

Филаретов В.В. Программа символьного анализа *cirsym*: история создания, структура и функции // Синтез, анализ и диагностика электронных цепей: международ. сб. науч. тр. – Ульяновск: УлГТУ, 2013. – Вып. 11. – С. 158–171.

ПРОГРАММА СИМВОЛЬНОГО АНАЛИЗА CIRSYM: ИСТОРИЯ СОЗДАНИЯ, СТРУКТУРА И ФУНКЦИИ

В. В. Филаретов

Программы символьного анализа на основе унисторного графа. Символьным анализом называется получение и представление результата анализа электрической цепи в виде аналитического выражения, в котором все или часть параметров схемы заданы буквами (символами). Если для анализа используется отображение схемы графом, то такой анализ называется символьно-топологическим.

Символьный анализ отличается от матрично-численного анализа высокой сложностью, как при получении, так и при представлении результатов. Для символьного анализа сложных схем приходится проводить анализ по частям, то есть делить схему на подсхемы, анализировать подсхемы в отдельности и объединять результаты анализа подсхем.

Наиболее эффективные программы символьного анализа электрических цепей в десятки узлов и элементов были разработаны на основе отображения схем унисторным (двунаправленным) графом проводимостей [1–6]. На унисторном графе можно было отображать источники напряжения, управляемые током (ИТУН). Остальные три типа управляемых источников (УИ): источник напряжения, управляемый напряжением (ИНУН); источник напряжения, управляемый током (ИНУТ) и источник тока, управляемый током (ИТУТ), а также идеальные операционные усилители (нуллоры) считаются нерегулярными для узлового базиса.

Электронная схема (ее граф) представляется как объединение подсхем (подграфов). Каждая подсхема (граф) анализируется в отдельности, затем осуществляется иерархическое объединение результатов анализа подсхем в том или ином порядке. Программы анализа [1–5] различаются, прежде всего, выбором способа деления графа на подграфы: делением по вершинам [1, 3, 5] или по дугам [2, 4]. В работе [6] показано, что деление графа по дугам является частным случаем деления по вершинам.

Эффективность программ [1–6] по сложности анализируемых схем и времени вычислений достигается за счет численного задания параметров схемы. Символьно задается только оператор дифференцирования – комплексная частота p . При этом результатами анализа подсхем и исходной схемы являются полиномы относительно p . Для задания символами некоторых (одного, двух, трех...) параметров схемы были разработаны

специальные версии программ, имеющие существенно сниженную эффективность. Наиболее эффективными программами признаны программы, реализующие диакоптический метод d -деревьев [1, 3, 5, 6].

Программа SYMBOL. Для полного символьного анализа электрических цепей В.В.Филаретовым была разработана программа SYMBOL, в которой также использовался метод унисторного графа [7, 8]. В отличие от программ [1–6] программа SYMBOL генерировала символьные выражения для полиномиальных коэффициентов в скобочной форме с оптимальным вынесением за скобки общих множителей [9]. Наиболее совершенные версии программы SYMBOL были внедрены в десятке проектных организаций и вузов в рамках кандидатской диссертации [10], а тексты на языке Фортран-4 опубликованы [11, 12].

Предельная сложность схем для программы SYMBOL ограничивалась 10 узлами и 5000 слагаемыми в знаменателе передаточной функции по напряжению (только такая схемная функция находилась программой). Время генерации потреблялось на нахождение слагаемых числителя и знаменателя в виде развернутого выражения и последующую его свертку по правилу выделения символа с максимальным участием в слагаемых выражения [9].

В начале 90-х годов программа SYMBOL была усовершенствована обобщением на нерегулярные типы УИ (ИНУН, ИНУТ и ИТУТ) на основе метода обобщенного унисторного графа [13, 14], что существенно повысило ее полезность для инженеров-разработчиков электронных схем и студентов, обучающихся на соответствующих специальностях.

Программа GRAPH. Повышение эффективности программы SYMBOL потребовало проведения кропотливых исследований в области оптимальной свертки символьных определителей ненаправленных и унисторных графов [15–21]. Эти исследования были положены в основу программы GRAPH для генерации оптимальных выражений определителей ненаправленных графов, которая была реализована Д.В.Шеиным [22, 23]. Показатели сложности (число операций умножения и сложения) оптимальных формул, генерируемых программой GRAPH, были рассчитаны теоретически, что показало полное совпадение теории и эксперимента [24–26].

Программа GRAPH формировала компактные скобочные выражения, минуя получение развернутого выражения и последующую его свертку, что многократно увеличило сложность анализируемых схем (до десятков–сотен узлов и элементов). Язык реализации – Си – обеспечил рекурсивные вызовы, что упростило структуру и текст программы, упростило ее отладку. Динамическое распределение памяти, предусматриваемое компиляторами Си, обеспечило многократное сокращение емкости требуемой во время вычислений памяти. Фактически память требуется только на хранение программы и символьного результата.

Технология формульного вывода, предложенная Д.В.Шеиным в программе GRAPH, была использована В.В.Филаретовым для реализации программ символьного анализа сложных электрических цепей с ИТУН и ОУ [27–31], а затем со всеми типами линейных элементов [32–37].

Программа CIRSYM (CIRcuit SYMbol). Программа CIRSYM и ее многочисленные модификации в отличие от известных программ символьного анализа, основанных на отображении схемы матрицами и графами, предусматривают работу на схемном уровне. При этом список элементов схемы служит как для задания исходных данных, так и для внутреннего формата данных об исходной схеме и производных от нее схем, что полностью исключает избыточность внутреннего (машинного) представления схемы.

В основе программы CIRSYM лежит метод схемных определителей. Понятие схемного определителя [38, 39] было введено В.В.Филаретовым по аналогии с понятием матричного определителя и развивает схемный подход к анализу электрических цепей, предложенный Фойснером [40– 42]. На методе схемных определителей основан ряд учебных курсов основан ряд учебных курсов [38, 43], этот метод популяризовался за рубежом [44–46]. О современном состоянии схемного подхода к символьному анализу электрических цепей можно узнать в [47, 48].

Для анализа электрических цепей с произвольным числом независимых источников напряжения и тока В.В.Филаретовым был предложен метод многомерного управляемого источника [46, 49], который легко реализовался в базовой версии CIRSYM. Таким образом, программа CIRSYM из программы генерации передаточных функций (для схем с одним независимым источником и одним откликом) стала программой символьного анализа электрических цепей с произвольным числом независимых источников и откликов.

Дальнейшая модернизация программы CIRSYM состояла в реализации базисной задачи диагностики, символьное решение которой разработал С.А.Курганов [50]. О современной версии программы CIRSYM можно получить информацию в [51]. К сожалению, до сих пор не удалось освободить символьный результат от небольшого числа избыточных скобок непосредственно в ходе генерации формул. Поэтому для исключения лишних скобок Д.В.Шейным была специально разработана программа LIKVID [52].

Сферу приложения программы CIRSYM существенно расширила реализация В.В.Филаретовым полиномиального режима вывода схемных функций [53]. Совсем недавно (в период написания этой статьи) полиномиальный режим был реализован в полном объеме (для нахождения произвольного числа откликов напряжения и тока от произвольного числа источников напряжения и тока).

Программа CIRSYM является анализирующим блоком системы схемотехнического анализа, диагностики и синтеза (SCADS) [54], а также используется в составе программы моделирования импульсных устройств FASTMEAN [55]. CIRSYM включена в отраслевой фонд алгоритмов и программ [56] и свободно распространяется [57]. Дополнительную информацию по обсуждаемому вопросу можно найти на сайтах [58–64]. Ниже обсуждаются особенности реализации программы CIRSYM.

Структуры исходных данных о схеме. Структура двухполюсных ветвей схемы имеет вид:

PASSIVE {первый узел; второй узел; кратность; идентификатор}.

Кратность – количество ветвей, входящих в двухполюсник. Для u -двухполюсника – это количество параллельных ветвей. Для z -двухполюсника – это количество последовательных ветвей. Идентификатором двухполюсника может быть имя отдельного сопротивления или проводимости, соответствующая сумма проводимостей или сопротивлений.

Структура управляемых источников (УИ) схемы:

SOURCE {начальный узел управляемой ветви; конечный узел управляемой ветви; начальный узел управляющей ветви; конечный узел управляющей ветви; кратность; идентификатор УИ}.

Кратность – количество отдельных УИ, образующих данный УИ. Для ИТУН – это количество параллельных ИТУН. Для ИНУТ – это количество последовательно соединенных ИНУТ. Идентификатором УИ является параметр отдельного УИ или соответствующая сумма параметров.

Специальные значения имеют кратности неудаляемых управляемых источников (НУИ) и многомерных НУИ. Кратность НУИ отрицательна и равна -1 . Под многомерным НУИ понимается многомерный УИ, управляемый идеальным элементом – нуллатором, то есть состоит из n обычных генераторов, управляемых единственным приемником, который не может быть удален из схемы в результате нейтрализации (обнуления).

Многомерный НУИ используется при анализе схемы с несколькими источниками воздействия, а также ее диагностики. Для решения этих задач нужен всего один многомерный НУИ, который помещается в схему автоматически и не задается в `sig`-файле. Внутри программы многомерный НУИ задается n управляемыми источниками, управляющие ветви которых имеют различные пары узлов, но управляющие ветви всех n управляемых источников одинаковы. Чтобы отличать многомерный НУИ от обычных УИ, кратность всех УИ многомерного НУИ уникальна и равна 999.

Структура графа, изоморфного схеме:

GRAPH {начальный узел; конечный узел; тип ветви}.

Типом ветви служит число 0 (для двухполюсника) или порядковый номер (для управляемого источника). Ребра графа, соответствующие управляемой и управляющей ветвям УИ, имеют одинаковый тип, равный номеру этого УИ.

Структура для диагностируемой схемы:

DIAG {номер диагностируемой ветви; идентификатор диагностируемой ветви};

Режимы работы программы CIRSYM:

Нахождение определителя схемы (при отсутствии приемников откликов) – **detan**.

Формирование схемной функции с разными сечениями для числителя и знаменателя – **cirfunst**.

Формирование схемной функции с единым сечением для числителя и знаменателя – **cirfunc**.

Формирование выражений для искомых токов и напряжений от нескольких источников напряжения и тока – **analys**.

Формирование полиномиальных выражений для искомых токов и напряжений от нескольких источников напряжения и тока – **analysp**.

Нахождение полиномиального определителя схемы (при отсутствии приемников откликов) – **detanp**.

Формирование полиномиальной схемной функции – **cirfunpa**.

Нахождение определителя схемы с компенсированными элементами – **detdia**.

Диагностика схемы – формирование выражений для неизвестных параметров – **diagnos**.

Разложение схемного определителя. Работа программы CIRSYM во всех режимах сводится к разложению схемного определителя. Используемые при этом функции языка Си подразделяются на 5 групп.

Выделение параметров элементов:

Рекурсивное динамическое разложение схемного определителя – **gggf**.

Вынесение за скобки общих множителей – **multipl**.

Рекурсивное динамическое разложение полиномиального схемного определителя – **gggp**.

Выделение общих множителей в полиномиальном схемном определителе – **multiplr**.

Нейтрализация (обнуление) УИ ($act[j].kol = 0$) с проверкой на вырождение схемы – **newtral**.

Объединение узлов $s1$ и $s2$ для УИ и нейтрализация ИТУН – **uniact**.

Объединение узлов $s1$ и $s2$ для двухполюсных элементов с удалением у-петель – **unipas**.

Проверка схемы на вырожденность:

Проверка графа (GRAPH) на связность – **bond1f**.

Проверка схемы (PASSIVE и SOURCE) на связность – **connec**.

Проверка наличия вырождающих схему сечений (из нораторов и генераторов тока или из нуллаторов и приемников напряжения – **hanged**.

Функции выбора ветвей для первоочередного выделения:

Нахождение висячей y -ветви (y -ветви, соединенной последовательно с генератором или приемником УИ) – **hangpas**.

Выбор оптимальной для разложения проводимости – **choiceg**.

Выбор оптимального для разложения сопротивления – **choicer**.

Проверка наличия и выбор УИ, оптимального для разложения определителя – **noideal**.

Выбор минимального числа из двух целых чисел – **minimum**.

Выбор максимального числа из двух целых чисел – **maximum**.

Частные случаи выделения:

Выделение z -петли – **pasloop**.

Выделение z -ветви, параллельной генератору напряжения или приемнику тока – **redur**.

Выделение параллельных норатора и нуллатора – **degener**.

Выделение y -ветви, последовательной генератору тока или приемнику напряжения – **pashang**.

Исключение узла, инцидентного нораторам, генераторам тока, нуллаторам и приемникам напряжения – **nodred**.

Операции с многомерным НУИ:

Нейтрализация многомерного НУИ – **allntr999**.

Нейтрализация k -го источника в многомерном НУИ – **autontr999**.

Нейтрализация оставшихся элементов многомерного НУИ – **ntr999**.

Функция проверки существования многомерного НУИ – **is999**.

Преобразование многомерного НУИ (с одним элементом) в обычный НУИ – **one999**.

Функции эквивалентных преобразований. Сложность анализируемых схем позволяют многократно увеличить упрощающие эквивалентные преобразования схем и формул, выполняемые перед анализом или в процессе анализа.

Упрощения путем введения подформул:

Свертывание последовательных проводимостей – **red2yy**.

Свертывание параллельных сопротивлений – **red2zz**.

Объединение последовательных сопротивлений – **seqzz**.

Объединение параллельных проводимостей – **really**.

Свертывание параллельных проводимости и сопротивления – **reallyz**.

Свертывание последовательных сопротивления и проводимости – **seqzy**.

Объединение ветвей:

Сведение параллельных y -ветвей – **reallf**.

Объединение параллельных ИТУН – **parall**.

Сведение последовательных z -ветвей – **seqzz0**.

Упрощения в единой формуле:

Реализация комплекса схемных упрощений – **simply**.

Удаление третьего элемента в треугольнике с двумя генераторами или приемниками – **triangle**.

Удаление y -ветвей, параллельных генераторам напряжения или приемникам тока – **redu**.

Упрощения при разомкнутых генераторах или приемниках УИ – **ejuihang**.

Упрощения при замкнутых генераторах или приемниках УИ – **actloop**.

Упрощения при параллельном соединении генераторов и приемников УИ – **redpar**.

Упрощения при последовательном соединении генераторов и приемников УИ – **redseq**.

Стягивание z -ветвей, последовательных генераторам тока или приемникам напряжения – **hangr**.

Вспомогательные функции. Введение вспомогательных функций позволяет уменьшить текст программы и сделать его более читаемым.

Нахождение числа узлов и вектора узлов:

Нахождение числа узлов p и вектора узлов в графе (GRAPH) – **verstr**.

Нахождение числа узлов в графе (GRAPH) – **ver_n**.

Нахождение числа узлов p и вектора узлов str схеме (PASSIVE и SOURCE) – **nodestr**.

Нахождение числа узлов в схеме (PASSIVE и SOURCE) – **node_n**.

Нахождение числа узлов и вектора узлов в активной схеме (SOURCE) – **nodes**.

Нахождение числа узлов в активной схеме (SOURCE) – **node**.

Операции со структурами:

Копирование структур PASSIVE – **pascopy**.

Копирование структур SOURCE – **actcopy**.

Преобразование PASSIVE в GRAPH – **copypas**.

Преобразование SOURCE в GRAPH – **copyact**.

Занесение в структуру SOURCE нового вырожденного НУИ (идеального проводника) – **nui**.

Занесение в структуру SOURCE нового НУИ – **nuiall**.

Занесение в структуру SOURCE нового идеального проводника вместо k -го элемента – **nuiintr**.

Занесение в структуру нового УИ – **actall**.

Занесение в структуру нового двухполюсника – **matrall**.

Освобождение структуры PASSIVE – **freematr**.

Освобождение структуры SOURCE – **freeact**.

Освобождение строк в PASSIVE – **frematr**.

Освобождение строк в SOURCE – **freact**.

Освобождение j -го элемента в PASSIVE – **frmatr**.

Освобождение j -го элемента в SOURCE – **fract**.

Функции проверки наличия:

Проверка наличия в схеме емкостной проводимости – **yesgc**.

Проверка наличия в схеме индуктивного сопротивления – **yesrL**.

Проверка наличия в схеме емкостной проводимости или индуктивного сопротивления – **yescl**.

Проверка наличия в схеме проводимостей g, y, Y – **yesg**.

Проверка наличия в схеме сопротивлений R, z, Z – **yesr**.

Проверка наличия в схеме ИТУН – **yess**.

Проверка наличия в схеме НУИ – **yesn**.

Функции диакоптического блока diakop. Делают возможным анализ схем, содержащих десятки-сотни узлов и элементов. Разложение схемного определителя по частям выполняется на основе рекурсивного половинного деления – бисекции схемы.

Раздельная бисекция схем числителя и знаменателя:

Деление графа схемы (GRAPH) по одному узлу – **bond1**.

Проверка независимости подграфов-подсхем, соединенных в одном узле – **indep1**.

Деление графа схемы (GRAPH) по двум узлам – **bond2**.

Проверка независимости подграфов-подсхем с двумя общими узлами – **indep2**.

Деление графа схемы (GRAPH) по трем узлам – **bond3**.

Проверка независимости подграфов-подсхем с тремя общими узлами – **indep3**.

Деление графа схемы (GRAPH) по четырем узлам – **bond4**.

Проверка независимости подграфов-подсхем с четырьмя общими узлами – **indep4**.

Деление графа схемы (GRAPH) по пяти узлам – **bond5**.

Проверка независимости подграфов-подсхем с пятью общими узлами – **indep5**.

Преобразование схемы (PASSIVE и SOURCE) в граф (GRAPH) – **transfor**.

Расчет количества элементов в PASSIVE и SOURCE – **fortest**.

Восстановление из графа (GRAPH) первой или второй подсхемы (PASSIVE и SOURCE) – **fortrans**.

Поиск половинного деления схемы на подсхемы по одному узлу – **bisec1**.

Разложение определителя схемы делением на две подсхемы по выбранному узлу – **form1**.

Поиск половинного деления схемы на подсхемы по двум узлам – **bisec2**.

Разложение определителя схемы делением ее на две подсхемы по выбранным двум узлам – **form2**.

Поиск половинного деления схемы на подсхемы по трем узлам – **bisec3**.

Поиск половинного деления схемы на подсхемы по четырем узлам – **bisec4**.

Поиск половинного деления схемы на подсхемы по пяти узлам – **bisec5**.

Разложение определителя схемы делением ее на две подсхемы по трем, четырем или пяти узлам выполняется одной функцией **form**.

Единая бисекция схем числителя и знаменателя:

Поиск половинного деления схемы на подсхемы по двум узлам – **bis2**.

Проверка независимости подграфов-подсхем с двумя общими узлами – **indep2m**.

Поиск половинного деления схемы на подсхемы по трем узлам – **bis3**.

Проверка независимости подграфов-подсхем с тремя общими узлами – **indep3m**.

Поиск половинного деления схемы на подсхемы по четырем узлам – **bis4**.

Проверка независимости подграфов-подсхем с четырьмя общими узлами – **indep4m**.

Поиск половинного деления схемы на подсхемы по пяти узлам – **bis5**.

Проверка независимости подграфов-подсхем с пятью общими узлами – **indep5m**.

Формирование схемной функции путем деления схемы на две подсхемы, одинаковые для схем числителя и знаменателя, по выбранным двум, трем, четырем или пяти узлам выполняется функциями **form2** и **form** (как при различном выборе подсхем для схем числителя и знаменателя).

В функциях **bis2–bis5** в отличие от функций **bisec2–bisec5** не предусмотрен вызов **form2** и **form**.

Функции ввода-вывода:

Ввод схемы и проверка корректности – **input**.

Разбор строки исходного **cir**-файла (файла схемы) – **extra**.

Удаление из формулы нулевых слагаемых – **ster**.

Занесение очередного элемента (параметра или суммы параметров) в формируемую строку – **printa**.

Занесение очередного элемента в строку сомножителей – **printi**.

Занесение сформированной строки в файл – **control**.

Пакетная обработка **cir-файлов: о новых версиях CIRSYM.**

Последние версии программы CIRSYM, предназначены для повышения эффективности работы системы SCADS и предусматривают пакетную обработку нумерованных **cir**-файлов, которые размещаются в текущем каталоге. При этом в одном **cir**-файле можно размещать несколько **cir**-файлов, разделяемых ключевым словом **.END**. Объединенный **cir**-файл оканчивается ключевым словом **.TOTAL**. Результаты обработки объединенного **cir**-файла помещаются в один выходной файл **out**.

Литература

1. Дмитришин Р. В., Шаповалов Ю. И. Диакоптический алгоритм анализа сложных линейных цепей на ЭВМ // Автоматизация проектирования в электронике. – Киев, 1975. – Вып. 12. – С. 42–46.

2. Матвийчук Я.Н. Разработка метода и программы анализа линейных схем по частям // Теоретическая электротехника.– Львов, 1980.– Вып. 29.– С. 41–52.

3. Березко Л.А., Шаповалов Ю.И. Реализация метода подсхем при символическом анализе линейных схем // Изв. Вузов. Радиоэлектроника.– 1980.– Т. 23, №6.– С. 21–25.

4. Оптимальная реализация линейных электронных *RLC*-схем / А.А. Ланнэ, Е.Д. Михайлова, Б.С. Саркисян, Я.Н. Матвийчук. – Киев: Наукова думка, 1981. – 208 с.

5. Шаповалов Ю.И., Давидюк Р.Д. Особенности реализации метода топологического анализа схем в программе AC13EC // Радиоэлектроника.– 1983.– № 6.– С. 79–81.

6. Ястребов Н.И. Повышение эффективности декомпозиционных алгоритмов символьного анализа // Радиоэлектроника.– 1985.– № 6.– С. 102–104.

7. Филаретов В.В., Кузьмин Н.П., Тимофеев В.Ф. и др. Комплекс программ анализа и оптимизации линейных электронных схем ПОЛЭС: Отчет о НИР / Ульян. политехн. ин-т.– № 0-100/85; № ГР01850073775; Инв. № 0286.0074623.– Ульяновск, 1985.– 68 с.

8. Филаретов В.В. Программа автоматического вывода аналитических выражений для функций аналоговых измерительных преобразователей // 8-я Всесоюз. конф. «Измерительные информационные системы»: Тез. докл.– Ташкент: Политехн. ин-т, 1987.– Ч. 3.– С. 115.

9. Филаретов В.В. Алгоритм формирования экономных буквенно-численных выражений для коэффициентов схемных функций // Электрон. моделирование.– 1987.– № 4.– С. 33–35.

10. Филаретов В.В. Алгоритмы символьно-топологического анализа электрических цепей: Дис. ... канд. техн. наук: 05.09.05 (Теорет. электротехника) / Ленингр. гос. техн. ун-т.– Л., 1990.– 148 с.

11. Филаретов В.В. Программа SYMBOL автоматического вывода передаточной функции электронной схемы произвольной структуры // Алгоритмы и устройства обработки сигналов и автоматизация проектирования.– Таллин: АН Эстонии, 1991.– С. 130–148.

12. Филаретов В.В. Алгоритм приведения развернутых алгебраических выражений топологических функций к скобочной форме и его реализация в программе SYMBOL // Алгоритмы и устройства обработки сигналов и автоматизация проектирования.– Таллин: АН Эстонии, 1991.– С. 149–166.

13. Филаретов В.В. Обобщенный унисторный граф электронной схемы и его анализ // Электричество.– 1993.– № 5.– С. 65–70.

14. Власенко О.Ф., Тухтаров В.Н., Филаретов В.В. Формульный вывод, оптимизационный расчет и графическое отображение передаточных функций электронных схем // Проблемы автоматизированного моделирования в электронике: Тез. докл. международ. конф.– Киев: Общество “Знание” Украины, 1993.– С. 29–30.

15. Филаретов В.В. Метод разложения определителя ненаправленного графа по ребрам с максимальным участием / Ред. журн. “Радиоэлектроника”.– Деп. в ВИНТИ 27.11.85, № 8184-B85.– 12 с.– Реф. в: Радиоэлектроника.– 1986.– № 3.– С. 110.

16. Филаретов В.В. Рекурсивные методы выражения определителя ненаправленного графа // Теорет. электротехника.– Львов, 1986.– Вып. 40.– С. 6–12.

17. Филаретов В.В. Алгоритм разложения определителя ненаправленного графа проводимостей по ребрам, входящим в пути между парами вершин // Теорет. электротехника.– Львов, 1988.– Вып. 45.– С. 28–34.

18. Филаретов В.В. Топологический анализ электронных схем методом выделения ветвей и дуг // Электричество.– 1992.– № 7.– С. 31–37.

19. Филаретов В.В. Исследования Вильгельма Фойснера в области теоретической электротехники // Электричество.– 1992.– № 9.– С. 64–67.

20. Филаретов В.В. Оптимизация формул схемных функций электрических цепей // Электричество.– 1993.– № 9.– С. 64–68.

21. Филаретов В.В. Приоритеты в науке, или еще об одном слагаемом успеха // Электричество.– 1994.– № 12.– С. 63–64.
22. Филаретов В.В., Шеин Д.В. Адаптивный алгоритм и программа генерации сверхкомпактных символьных выражений схемных функций // Проблемы автоматизированного моделирования в электронике: Тез. докл. международ. конф.– Киев: Об-во “Знание” Украины, 1993.– С. 28–29.
23. Филаретов В.В., Шеин Д.В. Машинная генерация оптимальных формул для функций пассивных схем // Проблемы автоматизированного моделирования в электронике: Сб. докл. международ. конф.– Киев: Политехн. ин-т, 1994.– С. 28–32.
24. Филаретов В.В. Синтез оптимальных формул схемных функций электрических цепей // Электричество.– 1995.– № 4.– С. 36–43.
25. Филаретов В.В. Оптимальная реализация дешифраторного метода анализа электрических цепей // Электричество.– 1996.– № 10.– С. 43–49.
26. Filaretov V.V. Improved decoding method for electrical network analysis // Electrical technology.– 1996.– N 4.– P. 41–53.
27. Филаретов В.В. Оптимальный синтез формул схемных функций // Проблемы автоматизированного моделирования в электронике: Сб. докл. международ. конф.– Киев: Политехн. ин-т, 1994.– С. 23–27.
28. Филаретов В.В., Шеин Д.В. Оптимальный синтез символьных функций для сложных схем с операционными усилителями // Проблемы физической и биомедицинской электроники: Сб. докл. международ. конф.– Киев: Политехн. ин-т, 1995.– С. 216–220.
29. Филаретов В.В., Шеин Д.В. Символьная машинная программа вывода формул для функций активных и пассивных электрических цепей // Тез. докл. 30-й науч.-техн. конф.– Ульяновск: Ульян. гос. техн. ун-т, 1996.– Ч. 1.– С. 43–44.
30. Филаретов В.В. Об иерархическом подходе к символьному анализу сложных электронных схем // Проблемы физической и биомедицинской электроники: Сб. докл. международ. конф.– Киев: Национальный техн. ун-т Украины, 1996.– С. 132–136.
31. Филаретов В.В. Схемный подход к символьному анализу активных электрических цепей // Электроника и связь: Науч.-техн. сб.– Киев, 1997.– Вып. 2.– Ч. 1.– С. 97–101.
32. Филаретов В.В. Топологический анализ электронных схем методом выделения параметров // Электричество.– 1998.– № 5.– С. 43–52.
33. Filaretov V.V. A topological analysis of electronic circuits by a parameter extraction method // Electrical technology.– 1998.– N2.– P. 46–61.
34. Филаретов В.В., Шеин Д.В. Компьютерный интеллектуальный справочник схемных функций активных электрических цепей // Методы и средства преобразования и обработки аналоговой информации: Тез. докл. международ. конф.– Ульяновск: Ульян. гос. техн. ун-т, 1999.– С. 13.
35. Филаретов В.В. Формирование символьных функций для активных электрических цепей методом стягивания и удаления ветвей // Электричество.– 2001.– № 4.– С. 43–51.

36. Филаретов В. В. Метод двоичных векторов для топологического анализа электронных схем по частям // *Электричество*.—2001.—№8.—С.33–42.

37. Филаретов В.В., Шеин Д.В. Формирование, интерпретация и компиляция символьных функций электронных схем // *Логико-алгебраические методы, модели, прикладные применения: Тр. междунар. конф.*— Ульяновск: Ульян. гос. техн. ун-т, 2001.—С. 10–12.

38. Филаретов В.В. Анализ электрических цепей методом схемных определителей: Метод. указания к практическим занятиям по электротехнике и электронике.— Ульяновск: УлГТУ, 2001.— 40с.

39. Филаретов В. В. Топологический анализ электрических цепей на основе схемного подхода: Дис. ... докт. техн. наук 05.09.05 (Теоретическая электротехника) / Ульяновский гос. техн. ун-т, Санкт-Петербургский гос. техн. ун-т. – Ульяновск–Санкт-Петербург, 2002. – 265 с.

40. Filaretov V.V., Gorshkov K.S. Friedrich Wilhelm Feussner und die Methode der Schemadeterminanten // *Synthesis, analysis and diagnosis of electronic circuits: Proceedings of international conference CLIN–2007.* – Ulyanovsk: ULSTU, 2007. – Т. 3.– P. 207–222.

41. Горшков К.С., Филаретов В.В. Жизнь и деятельность Вильгельма Фойснера, основоположника схемного подхода к анализу цепей // *Синтез, анализ и диагностика электронных цепей: Международный сб. науч. тр. / Под ред. В.В. Филаретова.* – Ульяновск: УлГТУ, 2009. – Вып. 7. – С. 200–216.

42. Горшков К.С., Филаретов В.В. Схемный подход Вильгельма Фойснера и метод схемных определителей / Под ред. В.В.Филаретова.— Ульяновск: Изд-во УлГТУ, 2009.— 189 с.

43. Горшков К.С., Токарев Ю.В., Филаретов В.В. Анализ и синтез линейных электрических цепей методом схемных определителей: Учебное пособие.— Ульяновск: УлГТУ, 2008.— 240 с.

44. Filaretov V.V., Korotkov A.S. Extension of Feussner's method to general active networks // *Direct topological models for active networks: Synthesis and analysis: Proceedings of international conference «Continual algebraic logic, calculus and neuralinformatics in science, technics and economics» (CLIN–2003).*— Ulyanovsk: ULSTU, 2003.— P. 95–101.

45. Filaretov V.V., Korotkov A.S. Generalized parameter extraction method in network symbolic analysis // *Proceedings of the European conference on circuit theory and design (ECCTD–2003).*— Kraków, Poland, 2003.— Vol. 2.— P. 406–409.

46. Filaretov V.V., Korotkov A.S. Generalized parameter extraction method in case of multiple excitation // *Proceedings of the 8–th international workshop on Symbolic Methods and Applications in Circuit Design.*—Wroclaw (September 23–24).—2004.—P. 8–11.

47. Филаретов В.В. Состояние и проблемы развития схемного подхода к символьному анализу электрических цепей // *Синтез, анализ и диагностика электронных цепей: Тр. междунар. конф. КЛИН–2007.* – Ульяновск: УлГТУ, 2007. – Т. 3.– С. 196–207.

48. Курганов С.А., Полях О.И., Филаретов В.В., Ястребов Н.И. От деревьев к нулловым схемам // Синтез, анализ и диагностика электронных цепей: Международный сб. науч. тр. / Под ред. В.В. Филаретова. – Ульяновск: УлГТУ, 2009. – Вып. 7. – С. 277–321.

49. Горшков К.С., Филаретов В.В. Схемно-символьный анализ электрических цепей с многомерными управляемыми источниками // Синтез, анализ и диагностика электронных цепей: Международ. сб. науч. тр. / Под ред. В.В. Филаретова. – Ульяновск: УлГТУ, 2008. – Вып. 6. – С. 72–79.

50. Курганов С.А. Символьный подход к решению задачи диагностики электрических цепей // Электричество.– 2002.– № 8.– С. 49–52.

51. Вольнов В.Е., Курганов С.А., Филаретов В.В. Символьный анализ сложных электрических цепей и сетей с помощью программы *CIRSYM* // Схемно-топологические модели активных электрических цепей: Синтез, анализ, диагностика: Тр. международ. конф. КЛИН–2004.–Ульяновск: УлГТУ, 2004.– Т. 4.– С. 27–30.

52. Шеин Д.В. Ликвидатор избыточных скобок и единиц в сложных алгебраических выражениях // Схемно-алгебраические модели активных электрических цепей: Синтез, анализ, диагностика: Тр. международ. конф. КЛИН–2006. – Ульяновск: УлГТУ, 2006. – Т. 3. – С. 153–161.

53. Курганов Д.С., Филаретов В.В. Алгоритм и программа формирования полиномиальных схемных функций электронных цепей // Синтез, анализ и диагностика электронных цепей: Международный сб. науч. тр. / Под ред. В.В. Филаретова. – Ульяновск: УлГТУ, 2008. – Вып. 6. – С. 152–165.

54. Березуев Р.И., Курганов С.А., Филаретов В.В., Шеин Д.В. SCAD – система символьного анализа и диагностики линейных электронных цепей // Схемно-топологические модели активных электрических цепей: Синтез и анализ: Тр. международ. конф. КЛИН–2005. – Ульяновск: УлГТУ. – 2005. – Т. 3. – С. 3–11.

55. Смирнов В.С., Филин В.А. Программа FASTMEAN для моделирования электронных цепей // Схемно-алгебраические модели активных электрических цепей: Синтез, анализ, диагностика: Тр. международ. конф. КЛИН–2005. – Ульяновск: УлГТУ. – 2005. – Т. 3. – С. 174–176.

56. Березуев Р.И., Курганов С.А., Филаретов В.В., Шеин Д.В. Символьный анализ и диагностика электронных цепей // Государственный координационный центр информационных технологий.– №ОФАП 3981; № ГР 50200401291. – 2004.

57. <http://levul.org/sm/symbolic.htm> – Официальный сайт системы *Symbolic Circuit Analysis, Diagnosis and Synthesis*.

58. <http://www.fastmean.ru> – Смирнов В. С., Филин В. А. Программа моделирования электрических цепей *FASTMEAN*.

59. <http://www.eng.uts.edu.au/~benr/symbolic/> – Rodanski B.S. *Symbolic Circuit Analysis: Library of Benchmark Circuits*.

60. <http://www.cirlab.unifi.it/Sapwin> – *Symbolic Analysis Program for Windows*.
61. http://ru.wikipedia.org/wiki/Фойснер,_Фридрих_Вильгельм
62. http://ru.wikipedia.org/wiki/Feussner,_Friedrich_Wilhelm
63. http://ru.wikipedia.org/wiki/Метод_схемных_определителей
64. http://ru.wikipedia.org/wiki/Method_circuit_determinants