а если возникнут вопросы по поводу значения переменных, то всегда есть в модели несколько строчек текста раскрывающие и описывающие эти константы.

В системе реализована возможность исследования модели и подбора параметров. В предыдущей главе говорилось о том, что можно создавать источник, плавно изменяющий исследуемую переменную. Интерфейс системы позволяет выводить графики функций, зависящих не только от времени, но и от других потенциалов – в таком случае время будет участвовать неявно.

У самой программы LT-SPICE IV хорошо и удобно развита система оптимизация вычислительных процессов и исследование процессов, а именно удобное отображения и множество режимов исследования, такие как амплитудно-частотный анализ, анализ передаточной функции и другие. В программе очень просто анализировать графики готовых симуляций. Можно, например, рассчитать RMS участка, провести быстрый фурье-анализ.

Одна из удобных возможностей системы это создавать источники, на которые можно загрузить реальные сигналы с осциллографа или компьютера в формате .wav. Это позволит симулировать модель конкретного модуля или компонента для действительного устройства. Например, если у пользователя уже есть осциллограмма действующего выпрямителя, он может подключить ее для последующего анализа инвертора, или же подбора оптимального значения емкости фильтра, если осциллограмма предполагает сигнал без сглаживающих конденсаторов.

## **5.2 Анализ проблем моделирования**

В данном виде, разработанная модель не учитывает некоторые явления. Это относится, прежде всего, к трехфазному трансформатору. Данная модель учитывает нелинейные искажения напряжения и тока, образованных под влиянием паразитных индуктивностей и магнитных характеристик сердечников, однако модель не учитывает саму магнитную связь между сердечниками. Куда большую сложность представляет собой сбор данных о действительных параметрах трансформаторов, в особенности величины коэрцитивной силы, остаточной индукции и индуктивности намагничивания. Даже зная тип материала сердечника, данные параметры у разных трансформаторов с одинаковыми по типу сердечниками будут отличаться.

Другая проблема - это время вычислений - чем больше деталей учитывает модель, тем дольше будет идти симуляция. Для использования этой программы на практике инженер должен понимать, что действительно важно, а чем можно пренебречь. Другая сложность, связанная с вычислительным процессом, - это “вес” полученного файла. Симуляция в LT-SPICE представляет собой просчет всех значений в каждом узле системы. Поэтому, даже несколько секунд симуляции займут несколько гигабайт памяти жесткого диска. Не смотря на это, автор полагает, что в ближайшем будущем эти проблемы будут решены в связи с развитием компьютерной индустрии.

### **5.3 Анализ возможностей дальнейшего развития данной системы**

Разработанная модель представляет собой сложную систему, позволяющую пользователю легко симулировать процессы, происходящие в настоящих преобразователях частоты. Потенциал такой системы огромен. Для начала можно было бы еще больше оптимизировать систему и предлагать пользователю значения для коэффициентов ПИД-регулирования, значения фильтров, в зависимости от требований по пульсациям, и другие. Это было бы удобно, т.к. многие величины рассчитываются исходя из известных законов и формул, а компьютер лучше человека умеет справляться с такими задачами.

Следующим шагом могло бы стать создание независимой программы-симулятора, в которую можно было бы ввести технические требования, а программа, в зависимости от выбранных опций, предлагала бы группу решений. Таким образом инженеру было бы достаточно взглянуть на предоставленные симуляции и отчет программы, что бы в кратчайшие сроки перейти к разработке наиболее выгодного решения.

Дальнейшим шагом могла бы стать интеграция с базами данных крупных дистрибьюторов электроники. Программа смогла бы не только строить систему, но и подыскивать и предлагать компоненты исходя из характеристик, цены, гарантийного срока и срока доставки. Сейчас эту функцию выполняют люди, однако, по

мнению автора, это значительная трата времени. Компьютер смог бы легко находить выгодные акции и пролистывать огромные каталоги за считанные секунды.

В дальнейшем развитие интернет-ботов позволило бы им анализировать отзывы и даже оставлять отзывы, создавая сеть по роботизированному обмену опытом. Все это значительно повысит скорость и, главное, качество производимого продукта.

## Заключение

Актуальность создания системы моделирования и симуляции преобразовательной техники увеличивается с каждым годом в связи с глобальной электрификацией и развитием рынка электромобилей. В данной работе была разработана открытая система преобразователя напряжения на базе программного обеспечения LT-SPICE IV. Эта система отвечает нескольким важным критериям: реалистичность симуляций, гибкость, простота в освоении и удобство в использовании.

Данная разработка уже была внедрена в производство и позволила сэкономить на макетировании и натурном моделировании. С помощью данной модели были выявлены причины искажения сигналов на настоящем преобразователе. Данная система позволила подобрать коэффициенты для ПИ-регулятора.

Данная работа также несет полезную образовательную функцию. Ввиду того, что на данном языке качественных учебных материалов по среде LT-SPICE IV крайне мало, данная работа может быть хорошим учебным пособием для инженеров электронщиков и электротехников.

В данной системе очень активно использовано понятие функционального источника напряжения, которое имеет широкие возможности применения, что в данной работе и показано. Именно на нем, используя несложные математические функции, возможно создавать определенного вида программы.

На основе функциональных источников были успешно реализованы несколько сложных алгоритмов, в том числе был описан алгоритм управления сигналами с помощью переключения других сигналов.

Система была разработана таким образом, чтобы пользователь легко смог разобраться в ней и подстроить ее под свою конкретную задачу. Сама же работа по-

строена таким образом, чтобы читатель смог легко воспроизвести, использовать и, возможно, усовершенствовать систему.

У данной системы есть несколько вопросов, решив которые можно вывести систему на качественно новый уровень. В целом же данная система решает поставленные задачи и может и в дальнейшем применяться в промышленности для сокращения расходов на разработку силовой преобразовательной техники. Ее главным преимуществом является ее открытость и доступность, что позволяет пользователям вносить усовершенствования и делиться с другими пользователями. На основе идей, представленных в данной работе можно создать и другие системы, решающие уже другие инженерные задачи. Если такие системы останутся открытыми, то достаточно скоро мы получим новое явление в инженерии.

## **Использованная литература**

Забродин, Ю.С. Промышленная электроника/ Ю.С. Забродин. – Москва: Высшая Школа, 1982, - 489 с.

Семенов, Б.Ю. Силовая Электроника: от простого к сложному/ Б.Ю. Семенов. – Москва: СОЛОН-Пресс, 2011, -417 с.

Володин, В. LTspice компьютерное моделирование электронных схем / В. Володин. – Санкт-Петербург: БХВ-Петербург, 2010. - 389 с.

Володин, В. Пополнение библиотеки схемных элементов симулятора LTspice. Создание модели ШИМ-контроллера TL494/ В. Володин // Компоненты и технологии. – 2009. – №4. – с. 127 – 137.

Володин, В. Моделирование сложных электромагнитных компонентов при помощи Spice-симулятора LTspice/SwCAD III/ В. Володин // Компоненты и технологии. – 2008. – №4. – с. 175 – 182.

Володин, В. Гистерезисная модель нелинейной индуктивности симулятора LTspice/ В.Володин// Силовая Электроника. – 2010. – №1. – с. 56 – 60.

Мазуренко Л.И. Моделирование и анализ трехфазного асинхронного генератора с вентильно-емкостным возбуждением и амплитудно-фазовым регулированием реактивного тока/ Л.И. Мазуренко, В.А. Лесник, А.В. Джура, Л.Н. Дынник// Вісник КДПУ. – 2006. - №3. – с. 116 – 119.

Сизганова Е.Ю. Моделирование работы однофазных трансформаторов напряжения в режиме холостого хода/ Е.Ю. Сизганова, Н.В. Сизганов, Л.А. Фоминцев// Сибирский Федеральный Университет. – 2013. – 4с.



Кружаев, А.В. Компьютерное моделирование и экспериментальное исследование переходных процессов в однофазном трансформаторе напряжения/ А.В. Кружаев, И.А. Елагин, М.А. Павлейно, В.А. Дмитриев, А.М. Чалый// Журнал технической физики. – 2015. – том 85. №2. – с. 31 – 38.

Шкуропат, И.А. Уравнения напряжения и схема замещения трансформатора/ И.А. Шкуропат// Электро. – 2004. - №4. – с.21 – 25.

Sung-Don Cho. Three-phase Transformer Model and Parameter Estimation for ATP/ Cho Sung-Don// Journal of Electrical Engineering and Technology. – 2006. – Vol.1. - №3. pp. 302 – 307.

Bruce A. Mork. Hybrid Transformer Model for Transient Simulation – Part I: Development and Parameters/ Bruce A. Mork, Francisco Gonzales, Dmitri Ishchenko, Joydeep Mitra, Don L. Stuehm// IEEE Transactions on Power Delivery. – 2007. – Vol.22. - №1. – pp. 248 – 254.

Engelhardt, Mike. Using Transformers in LTspice/SwitcherCAD III/ Mike Engelhardt// Linear Technology Magazine. – 2006. – septembe. – pp. 23 – 24.

PID Controller [eCircuit Center]. – <http://www.ecircuitcenter.com/circuits/pid1/pid1.htm> (27.01.2015).

AVR. Учебный курс. Использование ШИМ [easy electronics электроника для всех].- 2008. – <http://easyelectronics.ru/avr-uchebnyj-kurs-ispolzovanie-shim.html> (28.05.2016)

Transformers. – <http://www.ltwiki.org/?title=Transformers> (28.05.2016).

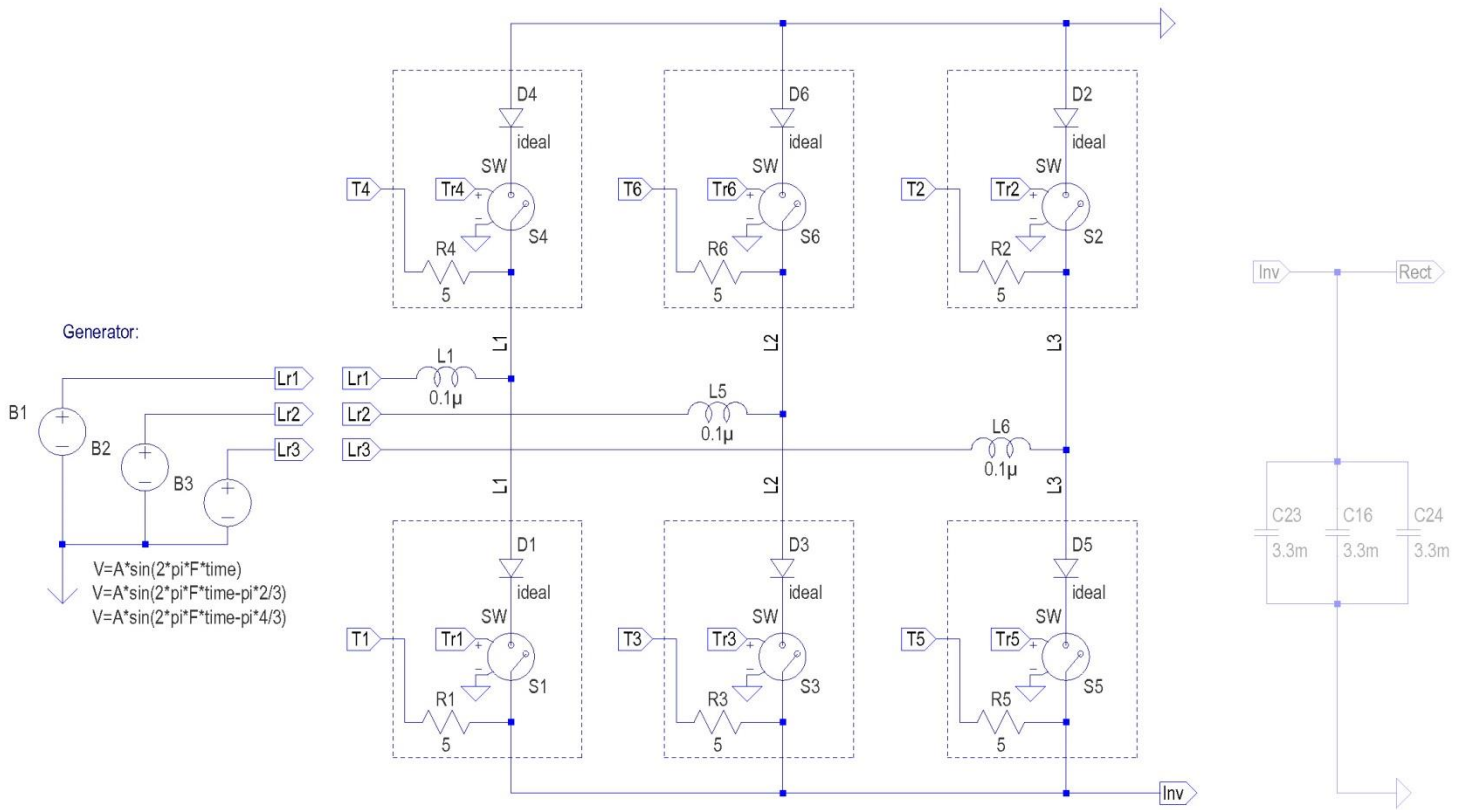
## **Приложения**

### **Приложение 1.**

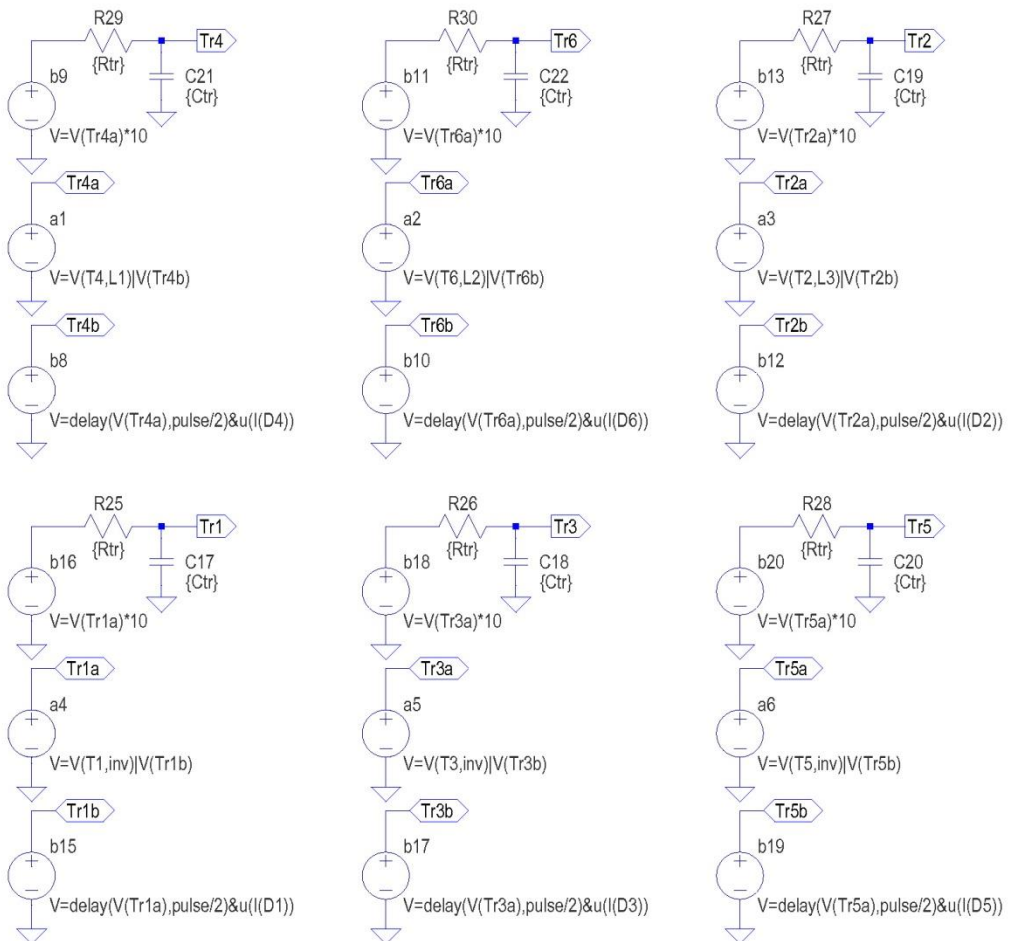
На первом листе – общий вид схемы, на последующих шести – более детальное изображение основной схемы.



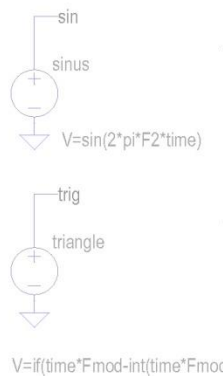
Rectifier:



Thyristors:



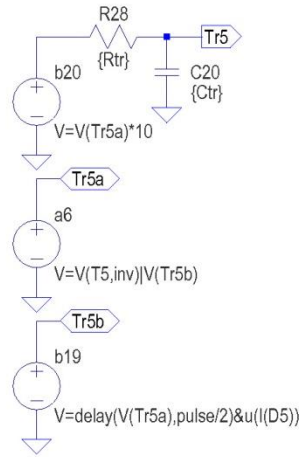
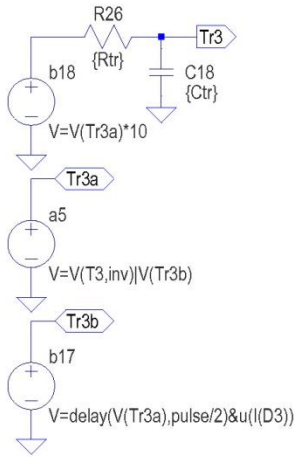
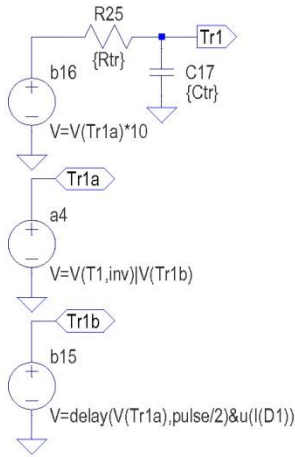
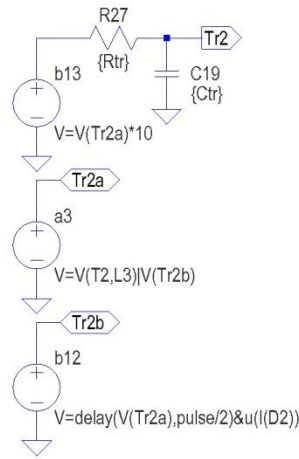
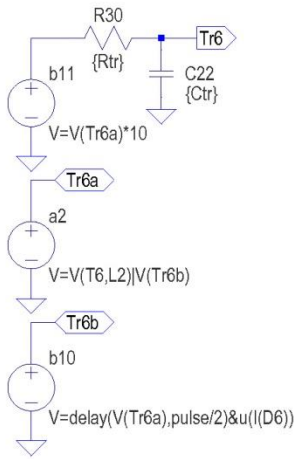
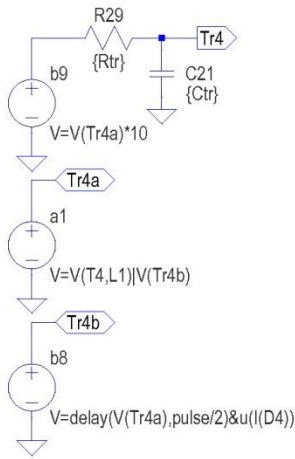
PWM generator:



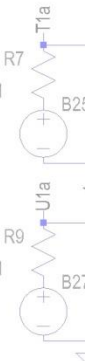
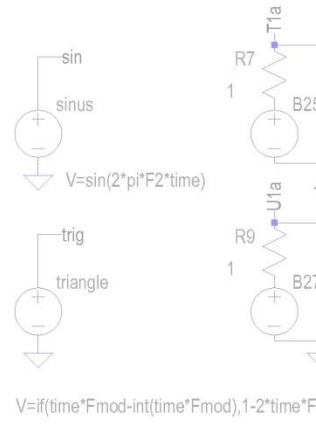
Rectifier control:



Thyristors:

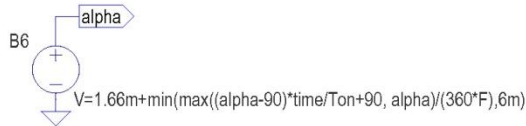


PWM generator:

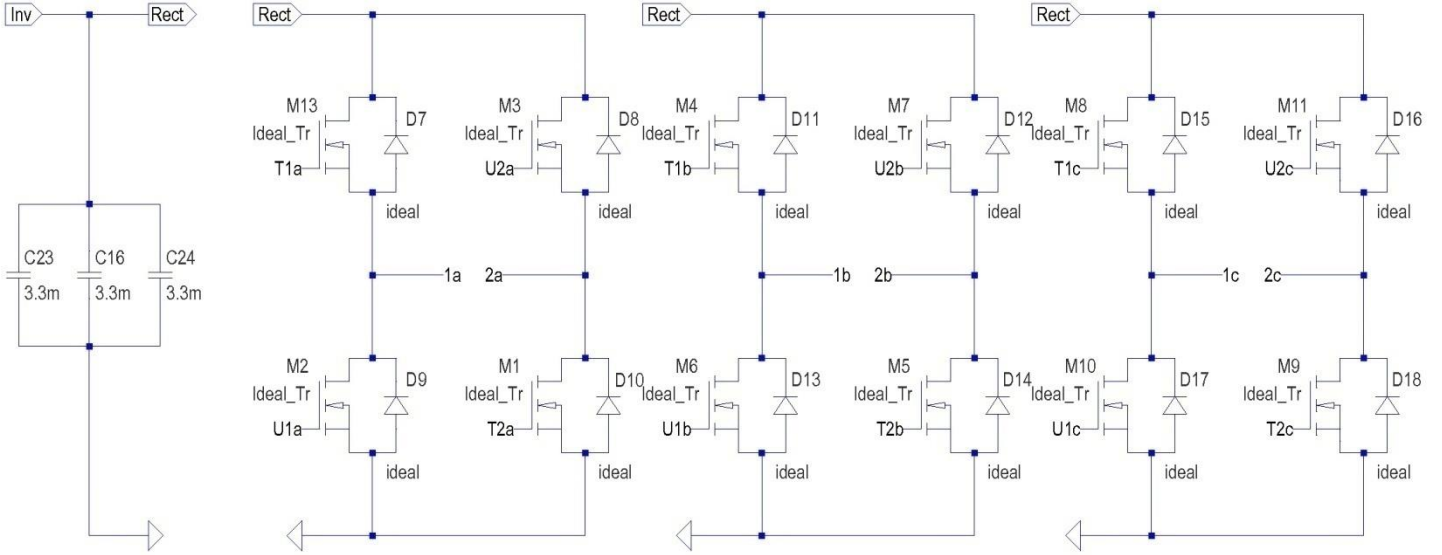


$V=Tr*(0.5+$   
 $V=Tr*(0.5+$   
 $V=Tr*(0.5+$   
 $V=Tr*(0.5+$

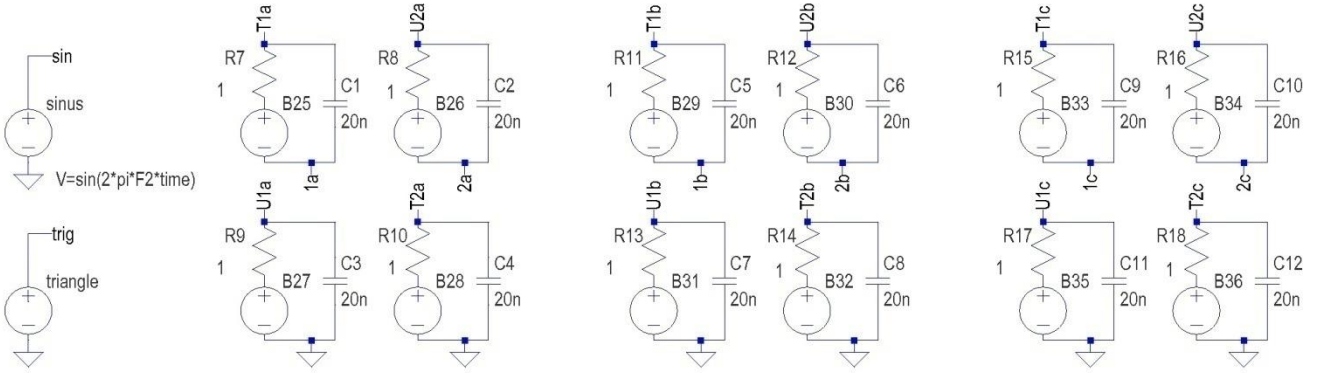
Rectifier control:



Inverter:



PWM generator:



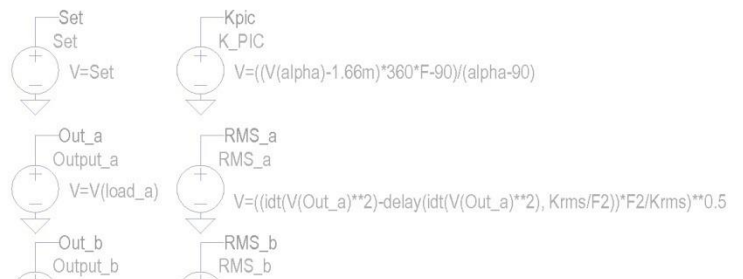
$$V = \text{if}(\text{time} * \text{Fmod} - \text{int}(\text{time} * \text{Fmod}), 1 - 2 * \text{time} * \text{Fmod} + \text{int}(2 * \text{time} * \text{Fmod}), 2 * \text{time} * \text{Fmod} - \text{int}(2 * \text{time} * \text{Fmod}))$$

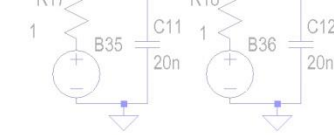
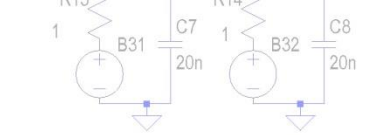
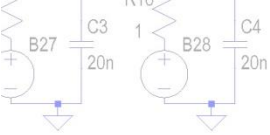
$$\begin{aligned} V &= \text{Tr} * (0.5 + V(\text{Kpwm\_a}) * V(\text{sin}) > V(\text{trig}) + \text{ds}) \\ V &= \text{Tr} * (0.5 + V(\text{Kpwm\_a}) * V(\text{sin}) < V(\text{trig}) - \text{ds}) \\ V &= \text{Tr} * (0.5 + V(\text{Kpwm\_a}) * V(\text{sin}) < V(\text{trig}) - \text{ds}) \\ V &= \text{Tr} * (0.5 + V(\text{Kpwm\_a}) * V(\text{sin}) > V(\text{trig}) + \text{ds}) \end{aligned}$$

$$\begin{aligned} V &= \text{delay}(\text{Tr} * (0.5 + V(\text{Kpwm\_b}) * V(\text{sin}) > V(\text{trig}) + \text{ds}), 1/(3 * \text{F2})) \\ V &= \text{delay}(\text{Tr} * (0.5 + V(\text{Kpwm\_b}) * V(\text{sin}) < V(\text{trig}) - \text{ds}), 1/(3 * \text{F2})) \\ V &= \text{delay}(\text{Tr} * (0.5 + V(\text{Kpwm\_b}) * V(\text{sin}) < V(\text{trig}) - \text{ds}), 1/(3 * \text{F2})) \\ V &= \text{delay}(\text{Tr} * (0.5 + V(\text{Kpwm\_b}) * V(\text{sin}) > V(\text{trig}) + \text{ds}), 1/(3 * \text{F2})) \end{aligned}$$

$$\begin{aligned} V &= \text{delay}(\text{Tr} * (0.5 + V(\text{Kpwm\_b}) * V(\text{sin}) > V(\text{trig}) + \text{ds}), 2/(3 * \text{F2})) \\ V &= \text{delay}(\text{Tr} * (0.5 + V(\text{Kpwm\_c}) * V(\text{sin}) < V(\text{trig}) - \text{ds}), 2/(3 * \text{F2})) \\ V &= \text{delay}(\text{Tr} * (0.5 + V(\text{Kpwm\_c}) * V(\text{sin}) < V(\text{trig}) - \text{ds}), 2/(3 * \text{F2})) \\ V &= \text{delay}(\text{Tr} * (0.5 + V(\text{Kpwm\_c}) * V(\text{sin}) > V(\text{trig}) + \text{ds}), 2/(3 * \text{F2})) \end{aligned}$$

RMS sensors and PI-system:





Y — 2b  
C — 1c  
Z — 2c

\*time\*Fmod+int(2\*time\*Fmod),2\*time\*Fmod-int(2\*time\*Fmod))

$V = \text{delay}(\text{Tr}*(0.5+V(\text{Kpwm\_a})*V(\sin)>V(\text{trig})+ds), 1/(3*F2))$   
 $V = \text{delay}(\text{Tr}*(0.5+V(\text{Kpwm\_a})*V(\sin)<V(\text{trig})-ds), 2/(3*F2))$   
 $V = \text{delay}(\text{Tr}*(0.5+V(\text{Kpwm\_a})*V(\sin)<V(\text{trig})-ds), 1/(3*F2))$   
 $V = \text{delay}(\text{Tr}*(0.5+V(\text{Kpwm\_a})*V(\sin)>V(\text{trig})+ds), 2/(3*F2))$

$V = \text{delay}(\text{Tr}*(0.5+V(\text{Kpwm\_b})*V(\sin)>V(\text{trig})+ds), 1/(3*F2))$   
 $V = \text{delay}(\text{Tr}*(0.5+V(\text{Kpwm\_b})*V(\sin)<V(\text{trig})-ds), 2/(3*F2))$   
 $V = \text{delay}(\text{Tr}*(0.5+V(\text{Kpwm\_b})*V(\sin)<V(\text{trig})-ds), 1/(3*F2))$   
 $V = \text{delay}(\text{Tr}*(0.5+V(\text{Kpwm\_b})*V(\sin)>V(\text{trig})+ds), 2/(3*F2))$

$V = \text{delay}(\text{Tr}*(0.5+V(\text{Kpwm\_c})*V(\sin)>V(\text{trig})+ds), 1/(3*F2))$   
 $V = \text{delay}(\text{Tr}*(0.5+V(\text{Kpwm\_c})*V(\sin)<V(\text{trig})-ds), 2/(3*F2))$   
 $V = \text{delay}(\text{Tr}*(0.5+V(\text{Kpwm\_c})*V(\sin)<V(\text{trig})-ds), 1/(3*F2))$   
 $V = \text{delay}(\text{Tr}*(0.5+V(\text{Kpwm\_c})*V(\sin)>V(\text{trig})+ds), 2/(3*F2))$

**RMS sensors and PI-system:**



3))&delay(u(V(com,Lr3)),pulse)),V(alpha))

n))&delay(u(V(Lr3,com)),pulse)),V(alpha))



.tran 0 2 0m 0.001m

.model SW SW(Ron=1m Vt=7 Vh=-3.5)

.model Ideal D(Ron=0.2 Roff=250k Vfwd=0)

.model Ideal\_Tr NMOS(KP=1)

.param pulse=1/(F\*6) Set=115 load=1g

.param A=325 F=50 A2=540 F2=400

.param alpha=0 Krms=1 Fmod=25k

.param Ton=50m Tr=30

.param Kp=100m Ki=200m Kpic=2 Npic=4

.param Rtr=2 Ctr=20n ds=5m

Set - output RMS

A - input amplitude

F - input frequency

A2 - DC rate in testing system

F2 - output frequency

alpha - rectifier control (from 0 to 90 degrees)

Krms - amount of periods to calculate RMS

Fmod - frequency of modulation

Ton - turning on time (alpha from 90 to set)

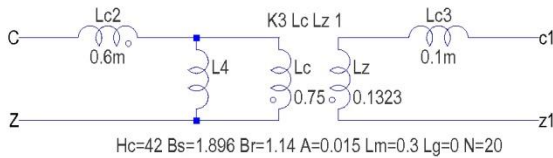
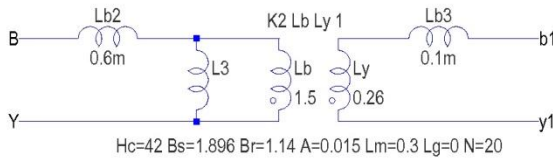
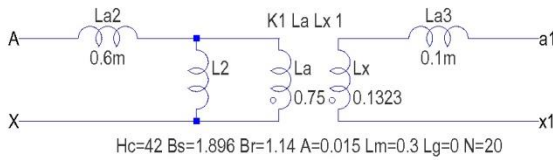
Tr - transistors control voltage

Kp - proportional coefficient

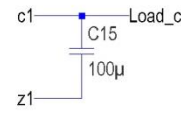
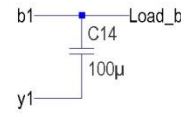
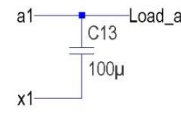
Ki - integral coefficient

Kpic - PI-controller coefficient (If 0 then off)

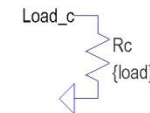
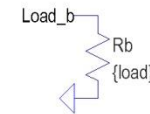
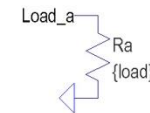
Transformer:



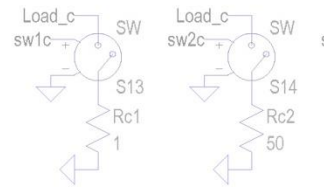
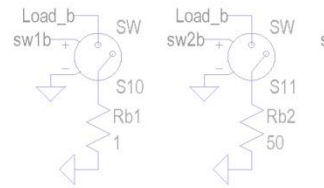
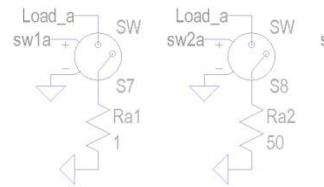
Filter:



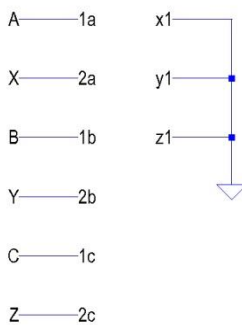
Load:



Testing load:

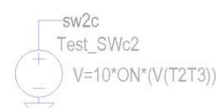
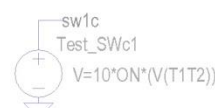
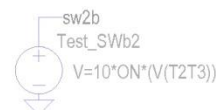
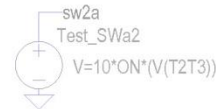
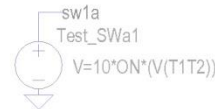


System:



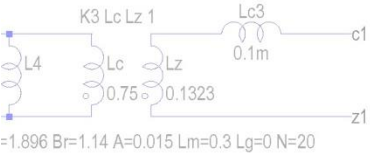
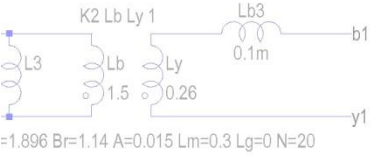
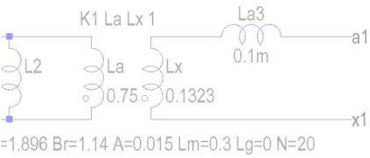
Testing load control:

.param ON=1 T1=0.3 T2=0.6 T3=0.8 T4=

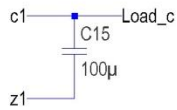
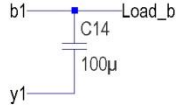
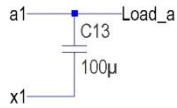




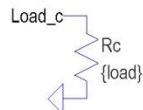
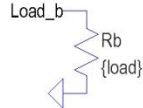
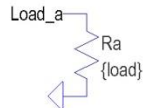
Kp - proportional coefficient  
 Ki - integral coefficient  
 Kpic - PI-controller coefficient (If 0 then off)



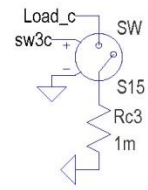
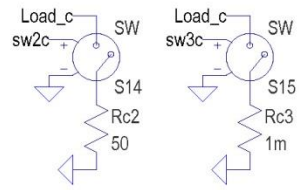
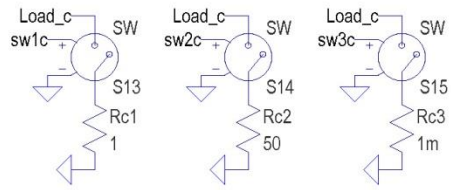
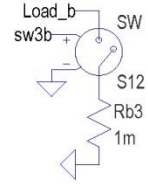
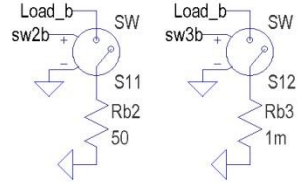
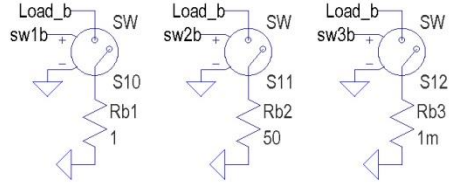
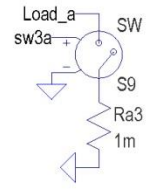
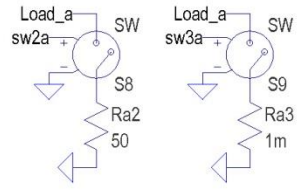
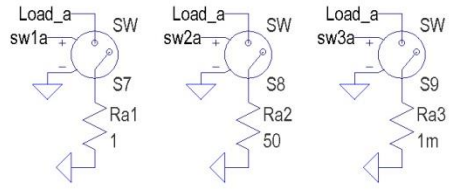
Filter:



Load:

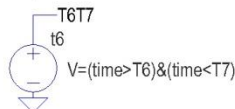
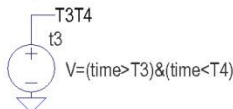
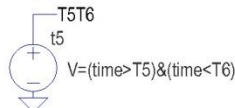
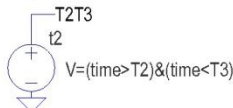
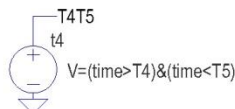
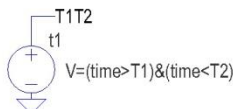
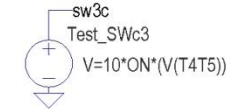
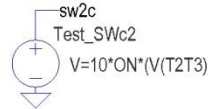
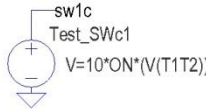
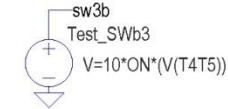
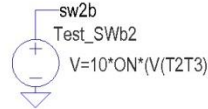
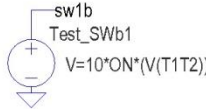
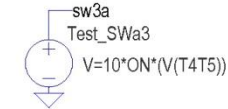
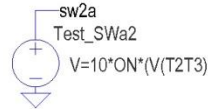
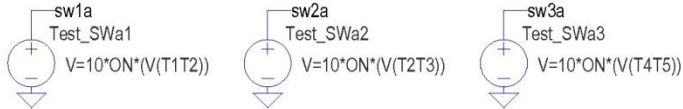


Testing load:

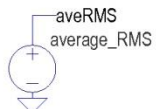


Testing load control:

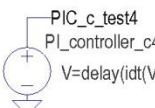
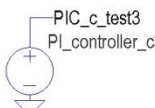
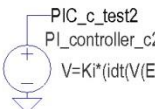
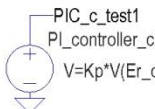
.param ON=1 T1=0.3 T2=0.6 T3=0.8 T4=1 T5=1.2 T6=1.5 T7=1.8



Testing:



$$V=(RMS\_a)+V(RMS\_b)+V(RMS\_c)/3$$



$$V=delay(idt(V(PIC\_c\_test1))+1,Krms/F2)$$

$$V=Kp*V(Er\_c)$$

$$V=Ki*(idt(V(Er\_c))-delay(idt(V(Er\_c)),Krms/F2))$$

$$V=delay(idt(V(PIC\_c\_test2))+1,Krms/F2)$$

## Приложение 2

Перечень операций и функций функционального источника напряжения (и тока) в LT-SPICE.

Операции	Описание
!	Преобразовать следующее выражение в один из двух логических размеров и инвертировать
**	Символ возведения в степень. Возвращается только вещественная часть, т. е. $-1^{**0.5}$ даст ноль, а не $i$
*	Умножение с плавающей точкой.
/	Деление с плавающей точкой
+	Сложение с плавающей точкой.
-	Вычитание с плавающей точкой
<	Истинно, если выражение слева меньше выражения справа, в противном случае ложно.
>	Истинно, если выражение слева больше выражения справа, в противном случае ложно.
<=	Истинно, если выражение слева меньше или равно выражению справа, в противном случае ложно.
>=	Истинно, если выражение слева больше или равно выражению справа, в противном случае ложно
&	Преобразовать выражение в один из двух логических размеров и затем AND
^	Преобразовать выражение в один из двух логических размеров и затем XOR
	Преобразовать выражение в один из двух логических размеров и затем OR

Функции от $x$	Определение	Описание
$abs(x)$	Абсолютное значение $x$	
$absdelay(x, t[, tmax])$	Задержать $x$ на $t$	Если задержка $t$ изменяется в процессе моделирования, то ее величина может быть ограничена с помощью параметра $tmax$
$acos(x)$	Аркосинус $x$	Результат в радианах. Возвращается вещественная часть аркосинуса $x$ , что снимает ограничения на величину аргумента $x$
$arccos(x)$	Аркосинус $x$	Синоним $acos(x)$
$acosh(x)$	Ареа-косинус $x$	Возвращает вещественную часть ареа-косинуса $x$ , что снимает ограничения на величину аргумента $x$
$asin(x)$	Арсинус $x$	Результат в радианах. Возвращает вещественную часть арксинуса $x$ , что снимает ограничения на величину аргумента $x$
$arcsin(x)$	Арсинус $x$	Синоним для $asin(x)$
$asinh(x)$	Ареа-синус $x$	
$atan(x)$	Арктангенс $x$	Результат в радианах
$arctan(x)$	Арктангенс $x$	Синоним для $atan(x)$
$atan2(y, x)$	Четырехквadrантный арктангенс $y/x$	Результат в радианах
$atanh(x)$	Ареа-тангенс $x$	

Функции от $x$	Определение	Описание
$bif(x)$		1 если $x > 0.5$ , иначе 0
$ceil(x)$	Целое равное или больше чем $x$	
$cos(x)$	Косинус $x$	$x$ в радианах
$cosh(x)$	Гиперболический косинус $x$	
$ddt(x)$	Производная $\frac{dx}{dt}$	
$delay(x, t[, tmax])$	Задержать $x$ на $t$	Синоним $absdelay(x)$
$exp(x)$	Число $e$ в степени $x$	
$floor(x)$	Целое равное или меньше $x$	
$hypot(x, y)$	$\sqrt{x^2 + y^2}$	
$idt(x[, ic[, assert]])$	Интеграл от $x$	В качестве дополнительных параметров указываются: начальное состояние $ic$ и сброс, если $assert$ истинно
$idtmod(x[, ic[, m[, o]])$	Интеграл от $x$	В качестве дополнительных параметров указываются: начальное состояние $ic$ , сброс по достижению моду- ля $m$ , смещение выхода на $o$
$if(x, y, z)$	Если $x > 0.5$ , то $y$ , иначе $z$	
$int(x)$	Преобразовать $x$ в целое	

Функции от $x$	Определение	Описание
$inv(x)$	0, если $x > 0.5$ , иначе 1	
$limit(x, y, z)$	Промежуточное значение $x$ , $y$ , и $z$	
$ln(x)$	Натуральный логарифм $x$	
$log(x)$	Альтернативный синтаксис для $ln(x)$	
$log10(x)$	Логарифм по основанию 10	
$max(x, y)$	Большее из $x$ или $y$	
$min(x, y)$	Меньшее из $x$ или $y$	
$pow(x, y)$	$x^{**}y$	Возвращает вещественную часть, что снимает ограничения на знак аргумента $x$
$pwr(x, y)$	$abs(x)^{**}y$	
$pwr(x, y)$	$sgn(x) \cdot abs(x)^{**}y$	
$rand(x)$	Случайное число между 0 и 1 в зависимости от целого значения $x$	
$random(x)$	То же, что и $rand(x)$	Обеспечивает гладкий переход между значениями
$round(x)$	Ближайшее целое $x$	

Функции от $x$	Определение	Описание
$sdt(x[ic[,assert]])$	Альтернативный синтаксис для $idt(x)$	
$sgn(x)$	Знак $x$	
$sin(x)$	Синус $x$	$x$ в радианах
$sinh(x)$	Гиперболический синус $x$	
$sqrt(x)$	Квадратный корень $x$	
$table(x,a,b,c,d)$	Интерполировать значение для $x$ , основанное на поиске в таблице, данной как набор пар точек	
$tan(x)$	Тангенс $x$	
$tanh(x)$	Гиперболический тангенс $x$	
$u(x)$	1 если $x > 0$ , иначе 0	
$uramp(x)$	$x$ если $x > 0$ , иначе 0	
$white(x)$	Случайное число между $-0.5$ и $0.5$	По сравнению с $random(x)$ обеспечивает более гладкий переход между значениями
$!(x)$	Альтернативный синтаксис для $inv(x)$	
$\sim(x)$	Альтернативный синтаксис для $inv(x)$	