

Время выполнения работы	Исполнители (фонд времени)				Остаток времени выполнения работы
	t_1	t_2	...	t_m	
w_1	x_{11}	x_{12}	...	x_{1m}	W_1
w_2	x_{21}	x_{22}	...	x_{2m}	W_2
...	...	...	...	...	...
w_n	x_{n1}	x_{n2}	...	x_{nm}	W_n
Остаток фонда времени исполнителей	T_1	T_2	...	T_m	

x_{ij} – допуск исполнителя до работы – будет записан в левом верхнем углу клетки, а c_{ij} – время, назначенное исполнителю на выполнение работы, будет записываться в правом нижнем углу клетки.

Анализируется остаток времени на выполнение работы. Если он равен нулю: $w_j = 0$, то эта строка исключается из дальнейшего рассмотрения.

Если текущая строка – последняя, то алгоритм завершается. Иначе, происходит переход к следующей строке и выполняется шаг 1.

Заключение

Был предложен алгоритм, оптимизирующий распределение задач между исполнителями при заданных ограничениях. Этот алгоритм можно применять не только в метрологических предприятиях, но и на предприятиях других отраслей, выполняющие много разных видов работ, требующих для их выполнения аккредитации исполнителя (станция технического обслуживания). Этот алгоритм позволит эффективно распределять задачи, а также, анализировать текущую нагрузку.

Список литературы

1. Горемыкин В.А. Планирование на предприятии: Учебник. – М.: ФИЛИНЪ, 2004. – 508 с.
2. Ильин А.И. Планирование на предприятии: Учебник. – М.: Новое знание, 2001. – 635 с.
3. Гасс С. Линейное программирование. – М.: Физико-математическая литература, 1961. – 300 с.
4. Гасс С. Путешествие в страну линейного программирования – М.: Мир, 1973. – 177 с.
5. Палий И.А. Линейное программирование. – М.: Эксмо, 2008. – 256 с.
6. Лунгу К.Н. Линейное программирование. Руководство к решению задач. – М.: ФИЗМАЛИТ, 2005. – 128 с.

ДЕТЕКТИРОВАНИЕ И КОРРЕКЦИЯ ЦИФРОВЫХ СИГНАЛОВ ДЛЯ УСТРОЙСТВ МАГНИТНОЙ ЗАПИСИ С МАКСИМАЛЬНЫМ ПРАВДОПОДОБИЕМ

Албутова Е.В., Шмокин М.Н.

Пензенский государственный технологический университет, Пенза, e-mail: oslit@yandex.ru

Из-за действия шумов, ограниченности полосы пропускания тракта магнитной записи-воспроизведения (МЗВ) и других причин форма воспроизводимых сигналов может существенно отличаться от записываемых.

Целью коррекции и детектирования является «очищение» воспроизводимого цифрового сигнала от помех и придание ему вида, пригодного для декодирования. При высокой плотности записи ограниченность АЧХ тракта МЗВ в области верхних частот приводит к взаимному воздействию воспроизводимых

импульсов, которое называют межсимвольной интерференцией (МСИ) [1].

Методы детектирования воспроизводимых импульсов можно разбить на две группы:

а) поэлементное детектирование, при котором решение 0 это или 1 принимается на основе последовательного анализа каждого воспроизводимого символа;

б) детектирование по алгоритму Витерби, при котором решение 0 это или 1 принимается в результате анализа группы воспроизводимых символов.

Амплитудный и другие методы поэлементного детектирования достаточно подробно описаны в литературе.

Рассмотрим некоторые аспекты парциального детектирования по алгоритму Витерби, которое применяется в цифровых системах форматов *Digital Betacam*, *MPEG IMX (D10)* [2].

Для примера рассмотрим систему *PR4*, в которой отсчеты имеют только три уровня $\{-1, 0, 1\}$ и предположим, что в канале воспроизведения действует аддитивная смесь сигнала $S(t)$ и Гауссовского шума $N(t)$. При поэлементном амплитудном детектировании происходит последовательное сравнение воспроизводимых отсчетов $Uk(t) = Sk(t) + Nk(t)$ с пороговыми уровнями $\pm 0,5$:

- если $Uk(t) > 0,5$, то отсчет равен 1,
- если $Uk(t) < -0,5$, то отсчет равен -1,
- если $Uk(t) \leq 0,5$, то отсчет равен 0.

Предположим, что на вход детектора поступает следующая последовательность зашумленных отсчетов: 0,3, -0,6, -0,4, 0,7, 0,9, 1,2, 0,2.

В соответствии с алгоритмом работы порогового детектора на его выходе будет последовательность $Sk = 0, -1, 0, 1, 1, 1, 0$.

Каждый класс сигналов *PR* имеет вполне определенные значения отсчетов. Из переходной и импульсной характеристик *PR4* следует, что в системах *PR4* не могут быть подряд три «единицы» [4]. Однако у поэлементного порогового детектора нет памяти, он ничего «не знает» о предыдущих и последующих отсчетах, он принимает решение 0 это или 1 по каждому отдельному отсчету.

Таким образом, три подряд 1 – это ошибочная комбинация: после двух 1 должен быть 0 или -1. Невозможно предложить несколько близких к Sk последовательностей (табл. 1).

Таблица 1

Отсчеты Sk	0,3	-0,6	-0,4	0,7	0,9	1,2	0,2
Последовательность N1	0	-1	0	1	1	0	0
Последовательность N2	0	-1	0	0	1	1	0
Последовательность N3	0	-1	0	0	0	1	1

Анализируя табл. 1, определим насколько последовательности N1, N2, N3 соответствуют воспроизводимым отсчетам. Последовательность N1 имеет отсчеты 0,7 и 0,9 соответствующие 1 (это правдоподобно), однако отсчету 1,2 соответствует 0, что маловероятно. Последовательность N2 имеет отсчет 0,7, соответствующий 0, это сомнительно, в то же время 0,9 и 1,2 соответствуют 1, что правдоподобно. В по-

добно), однако отсчету 1,2 соответствует 0, что маловероятно. Последовательность N2 имеет отсчет 0,7, соответствующий 0, это сомнительно, в то же время 0,9 и 1,2 соответствуют 1, что правдоподобно. В по-

следовательности N3 отсчеты 0,7 и 1,2 соответствуют 0 (весьма неправдоподобно), а отсчет 1,1 соответствует 1 (это правдоподобно). Из анализа следует, что наиболее правдоподобным результатом детектирования следует считать последовательность N2. Количественную оценку этого можно получить путем вычисления среднеквадратичного расстояния (*Mean-Squared Distance – MSD*) между отсчетами и возможными комбинациями последовательностей (1)

$$MSD = \sum_{k=1}^N [S(k) - a(k)]^2, \quad (1)$$

Вычисления по формуле (1) дают следующие результаты: последовательность N1 – $MSD=2,35$, последовательность N2 – $MSD=0,99$, последовательность N3 – $MSD=2,48$. Таким образом, последовательность N2 максимально правдоподобна воспроизводимым отсчетам.

Рассмотренный пример иллюстрирует принцип работы детектора Витерби:

– в отличие от поэлементного детектора решение принимается в результате анализа последовательности воспроизводимых отсчетов;

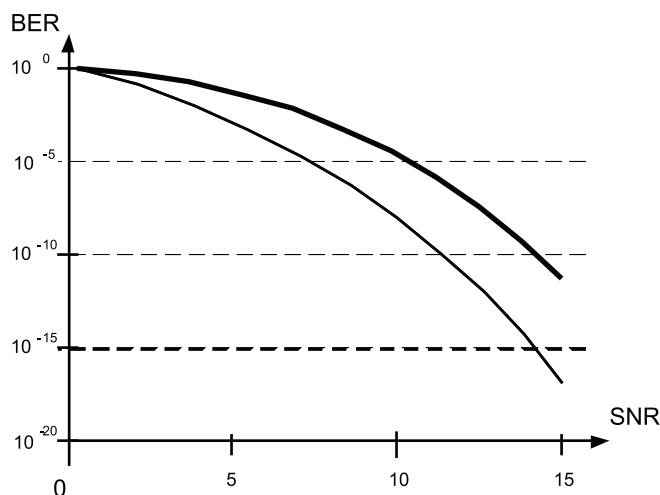
– для каждой последовательности воспроизводимых отсчетов создается возможный набор допустимых продетектированных последовательностей;

– каждая допустимая последовательность сравнивается с воспроизводимыми отсчетами и вычисляется MSD ;

– последовательность с минимальным MSD (максимально-правдоподобная) считается результатом детектирования.

Количественная оценка эффективности применения детектора Витерби в канале цифровой магнитной записи с аддитивным Гауссовским шумом дана в [3, 5].

Используя результаты этих работ, рассчитаны вероятности битовой ошибки BER (*Bit Error Rate*) для канала $PR4$ в зависимости от SNR (*Signal to Noise Ratio* – отношение сигнал/шум) (рис. 1): кривая (тонкая) соответствует максимально правдоподобному детектору по алгоритму Витерби, а кривая (жирная) – амплитудному пороговому детектору.



Вероятность битовой ошибки в зависимости от отношения сигнал/шум

Из рисунка видно, что детектор Витерби позволяет уменьшить вероятность ошибки на 1-2 порядка (зависит от SNR) по сравнению с амплитудным детектором [4]. Комбинацию системы PR определенного класса с максимально правдоподобным детектором (*Maximum Likelihood Detector*) принято называть сокращенно $PRML$ (*Partial Response Maximum Likelihood*) – каналом мли системой $PRML$. Технология $PRML$ существенно повышает энергетическую эффективность и эффективность использования полосы частот канала

записи-воспроизведения [3, 5]. Мерой увеличения энергетической эффективности $PRML$ в сравнении с амплитудным детектором является величина, показывающая, насколько можно снизить отношение сигнал/шум, чтобы обеспечить заданную достоверность записи-воспроизведения, определяемую вероятностью битовой ошибки. Например, система $PR4$ обеспечивает такую же вероятность битовой ошибки, как амплитудный детектор при меньшем на 3 дБ отношении сигнал/шум (табл. 2).

Таблица 2

Эффективность $PRML$ по сравнению с амплитудным детектором

Система $PRML$	Энергетический выигрыш	Плотность записи
$PR4$	1,41 (3 Дб)	1,1
$EPR4$	2 (6 Дб)	1,33
$E2PR4$	3,16 (10 Дб)	1,54

Мерой улучшения эффективности использования полосы частот системы $PRML$ по сравнению с амплитудным детектором является выигрыш в плотности записи (табл. 2.) Практически энергетический выигрыш и выигрыш в плотности записи несколько меньше, чем приведенный в табл. 2, так как при теоретическом анализе не учитывались ошибки син-

хронизации, нелинейные искажения, не идеальность формирования сигналов PR и ряд других факторов.

Список литературы

1. Харитонов М.И. Цифровая видеозапись: основные принципы, форматы, аппаратура. Часть 1, – М.: «625», №3, 2010.
2. Гончаров А.В., Харитонов М.И. Канал изображения видеоманитфона – М.: Радио и связь, 1987.
3. Kobayashi H., Tang D.T. Application of partial response channel coding in magnetic recording systems. IBM J. Res.Develop, July, 1970

4. Wood R.W. and Peterson D.A. Viterbi detection of class 4 partial response on magnetic recording channel –IEEE Trans. Commun, 34, 1986.

СЛОЖНОСТЬ АЛГОРИТМА ПРИ ВЫБОРЕ МИКРОКОНТРОЛЛЕРОВ

Башвеев Ю.А., Литвинская О.С.

Пензенский государственный технологический
университет, Пенза, e-mail: oslit@yandex.ru

В [1] был описан метод векторной оптимизации выбора микроконтроллера (МК) с использованием аппарата нечетких множеств. Было отмечено, что на конечный результат сильно влияет способ задания экспертом функций принадлежности (ФП). В настоящей работе предлагается в общем виде метод, опирающийся на информацию об алгоритме и времени его выполнения, позволяющий на этапе анализа технических требований к МК уйти от субъективизма при построении функций принадлежности.

Очевидно, что этап определения технических требований, в частности значения тактовой частоты МК оказывает сильное влияние на результат выбора модели МК. В связи с чем, разработка метода оценки временной сложности алгоритма представляется весьма актуальной.

Переход от оценки асимптотической сложности алгоритма к временной оценке сопряжен с некоторы-

ми объективными трудностями [4]: необходимость учёта особенностей используемого компилятора; различные времена выполнения реальных машинных команд; различие во времени выполнения одной команды в зависимости от значений операндов.

В настоящей работе для получения временных оценок предлагается использовать информацию по затратам машинного кода и числу тактов, необходимых для реализации операторов языка высокого уровня, полученную на основе статистических данных по частоте их употребления в типовых программах логического и цифрового употребления, где МК применяется наиболее широко [4], или же полученную после анализа алгоритма.

Предлагаемый метод включает в себя последовательность шагов:

- 1) разбиение программного алгоритма на конфигурационную и функциональную части;
- 2) определение количества высокоуровневых операций различных типов;
- 3) расчет количества тактов необходимых для реализации алгоритма с учетом имеющейся статистической информации по высокоуровневым операциям;
- 4) построение функции принадлежности частного критерия.

На рис. 1 представлена схема разрабатываемого метода.

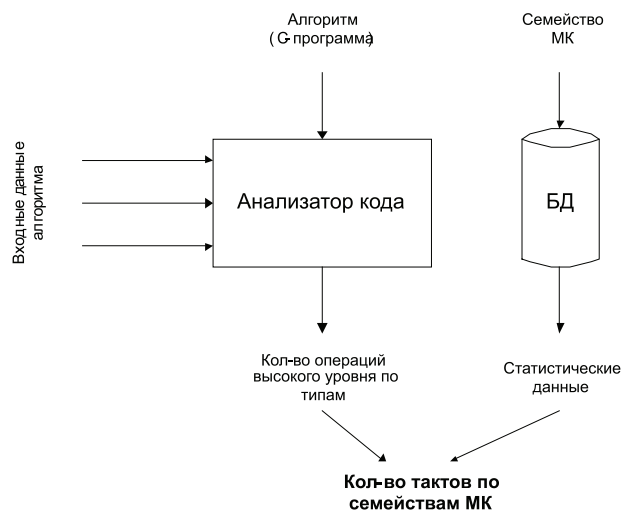


Рис. 1. Схема метода определения временных оценок алгоритма

Итак, программный алгоритм может быть разбит на две части: конфигурационную и функциональную. Нас в большей степени интересует функциональная часть, претендующая на многократное повторение. При таком подходе анализируемый алгоритм не будет привязан к конкретному МК, не будет зависеть от среды, и мы имеем полное право делать априорные оценки о времени выполнения алгоритма для любого МК.

После того, как функциональная часть алгоритма выделена, определяем количество высокоуровневых операций различных типов. В [4] предлагается использовать следующие элементарные операции:

- 1) простое присваивание $a \leftarrow b$;
- 2) одномерная индексация $a[i] = b$;
- 3) арифметические операции $a\{*, /, +, -\}b$;
- 4) операции сравнения $a\{>, <, =, \geq, \leq\}b$;
- 5) логические операции $a\{or, and, not\}b$;
- 6) операции адресации в сложных типах данных (name1.name2).

Рис. 1 отражает ситуацию, когда программный алгоритм задан в виде C-кода. Анализатор представляет собой программу, которая принимает на входе программный код и входные параметры алгоритма и рассчитывает кол-во высокоуровневых операций методом пооперационного анализа.

Основными алгоритмическими конструкциями являются конструкции следования, ветвления и цикла. С учетом введенных элементарных операций анализ таких конструкций сводится к следующим положениям.

а) конструкция «Следование». Трудоемкость конструкции есть сумма трудоемкостей блоков, следующих друг за другом

$$f_{\text{следование}} = f_1 + \dots + f_k;$$

б) конструкция «Ветвление». Общая трудоемкость конструкции «Ветвление» для построения функции трудоемкости в среднем случае требует ана-