

УДК 681.3(07)

**С.В. Ленков,**  
доктор технических наук, профессор,  
**В.О. Браун,**  
кандидат технических наук, доцент,  
**С.А. Пашков,**  
кандидат военных наук, доцент,  
**В. Н. Цыцарев,**  
кандидат технических наук, доцент

## ИМИТАЦИОННАЯ СТАТИСТИЧЕСКАЯ МОДЕЛЬ ПРОЦЕССА ОТКАЗОВ РЕЗЕРВИРОВАННЫХ РАДИОЭЛЕКТРОННЫХ СИСТЕМ

*Рассматриваются алгоритмы имитационной статистической модели, с помощью которой определяются оценки показателей надежности и стоимости эксплуатации радиоэлектронной системы, включающей в себя резервированные группы элементов. Резервирование может быть постоянным или замещающим. Восстановление отказавших элементов в резервированных группах может осуществляться сразу после отказа элемента или после отказа системы в целом. Закон распределения наработки до отказа отдельных элементов может быть произвольным (определяется выбором датчика случайных чисел).*

**Ключевые слова:** имитационная статистическая модель, показатель надежности, стоимость эксплуатации.

*Розглядаються алгоритми імітаційної статистичної моделі, за допомогою якої визначаються оцінки показників надійності і вартості експлуатації радіоелектронної системи, що включає резервовані групи елементів. Резервування може бути постійним або замінюючим. Відновлення елементів, що відмовили, в резервованих групах може здійснюватися відразу після відмови елементу або після відмови системи в цілому. Закон розподілу напрацювання повністю окремих елементів може бути вільним (визначається вибором датчика випадкових чисел).*

**Ключові слова:** імітаційна статистична модель, показник надійності, вартість експлуатації.

*The algorithms of simulation statistical model, which are determined by the assessment of reliability and cost-tis operation of radioelectronic system, comprising reserver-private group elements are discussed. Recovery of failed elements in redundant groups can be carried out immediately after a component failure, or after a failure of the system as a whole. The distribution of time to failure of individual elements can be arbitrary (determined by the choice of a random numbers generator).*

**Keywords:** statistical simulation model, indicator reliability, cost of operation.

### Введение

Для радиоэлектронных систем (РЭС) структурное резервирование является эффективным средством повышения уровня их надежности, поэтому

представляется важным иметь инструменты для прогнозирования показателей надежности (ПН) и стоимости эксплуатации (СЭ) с учетом введения в систему структурного резерва с теми или иными параметрами. Одной из особенностей РЭС как технических объектов является иерархическая организация их конструкции – система, как правило, состоит из шкафов, шкафы состоят из блоков, блоки из субблоков, и т.д. (неважно, как называются элементы различных конструктивных уровней). С учетом этого понятно, что структурное резервирование в РЭС может вводиться теоретически на любом конструктивном уровне. Поэтому поставим перед собой задачу построить математическую модель, с помощью которой устанавливается зависимость ПН и СЭ от параметров резервирования, вводимого на любом конструктивном уровне и для любого количества элементов системы.

Будем рассматривать только восстанавливаемые РЭС, предназначенные для эксплуатации в течение длительного периода времени. В качестве ПН и СЭ примем следующие показатели:

- $T_0$  – средняя наработка на отказ;
- $T_B$  – среднее время восстановления;
- $c_э$  – удельная стоимость эксплуатации.

Показатели  $T_0$  и  $T_B$  – это стандартные ПН и каких-либо пояснений не требуют [1]. В показателе учитывается стоимость заменяемых элементов и стоимость операций по их замене.

Известно, что при любых расчетах надежности принципиально важным является выбор способа задания условий работоспособности системы. В нашем случае нам еще нужно этот способ согласовать со способом представления конструктивной структуры системы.

Конструктивную структуру системы условимся задавать графом (деревом), корневая вершина которого представляет систему в целом, а висячие вершины графа представляют элементы нижнего конструктивного уровня. На рис. 1 а показан пример такого графа. Элементы нижнего конструктивного уровня будем называть *простыми* – их внутренний состав нас в данный момент не интересует. Все остальные элементы старших конструктивных уровней будем называть *составными*, они имеют в своем составе другие конструктивные элементы, которые могут быть как простыми, так и составными. Все простые элементы на рис. 1 а изображены кружками, составные – прямоугольниками.

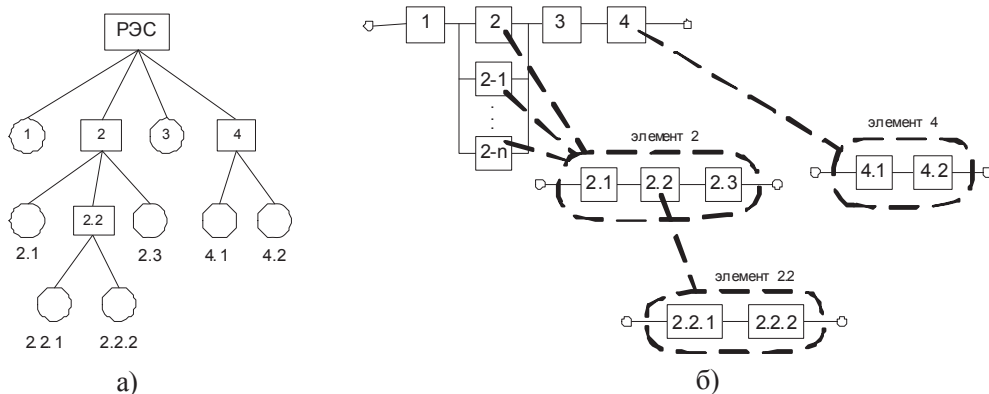


Рис. 1. Конструктивная и надежность структура системы с резервированным элементом 2

Надежностную структуру системы будем представлять структурной схемой надежности (ССН) [2]. Все элементы, входящие в состав составного элемента, считаются соединенными последовательно. Каждый элемент (простой или составной) может быть как единичным элементом, так и представлять собой резервированную группу (РГ) однотипных элементов. Резервирование в группах может быть постоянным (нагруженным) или замещающим (ненагруженным). Таким образом, надежностная структура системы в целом является последовательно-параллельной.

На рис. 1б изображена иерархия ССН для системы, конструктивная структура которой задана графом на рис. 1а, для случая, если элемент 2 имеет параллельную надежностную структуру, а элементы 2.2 и 4 – последовательную.

Будем рассматривать два режима восстановления:

а) **восстановление после отказа системы.** Восстановление начинается только после отказа системы в целом. Восстанавливаются все отказавшие к этому моменту времени элементы, включая резервные.

б) **постоянное восстановление.** Восстановление осуществляется непрерывно по мере отказов элементов. Если отказ произошел в резервированной группе, то система во время восстановления остается работоспособной за счет оставшихся исправных резервных элементов.

Восстановление начинается сразу после возникновения отказа системы или элемента в соответствии с выбранным режимом восстановления. Задержками времени индикации отказов и времени переключения резерва будем пренебрегать.

Для описанной системы задача построения математической модели для определения показателей  $T_0$ ,  $T_v$  и  $c_s$  является достаточно сложной, вряд ли представляется возможным решить данную задачу аналитическими методами. Поэтому для решения поставленной задачи предлагается использовать метод имитационного статистического моделирования [3].

В статье рассматривается имитационная статистическая модель (ИСМ), с помощью которой определяются оценки показателей  $T_0$ ,  $T_v$  и  $c_s$  для РЭС с иерархической конструктивной структурой и с резервированием части элементов при произвольном распределении наработки до отказа элементов.

## 1. Алгоритм имитационной статистической модели

### 1.1. Структуры данных, используемые алгоритмом

Исходная информация для модели содержится в базе данных (БД), в которую вводится вся необходимая информация о системе и ее элементах (показатели надежности и стоимости элементов, информация о конструктивной и надежностной структуре, и т.д.). По этой информации в памяти компьютера создается списковая структура данных, изоморфная конструктивной структуре системы. Списки создаются на базе компонента Delphi tList [4]. Элементами этой структуры являются объекты (экземпляры класса в терминологии объектно-ориентированного программирования), тип этих объектов имеет следующее описание:

```
tE = class(tObject)
    id      : Integer;    // идентификатор элемента
```

```

ur      : Byte;      // номер конструктивного уровня
te      : Byte;      // тип элемента
n       : Integer;   // число элементов в группе
name    : String;    // наименование элемента
ListE   : tList;     // список подчиненных элементов
ListE_rc : tList;    // список резервных элементов-копий
...
t       : Double;    // запланированное время отказа
...
constructor Create(AEpr:tE;Ate:Byte);
destructor Destroy; override;
end;

```

Приведенное описание представлено в нотации, принятой в языке программирования Delphi [4]. Здесь приведен только небольшой фрагмент описания, достаточный для понимания последующего материала. Назначение элементов этого описания мы будем объяснять ниже по мере необходимости их использования. Поясним только, что программный объект, созданный в соответствии с этим описанием, представляет собой определенную область оперативной памяти (ОП) компьютера, в которой хранятся данные, относящиеся к этому объекту, а сам такой объект является программным представлением отдельного конструктивного элемента моделируемой РЭС.

Объект типа tE создается с помощью конструктора Create, который представляет собой обычную программную процедуру. Уничтожение объекта (освобождение занимаемой памяти) осуществляется с помощью деструктора Destroy. Доступ к отдельным полям описания осуществляется по принципу “через точку”. Например, если E – это указатель на некоторый объект, то обратиться к имени конструктивного элемента, представляемого этим объектом, можно с помощью записи E.name.

Поле ListE в этом описании – это список указателей на элементы, входящие в данный конструктивный элемент (элемент, представленный данным объектом). Если элемент простой, то список ListE пуст (ListE=nil). Каждый элемент списка E ( $E \in \text{ListE}$ ) является указателем на такой же объект типа tE, в котором содержится информации о соответствующем конструктивном элементе.

Создание некоторого объекта E типа tE осуществляется с помощью вызова конструктора, имеющего следующий вид:

$$E:=tE.Create(E1,te), \quad (1)$$

где E – указатель на создаваемый элемент; E1 – указатель на элемент старшего конструктивного уровня, в состав которого входит создаваемый элемент E; te – тип создаваемого элемента (te=0 – составной элемент; te=1 – составной элемент, внутри которого только простые элементы; te=2 – простой элемент).

Для простоты далее мы будем говорить о создании элемента E, хотя, если говорить строго, в действительности создается объект типа tE, содержащий данные об элементе E, а E – это просто ссылка на создаваемый объект. Надеемся, что далее из контекста будет ясно, о чем идет речь.

Если создаваемый элемент  $E$  представляет резервированную группу элементов (в этом случае  $E.n > 1$ ), тогда дополнительно создается список резервных копий  $ListE\_rc$ , элементами которого являются указатели на элементы-копии (число элементов в списке  $E.ListE\_rc$  равно  $E.n-1$ ). Структура данных элементов-копий полностью повторяет структуру основного элемента  $E$ . Для приведенного выше на рис. 1 примера системы, в которой резервированным элементом является элемент 2, представленная в памяти конструктивная структура приобретает вид, показанный на рис. 2. В наименованиях резервных копий добавляется номер копии.

При создании с помощью вызова (1) некоторого элемента  $E$  создаются автоматически и все объекты, представляющие конструктивные элементы, входящие в состав элемента  $E$  (включая все внутренние резервные копии, если там имеются элементы, представляющие РГ). Это осуществляется путем рекурсивных вызовов конструктора  $Create$ .

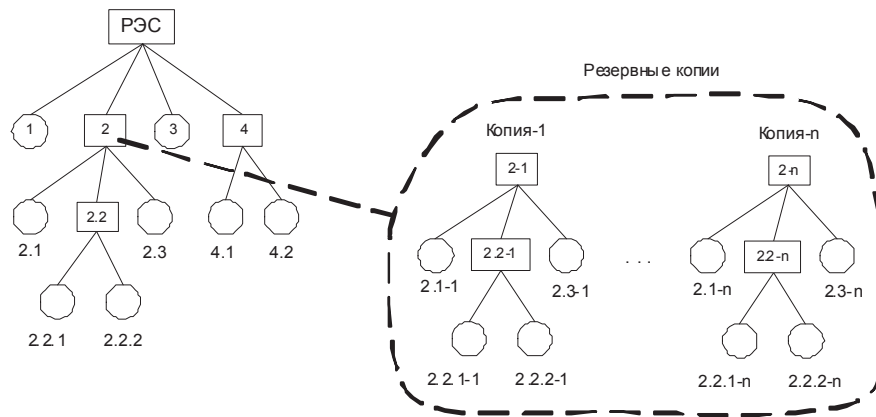


Рис. 2. Конструктивная структура системы с резервированными элементами (резерв имеется у элемента 2)

Таким образом, оказывается, что вся иерархическая структура данных, представляющая конструктивную структуру системы, создается в ОП с помощью единственного вызова следующего вида:

$$E0 := tE.Create(\mathit{nil}, 0), \quad (2)$$

где  $E0$  – указатель на созданные данные;

$\mathit{nil}$  – ссылка на “пустой” объект (это означает, что создаваемый элемент  $E0$  не входит ни в какие другие элементы).

Структура данных  $E0$  создается сразу при запуске программы модели, до начала процесса моделирования.

**Принципы моделирования.** Структуру моделируемого процесса можно представить графом состояний и переходов, представленным на рис. 3. Рассматриваются следующие состояния системы: 0 – исправное состояние; 1 – состояние частичного отказа; 2 – полный отказ системы. В состоянии 1 система переходит

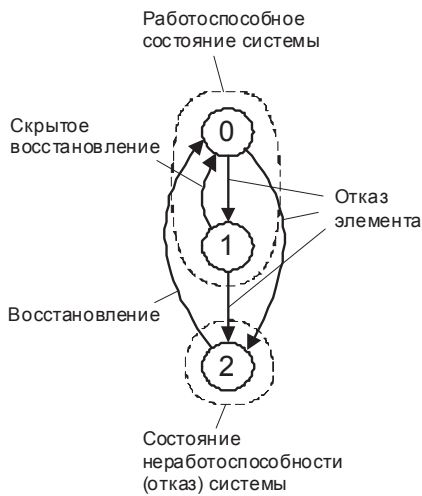


Рис. 3. Граф состояний и переходов моделируемого процесса.

запланированных значений времени. Найденное значение принимается в качестве текущего модельного времени, соответствующего текущему событию. Далее производится обработка текущего события, которая заключается в выполнении некоторых действий (вычислений), содержание которых зависит от типа события. При обработке события осуществляется накопление необходимой статистики и планирование следующего события. Новое значение времени данного события записывается в КС вместо прежнего значения.

Затем описанные действия повторяются – вновь осуществляется просмотр КС и определяется следующее значение модельного времени и связанное с ним событие, и т.д. Таким образом, осуществляется последовательное “продвижение” модельного времени по временной оси до тех пор, пока оно не достигнет заданного значения продолжительности эксплуатации системы. Описанная итерация процесса моделирования повторяется многократно до тех пор, пока не будет накоплен достаточный объем статистики. По накопленной статистике вычисляются оценки искомых показателей. Такова общая схема процесса моделирования.

**Структурная схема алгоритма модели.** Структурная схема алгоритма модели представлена на рис. 4. Оператор 1 осуществляет ввод исходной информации из БД и по введенной информации формирует в ОП структуру  $E_0$ , в которой содержится необходимая информация о системе и ее элементах. Одновременно создается список всех простых элементов  $ListE_0$  (элементами списка являются указатели на соответствующие элементы системы). Оператор 2 инициализирует значения всех переменных, в которых будет накапливаться статистика. Устанавливается также начальное значение переменной, которая используется для подсчета числа выполненных итераций моделирования. Оператором 3 генерируются и сохраняются в КС моменты времени первых событий – отказов всех простых элементов. Для генерирования случайной наработки до отказа элементов используется датчик случайных чисел, подчиняющийся  $DN$ -распределению [5].

Оператор 4 определяет текущее модельное время  $t$  следующим образом:

$$t := \min E.t; \quad (\forall E \in ListE_0)$$

в том случае, если у отказавшего элемента имеется исправный резервный элемент (в этом случае система остается работоспособной). Стрелками на рисунке показаны события, заключающиеся в переходе системы из одного события в другое.

Общая идея алгоритма моделирования состоит в следующем. Вначале генерируются случайные моменты времени первых событий модели, которыми являются отказы простых элементов. Эти моменты времени запоминаются в некоторой области ОП, которую будем называть календарем событий (КС). Затем начинается последовательный (циклический) “просмотр” КС и при каждом таком просмотре находится наименьшее из



то есть значение  $t$  находится как наименьшее из значений  $E.t$ , записанных в соответствующем поле элемента  $E$ , взятого из списка  $ListE0$ . Таким образом, КС реализуется в данном случае в виде множества  $\{E.t\}$ . Оператором 4 определяется также указатель на элемент  $E$ , отказ которого произошел в момент времени  $t$ .

Если текущее время  $t$  не вышло за пределы заданного периода эксплуатации  $T_3$ , выполняется оператор 6, который проверяет, не является ли текущее событие отказом. Если текущее событие "отказ" (элемента  $E$ ), то далее выполняются операторы 7-11, осуществляющие обработку этого события.

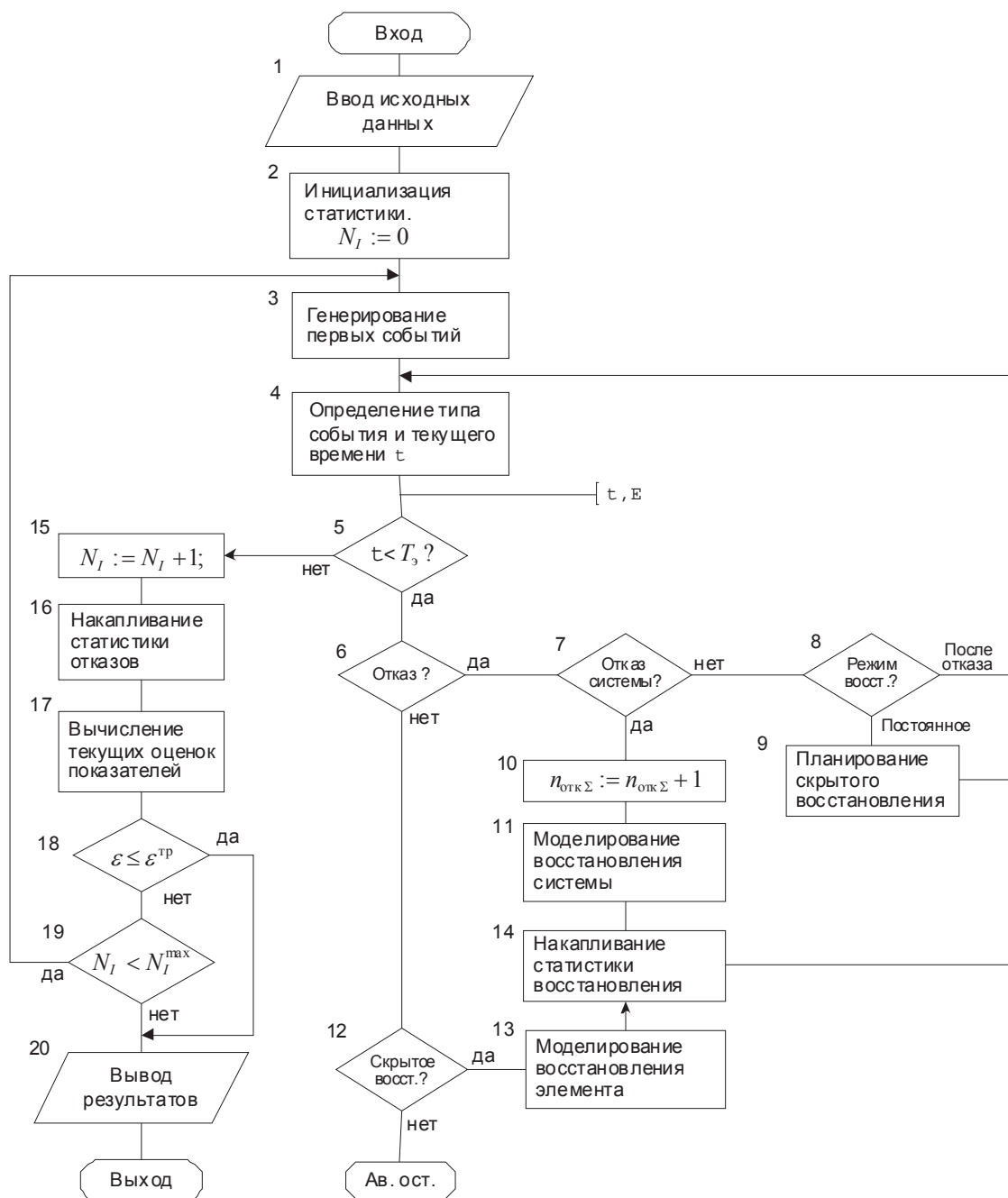


Рис. 4. Алгоритм имитационной статистической модели процесса отказов РЭС с резервированными элементами

Оператор 7 проверяет состояние системы после отказа элемента Е. Из-за ограниченный размера статьи мы не приводим его описание. Детальное описание алгоритма оператора 7 имеется в [6–8].

Если произошел отказ системы (произошел переход  $0 \rightarrow 2$  или  $1 \rightarrow 2$ ), выполняются операторы 10, 11. Оператор 10 подсчитывает суммарное количество отказов:  $n_{\text{отк}\Sigma} := n_{\text{отк}\Sigma} + 1$ . Оператор 11 имитирует восстановление системы. Описание алгоритма этого оператора подробно рассматривается в [6].

Если система не отказала (переход  $0 \rightarrow 1$ ), оператор 7 передает управление операторам 8, 9. Оператор 9 планирует скрытое восстановление элемента Е (только в режиме постоянного восстановления).

Оператор 12 проверяет, является ли текущее событие “скрытым восстановлением”. Если “да”, то выполняется оператор 13, имитирующий скрытое восстановление элемента Е. Отличие оператора 13 от оператора 11 состоит в том, что в нем имитируется восстановление только одного элемента (оператором 13 имитируется восстановление всех отказавших к данному моменту времени элементов).

Оператор 14 производит накопление статистики восстановления: накапливаются значения в переменных  $\tau_{\text{в}\Sigma}$  и  $C_{\text{в}\Sigma}$  – суммарное время восстановления и суммарные затраты стоимости на восстановление. Затем управление возвращается оператору 4 для продолжения поиска последующих событий в текущей итерации. Итерация завершается, как только текущее время  $t$  превысит значение  $T_3$  и оператор 5 передаст управление оператору 15.

Оператор 15 подсчитывает число выполненных итераций  $N_I$ . Оператор 16 накапливает статистику отказов:  $n_{\text{отк}\Sigma}$  и  $n_{\text{отк}\Sigma 2}$  – накопленные суммарные значения числа отказов и квадрата числа отказов<sup>1</sup> в итерациях моделирования.

Оператор 17 вычисляет текущие оценки результирующих показателей в соответствии со следующими выражениями:

$$\begin{aligned} T_0^* &:= (T_3 N_I) / n_{\text{отк}\Sigma} ; \\ T_{\text{в}}^* &:= \tau_{\text{в}\Sigma} / n_{\text{отк}\Sigma} ; \\ c_3^* &:= C_{\text{в}\Sigma} / (T_3 \cdot N_I), \end{aligned} \quad (3)$$

где  $T_0^*$ ,  $T_{\text{в}}^*$  и  $c_3^*$  – полученные в результате статистического моделирования оценки показателей  $T_0$ ,  $T_{\text{в}}$  и  $c_3$  соответственно.

Помимо этих показателей вычисляется также величина  $\varepsilon$  относительной ошибки оценки  $T_0^*$ . Величина  $\varepsilon$  определяется как половинный 95-процентный доверительный интервал для оценки относительной величины  $T_0 / T_0^*$ :

$$\text{Вер} \left\{ 1 - \varepsilon < \frac{T_0}{T_0^*} < 1 + \varepsilon \right\} = 0,95,$$

где  $T_0$  – неизвестное истинное значение средней наработки на отказ системы.

<sup>1</sup> Для вычисления оценки дисперсии.



Операторы 18, 19 обеспечивают завершение процесса моделирования. Оператор 18 проверяет условие достижения требуемой точности (относительная ошибка  $\varepsilon$  достигла требуемого значения  $\varepsilon^{\text{TP}}$ ). Оператор 19 завершает процесс моделирования в том случае, если текущая точность результатов моделирования еще не достигла заданного требования, но при этом достигнуто заданное ограничение на максимальное число итераций  $N_I^{\text{max}}$ . Оператор 20 выводит полученные результаты моделирования на экран дисплея.

Данный алгоритм реализован программно средствами системы программирования Delphi в рамках программы ISMPN, описание которой можно найти в [9].

**Пример моделирования.** Для примера возьмем систему, структура которой представлена на рис. 1. Показатели надежности и стоимости для всех простых элементов зададим одинаковыми:

- наработка до отказа подчинена  $DN$ -распределению с параметрами: математическое ожидание  $\mu_i = 10000$  ч; коэффициент вариации  $\nu_i = 1$ ;
- среднее время восстановления (замены) элемента  $T_{вi} = 1$  ч;
- стоимость элементов  $C_{0i} = 10$  у.е.; стоимость операции замены  $C_{\text{зам}i} = 1$  у.е.

Продолжительность эксплуатации системы  $T_0 = 20$  лет. Число итераций моделирования  $N_I^{\text{max}} = 500$ .

При этих исходных данных и при условии, что резервирование в системе отсутствует, были получены такие оценки ПН и СЭ:

$$T_0^* = 1248 \text{ ч}; \quad T_{в}^* = 1 \text{ ч}; \quad c_3^* = 0,00881 \text{ у.е./ч}; \quad \varepsilon = 0,048.$$

Теперь в систему введем резервирование для элемента 2 и произведем расчеты при различных параметрах резервирования и режимах восстановления. Полученные результаты расчетов сведены в табл. 1.

Таблица 1

**Показатели надежности и стоимости эксплуатации системы  
при резервировании элемента 2**

Режим восстановления	$n_{\text{рез}}$	Показатели надежности и стоимости эксплуатации							
		Нагруженное резервирование				Ненагруженное резервирование			
		$T_0^*$ , ч	$T_{в}^*$ , ч	$c_3^*$ , у.е./ч	$\varepsilon$	$T_0^*$ , ч	$T_{в}^*$ , ч	$c_3^*$ , у.е./ч	$\varepsilon$
Восстановление после отказа системы	1	1501	1,68	0,01232	0,040	2338	1,21	0,00568	0,051
	2	1640	2,32	0,01553	0,041	2459	1,25	0,00561	0,056
	3	1732	2,95	0,01874	0,044	2486	1,31	0,00582	0,055
	4	1802	3,60	0,02195	0,050	2494	1,33	0,00589	0,060
	5	1857	4,25	0,02517	0,056	2496	1,34	0,00590	0,057
Постоянное восстановление	1	2496	1,0	0,01321	0,060	2499	1,0	0,00949	0,062
	2	2497	1,0	0,01761	0,050	2497	1,0	0,00952	0,066
	3	2498	1,0	0,02201	0,065	2493	1,0	0,00952	0,061
	4	2498	1,0	0,02643	0,054	2497	1,0	0,00952	0,058
	5	2495	1,0	0,03085	0,069	2498	1,0	0,00951	0,059

На рис. 4 все эти результаты представлены для наглядности в виде графиков.

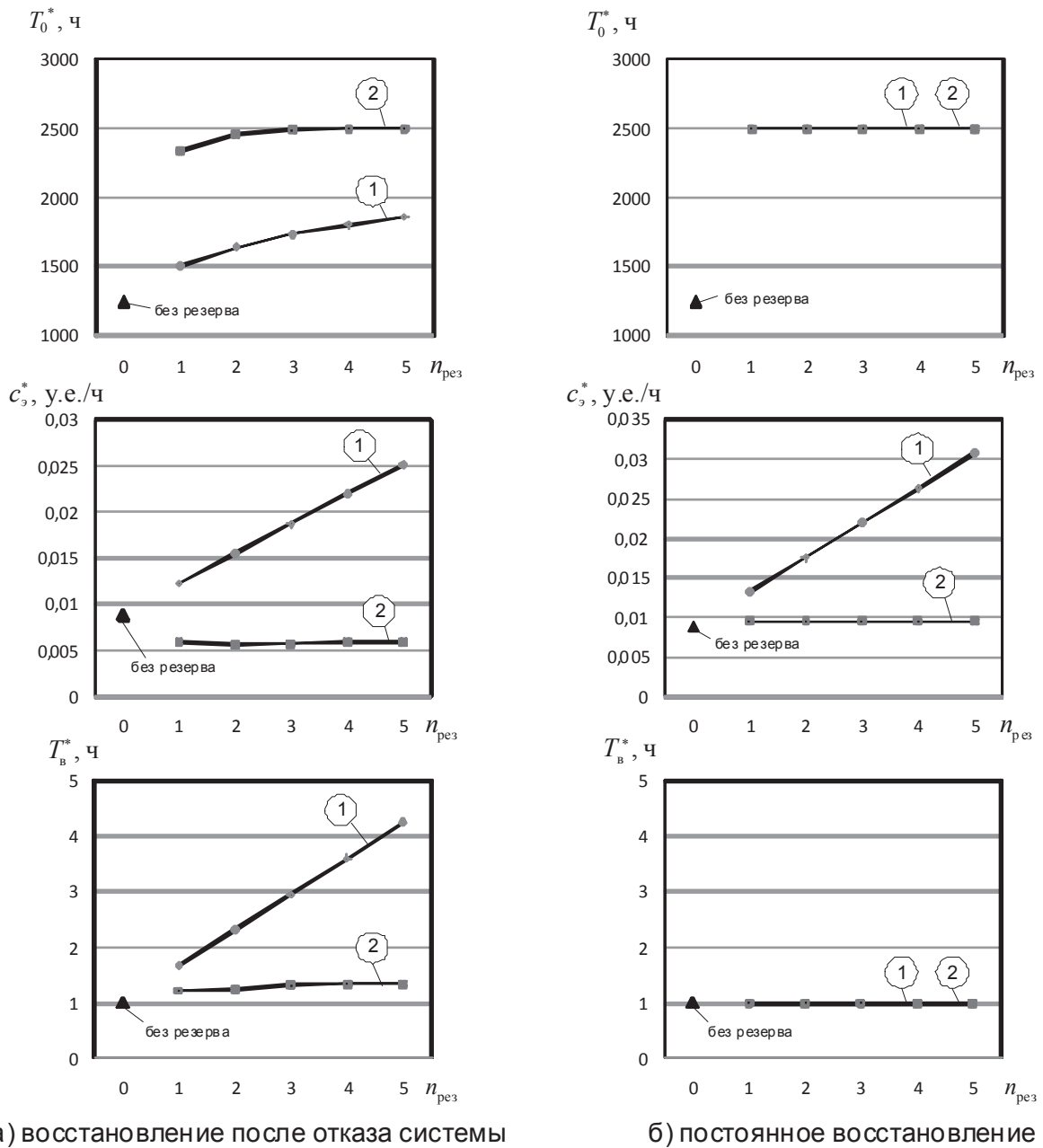


Рис. 4. Графики зависимости показателей  $T_0^*$ ,  $T_{в}^*$  и  $c_{э}^*$  от числа резервных элементов элемента 2 (1 – нагруженный резерв; 2 – ненагруженный резерв).

**Выводы**

Анализируя полученные результаты можно сделать следующие выводы:

- 1) ненагруженное резервирование является более эффективным (по критерию “надежность/стоимость”) по сравнению с нагруженным резервированием;
- 2) режим постоянного восстановления отказавших элементов является более эффективным по сравнению с режимом, когда восстановление производится только после отказа системы;

3) в режимі постійного відновлення показателі надійності системи  $T_0^*$  і  $T_B^*$  не залежать від кількості резервних елементів;

4) незважаючи на те, що при будь-якому резервуванні завжди потребуються додаткові витрати коштів, результуюча вартість експлуатації системи з резервуванням може виявитися нижче, ніж вартість експлуатації аналогічної системи без резервування (див. графік показателя  $c_3^*$  на рис. 4 а).

Ці висновки можуть показатися тривіальними, так як вони очевидні і повністю відповідають сенсу і змісту моделюваного процесу. Однак для нас вони все ж важливі, тому, що повністю підтверджуються конкретними результатами моделювання, отриманими з допомогою розробленої моделі. Це може слугувати підтвердженням адекватності розробленої ІСМ і правильності її програмної реалізації.

### СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ГОСТ 27.002-89. Надійність в техніці. Основні поняття. Термини і визначення. Введ. 01.07.1990.
2. ГОСТ 27.301-95. Надійність в техніці. Розрахунок надійності. Основні положення. Введ. 01.01.1997.
3. Бусленко Н.П. Моделювання складних систем / Н.П. Бусленко. – М. : Наука, 1978. – 400 с.
4. Дарахвелідзе П.Г. Програмування в Delphi 7 / П.Г. Дарахвелідзе, Е.П. Марков. – СПб. : БХВ-Петербург, 2004. – 784 с.
5. Стрельников В.П. Оцінка і прогнозування надійності електронних елементів і систем / В.П. Стрельников, А.В. Федухин. – К. : Логос, 2002. – 486 с.
6. Моделювання відмов структурно надлишкових радіоелектронних систем / С.В. Ленков, С.А. Пашков, В.А. Осьпа, В.Н. Цыцарев // Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка. – К., 2014. – № 47. – С. 23–26.
7. Оцінка показувачів безотказності складного відновлюваного об'єкта РЭТ при довільних законах розподілу наробітку до відмови елементів / С.В. Ленков, К.Ф. Боряк, В.Н. Цыцарев, Г.В. Банзак, В.В. Крихта // Сучасна спеціальна техніка. – 2010. – № 3(22). – С. 86–98.
8. Теоретичні можливості підвищення рівня безвідмовності складного об'єкта рет за рахунок проведення планових ремонтів / С.В. Ленков, В.О. Осипа, С.О. Пашков, Ю.В. Березовська // Сучасна спеціальна техніка. – 2014. – № 2.
9. Прогнозування надійності складних об'єктів радіоелектронної техніки і оптимізація параметрів їх технічної експлуатації з використанням імітаційних статистических моделей : монографія / С.В. Ленков, К.Ф. Боряк, Г.В. Банзак, В.О. Браун [и др.] ; під ред. С.В. Ленкова. – Одеса : Изд-во "ВМВ", 2014. – 256 с.

Отримано 4.10.2014

Рецензент Рибальський О.В., доктор технічних наук, професор.