

$$\begin{aligned} I_{out}(n) &= I(n)\cos(w_{QDDS}n) + Q(n)\sin(w_{QDDS}n), \\ Q_{out}(n) &= Q(n)\cos(w_{QDDS}n) - I(n)\sin(w_{QDDS}n), \end{aligned} \quad (8)$$

де w_{QDDS} частота виходу квадратурного прямого цифрового синтезатора частоти QDDS, а $I_{out}(n), Q_{out}(n)$ відповідно модулюючі сигнали [3]. Архітектуру такого модулятора наведемо на рис.3. Атенуатор використовується для компенсації спадання форми вихідного спектру у вигляді функції $\sin(x)/x$, та часткової компенсації спотворень фази та амплітуди в вихідних аналогових фільтрах.

Смуговий фільтр обмежує смугу випромінюваного сигналу, результатом чого є обмеження кількості каналів, що можуть бути використані, а в той самий час зменшує міжканальну інтерференцію. Інтерполюючий фільтр подавляє дзеркальні складові у сигналі спектру вихідного сигналу. Квадратурний синтезатор прямого синтезу за допомогою комплексного перемноження та додавання переносить квадратурні сигнали з основної смуги частот на проміжну частоту.

Архітектура такого модулятора значно простіша за аналогові рішення, і може використовуватись у портативній апаратурі [4, 5].

Висновки

Розглянуто принципи організації прямих цифрових синтезаторів частоти – DDS. Виконано математичний аналіз функціонування основних елементів такого синтезатора. Розглянуто можливості застосування синтезаторів прямого цифрового синтезу для побудови телекомунікаційних систем з різними видами модуляції. Отже застосування цифрових синтезаторів прямого синтезу DDS значно спрощує архітектуру радіоапаратури з точки зору спрощення побудови модуляторів, демодуляторів і наближає апаратуру до концепції SDR. Значно спрощується реалізація складних видів модуляції (QAM). Завданням подальших досліджень є аналіз впливу похибок прямого цифрового синтезу на характеристики вихідного сигналу QAM модулятора та методи зменшення впливу на якість спектру вихідного модульованого сигналу.

Література

1. Силян А. Технология Software Defined Radio. Теория, принципы и примеры аппаратных платформ / А. Силян // Технологии и стандарты. – 2007. – № 2. – С. 22 – 27.
2. Vankka J. Direct Digital Synthesizers: Theory, Design and Applications/ J.Vankka // Helsinki University of Technology. – 2000. – С. 192.
3. Манасевич В. Синтезаторы частот. Теория и проектирование / Манасевич В. ; Под ред. А.С. Галина ; [пер. с англ.]. – М. : Связь, 1979.
4. Побережский Е.С. Цифровые радиоприемные устройства / Побережский Е.С. – М. : Радио и связь, 1987.

Надійшла 6.5.2012 р.
Рецензент: д.т.н. Троцишин І.В.

УДК 681.518:681.326.7

Д.Е. ИВАНОВ

Институт прикладной математики и механики НАН Украины, г.Донецк

АЛГОРИТМЫ ДОСТИЖЕНИЯ СОСТОЯНИЙ В ЦИФРОВЫХ УСТРОЙСТВАХ И ИХ ПРИМЕНЕНИЕ В ЗАДАЧАХ ДИАГНОСТИКИ

В статье рассматривается задача достижения заданного состояния в синхронных последовательностных цифровых устройствах. Предлагается два метода решения данной задачи, которые основаны на применении R- и S-последовательностей и генетического алгоритма соответственно. Описываются возможные схемы применения метода в задачах идентификации цифровых устройств.

In paper the algorithms of achieving a given state in a synchronous sequential digital devices are considered. Two methods for solving this problem are proposed. The first method is based on the use of R- and S-sequences. Second method is based on genetic algorithm. The possible applications of this methods in problems of identification of digital devices are described.

Ключевые слова: синхронные последовательностные цифровые устройства, идентификация, достижение состояния, генетический алгоритм, генерация тестов.

Введение

Задача достижения заданного состояния в цифровых устройствах (ЦУ) (подтверждения состояния, в англ. терминологии state justification) часто возникает в задачах их диагностирования. Впервые она возникла при построении диагностических экспериментов с автоматами [1]. В настоящее время развивается

направление, в котором автоматные методы диагностирования ЦУ переносятся на структурный уровень [2–3]. Это связано с повышением вычислительной мощности рабочих станций, что позволяет проводить такую адаптацию.

Задача также часто возникает при разработке детерминированных методов построения тестов. В алгоритмах построения тестов данного рода [4–6] каждая неисправность первоначально должна быть активирована, а её влияние необходимо транспортировать на внешние выходы. При этом в последовательностных ЦУ это может потребовать от устройства находиться в некотором определённом состоянии для того, чтобы такая активация/распространение были возможны. Для этого требуется разработка процедур обратного распространения, которые для логического уровня представления последовательностных ЦУ усложняются двумя факторами: а) обратное распространение может потребоваться для нескольких тактов модельного времени; б) для многих логических элементов возникает проблема неоднозначности значений входов, заставляющая рассматривать множество вариантов. Дополнительные ограничения могут возникать из технологических особенностей: частичное или полное использование сканирующих регистров для установки начального состояния, предопределённость начального состояния, ограничение на число тактов для достижения состояния и т.д.

В противоположность обратному распространению, в подходах идентификации основанных на моделировании [7–8] влияние неисправности всегда распространяется только в прямом направлении. Оно заключается в генерации пробной последовательности и моделировании поведения ЦУ путём её приложения ко входам устройства. Сюда же следует отнести методы, которые основаны на генетическом алгоритме (ГА) [9–11].

Детерминированные алгоритмы чаще используются в схемах-контроллерах, тогда как подходы основанные на моделировании – для устройств обработки данных. Развитие сложных комплексных СБИС потребовало объединения этих двух направлений. В настоящее время известны методы построения тестов, которые совмещают детерминированный и основанный на моделировании подходы. Например, в [12] метод начинает работу с ГА. Если же в течении определённого времени не происходит улучшения ситуации, то алгоритм переключается на детерминированный метод построения тестов. Другим способом объединения двух указанных подходов является подтверждение состояний в детерминированных методах, которое реализуется, в свою очередь, основанным на моделировании алгоритмом.

Таким образом, задача построения последовательностей достижения состояний остаётся актуальной и в настоящее время. В данной работе будет предложено два алгоритма решения данной задачи. Первый алгоритм реализуется с помощью установочных последовательностей специального вида, тогда как второй основан на генетическом алгоритме. Будет предложено три способа применения данных методов в задачах идентификации ЦУ.

Алгоритм построения последовательностей достижения состояния с помощью R - и S -последовательностей

В качестве модели будем рассматривать синхронные последовательностные ЦУ с выделенной комбинационной частью и элементами состояний [13]. Пусть X – множество входов в ЦУ A , Y – множество выходов ЦУ, Z – множество элементов состояний ЦУ, при этом m – число элементов состояния. При моделировании поведения ЦУ A будем использовать трёхзначный алфавит $E_3 = \{0, 1, u\}$.

В [14, с.390] вводится понятие R - и S -последовательностей для элементов состояний синхронных последовательностных ЦУ.

Определение 1. Последовательность называется сбрасывающей R_i , если после её приложения ко входам ЦУ A , находящемуся в полностью неопределённом состоянии $Z = uu\dots u$, i -й элемент состояния $z_i \in Z$ получит значение 0 в алфавите моделирования E_3 .

Определение 2. Последовательность называется устанавливающей S_i , если после её приложения ко входам ЦУ A , находящемуся в полностью неопределённом состоянии $Z = uu\dots u$, i -й элемент состояния $z_i \in Z$ получит значение 1 в алфавите моделирования E_3 .

Данные R - и S -последовательности называются последовательностями типа A . Если же потребовать, что сброс (установление) значения на i -м элементе состояния достигается для ЦУ, в котором некоторые начальные элементы состояний уже фиксированы (имеют значение 0 или 1) и не изменяются, то такие последовательности называются R - и S -последовательностями типа B .

В [3] предложен генетический алгоритм построения R - и S -последовательностей. В той же работе они используются для ГА построения D -последовательностей различающих пару состояний путём их добавления в начальную популяцию, что позволяет ускорить сходимость метода.

В данной работе предлагается использовать данные типы последовательностей для решения задачи достижимости состояний. Задача достижения состояния в ЦУ может быть решена итеративным построением R - и S -последовательностей типа B для каждого элемента z_i состояния ЦУ A , $i = \overline{1, m}$. Отметим, что в такой постановке построение R - и S -последовательностей для элемента состояния z_i должно оставлять неизменными все остальные элементы состояний $z_j \in Z$, $j = 1, \dots, i-1, i+1, m$.

В практических задачах начальное состояние часто не является полностью неопределённым $Z = iii...i$, а задано. Для расширения метода на данный случай введём дополнительный входной параметр разрабатываемого алгоритма – начальное состояние ЦУ $Z_{нач}$, в котором некоторые элементы могут принимать значения 0, 1, а некоторые – i из алфавита моделирования E_3 . Случай с полностью неопределённым начальным состоянием будет являться частным при $Z_{нач} = iii...i$. Псевдокод алгоритмической реализации метода приведён ниже.

Алгоритм А1

```

Достижение_состояния( A , Zнач , Zкон )
{
    Последовательность = ∅ ;
    Инициализация_ЦУ( A , Zнач ) ;
    m = ЧислоЭлементовСостояний( A ) ;
    для( i=0 ; i<m ; i++ )
    {
        zi = Zкон[i] ;
        если( zi = 1 )
            S = Построение_S_последовательности( A , i ) ;
        иначе
            если( zi = 0 )
                S = Построение_R_последовательности( A , i ) ;
            иначе
                S = ∅ ;
        Последовательность = Последовательность ∪ S ;
    }
    return Последовательность ;
}

```

Данный подход имеет очевидные преимущества. Первым из них является прозрачность. Второе преимущество заключается в том, что если заранее построено множество возможных R - и S -последовательностей для данного ЦУ A (что может быть обеспечено разработчиком ЦУ), то построение последовательности достижения требуемого состояния является тривиальной задачей.

Очевидным недостатком такого метода решения задачи являются как итеративность вызова алгоритма метода построения R_i - и S_i -последовательностей, так и то, что для произвольного элемента состояния z_i не известно и не гарантировано существование такой последовательности, которая не изменяет остальные элементы состояний $z_j \in Z$, $j = 1, \dots, i-1, i+1, m$. В последнем случае алгоритм не применим полностью.

Отметим также, что в методе не акцентируется внимание на способе построения R - и S -последовательностей.

Генетический алгоритм построения последовательностей достижения состояния

Указанные недостатки для предыдущего метода с применением итеративного построения R - и S -последовательностей заставляют разработать новый метод. Как было отмечено ранее, в последнее время для построения идентифицирующих последовательностей различного типа успешно применяются генетические алгоритмы [10–11, 15]. Данные методы используют двухуровневую схему применения ГА, в которой ГА-метод построения последовательности вызывается после определения некоторой промежуточной цели (целевой неисправности, неразличимого множества неисправностей и т.п.) Для рассматриваемой задачи достижения состояния лучше использовать одноуровневую схему применения ГА [16–17]. В данной схеме поиск решения вызывается один раз и реализуется с помощью ГА.

Для построения ГА на основании общей структуры алгоритма [9] необходимо задать его компоненты. Поскольку необходимо найти входную двоичную последовательность S , то в качестве кодирования особей выбирается двоичная матрица [10], в которой каждая строка соответствует двоичному набору, подающемуся на вход ЦУ в один так модельного времени. Ширина такого двоичного набора соответствует числу внешних входов ЦУ. Традиционными при построении последовательностей с помощью ГА являются и генетические операции скрещивания и мутации [10–11]. Набор особей образует популяцию.

Наиболее важным при построении ГА является определение оценочной функции особей, определяющей их качество применительно к конкретной задаче. Семантически в нашей задаче оценочная функция показывает: насколько близко текущее состояние $Z_{тек}$ моделируемого ЦУ A находится от требуемого состояния $Z_{кон}$. Поскольку мы рассматриваем наиболее общий случай, когда граф переходов

ЦУ не известен, то для формализации оценки будем использовать расстояние по Хэммингу для двоичного представления $Z_{тек}$ и $Z_{кон}$, показывающее число совпадений элементов состояний. Таким образом, для заданного ЦУ A и заданной входной последовательности S оценка получается на основании результатов исправного моделирования:

$$o(A, S) = \sum_{Z_{тек}[i]=Z_{кон}[i]} 1, \quad (1)$$

где $Z[i] = z_i$ – показывает значение i -го элемента состояния.

Задание компонент ГА позволяет выполнить его алгоритмическую реализацию, которая в виде псевдокода приведена ниже.

Алгоритм А2

```

ГА_достижение_состояния( A , Zнач , Zкон )
{
    Pop = ПостроениеНачальнойПопуляции( ) ;
    ОценкаПопуляции( Pop ) ;
    while( не достигнут критерий останова( ) )
    {
        Popпром = ∅
        while( построение промежуточной популяции( ) )
        {
            Родители=ВыборРодителей( Pop ) ;
            if( rand() < Pскр )
                Потомок=Скрещивание( Родители ) ;
            if( rand() < Pмут )
                Потомок=Мутация( Потомок ) ;
            ДобавитьВПромежуточнуюПопуляцию( Popпром , Потомок ) ;
        }
        ПостроениеНовойПопуляции( Pop , Popпром ) ;
    } // конец цикла построения новых популяций
    return Pop[0] ;
} // конец алгоритма

```

Данный псевдокод показывает самую общую структуру ГА-метода построения последовательностей. Традиционно для эвристических алгоритмов необходимо выполнить предварительную его «настройку», которая заключается в экспериментальном обосновании выбора конкретных эволюционных операторов (скрещивания, мутации, построения новой популяции), а также основных параметров (вероятности скрещивания и мутации $P_{скр}$ и $P_{мут}$, предельных значений числа общих итераций и итераций без улучшения решений и т.д.) Например, в качестве верхней оценки длины строящихся последовательностей выбирается величина $4 \cdot d$, где d – структурная последовательностная глубина ЦУ [14].

Состояние $Z_{нач}$ необходимо для инициализации ЦУ A всякий раз перед моделированием, которое выполняется для текущей особи-последовательности S с целью её оценки. Состояние $Z_{кон}$ является результатом такого моделирования.

Варианты применения алгоритмов достижения состояний

Предложенные методы построения последовательностей достижения (подтверждения) состояний является по своей сути вспомогательными. В данном разделе будет показано возможное применение данных алгоритмов при решении некоторых задач, возникающих при проектировании и диагностировании поведения ЦУ.

Вариант 1. Диагностические эксперименты с автоматами.

Как было отмечено ранее, повышение вычислительной мощности рабочих станций породило направление исследований в технической диагностике, в котором на структурный уровень переносятся автоматные методы идентификации [1–3]. Диагностические эксперименты с автоматами часто состоят в построении деревьев обхода. В этом случае для выполнения одного перехода $Z_i \rightarrow Z_j$ в автоматном эксперименте необходимо вызвать один из рассмотренных алгоритмов, для которого Z_i и Z_j будут начальным и конечным состояниями соответственно.

Такое применение рассматриваемого алгоритма построения последовательностей достижения состояний является наиболее простым, при этом для оценки последовательности используется моделирование одного исправного ЦУ.

Вариант 2. Генерация тестов с помощью подтверждения состояний.

Одной из возможных схем применения алгоритма достижения состояний является гибридный (детерминированный+основанный на моделировании) алгоритм генерации тестов. Например в [18] описывается гибридный метод построения проверяющих тестов, который заключается в итеративном построении тестовой последовательности для каждой неисправности из заданного множества. Построение теста отдельной неисправности разбивается на два этапа:

активизация неисправности, распространение её влияния на внешние выходы, обратное распространение на входы и элементы состояния;

подтверждение состояния, полученного на этапе 1.

В данном случае на этапе 2 необходимо выполнять подтверждение состояния для двух ЦУ, первое из которых исправное ЦУ A , а второе A_f моделируется в присутствии некоторой неисправности f (выбранной на этапе 1), поскольку в последнем случае присутствие неисправности может изменять состояние ЦУ. Применение структурных методов подтверждения состояния в данном случае затруднено тем, что продвижение назад, возможно, придётся выполнять на несколько тактов модельного времени.

Однако такое подтверждение состояния можно выполнить на основании предложенных алгоритмов. В этом случае для алгоритмов необходимо задать начальное и конечное состояния исправного A и неисправного A_f устройств. Конечным состоянием для обоих ЦУ выбирается состояние, которое получено на этапе 1, т.е. то, которое необходимо подтвердить (достичь). Начальное состояние исправного ЦУ A выбирается таким, которое достигается в результате моделирования поведения ЦУ A на построенной к текущему моменту времени тестовой последовательности для предыдущих неисправностей. Для неисправного ЦУ A_f в качестве начального выбирается полностью неопределённое состояние $Z_{нач} = iii...m$.

Видно, что для оценки потенциальной тестовой последовательности S в этом случае необходимо выполнить моделирование поведения двух ЦУ с получением оценки в виде (1) для каждого из них. Таким образом, мы получаем, что оценка в данном случае является двухкритериальной. В том случае, когда оценочная функция ГА является многокритериальной при её формализации часто используются «наивные методы», в частности взвешенная сумма критериев [19]. В этом случае оценка качества последовательности S для исправного ЦУ A и ЦУ с неисправностью A_f будет иметь вид:

$$o(A, A_f, S) = c_1 \cdot o(A, S) + c_2 \cdot o(A_f, S), \quad (2)$$

где c_1 и c_2 – весовые коэффициенты, которые необходимо подобрать на основании экспериментов, а компоненты $o(A, S)$ и $o(A_f, S)$ вычисляются на основании (1).

Решение задачи достижимости в такой постановке с использованием оценочной функции в виде (2) требует изменения алгоритма выше. Для этого необходимо предусмотреть в нём обработку двух верифицируемых ЦУ.

Ещё одна модификация алгоритма для рассматриваемого случая позволит существенно ускорить его работу. Она заключается в параллельном моделировании неисправных ЦУ в разрядах машинного слова [20–21]. В этом случае в первом разряде необходимо выполнять моделирование поведения исправного ЦУ, что позволит вычислить компоненту $o(A, S)$ в (2). В остальных разрядах выполняется моделирование группы неисправных ЦУ, при этом для каждой неисправности вычисляется компонента $o(A_f, S)$. Такая модификация позволит оценивать сразу $k-1$ потенциальных решений, где k – разрядность инструментальной ЭВМ.

Вариант 3. Применение в методе построения тестов на основании различия пары состояний в исправном и неисправном ЦУ, который предложен в [3].

Данный метод заключается в построении входной последовательности, которая различает все пары возможных состояний в исправном и неисправном ЦУ. Метод также разработан с целью перенести автоматный подход на структурный уровень. В работе приводится оценка сложности такого метода $O(n^4)$, где n – число состояний рассматриваемого конечного автомата A . Оценка построена на основании того, что алгоритму необходимо различить n^2 пар состояний при том, что алгоритм такого различения имеет сложность $O(n^2)$.

В описанном алгоритме R - и S -последовательности включаются в стартовую популяцию ГА, который строит различающие D -последовательности выбранной пары состояний ЦУ A . Предлагается данную процедуру проводить с помощью предложенных в данной работе алгоритмов. Очевидно, значение

D 16-значного алфавита моделювання B_{16} [14] может трактуватися як кінцеве стання для запропонованих алгоритмів. Їх використання в даному контексті можна розділити на два випадки.

В першому випадку передбачається розширення алфавита моделювання ЦУ в розглянутих алгоритмах на універсальну 16-значну логіку. Тоді різниця станнь буде відповідати символу D по результатам моделювання послідовності.

Во другому випадку, як і в попередньому варіанті використання, передбачається паралельне моделювання поведінки несправних ЦУ. Оскільки моделювання в 16-значній логіці B_{16} відповідає моделюванню поведінки двох ЦУ (справного A і несправного A_f), то моделювання поведінки справного ЦУ A може бути обособлено і виконано в окремому разряді машинного слова. Тоді в залишених разрядах можна виконувати моделювання сразу групи несправних ЦУ, як в попередньому варіанті застосування. Аналогічно виконується ініціалізація станнь справного і несправних ЦУ. Фактично, сопоставлення значень першого разряду для справного ЦУ A і деякого вибраного разряду для несправного ЦУ A_f дає кодування значення в алфавіті B_{16} . При цьому моделювання поведінки справного ЦУ виконується один раз сразу для групи несправностей. Такий прийом паралельного моделювання також повинен суттєво підвищати загальну швидкість алгоритму тестування.

Висновки

В даній статті запропоновано два методи побудови послідовностей для досягнення заданного стання в СБИС. Перший метод оснований на ітеративній побудові R - і S - послідовностей для відповідних значень елементів стання. Другий метод реалізований за допомогою одноуровневої схеми застосування ГА побудови послідовності досягнення станнь. Описані способи застосування запропонованих методів в різних контекстах. Найбільш востребованими дані методи будуть в алгоритмах діагностування ЦУ структурного рівня, які базуються на експериментах з кінцевими автоматами.

Література

1. Гилл А. Введение в теорию конечных автоматов / А. Гилл. – М. : Наука, 1966. – 272 с.
2. Скобцов Ю.А. Генерация тестов для последовательностных схем с использованием кратной стратегии наблюдения выходных сигналов / Ю.А. Скобцов, В.Ю. Скобцов, Ш.Н. Хинди // Науковий вісник Чернівецького університету. – Фізика. Електроніка. – 2008. – Вип. 423. – С. 29–36.
3. Хинди Ш.Н. Иерархические эволюционные методы генерации тестов цифровых систем : дис. ... канд. техн. наук : 05.13.05 / Шукри Насри Али Хинди (Иордания) ; ГБУЗ "Донецкий национальный технический университет". – Донецк, 2010. – 141 с.
4. Lee D.H. A new test generation method for sequential circuits / D.H. Lee, S.M. Reddy // Proc. Int. Conf. Computer-Aided design.- 1991.- P.446-449.
5. Niermann T.M. NITEC: A test generation package for sequential circuits / T.M. Niermann, J.H. Patel // Proc. European Conf. Design Automation.- 1991.- P.214-218.
6. Ghosh A. Test generation for highly sequential circuits / A. Ghosh, S. Devadas, A.R. Newton // Proc Int. Conf Computer-Aided Design.- 1989.- P.362-365.
7. Wunderlich H.-J. Multiple distributions for biased random test patterns / H.-J. Wunderlich // IEEE Trans. Computer-Aided Design.- 1990.- Vol.9, №6.- P.584-593.
8. Agraval V.D. A directed search method for test generation using a concurrent simulator / V.D. Agraval, K.T. Cheng, P. Agraval // IEEE Trans. Computer-Aided Design.- 1989.- Vol.8, №2.- P.131-138.
9. Goldberg D.E. Genetic Algorithm in Search, Optimization, and Machine Learning / D.E. Goldberg.- Boston, MA:Addison-Wesley Longman Publishing Co.- 1989.- 412p.
10. Corno F. Experiences in the use of evolutionary techniques for testing digital circuits / F. Corno, M. Sonza Reorda, M. Rebaudengo // Proc. of Conf. Applications and science of neural networks, fuzzy systems, and evolutionary computation, San Diego CA.- 1998.- P.128-139.
11. Skobtsov Y.A. Distributed Genetic Algorithm of Test Generation For Digital Circuits / Y.A. Skobtsov, El-Khatib, D.E. Ivanov // Proceedings of the 10th Biennial Baltic Electronics Conference.- Tallinn Technical University.- 2006.- P.281-284.
12. Saab D.G. Iterative [simulation-based genetic + deterministic techniques] = complete ATPG / D.G. Saab, Y.G. Saab and J.A. Abraham // Proc. Int. Conf. Computer-Aided Design.- 1994.- P.40-43.
13. Барашко А.С. Моделирование и тестирование дискретных устройств / Барашко А.С., Скобцов Ю.А., Сперанский Д.В. – К. : Наукова думка, 1992. – 288 с.
14. Скобцов Ю.А. Логическое моделирование и тестирование цифровых устройств / Ю.А. Скобцов, В.Ю. Скобцов. – Донецк : ИПММ НАНУ, ДонНТУ, 2005. – 436 с.
15. Иванов Д.Е. Генетический алгоритм построения диагностических последовательностей цифровых устройств / Иванов Д.Е. // Вісник східноукраїнського національного університету ім. В.Даля, 2010. – № 10 (152). – С. 72–79.

16. Иванов Д.Е. Построение инициализирующих последовательностей синхронных цифровых схем с помощью генетических алгоритмов / Д.Е. Иванов, Ю.А. Скобцов, А.И. Эль-Хатиб // Проблемы информационных технологий, 2007. – №1. – С. 158–164.
17. Иванов Д.Е. Генетический подход проверки эквивалентности последовательностных схем / Д.Е. Иванов // Радиоелектроніка. Інформатика. Управління. – Запоріжжя, ЗНТУ, 2009. – № 1 (20). – С. 118–123.
18. Rudnick E.M. State Justification using Genetic Algorithms in Sequential Circuit Test Generation / Elizabeth M. Rudnick, Janak H. Patel // Tech. Report CRHC-96-01/UIIU-ENG-96-2201, Coordinated Science Laboratory, University of Illinois, Urbana, IL.- Jan.1996
19. Sean L. Essentials of Metaheuristics [Електронний ресурс] / L. Sean. – Режим доступу : [http://cs.gmu.edu/~sim\\$sean/book/metaheuristics/](http://cs.gmu.edu/~sim$sean/book/metaheuristics/)
20. Niermann T.M. PROOFS: A Fast, Memory-Efficient Sequential Circuits Fault Simulator / T.M. Niermann, W.-T. Cheng, J.H. Patel // IEEE Trans. CAD, 1992.- V.11.- №2.- P.198-207.
21. Иванов Д.Е. Параллельное моделирование неисправностей для последовательностных схем / Д.Е. Иванов, Ю.А. Скобцов // Искусственный интеллект, 1999. – № 1. – С. 44–50.

Надійшла 14.5.2012 р.
Рецензент: д.т.н. Скобцов Ю.А.

УДК 621.396.98

Ю.М. ВОЛОВИК, А.Ю. ВОЛОВИК, М.А. ШУТИЛО
Вінницький національний технічний університет
В.В. ЗАГОРСЬКИЙ
Вінницький технічний коледж

КВАЗІОПТИМАЛЬНА ФІЛЬТРАЦІЯ РІЗНОТОЧНИХ КУТОВИХ ВИМІРЮВАНЬ В СИСТЕМІ ПОСАДКИ САНТИМЕТРОВОГО ДІАПАЗОНУ

В статті розглядається задача фільтрації різноточних куткових вимірювань, які проводяться на борту повітряного судна в процесі заходу на посадку. Пропонуються квазіоптимальні методи фільтрації аномальних похибок куткових вимірювань, ефективність яких підтверджується результатами статистичного моделювання.

In article the problem of a filtration of angular measurements of the different accuracy which is carried out onboard an aircraft in the course of landing approach is considered. The suboptimal method's of a filtration of abnormal errors of the angular measurements which efficiency is confirmed by results of statistical modeling is offered.

Ключові слова: оптимальна нелінійна фільтрація, фільтр Калмана, статистичне моделювання

Вступ. Радіомаячні системи посадки сантиметрового діапазону (MLS-Microwave Landing System) розробляються на основі рішень Міжнародної організації цивільної авіації (ICAO) від 1972 р. Вони призначені для отримання на борту повітряного судна (ПС) та видачі екіпажу і в систему автоматичного керування польотом інформації про кути відхилення ПС від номінальної траєкторії посадки у вертикальній та горизонтальній площинах, віддалі до розрахованого місця приземлення, стану злітно-посадкової смуги, категорії метеомінімуму обслуговування та тощо. В якості міжнародного стандарту прийнята система TRSB (Time Reference Scanning Beam) [1, 2], параметри якої регламентовані ICAO. Стандартом ICAO до системи посадки сантиметрового діапазону передбачені підвищені вимоги щодо точності та надійності виведення ПС у напрямку на злітно-посадкову смугу, які забезпечуються новим принципом отримання куткових координат на борту ПС, надмірністю формату радіосигналу та методами обробки результатів радіовимірювань, що ґрунтуються на застосуванні новітніх інформаційних технологій.

Постановка задачі. Оскільки система посадки сантиметрового діапазону використовує більш високі частоти ніж СП-50, СП-70 та ILS, то численні завади, внутрішньо променеві відбиття радіосигналу, що пов'язані з топографією району посадки та наявністю приаеродромних споруд суттєво послаблюються, однак повністю усунути їх не вдалось. Інтерференція прямого та відбитого сигналів може призводити до падіння рівня корисного сигналу нижчого за припустимий і за таких умов упевнений прийом сигналів куткових радіомаяків стає неможливим, а в потоці посадочних даних з'являється випадкові пропуски окремих результатів вимірювань або вони будуть супроводжуватись нечастими аномальними похибками [4].

Проте, в процесі керування польотом ПС повинна використовуватись лише надійна інформація, а тому актуальною є розробка таких методів оцінювання куткових координат ПС, які поєднують високу точність та достовірність отриманих оцінок з підвищеною стійкістю до комплексу можливих порушень працездатності радіовимірювального тракту.

Однією з перших робіт цього напрямку можна вважати роботу [8], у якій підкреслювалось, що безпосереднє застосування результатів Калмана до задачі обробки малодостовірних результатів первинних радіовимірювань наштовхується на певні труднощі, так як згідно з [6], необхідна повнота апріорної інформації про динамічну модель інформативного параметра та статистичних властивостях супутніх збурень та завад. У тому випадку, коли така інформація відсутня, задача оптимізації обробки радіосигналів стає некоректно поставленою. Деякі автори [7] пропонують іти шляхом використання принципів адаптації з