

УДК 004.052

Ю.Н. Соколов<sup>1</sup>, В.С. Харченко<sup>1</sup>, Ю.Л. Поночовный<sup>2</sup><sup>1</sup> *Национальный аэрокосмический университет им. Н.Е. Жуковского "ХАИ", Харьков*<sup>2</sup> *Полтавский национальный технический университет им. Юрия Кондратюка, Полтава*

## ИНСТРУМЕНТИРОВАННОЕ ОЦЕНИВАНИЕ НАДЕЖНОСТИ ПРОГРАММНО-ТЕХНИЧЕСКИХ КОМПЛЕКСОВ ПРИ РОСТЕ ИНТЕНСИВНОСТИ ОТКАЗОВ

*Рассматриваются компьютерные методы оценивания надежности программно-технических комплексов (ПТК) на основе структурно-логических схем, разработанные в рамках проекта TEMPUS-SAFEGUARD National Safeware Engineering Network of Centres of Innovative Academia-Industry Handshaking (158886-TEMPUS-1-2009-1-UK-TEMPUS-JPCR). Излагаются элементы инструментально-ориентированной методики оценки надежности и готовности ПТК с использованием многофрагментных марковских моделей с учетом изменения параметров потоков отказов, вызванных проектными дефектами. Исследован случай роста интенсивности отказов вследствие внесения новых дефектов. Приведены компьютерные программы оценивания надежности и готовности ПТК. В качестве инструментария используется современный программный комплекс MATLAB.*

**Ключевые слова:** надежность, готовность, программно-технический комплекс, марковские модели, инструментальное оценивание, MATLAB.

### Введение

Функциональные возможности, надежность и безопасность информационно-управляющих систем (ИУС) как одного из классов технических систем существенно зависят от качества и надежности всех составляющих, прежде всего, программно-технического комплекса (ПТК). ПТК вместе с датчиковыми, коммуникационными средствами и исполнительными устройствами образует ИУС и состоит из аппаратных и программных средств (АС и ПС). Насколько весома роль ПС? Насколько важно учитывать их надежность при оценке ПТК?

Ответ на эти вопросы дают результаты исследований, приведенные в [1 – 3] на примере анализа отказов и аварий в ракетно-космической отрасли, ИУС АЭС, корпоративных информационных систем и других приложений: надежность ПС стала одним из основных факторов надежности и безопасности таких систем; оценивание надежности ПТК обязательно должно включать оценку показателей ПС; необходимы методические и инструментальные средства для решения этой задачи.

В данной статье рассматривается метод оценивания надежности простейшего дублированного одноверсионного (т.е. использующего одну программную версию в обоих каналах) ПТК с учетом отказов и восстановлений АС и ПС. Особенности метода являются [4]:

- метод базируется на аппарате марковских случайных процессов, позволяющем сделать оценивание формализованным и верифицируемым;
- при использовании метода обеспечивается повышение точности результатов, поскольку учиты-

вается взаимное влияние отказов АС и ПС, а также потоки отказов и восстановлений;

- применяемый показатель (коэффициент готовности) может быть использован для оценки рисков, затрат и прибыльности ПТК.

Дополнительной особенностью оцениваемого ПТК является рост интенсивности отказов ПС, который может иметь место вследствие:

- увеличения интенсивности отказов ПС при внесении новых дефектов (например, при расширении функций и повышении сложности);
- увеличение интенсивности отказов ПС при атаках на уязвимости.

В таких условиях надежность ПТК будет ухудшаться по сравнению с вариантом с постоянными интенсивностями. С практической точки зрения важно определить предельные значения коэффициента готовности и его изменений в нестационарном режиме.

Так как инструментальные средства MATLAB позволяют решать задачи моделирования и оценивания в разных вариантах [5], материал построен в порядке усложнения моделей: на первом этапе оценивается надежность двухканального одноверсионного ПТК с постоянными интенсивностями отказов (однофрагментная модель), а затем на этой основе оценивается ПТК с учетом роста интенсивности отказов (многофрагментная модель).

При оценивании будем придерживаться такой последовательности:

- принятие допущений и задание входных параметров модели;
- определение количества фрагментов и построение орграфа модели;

- составление уравнений Колмогорова по полученному оргграфу;
- решение уравнений модели в MATLAB;
- определение функции готовности, построение и анализ зависимостей.

### 1. Оценивание надежности ПТК при постоянной интенсивности отказов

На рис. 1 представлена наиболее распространенная архитектура ИУС (структурная схема надежности ПТК), включающая два резервированных аппаратных канала (например, бортовых цифровых вычислительных комплекса), в каждом из которых функционирует одинаковая версия ПС.

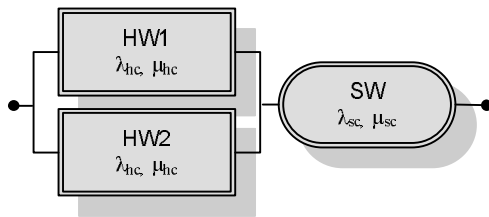


Рис. 1. Структурная схема надежности ПТК

Условное обозначение такой структуры системы –  $S_{21}$ , где 2 – количество аппаратных каналов, 1 – количество версий ПС.

При использовании однотипного оборудования в резервированных аппаратных (HW) каналах, модель имеет четыре параметра:

- интенсивности отказов и восстановлений одного аппаратного канала  $\lambda_{hc}$  (1/час) и  $\mu_{hc}$  (1/час);
- интенсивность отказов ПС (SW)  $\lambda_{sc}$  (1/час);
- интенсивность восстановления ПТК после проявления программного дефекта  $\mu_{sc}$  (1/час).

Считаем, что средства контроля идеальны: отказ обнаруживается мгновенно и точно, вероятность отказа средств равна нулю.

Для вышеуказанных параметров размеченный граф состояний (вершин графа)  $S_0, \dots, S_3$  и переходов – ребер графа (отмечены стрелками) модели представлен на рис. 2.

В каждом из состояний ПТК может находиться с вероятностью  $P_0, \dots, P_3$  соответственно. Работоспособными состояниями являются  $S_0$  и  $S_1$ . Первое из них соответствует исправному состоянию (оба канала и ПС работоспособны), второе – работоспособному состоянию ПТК при отказе одного канала и работе на резервном.

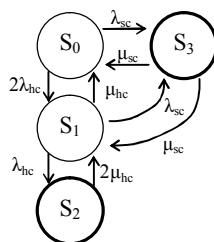


Рис. 2. Оргграф состояний ПТК

Используя оргграф, получим систему дифференциальных уравнений Колмогорова:

$$\begin{cases} dP_0(t)/dt = -(2\lambda_{hc} + \lambda_{sc})P_0(t) + \mu_{hc}P_1(t) + \mu_{sc}P_3(t), \\ dP_1(t)/dt = 2\lambda_{hc}P_0(t) - (\lambda_{hc} + \lambda_{sc} + \mu_{hc})P_1(t) + 2\mu_{hc}P_2(t) + \mu_{sc}P_3(t), \\ dP_2(t)/dt = \lambda_{hc}P_1(t) - 2\mu_{hc}P_2(t), \\ dP_3(t)/dt = \lambda_{sc}P_0(t) + \lambda_{sc}P_1(t) - 2\mu_{sc}P_3(t), \end{cases} \quad (1)$$

$$P_0(t) + P_1(t) + P_2(t) + P_3(t) = 1.$$

Функция готовности определяется как сумма вероятностей нахождения системы в работоспособных состояниях  $S_0$  и  $S_1$ :

$$P_g(t) = P_0(t) + P_1(t). \quad (2)$$

В начальный момент времени система находится в состоянии  $S_0$ .

Система дифференциальных уравнений (1) является однородной линейной системой, которую можно записать в матричной форме:

$$\frac{dP}{dt} = AP, \quad P(t_0), \quad (3)$$

где

$$P = \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix},$$

$$A = \begin{bmatrix} -(2\lambda_{hc} + \lambda_{sc}) & \mu_{hc} & 0 & \mu_{sc} \\ 2\lambda_{hc} & -(\lambda_{hc} + \lambda_{sc} + \mu_{hc}) & 2\mu_{hc} & \mu_{sc} \\ 0 & \lambda_{hc} & -2\mu_{hc} & 0 \\ \lambda_{sc} & \lambda_{sc} & 0 & -2\mu_{sc} \end{bmatrix}.$$

Получим численное решение уравнения (3) при  $\lambda_{hc} = 10^{-4}$ ;  $\mu_{hc} = 1,5$ ;  $\lambda_{sc} = 10^{-3}$ ;  $\mu_{sc} = 6$ .

Здесь и далее все интенсивности имеют размерность 1/ч. Время работы системы  $t_f = 15000$  (часов).

Тогда решение уравнения (3) можно получить с помощью следующей программы MATLAB в такой последовательности:

```
>> clear all % очистка памяти
%Ввод исходных параметров системы
lambda_sc= 5*10^-3; %интенсивность отказов ПС
lambda_hc=10^-4; %интенсивность отказов АС
mu_hc=1.5; %интенсивность восстановления АС
mu_sc= 6; %интенсивность восстановления ПС
tf=15000; % время работы системы (часов)
>>%Ввод и вычисление матрицы А
format short %установка формата чисел
A = [-(2*lambda_hc+ lambda_sc), mu_hc, 0, mu_sc
2*lambda_hc, -(lambda_hc+ lambda_sc+ mu_hc), ...
2* mu_hc, mu_sc
0, lambda_hc, -2* mu_hc, 0
lambda_sc, lambda_sc, 0, -2* mu_sc]
```

```
A =
-0.0052  1.5000   0   6.0000
 0.0002 -1.5051  3.0000  6.0000
 0        0.0001 -3.0000   0
 0.0050  0.0050   0  -12.0000
>>global A; %определение глобальной переменной
```

Благодаря использованию глобальной переменной A (одновременно в рабочей области – окне «Command Window» и в m-файле решения матричного уравнения – см. далее stiff.m) можно пропустить шаг копирования значений коэффициентов матрицы A в этом m-файле.

Далее проверим, существует ли устойчивое решение уравнения (3). Для этого вычислим собственные значения матрицы A:

```
>> vpa(eig(A),2) %здесь 2 - число значащих цифр
ans =
-12.0
(-3.4)*10^(-16)
-1.5
-3.0
```

Как видим, все собственные значения действительны и отрицательны, следовательно, решение устойчиво.

Для правильного выбора метода решения дифференциальных уравнений определим жесткость системы по известной формуле [4]:

$$S = \frac{\min \operatorname{Re} \lambda_i}{\max \operatorname{Re} \lambda_j} \gg 1, \quad i, j = 1, \dots, n, \quad i \neq j, \quad (4)$$

где  $\lambda$  – собственные значения системы.

Число S называют коэффициентом жесткости системы уравнений. Компонента с большим (по модулю) собственным значением вынуждает выбирать мелкий шаг и, одновременно, быстро перестает влиять на решение. Знак  $\gg$  ("значительно превосходит") на практике обычно означает, что  $S > 100$ .

```
>>%Определение жесткости
format short e
S=vpa(min(real(eig(A)))/max(real(eig(A))),2)
S =
3.5*10^16
```

Система уравнений сверхжесткая. Удобные и наглядные возможности для решения жестких систем предоставляет MATLAB, располагающий четырьмя специальными методами решения жестких систем:

1) ode15s (stiff/NDF) – метод конечных разностей в сочетании со схемой обратного дифференцирования (метод Гира), обеспечивает достаточно высокую точность интегрирования. NDF – numerical differentiation formulas – формулы численного дифференцирования;

2) ode23s (stiff/Mod.Rosenbrock) – модифицированный метод Розенброка 2-го порядка, обеспечива-

ет среднюю точность интегрирования со скоростью выше, чем ode15s;

3) ode23t (Mod. Stiff/Trapezoidal) – метод трапеций для интерполяции производных, рекомендуется для умеренно жестких систем, обеспечивает среднюю точность;

4) ode23tb (stiff/TR-BDF 2) – комбинированный метод Рунге-Кутты в сочетании с правилом трапеций и схемы обратного дифференцирования, обеспечивает среднюю точность. BDF – backward differentiation formulas – формулы обратного дифференцирования.

В дальнейших программах будем использовать первый метод (решатель ode15s).

Для решения матричного дифференциального уравнения создадим m-файл, показанный на рис. 3, и сохраним его под именем, например, stiff.m.

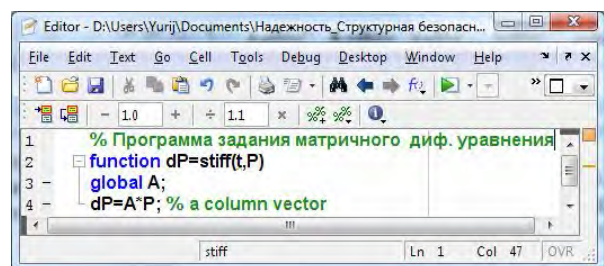


Рис. 3. m-файл матричного уравнения

При этом, если программа и m-файл находятся в разных текущих каталогах MATLAB, всплывает окно редактора, показанное на рис. 4, и дальнейшее решение не будет получено.

Для успешного продолжения решения необходимо в этом окне нажать кнопку Change Directory (изменить каталог) или Add to Path и запустить m-файл.

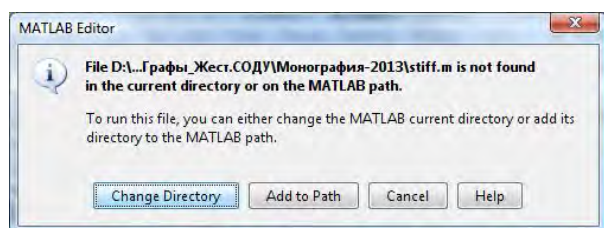


Рис. 4. Окно редактора MATLAB

Продолжая программу, введем в окне «Command Window» и выполним команды:

```
>> %Опции – задают относительную и абсолютную
%точности:
options = odeset('RelTol',1e-4, 'AbsTol',1e-9);
t0=0; %начальный момент времени
tf=15000; % время работы системы (часов)
Pt0=[1;0;0;0]; %начальные условия
[t,P] = ode15s(@stiff,[t0 tf],Pt0,options); %решение
```

Наконец, вычислим и построим функцию готовности с помощью программы:

```
>>Pg=P(:,1)+P(:,2); %извлечение элементов P0 и P1
figure(1)
plot(t,Pg,'-') %построение графика
xlabel('t'), ylabel('P_g') %метки осей
hold on
```

Точное конечное значение функции готовности можно найти, выбрав последний элемент вектора Pg и выполнив команду:

```
>> Pg1= Pg(size(Pg)) %где size – размер вектора Pg
Pg1 =
9.9958e-001
1.0000e+000
```

То есть  $Pg1(tf) = 0,99958$ .

Результат решения задачи показан на рис. 8 (пунктиром). Как видим, функция готовности  $Pg < 1$  и не зависит от времени, т.к. входные параметры модели надежности ПТК по условию задачи постоянны. Однако это не совсем так. На самом деле, в начальный момент времени  $t_0 = 0$  функция готовности  $Pg = 1$ , а затем быстро (около часа) уменьшается до некоторого установившегося значения. Убедиться в этом можно, заменив в предыдущей программе  $tf=15000$  на  $tf=1$ , т.е. выполнив следующую аналогичную программу:

```
>> options = odeset('RelTol',1e-4, 'AbsTol',1e-9);
t0=0; %начальный момент времени
tf=1; % время работы системы (часов)
Pt0=[1;0;0;0]; %начальные условия
[t,P] = ode15s(@stiff,[t0 tf],Pt0,options); %решение
Pg=P(:,1)+P(:,2);
figure(2)
plot(t,Pg,'-')
xlabel('t'), ylabel('P_g')
grid on
```

Результат решения показан на рис. 5.

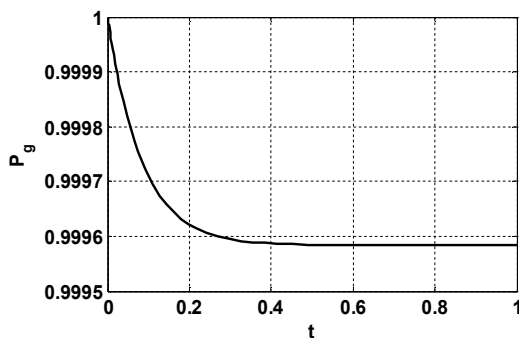


Рис. 5. Функция готовности одноверсионного ПТК на начальном этапе

## 2. Оценивание надежности ПТК при увеличении интенсивности отказов

Рассмотрим многофрагментную модель (МФМ) с учетом роста интенсивности отказов. Для построения такой модели примем ряд допущений:

- после проявления программного дефекта в программном реестре и оперативной памяти происходит накопление ошибок, вследствие чего интенсивность отказов ПС  $\lambda_{sc}$  будет увеличиваться на некоторую величину  $\Delta\lambda_{sc}$ ;

- увеличение  $\lambda_{sc}$  возможно до некоторого предела  $\lambda_{sc \max}$ .

Такая модель хорошо описывает поведение компьютеризированной системы с ограничением оперативной памяти под управлением требовательной к ресурсам операционной системы (например, Windows Vista), при одновременном запуске нескольких ресурсоемких приложений.

Так как эта модель представляет только исследовательский интерес, то в данном разделе опускаются вопросы определения значений  $\lambda_{sc \max}$ , а исследования ограничиваются расчетом модели для конкретных значений ее параметров.

Задача решается аналогично предыдущей с некоторыми особенностями.

Количество фрагментов модели  $N_{fr}$  зависит от предельного значения интенсивности отказов ПС  $\lambda_{sc \max}$ , интенсивности отказов одного фрагмента ПС  $\lambda_{sc}$  и шага изменения интенсивности отказов ПС  $\Delta\lambda_{sc}$ . Значение  $N_{fr}$  может быть вычислено по формуле:

$$N_{fr} = \frac{\lambda_{sc \max} - \lambda_{sc}}{\Delta\lambda_{sc}}$$

В общем случае орграф модели, содержащей  $k+1$  фрагментов, аналогичных орграфу на рис. 2, будет иметь вид, показанный на рис. 6.

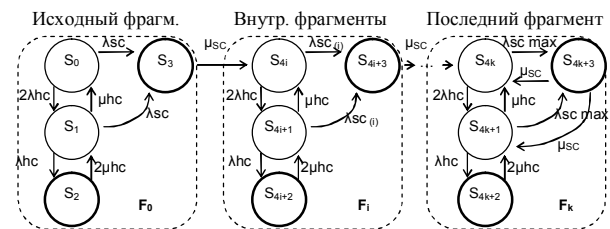


Рис. 6. Орграф ПТК с учетом роста интенсивности отказов ПС

Далее задача решается по аналогии с предыдущей. По приведенному орграфу составляем систему дифференциальных уравнений Колмогорова, которая состоит из следующих регулярных блоков:

– для исходного фрагмента F0:

$$\begin{cases} dP_0(t)/dt = -(2\lambda_{hc} + \lambda_{sc})P_0(t) + \mu_{hc}P_1(t), \\ dP_1(t)/dt = -(\lambda_{hc} + \lambda_{sc} + \mu_{hc})P_1(t) + 2\lambda_{hc}P_0(t) + \\ + 2\mu_{hc}P_2(t), \\ dP_2(t)/dt = -2\mu_{hc}P_2(t) + \lambda_{hc}P_1(t), \\ dP_3(t)/dt = -\mu_{sc}P_3(t) + \lambda_{sc}P_0(t) + \lambda_{sc}P_1(t). \end{cases}$$

– для внутренних фрагментов Fi:

$$\left\{ \begin{aligned} dP_{4i}(t)/dt &= -(2\lambda_{hc} + \lambda_{sc(i)})P_{4i}(t) + \mu_{hc}P_{4i+1}(t) + \\ &+ \mu_{sc}P_{4i-1}(t), \\ dP_{4i+1}(t)/dt &= -(\lambda_{hc} + \lambda_{sc(i)} + \mu_{hc})P_{4i+1}(t) + \\ &+ 2\lambda_{hc}P_{4i}(t) + 2\mu_{hc}P_{4i+2}(t), \\ dP_{4i+2}(t)/dt &= -2\mu_{hc}P_{4i+2}(t) + \lambda_{hc}P_{4i+1}(t), \\ dP_{4i+3}(t)/dt &= -\mu_{sc}P_{4i+3}(t) + \lambda_{sc(i)}P_{4i}(t) + \\ &+ \lambda_{sc(i)}P_{4i+1}(t). \end{aligned} \right. \quad (5)$$

– для последнего фрагмента Fk:

$$\left\{ \begin{aligned} \frac{dP_{4k}(t)}{dt} &= -(2\lambda_{hc} + \lambda_{sc\max})P_{4k}(t) + \mu_{hc}P_{4k+1}(t) + \\ &+ \mu_{sc}P_{4k-1}(t) + \mu_{sc}P_{4k+3}(t), \\ \frac{dP_{4k+1}(t)}{dt} &= -(\lambda_{hc} + \lambda_{sc\max} + \mu_{hc})P_{4k+1}(t) + 2\lambda_{hc} \times \\ &\times P_{4k}(t) + 2\mu_{hc}P_{4k+2}(t) + \mu_{sc}P_{4k+3}(t), \\ \frac{dP_{4k+2}(t)}{dt} &= -2\mu_{hc}P_{4k+2}(t) + \lambda_{hc}P_{4k+1}(t), \\ \frac{dP_{4k+3}(t)}{dt} &= -2\mu_{sc}P_{4k+3}(t) + \lambda_{sc\max}P_{4k}(t) + \\ &+ \lambda_{sc\max}P_{4k+1}(t). \end{aligned} \right.$$

Здесь  $i$  – номера внутренних фрагментов;  
 $k$  – номер последнего фрагмента.

Значение функции готовности определяется из выражения:

$$A(t) = \sum_{i=0}^k [P_{4i}(t) + P_{4i+1}(t)]. \quad (6)$$

Получим численное решение системы уравнений (5), которые, как и в предыдущем случае преобразуем к матричной форме (3) при  $\lambda_{hc} = 10^{-4}$ ;  $\mu_{hc} = 1,5$ ;  $\lambda_{sc} = 5 \cdot 10^{-3}$ ;  $\mu_{sc} = 6$ ;  $\lambda_{sc\max} = 10^{-2}$ ;  $\Delta\lambda_{sc} = 10^{-3}$ ; время работы системы  $t_f = 15000$  (часов).

Исходные параметры модели надежности такие же, как и в первом случае, что позволит произвести сравнение двух подсистем.

Решение уравнения (3) можно получить с помощью следующей программы MATLAB в такой последовательности.

В командном окне введем значения входных параметров модели и определим количество ее фрагментов:

```
>>clear all % очистка памяти
lambda_sc= 5*10^-3; %интенсивность отказов ПС
delta_lambda_sc=1*10^-3; %шаг изменения
lambda_sc_max= 1*10^-2; %макс. инт. отказов ПС
lambda_hc=10^-4; %интенсивность отказов АС
mu_hc=1.5; %интенсивность восстановления АС
mu_sc= 6; %интенсивность восстановления ПС
tf=15000; % время работы системы (часов)
%Количество фрагментов модели:
N_fr=round((lambda_sc_max-...
lambda_sc)/delta_lambda_sc)
N_fr =
    5
```

Здесь используется целочисленное округление (round) для того, чтобы при заполнении матрицы коэффициентов в последнем фрагменте интенсивность отказов ПС равнялась заданному значению lambda\_sc\_max.

>> %Определяем матрицу A из уравнений (5):

```
global A; %глобальная переменная
A=[];
for i=1:N_fr-1
A(4*i-3,4*i-3)=-2*lambda_hc-
(lambda_sc+delta_lambda_sc*(i-1));
A(4*i-3, 4*i-2)=mu_hc;
A(4*i-2, 4*i-3)=2*lambda_hc;
A(4*i-2,4*i-2)=-
1*(lambda_hc+(lambda_sc+delta_lambda_sc*(i-
1))+mu_hc);
A(4*i-2, 4*i-1)=2*mu_hc;
A(4*i-1, 4*i-2)= lambda_hc;
A(4*i-1, 4*i-1)= -2*mu_hc;
A(4*i, 4*i-3)= (lambda_sc+delta_lambda_sc*(i-1));
A(4*i, 4*i-2)= (lambda_sc+delta_lambda_sc*(i-1));
A(4*i, 4*i)= -1*mu_sc;
A(4*i+1, 4*i)= mu_sc;
end;
i=N_fr;
A(4*i-3, 4*i-3)=-2*lambda_hc-lambda_sc_max;
A(4*i-3, 4*i-2)=mu_hc;
A(4*i-3, 4*i)=mu_sc;
A(4*i-2, 4*i-3)=2*lambda_hc;
A(4*i-2,4*i-2)=-1*(lambda_hc+lambda_sc_max +mu_hc);
A(4*i-2, 4*i-1)=2*mu_hc;
A(4*i-2, 4*i)=mu_sc;
A(4*i-1, 4*i-2)= lambda_hc;
A(4*i-1, 4*i-1)= -2*mu_hc;
A(4*i, 4*i-3)= lambda_sc_max;
A(4*i, 4*i-2)= lambda_sc_max;
A(4*i, 4*i)= -2*mu_sc;
```

Полученная матрица A размера 20×20 сильно разрежена, заполнение которой можно увидеть, выполнив команду spy (рис. 7):

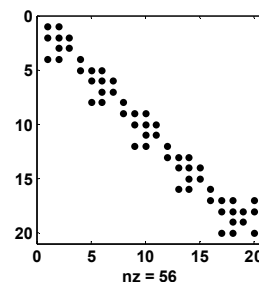


Рис. 7. Структура матрицы A

```
>> figure(3)
spy(A) %вид матрицы A (заполнение)
```

Из рис. 7 видно, что из 20×20=400 элементов матрица A содержит всего 56 ненулевых элементов. Далее найдем собственные значения матрицы A, продолжив программу:

```

>> %Собственные значения матрицы A
lambda=vpa(eig(A),9) %собственные значения
lambda =
    -12.0100001
    -6.00000236
    0.00000235985681*i-6.0
   -0.00000235985681*i-6.0
    -5.99999764
    -0.00799999996
    -0.00699999997
    -0.00599999997
    -0.00499999998
  (-2.59239423)*10^(-16)
    -1.50509933
    -1.5060992
    -1.50709906
    -1.50809893
    -1.51009904
    -3.00020067
    -3.0002008
    -3.00020107
    -3.00020094
    -3.0002009

```

Как видим, все собственные значения имеют отрицательные действительные части, следовательно, решение уравнений устойчиво.

Для правильного выбора метода решения дифференциальных уравнений определим жесткость системы по формуле (4):

```

>>%Коэффициент жесткости S
format short e
S=vpa(min(real(eig(A)))/max(real(eig(A))),2)
S =
  4.6*10^16

```

Система уравнений сверхжесткая. Поэтому для ее решения, как и в первом случае, используем решатель `ode15s`, а также ранее созданный `m`-файл (рис. 3). Подчеркнем, что `m`-файл и программа решения должны находиться в одном текущем каталоге MATLAB.

Продолжая программу, введем в окне «Command Window» и выполним команды:

```

>> options = odeset('RelTol',1e-4, 'AbsTol',1e-9);
P0=[1 zeros(1,4*N_fr-1)]; %вектор начал. условий
[t,P] = ode15s(@stiff,[0 tf],P0,options);%решение

```

Наконец, вычислим и построим функцию готовности с помощью программы:

```

>>Pg=[1:size(t,1)];
Pg(:,1)=0;
for i=1:N_fr
Pg(:,1)= Pg(:,1)+P(:,4*i-3)+ P(:,4*i-2);
end;
figure(1)
plot(t,Pg,'-')
xlabel('t'), ylabel('P_g')
grid on

```

Полученный график функции готовности показан на рис. 8 (сплошная кривая).

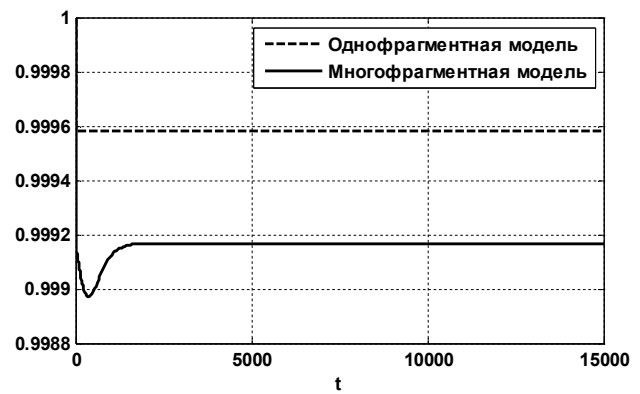


Рис. 8. Функции готовности

Точное конечное значение функции готовности, как и в первом случае, можно найти, выбрав последний элемент вектора  $P_g$ , выполнив команду:

```

>>Pg2=Pg(size(Pg)) %где size – размер вектора Pg
Pg2 =
  9.9917e-001
  1.0000e+000

```

То есть  $P_{g2}(t_f) = 0,9917$ .

### 3. Анализ результатов

Сравнение результатов оценивания ПТК в этих двух случаях позволяет сделать такие выводы:

- при увеличении интенсивности отказов имеет место ожидаемое снижение готовности; уровень снижения относительно невелик (в четвертом знаке после запятой при постепенном увеличении отказов ПС на порядок), а скорость перехода в стационарный режим велика вследствие высокой безотказности компонентов и их оперативного восстановления;
- при увеличении интенсивности получаем в отличие от исходного случая падение готовности (до 0,999 в течение тысячи часов);
- выход в стационарный режим объясняется ограниченным ростом интенсивности отказов ПС и локальной эргодичностью в последнем фрагменте.

### Выводы

Рассмотренная методика может быть использована для оценивания надежности и безопасности ПТК для разных случаев: постоянной и изменяющейся – возрастающей (вследствие обновления ПС и атак на уязвимости) или убывающей (при устранении дефектов сразу после проявления, после проведения обновлений, с учетом наложения патчей на уязвимости) интенсивности отказов.

Подробную информацию о решении задач можно найти в [4, 5].

Практический интерес представляют разработанные MATLAB-программы, которые можно использовать в инженерной практике.

Дальнейшее развитие методики может быть связано с встраиванием программ вычисления ин-

тенсивностей отказов и восстановлений ПС с использованием разных моделей и исходных данных, получаемых при тестировании и применении, а также из открытых баз данных, где хранится информация об уязвимостях программных компонент.

### Список литературы

1. *Безопасность критических инфраструктур: математические и инженерные методы оценки и обеспечения / под ред. В.С. Харченко. – Х.: Нац. аэрокосм. ун-т «ХАИ», 2011. – 603 с.*
2. *Vaidyanathan K. Extended Classification of Software Faults Based on Aging / K. Vaidyanathan, K.S. Trivedi // Proc. IEEE Software Reliability Engineering. – 2001. – Vol.6, №8. – P. 41-42.*
3. *Менаске Д.А. Производительность Web-служб. Анализ, оценка и планирование / Д.А. Менаске, А.Ф. Виргилио. – СПб.: ООО «ДиаСофтЮП», 2003. – 480 с.*
4. *Применение компьютерных технологий для оценивания надежности и безопасности программно-технических комплексов / под ред. Ю.Н. Соколова, В.С. Харченко – Х.: НАУ «ХАИ», 2013. – 462 с.*
5. *Персональный сайт Соколова Ю.Н. [Электронный ресурс]. – Режим доступа к ресурсу: <http://www.sokolov.5u.com>.*

Поступила в редколлегию 28.01.2014

**Рецензент:** д-р техн. наук, проф. В.А. Краснобаев, Полтавский национальный технический университет им. Ю. Кондратюка, Полтава.

### ІНСТРУМЕНТОВАНЕ ОЦІНЮВАННЯ НАДІЙНОСТІ ПРОГРАМНО-ТЕХНІЧНИХ КОМПЛЕКСІВ ПРИ ЗРОСТАННІ ІНТЕНСИВНОСТІ ВІДМОВ

Ю.М. Соколов, В.С. Харченко, Ю.Л. Поночовний

*Розглядаються комп'ютерні методи оцінювання надійності програмно-технічних комплексів (ПТК) на основі структурно-логічних схем, розроблені в рамках проекту TEMPUS - SAFEGUARD National Software Engineering Network of Centres of Innovative Academia – Industry Handshaking (158886 - TEMPUS - 1 -2009- 1 - UK- TEMPUS - JPCR). Викладаються методики оцінки надійності та готовності ПТК з використанням багатотрагментних марковських моделей з урахуванням зміни параметрів потоків відмов, викликаних проектними дефектами. Досліджено випадок зростання інтенсивності відмов внаслідок внесення нових дефектів. Наведено комп'ютерні програми вирішення задач оцінки надійності та готовності ПТК. Як інструментарій використовується сучасний програмний комплекс MATLAB.*

**Ключові слова:** надійність, готовність, програмно-технічний комплекс, марковські моделі, інструментоване оцінювання, MATLAB.

### TOOL-BASED ESTIMATION OF SOFTWARE-HARDWARE COMPLEXES AVAILABILITY IN CASE OF FAILURE RATE GROWTH

Y.N. Sokolov, V.S. Kharchenko, Y.L. Ponochozny

*The methods of estimating the software and hardware complexes (SHC) availability based on the structural-logic, which were developed during the project TEMPUS-SAFEGUARD (158886 - TEMPUS- 1-2009 -1 -UK-TEMPUS-JPCR) are discussed. The technique for SHC availability and safety estimation the using a multi-fragmental Markov's models taking into consideration the changes in failure rate caused by design defects and vulnerabilities is described. Model of SHC availability in conditions of the failure rate increasing is researched. MATLAB-programs for SHC availability and safety estimation are developed.*

**Keywords:** availability, software and hardware failures, Markov's models, tool-based estimation, MATHLAB.