

А.Н. Тихменев, В.В. Жаднов

# ПРОГНОЗИРОВАНИЕ НАДЕЖНОСТИ СТРУКТУРНО-СЛОЖНЫХ РАДИОЭЛЕКТРОННЫХ СРЕДСТВ МЕТОДАМИ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

В статье рассматривается применение имитационного моделирования для оценки показателей надежности радиоэлектронных средств со сложными алгоритмами резервирования и реконфигураций. Описываются модели и возможности специализированного языка описания отказов радиоэлектронных средств на примере нескольких вариантов сложных структур.

**Ключевые слова:** радиоэлектронные средства, надежность, имитационное моделирование

**И**сследование осуществлено в рамках Программы фундаментальных исследований НИУ ВШЭ в 2013 году. Предприятия отечественной радиопромышленности, занимающиеся разработкой и производством радиоэлектронных средств (РЭС) для космических аппаратов (КА), сталкиваются с проблемами обеспечения надежности и, в первую очередь, безотказности. Об этом свидетельствуют как отказы при проведении приемо-сдаточных испытаний РЭС, так и аварии при эксплуатации КА. Одной из причин такой ситуации является применение устаревших и неточных методов оценки надежности РЭС КА на этапе проектирования.

С другой стороны, использование "нижних" оценок показателей безотказности может привести к снижению конкурентоспособности РЭС КА, так как в этом случае для повышения надежности необоснованно используют различные дополнительные способы, что ведет к ухудшению экономических, массогабаритных и др. показателей. Поэтому повышение точности расчетной оценки надежности РЭС КА с длительными сроками активного существования является актуальной задачей.

Кроме изменения элементной базы и введения резервирования для повышения надежности в бортовых РЭС ответственного применения используется и реконфигурация структуры при отказах составных частей (СЧ). Основная идея этого метода состоит в перераспределении функций отказавших СЧ между исправными. Под реконфигурируемыми системами понимаются такие системы, в процессе функционирования которых происходят перераспределения нагрузок, подключение и отключение дополнительных модулей, в зависимости от приоритетов решаемых задач или в случае отказов. Использование таких систем технически выгодно, так как позволяет эффективно расходовать энергию и ресурсы СЧ. Особенно широкое распространение такой метод повышения надежности получил в РЭС КА в силу высоких требований по надежности и автономного режима работы (без технического обслуживания).

Возможность реконфигурации структуры закладывается на этапе проектирования РЭС и тесно пере-

плетена с особенностями выполняемых задач, общей структурой РЭС и многими другими факторами. В силу этого многие современные бортовые РЭС имеют индивидуальные алгоритмы работы и уникальную структуру, оценка надежности которых аналитическими методами весьма затруднительна, а в большинстве случаев - просто невозможна. "Нижние" оценки надежности таких РЭС могут быть получены по моделям ОСТ 4Г.0.012.242 [1], однако в большинстве случаев структуру РЭС КА не удается свести к набору "типовых" групп.

Для решения таких задач, в соответствии с рекомендациями ГОСТ 27.301 [2], следует применять метод имитационного (статистического) моделирования. Однако, как отмечено в [3], "...статистическое моделирование (метод Монте-Карло), являющееся очень эффективным средством анализа различных сложных систем, остается пока искусством, овладение которым под силу только профессионалам в данной области. Возможно, создание простых и доступных специальных языков высокого уровня поможет более широкому внедрению методов статистического моделирования в инженерную практику". Вопросы применения имитационного моделирования для решения задач обеспечения надежности достаточно часто затрагиваются в различных статьях. Но в них, как правило, рассматриваются аспекты обеспечения надежности, связанные с техническим обслуживанием и ремонтом, то есть в той сфере, где можно использовать модели теории систем массового обслуживания (СМО). Модели СМО хорошо разработаны теоретически, а для автоматизации расчетов созданы программные средства.

## 1. Постановка задачи

Задачи расчета надежности структурно-сложных РЭС на основе данных о надежности ее СЧ методами имитационного моделирования поднимаются в литературе реже и носят частичный характер [4-6]. В этих работах рассматриваются частные случаи и теоретические подходы. Универсального метода, применимого к разнообразным структурам РЭС, в них не описывается. В связи с этим была проведена разработка типизированной настраиваемой модели

для описания функционирования РЭС с точки зрения надежности.

В основу моделирования положено представление РЭС как множества элементарных компонентов со своими распределениями наработок и, соответственно, некоторым временем жизни. При этом каждый компонент характеризуется некоторым набором состояний и режимов работы, которым соответствуют свои законы распределения времени до отказа компонента. В единую структуру эти разрозненные компоненты объединяются через события реконфигурации и критерии отказов. Первые описывают изменения, которым подвергается РЭС, вторые позволяют разграничить множество состояний компонентов, при котором РЭС остается в работоспособном состоянии, от множества, в котором функционирование исследуемого РЭС будет невозможно.

Через вариации числа компонентов, их состояний, режимов, законов распределений времени до отказа, а также событий и критериев отказов можно составить модели для широкого класса РЭС с разнообразными алгоритмами функционирования и реконфигурации [7]. При этом, для повышения "гибкости" описания используется инструментарий классических языков программирования с их возможностями по ветвлению алгоритмов и вычислению логико-математических выражений. Аргументами для выражений могут являться как переменные (глобальные и локальные), так и состояния компонентов, входящих в РЭС, что, естественно, используется гораздо чаще.

Основным элементом формальной модели является компонент (knot). Компоненты делятся на два типа: элементарный и составной. Элементарный компонент - это модель нижнего уровня разукрупнения структуры РЭС, в нем нет составных компонентов, и исходными данными для него являются законы распределения (в частном случае - закон распределения наработки). Составной компонент, с точки зрения структуры РЭС, это компонент более высокого уровня, у него есть дочерние элементы, и законы распределения наработок в явном виде для него неизвестны.

Внешне для модели любой компонент описывается состоянием и режимом. Для элементарных компонентов переход из состояния в состояние моделируется на основе законов распределения наработок. В общем виде диаграмма переходов показана на рис. 1.

Переход в другое состояние с тем же режимом характеризуется некоторой плотностью вероятности во времени. Плотность в формальной модели задается через закон распределения времени нахождения компонента в каждом состоянии (обычно это распределение времени до отказа компонента).

Переход из состояния в состояние может произойти в двух случаях: в результате истечения времени нахождения в данном состоянии или в результате какого-либо события в РЭС. Причем переход по наработке не может изменить режима, изменяется

только состояние. Событие же может перевести компонент в любую пару режим-состояние (рис. 1). Для удобства будем называть изменение состояния компонента по интенсивности безусловным переходом, а в результате события - условным. Семантика стандартной модели компонента предполагает, что состояние "1" - это состояние отказа, из него переход в другое состояние невозможен. Более подробно семантика модели компонента приведена в [8].

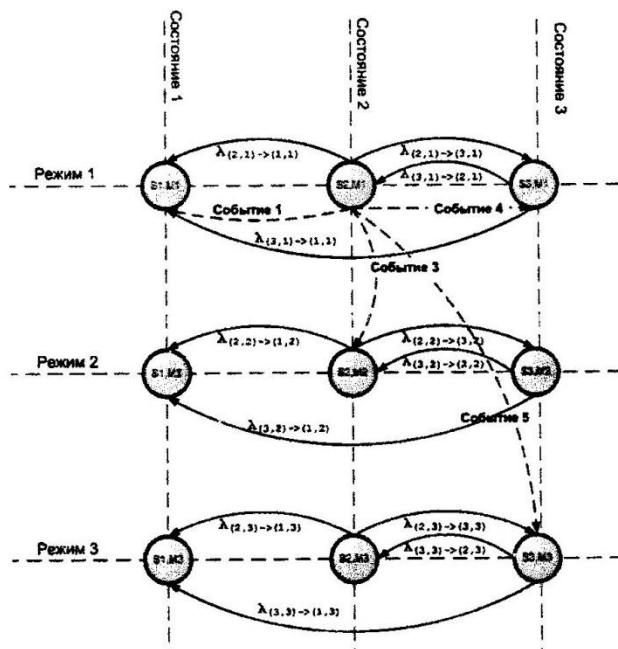


Рис. 1. Обобщенная диаграмма переходов компонента

## 2. Примеры формальных моделей структурно-сложных РЭС

Покажем возможности языка и разработанные формальные модели на примере различных структур РЭС. Для наглядности будем моделировать простые структуры, но с разнообразными алгоритмами функционирования и реконфигурации.

Рассмотрим резервированное РЭС, структурная схема надежности (ССН) которого приведена на рис. 2.

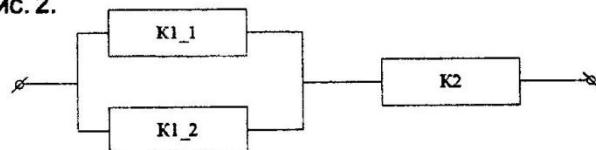


Рис. 2. Структурная схема надежности (пример 1)

Компоненты ССН имеют постоянную интенсивность отказов на всем протяжении функционирования. Такие компоненты описываются диаграммой переходов из двух состояний "работа" и "отказ", и одного режима. Перед описанием каждого компонента необходимо задать закон распределения его наработки. Для этого используется следующий оператор:

`distribution Dis_K_example (1e-5);`

В данном случае объявляется экспоненциальный закон распределения с параметром  $\lambda$  (интенсивность отказов),  $\lambda = 1 \cdot 10^{-5}$  и именем `Dis_K_example`. Далее это имя можно использовать при описании компонента:

```

knot K1_1
{
state: Fail, Work;
mode: Normal;
startState: Work;
startMode: Normal;
cntrlMode: unDistribution;
tableDistribution:
    | Normal           |
Work   | Dis_K_example;
tableStateChange:
    Normal          |
Work   | Fail           ;
}

```

где *knot* - ключевое слово, означающее объявление компонента; *BP\_1* - имя компонента; *state: Fail, Work;* - перечисление состояний компонента (состояние, указанное первым, считается необратимым отказом; если оно присвоено компоненту, то изменить его уже нельзя); *mode: Normal;* - перечисление режимов работы компонента; *startState: Work;* - состояние при начале эксперимента; *startMode: Normal;* - режим при начале эксперимента; *ctrlMode: unDistribution;* - тип контроля (значение *unDistribution* означает, что данный компонент изменит состояние по прошествии времени, разыгранного генератором случайных чисел); таблица, определяющая по какому распределению разыгрывается время, которое компонент будет находиться в данном состоянии:

```
tableDistribution:  
    | Normal  
Work   | Dis K example;
```

таблица, указывающая, в какое состояние перейдет данный компонент по истечении разыгранного времени. Т.к. РЭС в примере 1 не восстанавливаемое, то, следовательно, указано состояние *Fail*:

tableStateChange:	Normal	
Work	Fail	:

Для построения модели РЭС необходимо описать каждый компонент отдельно, но для рассматриваемого примера модели компонентов идентичны и отличаются только именами. Заметим, что часто в модели РЭС встречается много одинаковых по смыслу моделей компонентов, отличающихся лишь значениями интенсивностей отказов.

После описания каждого компонента необходимо описать составной компонент, моделирующий РЭС

в целом. Текст модели такого компонента представлен ниже

```

general knot RES
{
state: Fail, Work;
mode: Normal;
startState: Work;
startMode: Normal;
cntrlMode: unFunction;
tableStateChange:
    Normal
Work  |FunctRES ;
};

```

Главное отличие в параметре *ctrlMode*, который для элементарного узла имеет значение *unDistribution*, а для составного узла указывается *ifFunction*. Компонент РЭС также имеет два состояния и один режим, но текущее состояние определяется состояниями других компонентов. Ключевое слово *general* означает, что это модель центрального компонента РЭС, и моделирование имеет целью оценить именно его показатели надежности. Поэтому при его отказе (переходе в состояние, указанное первым в списке *state*) имитационный эксперимент завершается.

Для сложного узла не задается *tableDistribution*, а в таблице *tableStateChange* указывается не состояние, а имя функции, которая вычисляет состояние сложного узла. Функция должна вернуть номер состояния, т.е. для *RES* функция должна вернуть "0", если *RES* в состоянии отказа, и "1" - если в работоспособном состоянии. Для вычисления состояния сложного узла используются состояния других компонентов, математические и логические операции над ними. Для рассматриваемого примера функция *FunctRES* будет выглядеть так:

### function FunctRES

```
return (K1_1|K1_2) & K2;
```

При вычислении состояния имена компонентов заменяются на "1", если компонент в работоспособном состоянии (первом, указанном в списке состояний) и "0" - в противном случае. Также можно использовать следующую запись: **K1\_1:Work**. Такой оператор возвращает "1", если компонент находится в состоянии **Work**, и "0", если в любом другом. Выражение в **(K1\_1|K1\_2)& K2** будет равняться "1" до тех пор, пока функционирует хотя бы один из компонентов **K1\_1**, **K1\_2** и компонент **K2**.

Теперь рассмотрим процесс моделирования. После компиляции модели и запуска имитационного эксперимента разыгрываются времена до отказа для всех элементарных компонентов (в нашем примере  $K1\_1, K1\_2, K2$ ). Далее модельное время "перематывается" к первому отказу, и изменяется состояние отказавшего узла. После этого вызывает-

ся функция `FunctRES`, которая определяет состояние узла `RES`. Если `RES` не отказал, то время "перематывается" к следующему отказу. В случае отказа `RES`, фиксируется текущее модельное время как время отказа `RES`, и эксперимент завершается. Повторив многократно такие эксперименты можно оценить наработку на отказ РЭС.

Более сложным случаем является моделирование РЭС, представляющее собой группу "ненагруженное резервирование", ССН которой приведена на рис. 3.

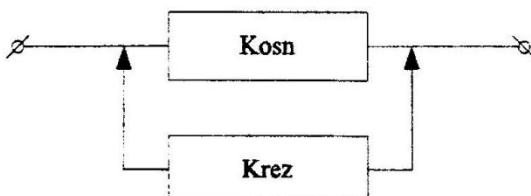


Рис. 3. Структурная схема надежности (пример 2)

Основной компонент, который изначально находится в рабочеспособном состоянии, описывается идентично компонентам из примера 1. В модели резервного компонента необходимо ввести дополнительный режим, моделирующий отказы в облегченном режиме. Для этого объявляются два закона распределения наработок и соответствующие им интенсивности отказов в облегченном и нагруженном режимах. В этом случае описание резервного компонента будет выглядеть следующим образом:

```

knot Kres
{
  state: Fail, Work;
  mode: Light, Hard;
  startState: Work;
  startMode: Light;
  cntrlMode: unDistribution;
  tableDistribution:
    | Hard       | Light      |
    Work   | Dis_Klight | Dis_Khard;
  tableStateChange:
    | Hard       | Light      |
    Work   | Fail       | Fail       ;
}
  
```

Данная модель описывает два возможных рабочеспособных состояния с разными режимами и, соответственно, с разными интенсивностями отказов. Описание РЭС аналогично примеру 1:

```

function FunctRES
{
  return (Kres | Kosn);
}
  
```

Для полного соответствия модели реальному РЭС необходимо смоделировать переход резервного компонента из облегченного режима в нагруженный. Это делается при помощи следующей конструкции:

```

switch_Event Kosn_Fail ( -> Kosn:Fail)
{
  set_mode (Kres :Hard );
}
  
```

Здесь `switch_Event` - ключевое слово языка, означающее событие; `Kosn_Fail` - имя события. В

круглых скобках указывается условие выполнения (в виде логического выражения, где также можно использовать состояния узлов).

Однако удобнее пользоваться более сложными операторами, определяющими момент перехода компонентов из состояния в состояние, т.е. условие события `Kosn_Fail` выполняется в тот момент времени, когда узел `Kosn` переходит в состояние `Fail`. В этот момент выполняется тело события, указанное в фигурных скобках. Оператор `set_mode (Kres :Hard )`; переводит узел `Kres` в нагруженный режим, который обозначен именем `Hard`. Аналогично, оператором `set_state` можно изменить состояние узла.

Еще более сложные модели используются при моделировании восстанавливаемых компонентов. Так как в семантике модели компонента переходов из состояния отказа в какое-либо другое состояние нет, то восстанавливаемый компонент в него попадать никогда не должен. Имитация отказа и восстановления достигается за счет введения (кроме фиктивного состояния отказа) двух других состояний - "работы" и "восстановления". В этом случае компонент последовательно меняет состояние "работа" на состояние "восстановление" с интенсивностью, соответствующей интенсивности отказов, и наоборот, состояние "восстановление" на состояние "работа" с интенсивностью восстановления.

Рассмотрим модель восстанавливаемого РЭС, ССН которого приведена на рис. 4.

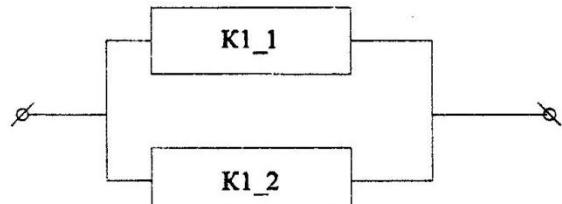


Рис. 4. Структурная схема надежности (пример 3)

Допустим, что компоненты одинаковые, интенсивность отказов каждого  $\lambda = 1 \cdot 10^{-5}$ , а интенсивность восстановления  $\mu = 1 \cdot 10^{-4}$ .

Аналогично предыдущим примерам, определяем законы распределения, и далее описываем первый узел с тремя состояниями и одним режимом:

```

knot K1_1
{
  state: Fail, Recovery, Work;
  mode: Normal;
  startState: Work;
  startMode: Normal;
  cntrlMode: unDistribution;
  tableDistribution:
    | Normal          |
    Recovery | Dis_K_recovery |
    Work     | Dis_K_fail      ;
  tableStateChange:
    | Normal          |
    Recovery | Work          |
    Work     | Recovery        ;
}
  
```

Основное отличие в таблице направлений переходов `tableStateChange` в том, что данный компонент "зациклен" между состояниями работы и восстановления. Диаграмма переходов для компонента `K1_1` представлена на рис. 5.

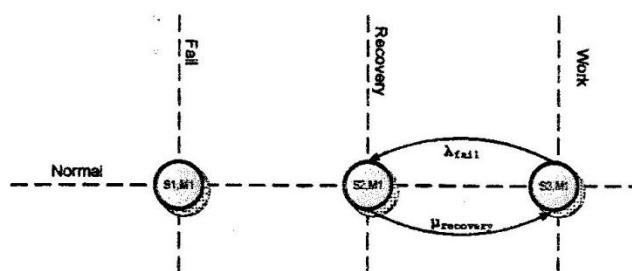


Рис. 5. Диаграмма переходов восстанавливаемого компонента

Т.к. само РЭС не восстанавливаемое, то для него только усложняются критерии отказа. Поэтому текст формальной модели главного компонента так и остается без изменений, зато критерий работоспособности необходимо преобразовать к следующему виду:

```

function FunctRES
{
    return (K1_1:Work | K1_2:Work);
}
    
```

Т.е. РЭС считается работоспособным, если хотя бы один из компонентов находится в работоспособном состоянии. Поэтому условия, описанные в предыдущих примерах, будут неадекватны, т.к. компоненты в данной модели не достигают состояния "полного" отказа, и, в связи с этим, необходимо использовать имя работоспособного состояния.

Рассмотрим описание условия работоспособности группы компонентов типа "n из m". Для определенности рассмотрим группу из 5 компонентов `K1_1, K1_2, K1_3, K1_4, K1_5`, для работы которой необходимо, чтобы функционировали два любых компонента. Наиболее удобной формой записи условия функционирования этой группы является следующее логическое выражение:

$$(K1\_1 + K1\_2 + K1\_3 + K1\_4 + K1\_5) \geq 2$$

Т.к. каждый из операторов `<KnotName>` возвращает "1", если узел в работоспособном состоянии, то значение, вычисленное в скобках, будет равняться числу работоспособных элементов, поэтому сравнив это число с числом "2" можно определить, работает РЭС или отказалось.

Следующим интересным аспектом является моделирование условий применения компонентов, частным случаем которых является "сессийный" режим работы. Для его описания используются события `switch_Event` и дополнительный компонент, "зацикленный" между двумя состояниями, аналогично восстанавливаемому компоненту (см. пример 3). Рассмотрим модель РЭС, работающего в "сессионном" режиме, ССН которого приведена на рис. 6.



Рис. 6. Структурная схема надежности (пример 4)

Вначале формальной модели объявляются временные параметры, распределения наработок компонентов в режимах работы и ожидания, а также распределения интервалов времени режимов работы и ожидания. Допустим, что РЭС имеет следующий временной график: одни сутки - работа, трое суток - ожидание (т.е. время работы и ожидания - константы). Описываются такие распределения следующим образом:

- время работы:

`distribution Dis_RES_Work (const 24);`

- время ожидания:

`distribution Dis_RES_Relax (const 72);`

Время, разыгранное по распределению с параметром `const x`, где `x` целое положительное число, всегда будет равно `x`. Далее описываем компоненты `K1` и `K2`. Для них определяем два состояния: отказ и работоспособное, и два режима: ожидание и работа. Так как имена состояний и режимов не должны совпадать, обозначим работоспособное состояние `Work`, а режим работы `Operation`:

```

knot K1
{
    state: Fail, Work;
    mode: Relax, Operation;
    startState: Work;
    startMode: Relax;
    ctrlMode: unDistribution;
    tableDistribution:
        | Relax           | Operation          |
        Work   | Dis_K_fr_relax | Dis_K_fr_work;
    tableStateChange:
        | Relax           | Operation          |
        Work   | Fail            | Fail              ;
}
    
```

Модель второго компонента в целом аналогична, и отличаться может только другими законами распределения. В модели принято, что в начальный момент времени компонент находится в режиме ожидания (`startMode:Relax`), поэтому описываем стартовое состояние компонента-переключателя состоянием `off` и "зацикливаляем" его между двумя состояниями, аналогично модели восстанавливаемого компонента:

```

knot Switcher
{
    state: Fail, On, Off;
    mode: Normal;
    startState: Off;
    startMode: Normal;
    ctrlMode: unDistribution;
    tableDistribution:
        | Normal          |
        On    | Dis_RES_Work |
        Off   | Dis_RES_Relax;
    tableStateChange:
        | Normal          |
        On    | Off            |
        Off   | On             ;
}
    
```

Такая модель компонента будет находиться в состоянии `On` 24 часа (в соответствии с параметром

распределения *Dis\_RES\_Work*) и переключаться состояние *Off* на 72 часа (опять же в соответствии с параметром распределения *Dis\_RES\_Relax*), после чего опять возвращаться в состояние *On*.

Далее необходимо связать переключения компонента *Switcher* с режимами работы РЭС, то есть по переключению *Switcher* в состояние *On K1* и *K2* должны переключаться в режим *Operation*, и, наоборот, по переключению в состояние *Off K1* и *K2* переключаются в режим *Relax*. Это моделируется при помощи двух событий:

```
switch_event toON (->Switcher:On)
{
    set_mode (K1:Operation);
    set_mode (K2:Operation);
}
switch_event toOFF (->Switcher:Off)
{
    set_mode (K1:Relax);
    set_mode (K2:Relax);
}
```

Опять же модель РЭС остается без изменений, за исключением критериев отказов. Логическое выражение обычно не включает в себя зависимость от состояния переключателя, в этом случае все просто, но иногда (особенно для восстанавливаемых компонентов) условием отказа является неработоспособность именно во время сеанса работы. Т.е., пока РЭС не работает, условие работоспособности должно быть всегда истинно:

```
function FunctREA
{
    return Switcher:Off | (K1&K2);
};
```

### **Заключение**

В работе были рассмотрены примеры формальных моделей наиболее часто встречающихся элементов структурно-сложных РЭС и возможности специализированного языка описания отказов для программирования таких моделей. Благодаря введению иерархической структуры и за счет особенностей описания ССН РЭС формальные модели легко подвергаются модификации при рассмотрении более сложных ССН, а встроенные в программное средство инструменты верификации моделей позволяют подтвердить их соответствие исследуемому РЭС.

Результаты верификационных расчетов показали приемлемую для инженерных расчетов точность оценки показателей надежности [9, 10]. Кроме того, овладеть средствами языка можно достаточно быстро, поэтому при разработке модели на первый план выходит доскональное знание исследуемого РЭС. Это позволяет сделать вывод о том, что разработанный язык и средства моделирования являются перспективными для прогнозной оценки надежности структурно-сложных и реконфигурируемых РЭС на ранних этапах проектирования и вполне доступны инженерам-проектировщикам.

### **Литература:**

1. ОСТ 4Г.0.012.242-84. Аппаратура радиоэлектронная. Методика расчета показателей надежности.
2. ГОСТ 27.301-95. Надежность в технике. Расчет надежности. Основные положения.
3. Ушаков И.А. Надежность: прошлое, настоящее, будущее. Обзор // Методы менеджмента качества. 2001. № 6.
4. Федухин А.В. Моделирование надежности невосстанавливаемой нерезервированной системы с последовательной структурой элементов // Математичні машини і системи. 2008. № 1.
5. Федухин А.В. Моделирование надежности восстанавливаемой резервированной системы со структурой типа "к из n" // Математичні машини і системи. 2008. № 4.
6. Рафалович С.А., Задорожный В.Н. Решение уравнений в переключательных функциях на GPSS WORLD [электронный ресурс]. - Режим доступа: <http://imlab.acouy.ru/default.aspx?page=articles>
7. Жаднов В.В., Полесский С.Н., Тихменев А.Н. Разработка моделей надежности для проектных исследований надежности радиоэлектронной аппаратуры / Радиовысотометрия-2010: Сб. трудов Третьей Всероссийской научно-технической конференции. // Под ред. А.А. Иофина, Л.И. Пономарева. - Екатеринбург: Форт Диалог-Исеть, 2010. С. 200-201.
8. Жаднов В.В. Тихменев А.Н. Моделирование компонентов электронных средств с реконфигурируемой структурой / Надёжность и качество: Труды международного симпозиума в 2-х т. Т.1 // Под ред. Н.К. Юркова. - Пенза: ПГУ, 2010. С. 330-331.
9. Жаднов В.В., Авдеев Д.К., Тихменев А.Н. Проблемы расчета показателей достаточности и оптимизации запасов в системах ЗИП // Надежность. 2011. № 3. С. 53-60.
10. Жаднов В.В., Полесский С.Н., Тихменев А.Н. Современные подходы к исследованию безотказности электронных средств циклического применения / Надежность и качество-2012: труды Международного симпозиума: в 2-х т. // Под ред. Н.К. Юркова. - Пенза: ПГУ, 2012. С. 70-74.

**Тихменев Александр Николаевич,**  
магистр техники и технологии,  
аспирант МИЭМ НИУ ВШЭ.  
e-mail: alextikhmenev@gmail.com

**Жаднов Валерий Владимирович,**  
канд. техн. наук, доцент кафедры  
"Радиоэлектроника и телекоммуникации"  
МИЭМ НИУ ВШЭ.  
e-mail: vzhadnov@hse.ru

A.N. Tikhmenev, V.V. Zhadnov

# **RELIABILITY PREDICTION STRUCTURAL-DIFFICULT RADIO-ELECTRONIC DEVICES BY METHODS OF IMITATING MODELLING**

This paper review the use of simulation to assess the reliability of electronic means with complex redundancy and reconfiguration algorithms. The models and the resources of a specialized language used to describe failures of electronic means with reconfigurable structure are illustrated by several variants of electronic means with complex structures.

**Keywords:** radio-electronic devices, reliability, imitating modeling

#### **References:**

1. 4G.0.012.242 OCT-84. Radio-electronic equipment. The method of calculating the reliability.

2. GOST 27.301-95. Reliability engineering. Calculation of reliability. The main provisions.

3. Ushakov I.A. Reliability: Past, present, future. Review // Methods of Quality Management. 2001. № 6.

4. Feduhin A.V. Simulation of non-redundant system reliability unrecoverable with consecutive structure elements // Mathematical Machines i system. 2008. Number 1.

5. Feduhin A.V. Modeling the reliability of renewable redundant system with a structure of "k of n" // Mathematical Machines i system. 2008. Number 4.

6. Rafalovich S.A., Zadorozhnyj V.N. Solution of the equations in the switching functions on GPSS WORLD [electronic resource]. - Mode of access: <http://imlab.acouy.ru/default.aspx?page=articles>

7. Zhadnov V.V. Polessky S.N. Tikhmenev A.N. Development of reliability models for proektny reliability studies of electronic equipment / radio altimeter-2010: Sat. Proceedings of the Third All-Russian Conference. // Ed. A.A. Iofina, L.I. Ponomarev. - Ekaterinburg Fort Dialog-Iset, 2010. PP. 200-201.

8. Zhadnov V.V. Tikhmenev A.N. Modeling of electronic components with re-configurable structure / Reliability and Quality: Proceedings of the International Symposium in 2 volumes, Vol.1 // Ed. N.K. Yurkova. - Penza: PSU. 2010. PP. 330-331.

9. Zhadnov V.V. Avdeev D.R., Tikhmenev A.N. Problem of calculating sufficiency and optimize inventory systems spare parts.// Reliability. 2011. № 3. PP. 53-60.

10. Zhadnov V.V. Polessky S.N., Tikhmenev A.N. Modern approaches to the study of non-repudiation of electronic cyclic application / Reliability and Quality-2012: Proceedings of the International Symposium: in 2 volumes // Ed. N.K. Yurkova. - Penza: PSU, 2012. PP. 70-74.

*Tikhmenev Alexander Nikolaevich,  
Master of Engineering and Technology,  
Postgraduate MIEM HSE.  
e-mail: alextikhmenev@gmail.com*

*Zhadnov Valeriy Vladimirovich,  
Candidate. tech., assistant professor of  
"Electronics and Telecommunications"  
MIEM HSE.  
e-mail: vzhadnov@hse.ru*

**Tikhmenev Alexander Nikolaevich,**  
**Master of Engineering and Technology,**  
**Postgraduate MIEM HSE**  
**e-mail: alextikhmenev@gmail.com**

**Zhadnov Valeriy Vladimirovich,**  
Candidate. tech., assistant professor of  
"Electronics and Telecommunications"  
MIEM HSE.  
e-mail: vzhadnov@hse.ru