

MathSpice – аналитический PSpice-движок для OrCAD и MicroCAP

Часть 1. Загрузка пакета MSpice и получение аналитического решения в Maple

Олег Петраков (Москва)

Пакет расширений Maple под названием MathSpice (MSpice) предназначен для аналитического решения электронных цепей и функциональных схем, но может быть использован как инструмент создания Spice-моделей сигналов и электронных приборов для различных симуляторов. Предлагаемый цикл статей посвящён применению MSpice для автоматизации аналитических расчётов.

ВВЕДЕНИЕ

Чтобы получить правильные аналитические выражения для практически ценных цепей, необходимо приложить немало труда и потратить изрядное количество времени, даже в случае использования MathCAD. Применение MathSpice для этих целей в корне меняет ситуацию: достаточно без ошибок нарисовать схему в MicroCAP или OrCAD Capture (Schematics), несколько секунд ожидания, и все решения у вас на экране.

Программа PSpice не является языком официальной математики и предназначена для написания заданий для численного моделирования схем. Однако MathSpice транслирует PSpice в математическую нотацию Maple, т.е. математические задачи теперь можно ставить на языке PSpice. Поскольку электронные САПР обычно генерируют NET-лист в формате PSpice, то аналитические задачи можно рисовать! Следует отметить, что однажды решённая задача является шаблоном для решения всех задач данного типа, что позволяет отказаться от хранения множества документов с расчётами и хранить только схемы.

Символьные расчёты в электронике весьма эффективны, но требуют феноменального трудолюбия. Это – серьёзная преграда для их оперативного использования. В процессе расчётов легко совершить опisku, и тогда весь последующий труд пропадёт. Всё изменилось с появлением программы MathSpice: чтобы получить решение, схему достаточно на-

рисовать в графическом редакторе MicroCAP или OrCAD. После запуска MSpice вы сразу получаете аналитические выражения для всех напряжений и токов. Дальнейшая работа сводится к анализу этих выражений либо производных от них (коэффициентов передачи, входных или выходных сопротивлений).

Поскольку MSpice и PSpice используют один и тот же источник исходных данных, это позволяет осуществлять высочайший уровень контроля достоверности получаемых аналитических результатов. Если у вас возникли сомнения в полученных формулах – запустите PSpice-симулятор MicroCAP (или OrCAD) с той же схемой: сравнение вернёт вам уверенность и убержёт от публикации ошибочных результатов.

Программа MathSpice имеет три варианта: ElectronicsSolver, ESolver и MSpice. ElectronicsSolver поддерживает базовые модели OrCAD или MicroCAP и достаточно для решения практически всех задач курсового и дипломного проектирования. В принципе можно решать все задачи при помощи ESolver Pro, но в нём меньше сервиса и требуется больше работы руками. ESolver – профессиональная версия ElectronicsSolver – имеет расширенную поддержку моделей и функций OrCAD или MicroCAP и подходит для свободного творчества технических писателей, дипломников и диссертантов. Множество стандартных моделей MicroCAP и OrCAD позволяют мгновенно «лепить» проекты. Пакет MSpice предна-

значен для «пашущих целину» разработчиков (поддерживаются все функции предыдущих пакетов), специалистов в области моделирования нелинейных цепей и дифференциальных уравнений; подходит создателям моделей приборов для симуляторов различного типа.

Программа MSpice является не столько альтернативой, сколько дополнением Spice-систем. Очевидно, что система символьной математики обеспечивает большие возможности анализа схем. Там, где это возможно, MSpice получает аналитический отклик цепи или его аналог в виде системы дифференциальных уравнений для последующего математического анализа. Там, где это бесперспективно, задача будет решаться численными методами.

Применение MSpice вместо PSpice может быть оправдано при решении задач, которые требуют особо высокой точности решения, поскольку в Maple ограничений на точность практически не существует. Имея решение в Maple, вы можете воспользоваться инструментами математического анализа высочайшего класса, разработанными для математиков-профессионалов. MSpice позволяет преобразовывать аналитические выражения Maple в формат PSpice и обратно, что превращает его в инструмент разработчика PSpice-моделей.

Программы работают совместно с OrCAD или MicroCAP, в которых вы рисуете схемы и создаёте NET-лист. Программы читают NET-лист, формируют уравнения и решают их относительно всех переменных. Дальнейшие расчёты могут выполняться произвольно или с использованием набора специализированных функций, входящих в состав пакета.

С ростом сложности задачи время выполнения возрастает экспоненциально, поэтому, если задачу можно разделить на независимые части (выпол-

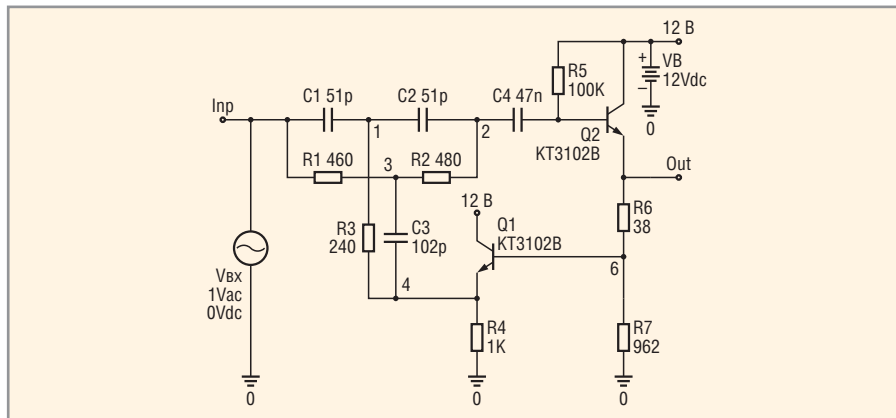


Рис. 1. Схема узкополосного режекторного фильтра на частоту 6,5 МГц

нить декомпозицию), это следует сделать. Разумная сложность задачи ограничена, на наш взгляд, 30 – 50 компонентами, чтобы не переусложнять выходные формулы. Слишком большая схема, решённая аналитически, имеет грандиозный размер решения! Вряд ли кого интересуют решения в виде формул на 20 страницах, что возможно уже при наличии в схеме с обратными связями 20 компонентов. Всегда интересно решить задачу просто или упростить сложное решение до простого.

Интерес для аналитического анализа представляют, как правило, функциональные узлы и каскады, на которые можно разделить сложную схему. Поэтому перед решением задачи в ElectronicSolver необходимо предельно упростить схему узла и избавиться от всех несущественных элементов, которые загромождают результат в виде формулы (потом это будет гораздо сложнее). Целесообразно предельно коротко обозначать компоненты и узлы, поскольку их названия войдут в состав формул.

Более подробную информацию можно найти на интернет-странице www.pspicelib.narod.ru. Для загрузки пакета расширений MSpice используется следующая конструкция Maple:

```
> read('F:/PMAPLE/ESolver.m');
with(MSpice):
```

MSpice состоит всего из двух основных директив:
 ESolve() – чтение NET-листа схемы, вывод системы уравнений цепи, вывод решения;
 Values() – ввод номиналов компонентов схемы.

Остальные директивы имеют характер Smart-утилит, ускоряющих выполнение некоторых типовых задач. Чтобы упростить дальнейшее вос-

приятие, приведём типовой пример с расчётами.

ПРИМЕР РАСЧЁТА ТРАНЗИСТОРНОГО ФИЛЬТРА

В качестве узкополосных режекторных фильтров используются резисторно-конденсаторные двойные T-образные мостовые фильтры. Недорогой фильтр на высокие частоты можно сделать на транзисторах (см. рис. 1). Однако расчёт транзисторных схем гораздо сложнее, чем схем на операционных усилителях. Применение программы MSpice устраняет эту проблему, позволяя быстро получить необходимые формулы.

Большинством паразитных параметров транзисторов можно пренебречь много ниже частоты единичного усиления, поэтому для расчётов использовалась простейшая схема замещения транзистора (см. рис. 2).

```
> restart:read('F:/PMAPLE/ESolver.m');with(MSpice):
```

Применение директивы ESolve() для вызова NET-листа схемы фильтра:

```
> Devices:= [Одинаковые, [BJT,DC1,2]] :
ESolve(Q, 'BJT-PSpiceFiles/SCHEMATIC1/SCHEMATIC1.net') :
```

DC1 модель ВТТ ТРАНЗИСТОРА

Система Кирхгофа-Лапласа

$$\frac{V6 - VOUT}{R6} + \frac{(V5 - VOUT)\beta}{R\beta} - \frac{VOUT - V5}{R\beta},$$

$$H := \frac{(R^2 C^2 R6 + C^2 R^2 R7)s^2 + R6 + R7}{(R^2 C^2 R6 + C^2 R^2 R7)s^2 + 4sCRR6 + R6 + R7}$$

$$-\frac{V4}{R4} + (V3 - V4)sC3 - \frac{V4 - V1}{R3} + \frac{(V6 - V4)\beta}{R\beta} - \frac{V4 - V6}{R\beta},$$

$$(V5 - V2)sC3 - \frac{V3 - V2}{R2} - (V2 - V1)sC2,$$

$$\frac{V4 - V1}{R3} + (V2 - V1)sC2 -$$

$$-(V1 - V6x)sC1,$$

$$\frac{V6x - V3}{R1} - \frac{V3 - V2}{R2} - (V3 - V4)sC3,$$

$$\frac{VOUT - V5}{R\beta} - \frac{V5 - VB}{R5} - (V5 - V2)sC4$$

$$-\frac{V6}{R7} + \frac{V4 - V6}{R\beta} - \frac{V6 - VOUT}{R6}$$

Решения

{VOUT,V1,V2,V5,V4,V6,V3}

MSpice v8.27

Заданы узлы: {V12V, VINP}

Источники: [V6x, VB, J\beta]

Решения V_NET: {VOUT, V1, V2, V5, V4, V6, V3}

J_NET: {J\beta, JC3, JR6, JR2, JR3, JC2, JR7, JC1, JR\beta, JFt, J\beta, JV6x, JR5, JVB, JC4, JR1, JR4, Jk, JT}

Для упрощения формул учтём, что для фильтра с мостом Вина должны выполняться следующие соотношения:

```
> C1:=C: C2:=C: C3:=2*C: R1:=R: R2:=R: R3:=R/2:
```

Найдём передаточную функцию фильтра, предположив, что $\beta = \infty$, $C4 = \infty$, $R5 = \infty$. Это позволяет получить простые формулы для предварительного расчёта. Точные формулы с помощью программы Maple получить можно, но они будут очень сложными. При настройке параметры схемы (добротность) легко скорректировать подбором резистора R6.

```
> VB:=0: beta:=x: C4:=x: R5:=x:
H:=collect(limit(VOUT/V6x,x=infinity),s);
```

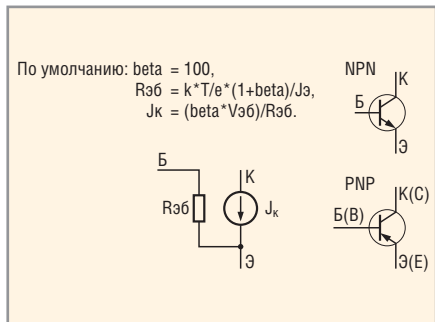


Рис. 2. DC1-модель биполярного транзистора

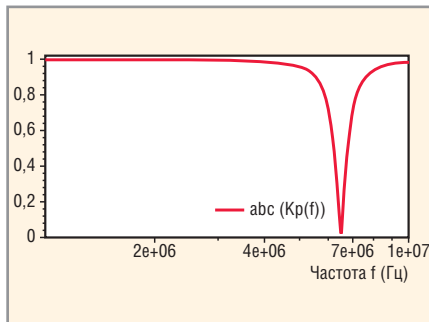


Рис. 3. АЧХ режекторного фильтра

Теперь найдём коэффициент передачи в частотной области, $K = K(f)$, выполнив подстановку $s = I \cdot 2 \cdot \pi \cdot f$:

```
>
K:=simplify(subs(s=I*2*Pi*f,H))
;
```

$$K := \frac{4\pi f^2 R^2 C^2 R_6 + 4\pi f^2 C^2 R^2 R_7 - R_6 - R_7}{4\pi f^2 R^2 C^2 R_6 + 4\pi f^2 C^2 R^2 R_7 - 8I\pi f C R R_6 - R_6 - R_7}$$

Частота режекции:

```
> Fp:=I*solve(diff(K,f)=0,f)[2];
```

$$Fp := \frac{1}{2\pi C R}$$

Полоса режекции по уровню -3 дБ:

```
> solve(evalc(abs(K))=subs(f=0,K)/sqrt(2),f): П:=simplify(%[4]-%[2]);
```

$$\Pi := \frac{2R_6}{\pi(R_7 + R_6)C R}$$

Добротность определяется как $Q = Fp / \Pi$, отсюда

```
> Q:=Fp/П;
```

$$Q := \frac{R_7 + R_6}{4R_6}$$

Построим АЧХ коэффициента передачи фильтра:

```
Values(AC,RLCVI,[1]); C:=C1;
R:=R1;
```

Ввод номиналов компонентов:

$$H := \frac{0.5992704 \cdot 10^{-12} s^2 + 1000}{0.5992704 \cdot 10^{-12} s^2 + 0.374096 \cdot 10^{-5} s + 1000}$$

$$[F_p = 0,650142107, Q = 6,578947368, \Pi = 988216,9668]$$

На рисунке 3 приведена АЧХ режекторного фильтра. На этом решение нашей задачи заканчивается. Ручной труд, при котором возможен ошибоч-

```
R5 := .100e6, "100к"
C4 := .47e-7, "47н"
R4 := .1e4, "1к"
R1 := 480, "480"
C1 := .51e-10, "51п"
R6 := 38, "38"
R2 := 480, "480"
R7 := 962, "962"
R3 := 240, "240"
C2 := .51e-10, "51п"
C3 := .102e-9, "102п"
beta := 100, "100"
`Fт` := .100e6, "100e3"
`Jэ` := .1e-2
`Rэб` := k*T/e*(beta+1)/`Jэ`
`Rэб` := k*T/e*(beta+1)/`Jэ`
e := .1602176462e-18
k := .1380650277e-22
T := 300, "300"
AC источник: DC: Vвх:=0 AC:
Vвх:=1 Pphase(degrees):=0
DC источник: DC: VB:=12 AC: VB:=0
H_Q2_КТ3102В := (V5-VOUT)*beta/`Rэб`
H_Q1_КТ3102В := (V6-V4)*beta/`Rэб`
> H:=H; ['Fp'=evalf(Fp),
'Q'=evalf(Q), 'П'=evalf(П)];
H SF ([H], f=1e6..10e6, "3)
semi[abs(Kp(f))] $500 режекторного
фильтра |Kp(f)|";
```

ный ввод, свёлся к рисованию схемы. В следующей статье мы продемонстрируем описание управляющих директив MSpice, которые делятся на основные и вспомогательные. ☺

Продолжение следует