

Квазиоптимальный по времени алгоритм проектирования аналоговых цепей

А.М. Земляк

Национальный технический университет Украины «КПИ», Украина, Киев

Автономный университет Пуэбла, Мексика, Пуэбла, azemliak@yahoo.com

Аннотация — Разработанная ранее методология проектирования аналоговых цепей, базирующаяся на применении теории управления, является основой для конструирования оптимального или квазиоптимального алгоритма проектирования. Важнейшим критерием при этом, позволяющим выявить необходимую структуру алгоритма, является поведение функции Ляпунова, определенной для процесса оптимизации цепи. Характеристики функции Ляпунова и ее производной являются основой для поиска оптимальной структуры управляющего вектора, определяющего, в свою очередь, структуру алгоритма. Построена структурная схема квазиоптимального алгоритма, реализующего основные идеи разработанной ранее методологии, и определены основные характеристики этого алгоритма в сравнении с традиционным подходом и с квазиоптимальной стратегией проектирования.

Ключевые слова — аналоговые цепи, обобщенная методология оптимизации, функция Ляпунова, квазиоптимальный алгоритм.

I. ВВЕДЕНИЕ

Идея отказа от выполнения законов Кирхгофа при проектировании электронных цепей была высказана в работе [1] и реализована практически в двух системах проектирования [2]-[3]. Наиболее общий подход был реализован при разработке обобщенной методологии процесса оптимизации электронной цепи, определенного как управляемая динамическая система [4]-[5]. Эта система определяется дифференциальным или разностным уравнениями для переменных состояния и системой ограничений, в качестве которых выступает математическая модель электронной цепи [4].

Будем считать, что процесс проектирования электронной цепи сформулирован как задача минимизации некоторой, не отрицательно определенной целевой функции $C(X)$. Предполагается при этом, что в точке минимума целевой функции $C(X)$ достигаются все цели проектирования.

II. ПОСТАНОВКА ЗАДАЧИ

В случае применения дифференциальной формы для оптимизационной процедуры система уравнений для переменных состояния в процедуре оптимизации может быть записана [5] в виде:

$$\frac{dx_i}{dt} = f_i(X, U), \quad i = 1, 2, \dots, N, \quad (1)$$

где N – общее число переменных в задаче оптимизации электронной цепи, $U = (u_1, u_2, \dots, u_M)$ – вектор управляющих функций, $u_j \in \Omega$, $\Omega = \{0; 1\}$. Система ограничений в процедуре оптимизации, являющаяся по смыслу математической моделью электронной цепи, может быть определена следующей системой уравнений:

$$(1 - u_j) g_j(X) = 0, \quad j = 1, 2, \dots, M, \quad (2)$$

где M – число зависимых переменных, совпадающее с числом узлов схемы.

Функции $f_i(X, U)$ определяются тем или иным методом оптимизации и, в частности, для градиентного метода оптимизации функции $f_i(X, U)$ задаются следующим образом:

$$f_i(X, U) = -\frac{\delta}{\delta x_i} F(X, U), \quad (3)$$

$$i = 1, 2, \dots, K$$

$$f_i(X, U) = -u_{i-K} \frac{\delta}{\delta x_i} F(X, U) + \frac{(1 - u_{i-K})}{dt} [-x_i' + \eta_i(X)],$$

$$i = K + 1, K + 2, \dots, N, \quad (3')$$

где K – число независимых переменных в традиционной постановке задачи ($N = K + M$), функция $\eta_i(X)$, записанная в неявном виде, определяет

текущее значение переменной x_i из системы (2), $(x_i = \eta_i(X))$, x_i' – предыдущее значение переменной x_i . Функция $F(X, U)$ является обобщенной целевой функцией процесса оптимизации цепи и может быть определена следующим выражением:

$$F(X, U) = C(X) + \frac{1}{\varepsilon} \sum_{j=1}^M u_j g_j^2(X). \quad (4)$$

В такой постановке задача оптимизации цепи формулируется как управляемая динамическая система, которую надо привести в состояние равновесия. При этом время переходного процесса для системы ассоциируется с временем оптимизации электронной цепи. В этом случае основным инструментом является управляющий вектор U , который меняет внутреннюю структуру задачи оптимизации цепи (1)-(4) и осуществляет перераспределение затрат процессорного времени между двумя основными блоками задачи (1) и (2).

III. КВАЗИОПТИМАЛЬНЫЙ АЛГОРИТМ

Отметим две характерные стратегии оптимизации. Первая стратегия определяется управляющим вектором $U = (0, 0, \dots, 0)$. Эта стратегия соответствует традиционному подходу к оптимизации цепи, и в этом случае система (2), являющаяся системой ограничений, должна решаться на каждом шаге процедуры оптимизации. Назовем эту стратегию Традиционной Стратегией Оптимизации (ТСО). Вторая стратегия определяется управляющим вектором $U = (1, 1, \dots, 1)$. В этом случае система (2) исчезает, но информация о цепи появляется в обобщенной целевой функции (4). Эту стратегию назовем Модифицированной Традиционной Стратегией Оптимизации (МТСО). Блок-схемы как ТСО, так и МТСО изображены на рис. 1 и 2.

В работе [6] были проанализированы динамические свойства процесса проектирования на основе введенной функции Ляпунова процесса оптимизации. Было отмечено наличие выраженной корреляции между процессорным временем оптимизации цепи и свойствами функции Ляпунова процесса оптимизации. В этой работе было показано, что функцию Ляпунова можно определить на основе обобщенной целевой функции процесса оптимизации $F(X, U)$ посредством следующей формулы:

$$V(X, U) = [F(X, U)]^r, \quad (5)$$

где степень $r > 0$.

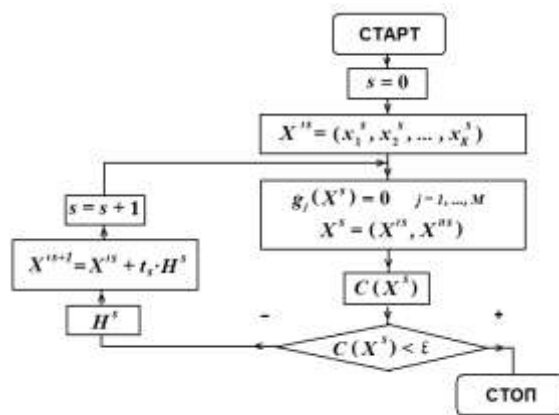


Рис. 1. ТСО

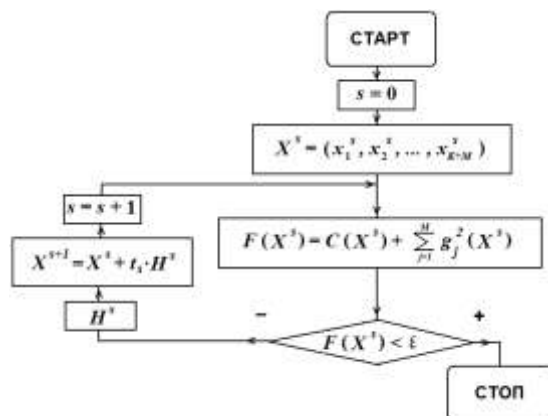


Рис. 2. МТСО

Вопрос устойчивости траектории проектирования связан с анализом поведения производной по времени

от функции Ляпунова \dot{V} . Более информативной, однако, является нормированная производная по времени, задаваемая выражением $W = \dot{V}/V$.

Было показано [7], что на основе анализа поведения производной функции Ляпунова можно найти оптимальные точки переключения управляющего вектора U , являющегося основой квазиоптимальной стратегии проектирования и позволяющего минимизировать процессорное время.

Для построения квазиоптимальной стратегии требуется выбрать как минимум две различные стратегии, одна из которых принадлежит подмножеству стратегий, имеющих траектории подобные МТСО, а другая принадлежит подмножеству стратегий, имеющих траектории подобные ТСО [7].

Определим элементарные структуры, из которых будет строиться блок-схема квазиоптимального алгоритма. Таких структур две. Первую структуру определим как традиционную стратегию оптимизации

для одного шага оптимизационной процедуры (ТСО₁). Эта структура включает:

1) Один шаг процедуры безусловной оптимизации в пространстве R^K независимых переменных

$$X'^{s+1} = X'^s + t_s \cdot H^s, \quad (6)$$

где $X' \in R^K$, а вектор H есть направление уменьшения целевой функции $C(X)$, определяемое тем или иным методом спуска. Вектор H является функцией целевой функции $C(X)$ и его компонентами являются правые части системы (1).

2) Решение системы нелинейных уравнений, являющейся математической моделью электронной цепи

$$g_j(X) = 0, \quad j = 1, 2, \dots, M. \quad (7)$$

В результате воспроизведения обоих шагов мы получаем новые значения всех координат вектора X .

Вторая элементарная структура представляет собой модифицированную традиционную стратегию оптимизации для одного шага оптимизационной процедуры (МТСО₁) и включает один шаг процедуры безусловной оптимизации в пространстве R^N независимых переменных

$$X^{s+1} = X^s + t_s \cdot H^s, \quad (8)$$

где H есть направление уменьшения обобщенной целевой функции $F(X)$.

Вектор H является векторной функцией целевой функции $F(X)$. В результате воспроизведения этого шага мы получаем новые значения всех координат вектора X .

На основе идей, представленных в работе [7] и определенных данных выше, разработан один из возможных вариантов квазиоптимального алгоритма. На рис. 1 представлены блок-схемы алгоритмов ТСО и МТСО, соответственно. Понятно, что ТСО₁ и МТСО₁ реализуют свою собственную стратегию на одном шаге соответствующего метода оптимизации. В дальнейшем будем считать, что эти стратегии оформлены в виде отдельных блоков и могут быть использованы в вызывающих алгоритмах.

Блок-схема квазиоптимального алгоритма представлена на рис. 3.

Предполагается, что данный вариант алгоритма основан только на двух переключениях управляющего вектора.

Задаются начальные точки переключения управляющего вектора S_1 и S_2 , где, например, $S_1=1$ и $S_2=S_1+n$. Параметр n , в свою очередь, может принимать значения 1, 2, Алгоритм начинается с МТСО, которая выполняется до момента переключения на ТСО в точке S_1 . Затем выполняется ТСО до следующего переключения на МТСО в точке S_2 . Далее выполняется МТСО до момента $T_1 + j_{max}$, где T_1 задается числом в диапазоне 10 – 40, а j_{max} – числом в диапазоне 20 – 80.

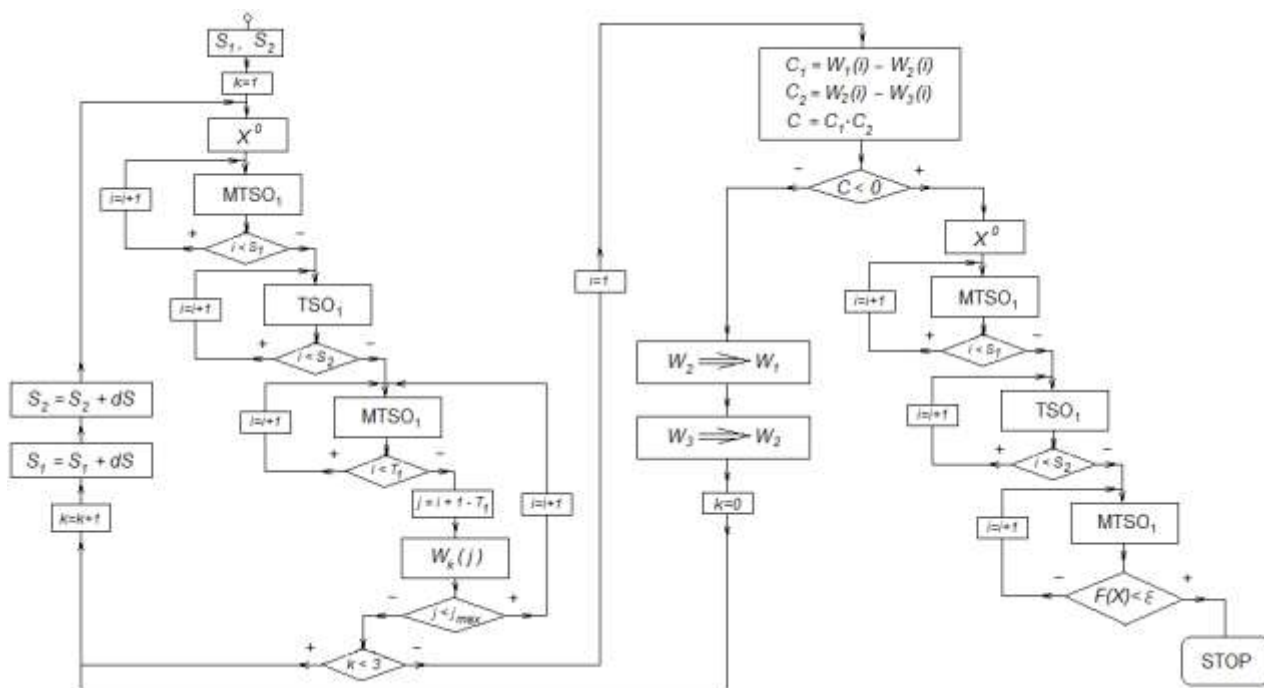


Рис. 3. Структура квазиоптимального алгоритма

В диапазоне изменения счетчика числа шагов оптимизационной процедуры от T_1 до T_1+j_{max} рассчитываются и запоминаются значения относительной производной функции Ляпунова $W_j(t)$.

Расчет повторяется при сдвиге точек переключения на шаг dS . Вновь запоминаются значения относительной производной функции Ляпунова, но это уже $W_2(t)$. Расчет повторяется при следующем сдвиге на шаг dS и определяется $W_3(t)$. Затем производится анализ полученных результатов, и вырабатывается значение основного критерия C . Подобный подход, реализуемый при оптимизации цепей, позволяет уменьшить затраты машинного времени в несколько сот раз по сравнению с традиционным подходом [8].

IV. ОСНОВНОЙ КРИТЕРИЙ

Рассмотрим детально результаты анализа процесса оптимизации цепи, изображенной на рис. 4.

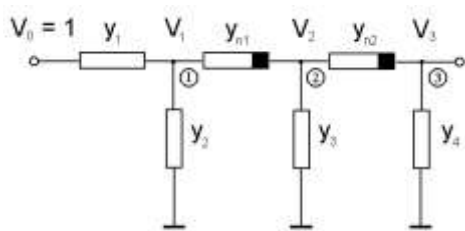


Рис. 4. Нелинейная цепь с тремя узлами

Нелинейные элементы определены следующими зависимостями $y_{n1} = a_{n1} + b_{n1} \cdot (V_1 - V_2)^2$, $y_{n2} = a_{n2} + b_{n2} \cdot (V_2 - V_3)^2$. Вектор X включает семь компонент: $x_1^2 = y_1$, $x_2^2 = y_2$, $x_3^2 = y_3$, $x_4^2 = y_4$, $x_5 = V_1$, $x_6 = V_2$, $x_7 = V_3$. Модель цепи (2) включает три уравнения ($M=3$), и оптимизационная процедура (1) включает семь уравнений. Цепь характеризуется тремя зависимыми переменными, и управляющий вектор состоит из трех компонент: $U = (u_1, u_2, u_3)$. Целевая функция определена формулой $C(X) = (V_3 - V_{out})^2$, где V_{out} – заданное напряжение на выходе цепи. Полный структурный базис стратегий проектирования содержит 8 различных стратегий со следующими значениями управляющего вектора: (000), (001), (010), (011), (100), (101), (110), (111).

Принимая во внимание ранее сформулированные условия поиска квазиоптимальной структуры управляющего вектора [8], определим структуру квазиоптимальной стратегии, состоящей из двух частей. Первая часть определяется управляющим вектором (111) и соответствует МТСО, вторая часть определяется управляющим вектором (000), что соответствует ТСО. Нахождение оптимальной точки переключения является основной целью анализа. Последовательное изменение точки переключения осуществлялось, начиная со 2-го шага оптимизационной процедуры (1) и до 20-го.

Результаты детального исследования зависимости процессорного времени от точки переключения в окрестности оптимальной точки приведены в табл. 1.

Таблица 1

Зависимости процессорного времени от точки переключения для схемы с тремя узлами

N	Точка перекл.	Число итераций	Процессорное время (сек)
1	6	8409	0,659
2	7	6408	0,502
3	8	3141	0,246
4	9	1234	0,096
5	10	3310	0,259
6	11	5918	0,464
7	12	7404	0,581

Компьютерные программы реализованы на языке C++. Компилятор этого языка содержит операторы, позволяющие рассчитать время выполнения как всей программы, так и отдельных ее частей.

Видно, что точка переключения, соответствующая 9-му шагу итерационной процедуры, является оптимальной. Точки переключения, как до, так и после оптимальной, приводят к возрастанию полного процессорного времени проектирования. Поведение функций $V(t)$ и $W(t)$ в течение процесса проектирования, соответствующее данным этой таблицы, показано на рис. 5 для семи различных точек переключения, находящихся вблизи оптимальной.

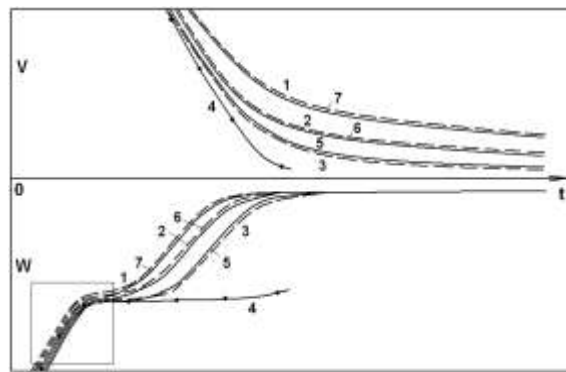


Рис. 5. Поведение функций $V(t)$ и $W(t)$ в течение процесса оптимизации для семи последовательных значений номера шага точки переключения (с 6 до 12)

Пунктирные линии соответствуют зависимостям, полученным для точек переключения, находящихся до оптимальной точки, а сплошные линии соответствуют точкам переключения, находящимся за оптимальной. Кривая 4, соответствующая оптимальной точке переключения, отмечена дополнительными точками. Зависимости функций $W(t)$ на начальном интервале процесса проектирования, выделенном прямоугольником, в большом масштабе приведены на рис. 6. Можно отметить сильную корреляцию между поведением функции $W(t)$ и полным процессорным

временем оптимизации цепи. В случае, когда номер шага n , соответствующий точке переключения управляющего вектора, находится достаточно далеко от оптимального, соответствующего номеру шага 9, поведение функции $W(t)$ при различных точках переключения определяется простым сдвигом вправо кривой, описывающей эту функцию.

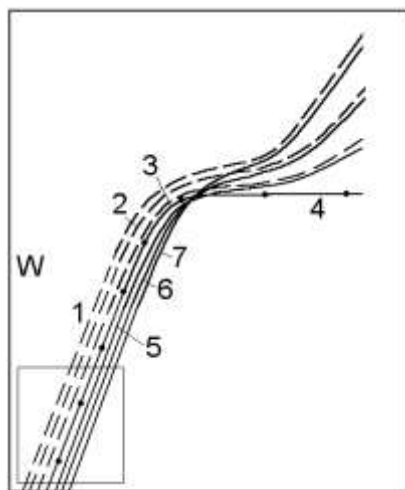


Рис. 6. Поведение функции $W(t)$ в течение процесса оптимизации

При $n < 7$ наблюдается семейство как бы «параллельных» кривых, в том смысле, что сдвинутые кривые не пересекаются. При приближении слева к оптимальной точке переключения, соответствующей шагу 9, кривые по-прежнему не пересекаются, однако они отходят друг от друга на значительно большее расстояние в области, предшествующей достижению функции Ляпунова $V(t)$ минимальных значений (кривые 1, 2 и 3, рис. 5). При этом справедливым является вывод, что чем ниже проходит функция $W(t)$ для некоторой стратегии (т.е. чем больше ее абсолютное значение), тем меньшее процессорное время соответствует этой стратегии, что видно из табл. 1. Кривая 4 соответствует оптимальной точке переключения, и функция $W(t)$ при этом проходит ниже остальных. При дальнейшем увеличении номера шага точки переключения зависимость $W(t)$ изменяется существенно. Кривые 5, 6, 7, соответствуют точкам переключения, находящимся на шагах, превышающих оптимальное значение, и в этом случае наблюдается взаимопересечение кривых. Кривые 5, 6, 7, начиная с некоторого момента проходят выше кривой 4, т.е. оптимальная точка переключения соответствует наибольшему по абсолютной величине значению функции $W(t)$. Это означает, что переключение с одной стратегии на другую после оптимальной точки приводит к качественному изменению в поведении функции $W(t)$, т.е. оптимальная точка переключения является точкой бифуркации. Эта характеристика оптимальной точки переключения является определяющей при ее поиске.

Однако, несмотря на то, что взаимопересечения являются признаком нахождения оптимальной точки переключения управляющего вектора, их использование не является желательным при построении квазиоптимального алгоритма. Дело в том, что для получения пересечений кривых $W(t)$ необходимо производить расчет по соответствующей оптимизационной процедуре в течение 100-200 шагов после точки переключения. Поскольку этот расчет требуется производить при каждом последовательном изменении точки переключения управляющего вектора, потери машинного времени становятся довольно большими и полный выигрыш квазиоптимального алгоритма уменьшается по сравнению с выигрышем квазиоптимальной стратегии в 2-3 раза. Для устранения этого недостатка требуется производить детальный анализ кривых $W(t)$ на более ранней стадии.

Рассмотрим зависимости функций $W(t)$ на начальном интервале процесса проектирования, выделенном прямоугольником на рис. 6, в большом масштабе. Эти зависимости приведены на рис. 7.

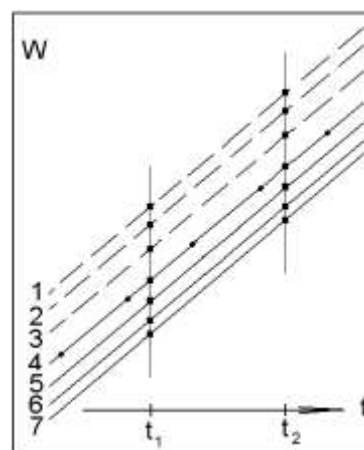


Рис. 7. Поведение функции $W(t)$ в течение начального интервала процесса оптимизации

Как видно из рисунка, в отмеченном интервале нет взаимопересечений кривых, соответствующих различным точкам переключения управляющего вектора. Однако можно отметить, что расстояния между этими кривыми различное. Расстояние между пунктирными кривыми, соответствующими переключению до оптимальной точки, увеличивается по мере приближения к оптимальной точке переключения. Напротив, расстояние между сплошными кривыми, соответствующими переключению после оптимальной точки, уменьшается по мере удаления от оптимальной точки. Это является хорошим критерием, позволяющим выявить оптимальную точку переключения управляющего вектора без больших дополнительных затрат машинного времени.

Введем функцию P , определяемую как разность значений функции $W(t)$ для двух соседних кривых

рисунка 7 в определенный момент времени t . Аргументом функции P является номер m соответствующей кривой (1, 2, и т.д.), т.е. эта функция является функцией дискретного аргумента. Построим эту функцию для двух различных значений времени t_1 и t_2 , определенных на рис. 7. Время t_1 соответствует 20-му шагу оптимизационной процедуры после точки переключения, и время t_2 соответствует 40-му шагу. Соответствующие графики в виде сплошных кривых изображены на рис. 8. Время является параметром.

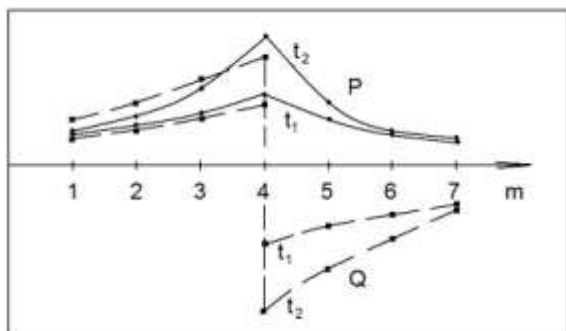


Рис. 8. Поведение функций P и Q

Эта функция возрастает при приближении к оптимальной точке переключения управляющего вектора. Номер 4 соответствует 9-му шагу процедуры оптимизации, являющемуся оптимальным шагом переключения (табл. 1). Достигнув максимума функция P далее уменьшается. Видно, что функциональная зависимость выражена сильнее для точки t_2 , однако и для точки t_1 эта зависимость дает возможность делать заключение относительно оптимальности точки переключения.

Определим еще одну функцию Q , являющуюся дискретной производной от функции P ($Q=P(m+1)-P(m)$), и которая построена на этом же рисунке. Поведение функции Q таково, что она может служить базисом для выработки основного критерия квазиоптимального алгоритма. Если в качестве критерия S выбрать произведение значений функции Q в двух соседних точках своего аргумента ($S=Q(m) \times Q(m+1)$), то положительное значение S свидетельствует о том, что оптимальная точка переключения еще не достигнута, а отрицательное как раз соответствует оптимальной точке переключения. Таким образом, рассчитывая значение функций P и Q и отслеживая внутри алгоритма значение критерия S , мы можем точно определить оптимальную точку переключения управляющего вектора. Подобная стратегия реализуется и в случае двух точек переключения, т.е. когда в точке S_j осуществляется переключение с МТСО на ТСО, а в точке S_{j+1} обратное переключение с ТСО на МТСО.

После идентификации оптимальных точек переключения управляющего вектора реализуется квазиоптимальная стратегия, соответствующая этим

точкам. Все эти детали отражены в алгоритме, приведенном на рис. 2. Подобный подход реализуется на этапе оптимизации электронных цепей и позволяет уменьшить затраты машинного времени в несколько сот раз по сравнению с традиционным подходом [8].

Дополнительные затраты процессорного времени, связанные с поиском оптимальных точек переключения управляющего вектора, составляют от 1% до 40% полного времени реализации квазиоптимального алгоритма. Эти оценки получены на основе анализа как приведенного примера (рис. 4), так и проектирования транзисторных усилителей, представленных в [4]. Например, при оптимизации цепи на рис. 4 эти затраты составили 16%.

V. ЗАКЛЮЧЕНИЕ

Дополнительный расход машинного времени, связанный с поиском оптимальных точек переключения управляющего вектора уменьшает выигрыш во времени, соответствующий квазиоптимальной стратегии. Однако, учитывая тот факт, что квазиоптимальная стратегия позволяет получить общий выигрыш в процессорном времени по сравнению с ТСО в несколько сот раз для исследованных электронных цепей, можно утверждать, что квазиоптимальный алгоритм, т.е. практическая реализация квазиоптимальной стратегии, дает выигрыш такого же порядка.

ЛИТЕРАТУРА

- [1] Каширский И.С., Трохименко Я.К. Обобщенная оптимизация электронных схем. Киев: Техника. 1979. 192 с.
- [2] Rizzoli V., Costanzo A., Cecchetti C. Numerical optimization of broadband nonlinear microwave circuits // IEEE MTT-S Int. Symp. 1990. Vol. 1. P. 335-338.
- [3] Ochotta E.S., Rutenbar R.A., Carley L.R. Synthesis of High-Performance Analog Circuits in ASTRX/OBLX // IEEE Trans. on CAD. 1996. Vol. 15. № 3. P. 273-294.
- [4] Zemliak A.M. Analog System Design Problem Formulation by Optimum Control Theory // IEICE Trans. Fundamentals Electron., Commun. and Computer Sciences. 2001. Vol. E84-A. № 8. P. 2029-2041.
- [5] Земляк А.М. Проектирование аналоговых цепей методами теории управления. I. Теория // Изв. Вузов. Радиоэлектроника. 2004. Т. 47. № 5. С. 18-28.
- [6] Земляк А. М. Сравнительный анализ функции Ляпунова различных стратегий проектирования аналоговых цепей // Изв. Вузов. Радиоэлектроника. 2008. Т. 51. № 5. С. 3-11.
- [7] Земляк А.М. Структура минимальной по времени стратегии оптимизации аналоговых цепей // Радиоэлектроника. 2009. Т. 52. № 1. С. 47-56.
- [8] Zemliak A.M. Acceleration Effect of System Design Process // IEICE Trans. Fundamentals Electron., Commun. and Computer Sciences. 2002. Vol. E85-A. № 7. P. 1751-1759.