

УДК 004.94:519.217

Применение программной среды Matlab для имитационного моделирования сложных систем военного назначения

Горин А. Н., Будников С. А.

Постановка задачи: математические модели функционирования сложных систем военного назначения, построенных по блочно-модульному принципу, определяют последовательность смены дискретных состояний, характеризующих частичные или полные потери функциональных возможностей систем. Адекватной формой описания таких систем являются полумарковские модели, а в качестве показателей эффективности обычно используют значения вероятностей пребывания системы в каждом из заданных состояний в стационарном режиме. Определение стационарных вероятностей состояний для полумарковских процессов представляет собой достаточно сложную вычислительную задачу, сводящуюся к многоэтапным расчетам и неудобную для программирования. **Цель работы:** упрощение процедуры получения оценок для стационарных вероятностей состояний систем, описываемых полумарковскими моделями. **Используемые методы:** применение методов имитационного моделирования полумарковских систем, формирование в моделях средствами Matlab потоков событий с разными законами распределения, инициирующих переходы из текущих состояний в другие состояния и определяющих времена пребывания в состояниях, получение эмпирических оценок вероятностей состояний. **Новизна:** использование структурных элементов полумарковских моделей в имитационных моделях, применение рекуррентного усреднения для получения «точечных» оценок вероятностей состояний. **Результат:** имитационные модели на языке Matlab, легко настраиваемые для исследования систем с произвольным числом дискретных состояний и разными случайными процессами, протекающими в системах. **Практическая значимость:** разработанные имитационные модели позволяют достаточно просто получить оценки вероятностей состояний, задавая структурные параметры системы, типы и параметры распределений для потоков событий, имеющих место в реальных системах. Возможные пути использования оценок вероятностей – помощь в принятии решений о соответствии систем предъявляемым к ним требованиям, проверка корректности аналитических моделей.

Ключевые слова: имитационное моделирование, полумарковские модели, стационарные вероятности состояний.

Введение

Анализ структур систем и комплексов радиосвязи военного назначения, командных пунктов и пунктов управления войсками и оружием, сложных систем вооружения показывает, что основой их построения является блочно-модульный принцип. Функционирование таких систем в ходе боевых действий или учебы можно представить, как последовательность смены дискретных состояний, характеризующих частичные или полные потери функциональных возможностей. Восстановление функциональности систем происходит путем

Библиографическая ссылка на статью:

Горин А. Н., Будников С. А. Применение программной среды Matlab для имитационного моделирования сложных систем военного назначения // Системы управления, связи и безопасности. 2019. № 1. С. 221-251. DOI: 10.24411/2410-9916-2019-10114

Reference for citation:

Gorin A. N., Budnikov S. A. Use of Matlab Software Environment for Simulation Modeling of Complex Military Systems. *Systems of Control, Communication and Security*, 2019, no. 1, pp. 221-251. DOI: 10.24411/2410-9916-2019-10114 (in Russian).

замены неисправных блоков исправными. Эффективность выполнения основных функций такими системами может быть оценена только с помощью вероятностных методов. В качестве показателей эффективности обычно используют значения вероятностей пребывания системы в каждом из заданных состояний в стационарном режиме.

При разработке и исследовании сложных систем с дискретными состояниями достаточно широко используются аналитические полумарковские модели их функционирования, например [1, 2], при этом одним из результатов моделирования являются оценки вероятностей пребывания системы в различных состояниях (вероятностей состояний) в установившемся режиме.

Однако, определение стационарных вероятностей состояний для полумарковских процессов представляет собой достаточно сложную вычислительную задачу [3-6], сводящуюся к многоэтапным расчетам. Кроме того, сами результаты могут содержать погрешности как из-за сложностей моделей, так и применения численных методов для получения оценок вероятностей.

Подтвердить результаты, полученные с использованием аналитических моделей, можно с помощью имитационных моделей [7], позволяющих учесть все особенности функционирования систем. Имитационное моделирование может быть использовано и как основной инструмент при исследовании сложных систем [8, 9].

Следует отметить, что при имитационном моделировании широко используется вычислительная система Matlab [2, 7].

В то же время существует предубеждение в применении имитационных моделей для практических расчетов, вызванное двумя факторами:

- большим количеством вычислений при изменении параметров модели;
- неопределенностью результатов моделирования из-за стохастического характера задачи.

В настоящее время большой объем вычислений при определении эмпирических вероятностей не является лимитирующим фактором. И как следствие, результаты многократных выполнений моделей можно использовать для уточнения оценок вероятностей, что делает имитационные модели сравнимыми по сходимости результатов с аналитическими моделями.

Постановка задачи

Учитывая сказанное, в статье ставится задача разработки имитационных моделей сложных систем, формализующих реальные комплексы военного назначения с дискретными состояниями для разных видов случайных процессов, протекающих в этих системах. Результатом работы моделей должны быть оценки стационарных вероятностей состояний системы, при этом имитационные модели должны иметь преимущества в простоте использования по сравнению с аналитическими моделями.

При разработке моделей должна быть реализована возможность уменьшения разброса вероятностей состояний, обусловленного стохастическим ха-

рактором имитационной модели, до практически приемлемого уровня путем использования современных методов обработки измерений.

На этапе разработки моделей должна быть подтверждена их правильность путем сравнения результатов с результатами, полученными аналитическими методами.

Допущения при построении имитационных моделей

Имитационные модели должны строиться при следующих допущениях достаточно общего характера:

1) системы являются стохастическими, т. е. изменение дискретных состояний систем происходит случайным образом;

2) статистические свойства процессов, изменяющих состояния и определяющих время пребывания системы в состояниях, не зависят от числа уже сделанных переходов (системы являются однородными) и от способа попадания системы в каждое из возможных состояний;

3) изменение состояний системы происходит мгновенно, т. е. в моменты смены состояний системы являются марковскими.

Имитационное моделирование дискретных марковских цепей

Для дискретных марковских цепей разработаны аналитические модели, позволяющие получить оценки вероятностей состояний как в динамике, так и в установившемся (стационарном) режиме [10, 11]. Наличие достаточно простых аналитических моделей позволяет проверить корректность результатов, полученных с использованием имитационных моделей.

В дальнейшем дискретные марковские цепи будут использоваться как инструмент смены состояний в имитационных моделях марковских и полумарковских процессов.

Рассмотрим однородную дискретную марковскую цепь, описывающую систему с n дискретными состояниями и дискретным временем, в которой смена состояний происходит достаточно быстро через одинаковые промежутки времени. Требование смены состояний через одинаковые промежутки времени может быть заменено менее жестким требованием контроля состояний. Примером может служить таксопарк, в котором в конце рабочего дня анализируется состояние машин.

При аналитическом моделировании таких систем предполагается известной матрица переходных вероятностей Q размером $n \times n$ [3, 4, 10, 11]

$$Q = [q_{ij}], \quad i, j = 1, 2, \dots, n,$$

где n – число состояний системы; q_{ij} – вероятность перехода системы из состояния S_i в состояние S_j или, иначе, из состояния i в состояние j , не зависящая от времени.

Переходные вероятности должны удовлетворять условию нормировки

$$\sum_{j=1}^n q_{ij} = 1, \quad i = 1, 2, \dots, n.$$

Для определения вероятности $P_j(k)$ пребывания системы в каждом j -ом состоянии на k -ом шаге (через k интервалов времени, определяющих моменты изменения состояний) используется рекуррентная формула

$$P_j(k) = \sum_{i=1}^n P_i(k-1)q_{ij}, \quad k=1,2,\dots \quad (1)$$

где начальные условия $P_j(0)$, $j = 1, 2, \dots, n$ считаются известными.

Для стационарного режима $P_j(k) = P_j(k-1)$, $j = 1, 2, \dots, n$. Тогда уравнение (1) можно переписать в виде

$$P_j = \sum_{i=1}^n P_i q_{ij}, \quad j=1,2,\dots, n$$

или

$$\sum_{\substack{i=1 \\ i \neq j}}^n P_i q_{ij} + P_j (q_{jj} - 1) = 0, \quad j=1,2,\dots, n \quad (2)$$

Система уравнений (2) с нулевыми свободными членами имеет бесконечное множество решений. Для получения единственного решения одно из уравнений, например последнее, заменяется условием нормировки

$$\sum_{i=1}^n P_i = 1.$$

Тогда можно записать

$$\begin{cases} \sum_{\substack{i=1 \\ i \neq j}}^n P_i q_{ij} + P_j (q_{jj} - 1) = 0, \quad j=1,2,\dots, n-1, \\ \sum_{i=1}^n P_i = 1. \end{cases} \quad (3)$$

Искомые вероятности P_i , $i = 1, 2, \dots, n$ в стационарном режиме являются решением системы линейных алгебраических уравнений (3).

В Matlab это решение может быть получено с помощью оператора

$$psost=M^{(-1)} * transpose (B) ;$$

где $psost$ – вектор вероятностей состояний; M , B – матрица коэффициентов и вектор-строка свободных членов системы уравнений (3); $transpose$ – функция для транспонирования вектора (матрицы).

Матрица переходных вероятностей задается в программе Matlab в виде двумерного массива q .

Массив q , например для числа состояний $n = 4$, имеет вид

$$q=[0.35 \ 0.3 \ 0.1 \ 0.25; \ 0.4 \ 0.1 \ 0.2 \ 0.3; \dots \\ 0.5 \ 0.2 \ 0.1 \ 0.2; \ 0.35 \ 0.2 \ 0.2 \ 0.25];$$

Далее разработаем имитационную модель для дискретной марковской цепи.

Имитационная модель, по сути, представляет собой самостоятельный объект (систему), на котором проводятся статистические испытания. Каждое испытание представляет собой одну *реализацию* модели при некоторых значениях параметров. В результате испытания система может перейти в другое состояние или остаться в прежнем.

Проведя достаточно большое число испытаний N и подсчитав суммарное число m_i переходов системы в каждое состояние S_i , $i = 1, 2, \dots, n$, искомые вероятности можно определить по формуле

$$P_i = \frac{m_i}{N}. \quad (4)$$

Основой имитационной модели является модель случайных событий, включающая в себя генерирование равномерно распределенных в интервале $[0; 1]$ случайных чисел и проверку их попадания в интервалы, на которые делится интервал $[0; 1]$. В Matlab для генерирования таких чисел используется функция `rand`.

Алгоритм имитационной модели дискретной марковской цепи включает в себя следующие операции:

- определение номера k исходного состояния по известным начальным вероятностям состояний $P_i(0)$, $i = 1, 2, \dots, n$ и увеличение соответствующего счетчика числа переходов на единицу;
- организацию большого числа N испытаний модели, при этом для каждого испытания определяется, в какое состояние должна перейти система, и увеличивается значение счетчика числа переходов в соответствующее состояние.

После завершения N испытаний вычисляются оценки стационарных вероятностей в соответствии с формулой (4).

Номер начального состояния k определяется путем «розыгрыша» следующим кодом

```
s=p(1); k=1;  
gg=rand; %Получение случайного числа, равномерно  
%распределенного в [0;1]  
while (gg>s), k=k+1; s=s+p(k);
```

где p – одномерный массив начальных вероятностей состояний.

Пример массива p для числа состояний $n = 4$: $p=[0.3 \ 0.2 \ 0.4 \ 0.1]$;

Для моделирования перехода из состояния S_i в некоторое другое состояние интервал $[0; 1]$ разбивается на n отрезков, определяемых значениями переходных вероятностей i -й строки матрицы Q . Правые граничные значения этих отрезков L_{ik} определяются согласно выражению

$$L_{ik} = \sum_{j=1}^k q_{ij}, \quad k = 1, 2, \dots, n. \quad (5)$$

В модели производится обращение к датчику равномерно распределенных в интервале $[0; 1]$ случайных чисел для получения случайного значения x , которое проверяется на принадлежность одному из отрезков, на которые разбит интервал $[0; 1]$. Если значение x попадает в некоторый m -й отрезок, т. е. удовлетворяется условие

$$L_{i,m-1} < x \leq L_{im},$$

то считается, что система переходит из состояния S_i в состояние S_m . Факт перехода в состояние S_m фиксируется в модели увеличением на единицу значения соответствующего счетчика.

Результаты вычислений по формуле (5) объединяются в матрице вероятностных интервалов [11], элементы которой вычисляются по формуле

$$mvi_{ij} = \sum_{k=1}^j q_{ik}, \quad i, j = 1, 2, \dots, n,$$

где q_{ik} – вероятность перехода из состояния S_i в состояние S_k .

Рассмотренные операции реализуются в Matlab следующим кодом:

```
%Формирование матрицы вероятностных интервалов
for i=1:n, sper=0;
    for j=1:n,
        for k=1:j, sper=sper+q(i, k); end,
        mvi(i, j)=sper; sper=0;
    end,
end
%Моделирование процесса смен состояний
ktek=k; %ktek - текущее состояние
for i=1:N,
    count(ktek)=count(ktek)+1;
    sl=rand; k=1;
    while(sl>mvi(ktek, k)), k=k+1; end, ktek=k;
end
```

Ниже приведены исходные данные и значения стационарных вероятностей, полученные по аналитической (psostteor) и имитационной (psostmod) моделям.

```
%Вероятности состояний в начальный момент времени
p=[0.3 0.2 0.4 0.1];
%Матрица переходных вероятностей
q=[0.35 0.3 0.1 0.25; 0.4 0.1 0.2 0.3; ...
    0.5 0.2 0.1 0.2; 0.35 0.2 0.2 0.25];
N =10000
psostteor=[0.3829 0.2166 0.1470 0.2535];
psostmod=[0.3826 0.2184 0.1476 0.2514].
```

Близость значений стационарных вероятностей (с точностью до двух значащих цифр после запятой) для обоих видов моделей подтверждает корректность подхода при построении имитационной модели.

В качестве примера системы, описываемой дискретной марковской цепью, рассмотрим группировку из трех специальных радиоэлектронных средств (РЭС), подвергающуюся последовательным огневым ударам [11]. Состояния, в которых может находиться группировка РЭС, следующие: S_1 – все РЭС находятся в боевой готовности, S_2 – два РЭС находятся в боевой готовности, S_3 – одно РЭС находится в боевой готовности, S_4 – все РЭС выведены из строя в результате огневого поражения.

Пусть матрица переходных вероятностей имеет вид

$$Q = \begin{pmatrix} 0,4 & 0,25 & 0,2 & 0,15 \\ 0 & 0,35 & 0,3 & 0,35 \\ 0 & 0 & 0,45 & 0,55 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

т. е. система является поглощающей (невосстанавливаемой).

Оценим вероятности состояний в стационарном режиме. Результаты, полученные с помощью аналитической модели

$$psostteor = [0.0000 \ 0.0000 \ 0.0000 \ 1.0000].$$

Результаты имитационного моделирования при $N = 10000$ испытаниях модели следующие:

$$psostmod = [0.0001 \ 0 \ 0 \ 0.9999].$$

Как и ожидалось, результаты оказались практически идентичными.

Имитационное моделирование непрерывных марковских цепей

Построим имитационную модель однородной непрерывной марковской цепи. В этой цепи смена состояний происходит мгновенно в *случайные* моменты времени, формируемые экспоненциальной плотностью вероятности.

Рассмотрим сначала аналитическую модель непрерывной марковской цепи, представляющую собой систему дифференциальных уравнений Колмогорова-Чепмена вида [10, 11]

$$\frac{dP_j(t)}{dt} = -P_j(t) \sum_{k=1}^n \lambda_{jk} + \sum_{\substack{i=1 \\ i \neq j}}^n P_i(t) \lambda_{ij}, \quad j = 1, 2, \dots, n, \quad (6)$$

где: $P_j(t)$ – вероятность пребывания системы в j -ом состоянии; λ_{ij} – известная интенсивность потока событий, переводящего систему из состояния i в состояние j .

Кроме того, считаются известными вероятности состояний в начальный момент времени $P_1(0), P_2(0), \dots, P_n(0)$, при этом сумма вероятностей всех состояний должна равняться единице.

Вероятности состояний, определяющие поведение моделируемой системы во времени, находятся путем интегрирования системы дифференциальных уравнений (6) при заданных начальных условиях.

Для однородной непрерывной марковской цепи с конечным числом состояний стационарный режим существует и не зависит от начального состояния системы, если из каждого состояния можно за конечное число шагов перейти в любое другое состояние.

Модель однородной непрерывной марковской цепи в установившемся режиме представляет собой систему линейных алгебраических уравнений

$$-P_j \sum_{k=1}^n \lambda_{jk} + \sum_{\substack{i=1 \\ i \neq j}}^n \lambda_{ij} P_i = 0, \quad j = 1, 2, \dots, n, \quad (7)$$

которая получается из системы уравнений (6) приравниванием производных нулю и исключением зависимости вероятностей от времени.

Эта система уравнений имеет бесконечное множество решений, поскольку вектор свободных членов имеет нулевые значения. Для получения единственного решения одно из уравнений системы (7) заменяется условием нормировки

$$P_1 + P_2 + \dots + P_n = 1.$$

В непрерывной марковской цепи случайные моменты смены состояний определяются потоками событий с экспоненциальными плотностями вероятностей.

При имитационном моделировании «розыгрыш» смены состояний может быть реализован в Matlab путем генерирования случайных чисел, подчиняющихся такому распределению.

Процедура выбора состояния, в которое должна перейти система из текущего состояния, следующая:

- генерируются случайные числа в соответствии с плотностями вероятностей для всех потоков событий, переводящих систему из текущего состояния в другие возможные состояния;
- выбирается минимальное случайное число, соответствующее наступлению ближайшего события, и выполняется переход в соответствующее состояние;
- время до смены текущего состояния суммируется в соответствующей переменной.

Номер начального состояния $k_{тек}$ определяется тем же способом, что и в имитационной модели дискретной марковской цепи.

Формирование переходов и времени пребывания системы в конкретных состояниях осуществляется следующим кодом:

```
for i=1:N, %Цикл по количеству переходов
    for m=1:n,
        sl(m)=0;
        if(ff(ktek,m)~=0),
            sl(m)=random('exp',ff(ktek,m));
        end
    end
    %Определение наиболее раннего события
    %для формирования перехода
    mint=nval; % nval=1e6 - нач. значение для поиска
               %минимума
    for m=1:n,
        if((m ~= ktek) && (sl(m)< mint) && (sl(m)~=0)),
            mint=sl(m);knew=m; %Переход в состояние knew
        end
    end
    Tsost(ktek)=Tsost(ktek)+mint;
    ktek=knew;
end
```

Суммарное время пребывания системы в i -ом состоянии содержится в соответствующем элементе массива T_{sost} .

Оценки вероятностей состояний P_i , $i = 1, 2, \dots, n$ в установившемся режиме вычисляются как отношения времени пребывания системы в конкретных состояниях ко времени пребывания системы во всех состояниях при достаточно большом количестве испытаний модели кодом

```
%Определение времени пребывания системы во всех состояниях
sumps=0;
for j=1:n, sumps=sumps+Tsost(j); end
```

```
%Определение вероятностей пребывания системы в каждом
%из состояний
for j=1:n, psost(j)=Tsost(j)/sumps; end
```

Генерирование случайных чисел с различными законами распределения в Matlab осуществляется с помощью функции `random`. Обращение к этой функции имеет вид

```
x = random('name', value1 [, value2, ...]),
```

где x – случайное число; `name` – название закона распределения, принятое в Matlab (приложение 1); `value1`, `value2` – значения параметров плотности вероятности.

В соответствии с соглашениями Matlab в матрицу `ff` записываются *средние* значения времени между сменами состояний.

В качестве примера системы, описываемой непрерывной марковской цепью, рассмотрим специальное радиоэлектронное средство (РЭС), поведение которого задано графом, представленным на рис. 1 [11].

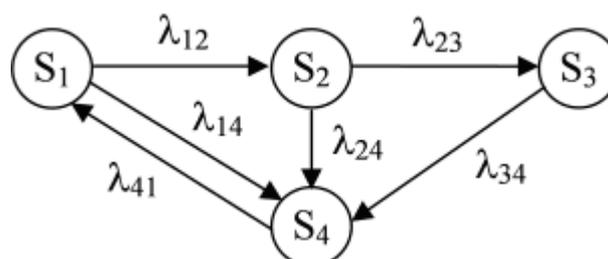


Рис. 1. Граф состояний и переходов РЭС

На рис. 1 приняты следующие обозначения: S_1 – РЭС находится в боевой готовности; S_2 – РЭС ведет разведку целей; S_3 – РЭС ведет подавление целей; S_4 – РЭС находится в неисправном состоянии; λ_{ij} – интенсивность перехода из состояния S_i в состояние S_j .

Поведение такой системы описывается моделью в виде непрерывной марковской цепи.

Оценим вероятности состояний в стационарном режиме при средних значениях времени смены состояний, приведенных в матрице (здесь и далее форма записи для матриц и векторов соответствует принятой в Matlab)

$$ff = [0 \ 4 \ 0 \ 3.9; \ 0 \ 0 \ 1.5 \ 2.5; \ 0 \ 0 \ 0 \ 2.5; \ 1.4 \ 0 \ 0 \ 0]. \quad (8)$$

Интенсивности переходов определяются по формуле $\lambda_{ij} = 1/ff_{ij}$. Нулевые значения в матрице `ff` означают, что соответствующий переход отсутствует.

Результаты, полученные с помощью аналитической (`psostteor_con`) и имитационной (`psostimit_con`) моделей, следующие:

$$\begin{aligned} psostteor_con &= [0.4285 \ 0.1004 \ 0.1674 \ 0.3038] \\ N=5000 \ psostimit_con &= [0.4251 \ 0.1028 \ 0.1667 \ 0.3055] \end{aligned}$$

Близость значений стационарных вероятностей для обоих видов моделей подтверждает корректность подхода, принятого при построении имитационной модели.

Визуализация результатов моделирования и повышение точности оценок

В Matlab имеются различные средства визуализации результатов расчетов. Исследователь может использовать возможности визуализации для принятия решения о необходимом числе испытаний имитационной модели, необходимости применения каких-либо методов для обработки результатов моделирования.

В имитационной модели можно достаточно легко дополнительно организовать цикл по количеству испытаний (переходов) N и сохранить результаты в двумерном массиве xx .

```
%NN - количество сеансов моделирования
NN=50
for ii=1:NN,
    N=ii*1000;
    for i=1:N,
        ...
        for j=1:n, xx(ii,j)=x(j); end
    end
end
...
end
```

В приведенном коде массив x содержит значения стационарных вероятностей для текущего числа испытаний модели N .

График зависимостей вероятностей от числа испытаний модели, например для четырех состояний, может быть сформирован следующим кодом:

```
ii=1:1:NN;
plot(ii*1000, xx(ii,1), 'r', ii*1000, xx(ii,2), 'g', ...
     ii*1000, xx(ii,3), 'b', ii*1000, xx(ii,4), 'k')
xlabel('Количество реализаций модели')
ylabel('Вероятности состояний')
```

Зависимости вероятностей состояний от количества переходов (шаг равен 1000) для непрерывной марковской цепи с параметрами, указанными в матрице (8), приведены на рис. 2.

В имитационной модели эмпирические значения вероятностей определяются по формуле вида (4). Естественно, оценки вероятностей могут считаться относительно достоверными только при больших значениях N . Но даже в этом случае будут наблюдаться флуктуации в значениях вероятностей при повторных сеансах моделирования из-за стохастического характера модели (рис. 2).

Сглаживание «зашумленных» измерений может выполняться различными методами, например, усреднением оценок вероятностей, полученных для множества сеансов моделирования при одинаковых количествах переходов между состояниями.

В рассматриваемом случае модель системы представляет собой самостоятельный объект, на котором проводятся статистические эксперименты. Результаты этих экспериментов, полученные на модели при разных значениях переходов N , могут быть использованы для уточнения оценок вероятностей с помощью рекуррентного определения математического ожидания для вероятности j -го состояния.

$$m_j(k) = m_j(k-1) + \frac{1}{k}[x_j(k) - m_j(k-1)], \quad k = 1, 2, \dots, NN, \quad (9)$$

где: NN – количество сеансов моделирования; m_j – среднее значение параметра x_j ; $x_j(k)$ – значение параметра x_j при k -ом сеансе моделирования.

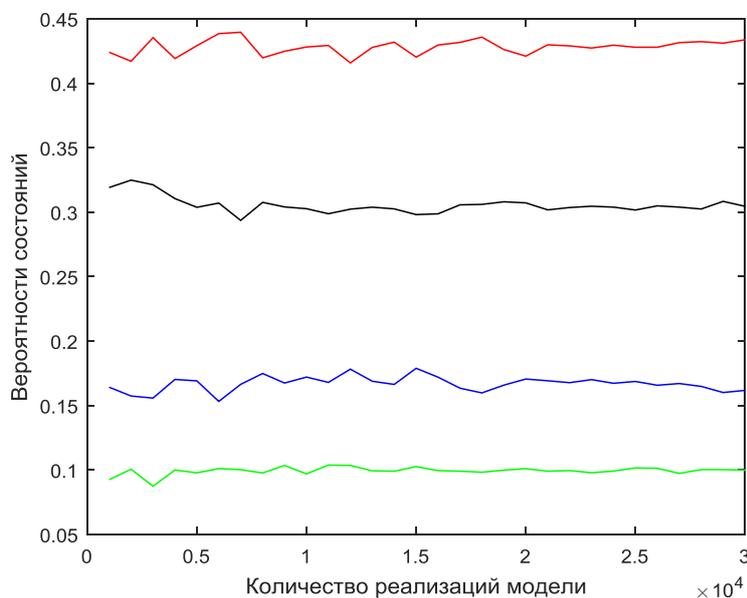


Рис. 2. Зависимости вероятностей состояний от количества переходов

Зависимости вероятностей состояний от количества переходов (шаг равен 1000) для непрерывной марковской цепи при сглаживании вероятностей с помощью рекуррентного усреднения (9) приведены на рис. 3. Исходные данные те же, что и для результатов, представленных на рис. 2.

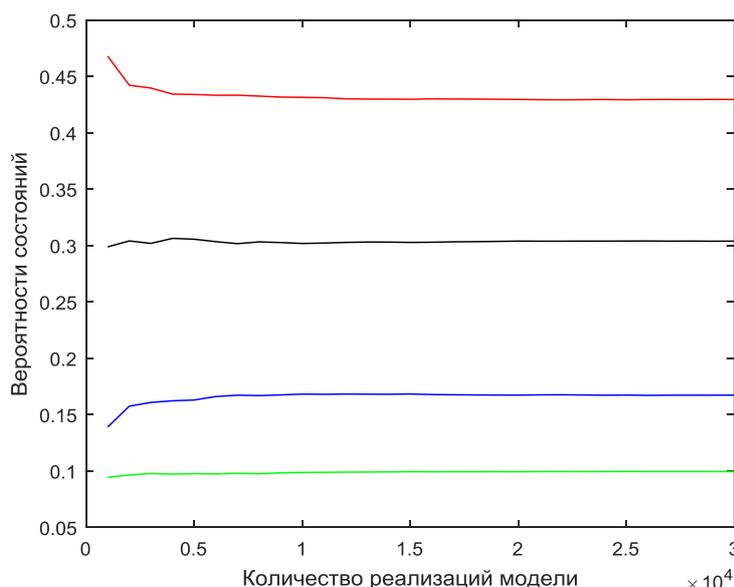


Рис. 3. Зависимости вероятностей состояний от количества переходов при рекуррентном усреднении

Определение характеристик системы на модели

Из рис. 3 следует, что оценки вероятностей, по сути, являются «точечными», следовательно, имитационные модели можно использовать и для исследования характеристик системы при изменениях ее параметров.

На практике каждое состояние обычно характеризуется некоторым набором параметров. Эти параметры тем или иным образом влияют на параметры плотности вероятности, определяющей либо процесс смены состояний, либо значение времени пребывания системы в состоянии.

В качестве примера исследуем влияние среднего времени для перехода из состояния 4 в состояние 1 (значения элемента $ff(4, 1)$ в матрице (8)) на стационарные вероятности системы.

Проведя серию расчетов, построим зависимости вероятностей состояний от варьируемого параметра с помощью следующего кода:

```
i=1:1:8;
plot(p(1,i),p(2,i),'r',p(1,i),p(3,i),'k',...
     p(1,i),p(4,i),'b',p(1,i),p(5,i),'g')
    %p(1,i) содержит значения ff(4,1)
xlabel('Параметр ff(4,1)')
ylabel('Вероятность состояний')
```

Полученные зависимости представлены на рис. 4.

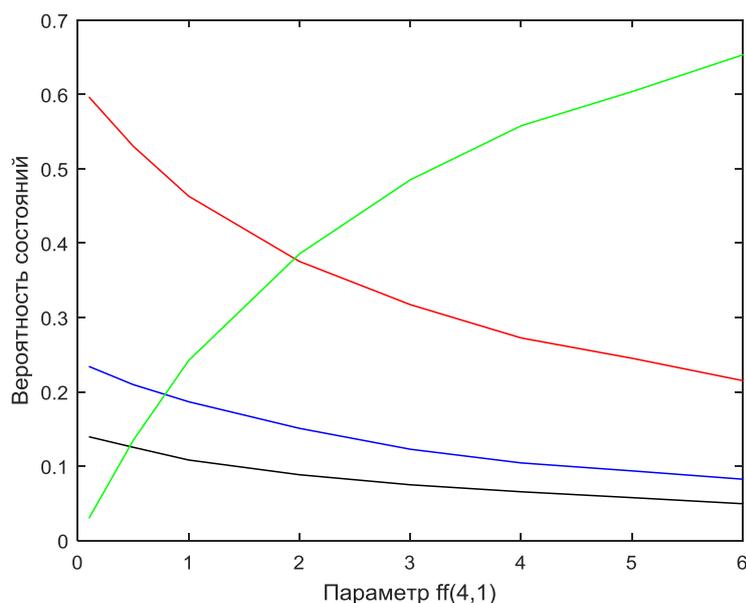


Рис. 4. Зависимости вероятностей состояний от среднего времени перехода из состояния 4 в состояние 1 для непрерывной марковской цепи

Имитационное моделирование полумарковских процессов

Рассмотрим построение имитационной модели процесса, в котором изменение состояний, как и в непрерывной марковской цепи, происходит мгновенно в случайные моменты времени, но для них плотности вероятностей отличаются от экспоненциальных. В этом случае говорят о полумарковском процессе.

При имитационном моделировании «розыгрыш» смены состояний может быть реализован в Matlab путем генерирования случайных чисел, подчиняющихся соответствующему распределению.

Процедура выбора состояния, в которое должна перейти система из текущего состояния, следующая:

- генерируются случайные числа в соответствии с плотностями вероятностей для всех потоков событий, переводящих систему из текущего состояния в другие возможные состояния;
- выбирается минимальное случайное число, соответствующее наступлению ближайшего события, и выполняется переход в соответствующее состояние;
- время до смены текущего состояния суммируется в соответствующей переменной.

В качестве исходных данных для имитационной модели должны использоваться три матрицы, если ограничиться двухпараметрическими плотностями вероятностей.

Одна матрица vf должна определять законы распределений, а две других – $param1$ и $param2$ – параметры плотностей вероятностей для моментов перехода из состояний в другие состояния.

Основная часть имитационной модели полумарковского процесса для трех наиболее часто используемых распределений (экспоненциального, Рэлея и Вейбулла) имеет следующий вид:

```
for i=1:N, %Цикл по количеству переходов
    for m=1:n,
        sl(m)=0;
        if((vf(ktek,m)~=0) && (vf(ktek,m)==1))
            sl(m)=random('exp',param1(ktek,m));
        end
        if((vf(ktek,m)~=0) && (vf(ktek,m)==2))
            sl(m)=random('rayl',param1(ktek,m));
        end
        if((vf(ktek,m)~=0) && (vf(ktek,m)==3))
            sl(m)=random('wbl',param1(ktek,m),param2(ktek,m));
        end
    end
    %Определение наиболее раннего события
    mint=nval; % nval - нач. значение для поиска минимума
    for m=1:n,
        if((m ~= ktek) && (sl(m)< mint) && (sl(m)~=0)),
            mint=sl(m);knew=m; %Переход в состояние knew
        end
    end
    Tsost(ktek)=Tsost(ktek)+mint;
    ktek=knew;
end
```

Естественно, что количество видов распределений может быть легко увеличено путем добавления в код соответствующих условных операторов.

Аналитические методы определения стационарных вероятностей таких процессов достаточно сложны, поэтому использование имитационных моделей в качестве основного рабочего инструмента в этом случае является оправданным.

Для проверки корректности имитационной модели сравним результаты расчетов, полученные с использованием аналитических методов, с результатами имитационного моделирования.

Стационарные вероятности полумарковского процесса, представляемого в виде двумерного процесса, определяются по формуле [3]

$$p_j = \frac{a_j T_j}{\sum_{i=1}^n a_i T_i}, j = 1, 2, \dots, n, \quad (10)$$

где: $a_i, i = 1, 2, \dots, n$ – стационарные вероятности для вложенной непрерывной марковской цепи; $T_i, i = 1, 2, \dots, n$ – среднее время пребывания в i -ом состоянии при произвольном распределении.

Среднее время пребывания процесса в i -ом состоянии определяется по формуле [6]

$$T_i = \int_0^{\infty} \prod_{j=1}^n (1 - F_{ij}(t)) dt, \quad (11)$$

где: n – число состояний; $F_{ij}(t)$ – функция распределения для случайного времени перехода из состояния i в состояние j .

Стационарные вероятности для вложенной непрерывной марковской цепи вычисляются из системы линейных уравнений вида (3)

$$a_i = \sum_{j=1}^n a_j q_{ji}, i = 1, 2, \dots, n, \quad (12)$$

где $q_{ij}, i, j = 1, 2, \dots, n$ – переходные вероятности для стационарного режима, вычисляемые по формуле [6]

$$q_{ij} = \int_0^{\infty} \prod_{\substack{k=1 \\ k \neq i}}^n (1 - F_{ik}(t)) dF_{ij}(t), i, j = 1, 2, \dots, n. \quad (13)$$

Для подтверждения правильности имитационной модели сравним результаты расчетов по формулам (10)-(13) с результатами имитационного моделирования полумарковского процесса для системы, граф которой изображен на рис. 1.

Исходные данные для моделирования приведены в таблице 1. Обозначения параметров плотностей вероятностей соответствуют обозначениям в приложении 1.

Среднее значение времени между сменами состояний вычислено в соответствии с приложением 1.

Таблица 1 – Виды и параметры плотностей вероятностей

Переход между состояниями	Название и код распределения	Параметры плотности вероятности	Среднее значение времени между сменами состояний
1-2	Рэля – 2	$b = 5$	6,267
1-4	Рэля – 2	$b = 6$	7,52
2-3	Вейбулла – 3	$a = 2,8; b = 2$	2.481
2-4	Экспоненциальное – 1	$m = 2,5$	2,5
3-4	Экспоненциальное – 1	$m = 2,5$	2,5
4-1	Экспоненциальное – 1	$m = 1,4$	1,4

Матрицы с исходными данными для имитационной модели имеют вид
 $vf = [0 \ 2 \ 0 \ 2; \ 0 \ 0 \ 3 \ 1; \ 0 \ 0 \ 0 \ 1; \ 1 \ 0 \ 0 \ 0];$
 $param1 = [0 \ 5 \ 0 \ 6; \ 0 \ 0 \ 2.8 \ 2.5; \ 0 \ 0 \ 0 \ 2.5; \dots$
 $1.4 \ 0 \ 0 \ 0];$
 $param2 = [0 \ 0 \ 0 \ 0; \ 0 \ 0 \ 2 \ 0; \ 0 \ 0 \ 0 \ 0; \ 0 \ 0 \ 0 \ 0];$

Результаты, полученные с помощью имитационной (psostmods) модели при числе переходов $N = 50000$, следующие:

$psostmods = [0.625 \ 0.112 \ 0.0814 \ 0.1815].$

Результаты расчетов по формулам (10)-(13) приведены в таблице 2.

Таблица 2 – Результаты расчетов по аналитическим формулам

Среднее время T_i пребывания в состоянии	Переходные вероятности q_{ij}	Стационарные вероятности для вложенной непрерывной марковской цепи – a_i	Стационарные вероятности состояний p_i
4,814	0 0,59 0 0,41	0,352	0,626
1,455	0 0 0,418 0,582	0,208	0,112
2,5	0 0 0 1	0,087	0,08
1,4	1 0 0 0	0,352	0,182

Близость результатов позволяет сделать вывод о корректности имитационной модели.

Результаты, приведенные в табл. 2, получены в Mathcad. Ниже приведены фрагменты расчетов.

$$b12 := 5 \quad b14 := 6 \quad F12(t) := 1 - e^{-\frac{t^2}{2 \cdot b12^2}} \quad F14(t) := 1 - e^{-\frac{t^2}{2 \cdot b14^2}}$$

$$T_1 := \int_0^{\infty} (1 - F12(t)) \cdot (1 - F14(t)) \, dt = 4.814$$

$$\begin{aligned}
 a_{23} &:= 2.8 & b_{23} &:= 2 & m_{24} &:= 2.5 \\
 F_{23}(t) &:= 1 - e^{-\left(\frac{t}{a_{23}}\right)^{b_{23}}} & F_{24}(t) &:= 1 - e^{-\frac{t}{m_{24}}} \\
 q_{23} &:= \int_0^{\infty} (1 - F_{24}(t)) \frac{d}{dt} F_{23}(t) dt = 0.418 \\
 \underline{A} &:= \begin{pmatrix} 1 & 0 & 0 & -q_{41} \\ -q_{12} & 1 & 0 & 0 \\ 0 & -q_{23} & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} & \underline{B} &:= \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} & \underline{P}_a &:= \underline{A}^{-1} \cdot \underline{B} = \begin{pmatrix} 0.352 \\ 0.208 \\ 0.087 \\ 0.352 \end{pmatrix} \\
 n &:= 4 & \underline{s}_i &:= \sum_{i=1}^n (P_{a_i} \cdot T_i) \quad i := 1..4 & P_{v_i} &:= \frac{P_{a_i} \cdot T_i}{s} \\
 \underline{P} &:= \underline{P}_v^T = (0.626 \quad 0.112 \quad 0.08 \quad 0.182)
 \end{aligned}$$

Имитационное моделирование полумарковских процессов с вложенными дискретными марковскими цепями

В дискретных марковских цепях предполагается, что изменение состояния происходит мгновенно через равные интервалы времени, а в течение этих интервалов состояние не меняется.

В реальных системах время пребывания в состоянии после перехода может быть случайным и определяться экспоненциальной или иной плотностью вероятности, а сами переходы задаваться матрицей переходных вероятностей. В этом случае говорят, что переходы осуществляются *вложенной* дискретной марковской цепью, а сам процесс в системе является *двумерным*.

В имитационной модели такого процесса состояние, в которое перейдет система, следует определить с помощью матрицы переходных вероятностей, а затем сформировать значение времени пребывания системы в состоянии, из которого осуществляется переход.

В общем случае закон распределения для пребывания системы в состоянии может быть разным и определяться направлением перехода. В этом случае говорят о *полумарковском* случайном процессе [4].

Имитационная модель для двумерных процессов будет представлять собой объединение алгоритмов, использованных при имитационном моделировании дискретной и непрерывной марковских цепей.

В случае если плотность вероятности для времени пребывания системы в состоянии является экспоненциальной, код основной части имитационной модели имеет вид

```

for i=1:N,
    %Определение номера нового состояния
    s1=rand;k=1;
    while(s1>mvi(ktek,k)),k=k+1; end, knew=k;
    Ttek=random('exp',ff(ktek,knew));%Время пребывания
    
```

```
    % системы в состоянии ktek при переходе
    % в состояние knew
    Tsost(ktek)=Tsost(ktek)+Ttek; %Суммарное время
    % пребывания системы в состоянии
    ktek=knw;%ktek - номер текущего состояния системы
end
```

В этом коде время пребывания в данном состоянии зависит от последующего состояния процесса, т. е. код представляет собой имитационную модель полумарковского процесса [4]. В общем случае все элементы в матрице ff , определяющей параметры плотностей вероятностей для экспоненциального закона, являются различными. Элементами матрицы в Matlab являются средние значения времени пребывания системы в состояниях.

Если плотность вероятности для времени пребывания системы в состоянии отличается от экспоненциальной, код основной части имитационной модели имеет вид

```
for i=1:N,
    %Определение номера нового состояния
    sl=rand;k=1;
    while(sl>mvi(ktek,k)),k=k+1; end,
    knew=k;%knw - номер нового состояния системы
    %Формирование значения времени пребывания системы
    % в состоянии до перехода
    if (vf(ktek,knew)==1)
        ssl=random('exp',param1(ktek,knew)); end
    if (vf(ktek,knew)==2)
        ssl=random('rayl',param1(ktek,knew)); end
    if (vf(ktek,knew)==3)
        ssl=random('wbl',param1(ktek,knew),...
            param2(ktek,knew)); end
    if (vf(ktek,knew)==4)
        ssl=random('gam',param1(ktek,knew),...
            param2(ktek,knew)); end
    Ttek=ssl; %Время пребывания системы в состоянии
    %до перехода
    Tsost(ktek)=Tsost(ktek)+Ttek; %Суммарное время
    % пребывания системы в состоянии
    ktek=knw;%ktek -номер текущего состояния системы
end
```

Переменная $Tsost(j)$ содержит суммарное время пребывания системы в j -ом состоянии.

В приведенном коде имитационной модели предусмотрено применение для времени пребывания системы в состоянии четырех наиболее часто используемых распределений: экспоненциального (*exp* – код 1), Рэлея (*rayl* – код 2), Вейбулла (*wbl* – код 3), Гамма-распределения (*gam* – код 4) (приложение 1).

В матрице vf указывается код вида плотности вероятности для каждого варианта перехода из состояния, а в матрицах $param1$ и $param2$ – параметры плотностей вероятностей.

Проверку правильности имитационной модели выполним путем сравнения значений стационарных вероятностей полумарковского процесса, полученных по аналитическим формулам [3], и значений, являющихся результатом работы имитационной модели.

$$p_j = \frac{a_j T_{sj}}{\sum_{i=1}^n a_i T_{si}}, j = 1, 2, \dots, n, \quad (14)$$

где: $a_j, j = 1, 2, \dots, n$ – стационарные вероятности для вложенной дискретной марковской цепи; $T_{si}, i = 1, 2, \dots, n$ – время пребывания в i -ом состоянии, вычисляемое по формуле

$$T_{si} = \sum_{j=1}^n q_{ij} T_{ij}, \quad (15)$$

где $T_{ij}, i, j = 1, 2, \dots, n$ – среднее время пребывания в i -ом состоянии до перехода в j -е состояние.

Выполним расчеты для исходных данных, приведенных в таблице 3.

Таблица 3 – Исходные данные для расчета по аналитическим формулам

Переход $i-j$	Код плотности вероятности	Параметры плотности вероятности	T_{ij}	T_{si}
1-1	1	$m = 7,5$	7,5	5,01
1-2	2	$b = 2$	2,507	
1-3	3	$a = 1,5; b = 2$	1,329	
1-4	4	$a = 2; b = 3$	6	
2-1	2	$b = 4$	5,013	3,335
2-2	2	$b = 2,5$	3,133	
2-3	3	$a = 2,1; b = 1,5$	1,896	
2-4	3	$a = 2,4; b = 2,2$	2,125	
3-1	3	$a = 3; b = 2$	2,659	2,735
3-2	3	$a = 1,5; b = 2$	1,329	
3-3	4	$a = 3; b = 1,8$	5,4	
3-4	4	$a = 2,5; b = 1,2$	3	
4-1	4	$a = 2; b = 2$	4	3,14
4-2	4	$a = 3; b = 1,6$	4,8	
4-3	1	$m = 1,4$	1,4	
4-4	1	$m = 2$	2	

Значения T_{ij} рассчитаны в соответствии с приложением 1, а значения T_{si} – по формуле (15) для следующей матрицы переходных вероятностей:

$$q = [0.35 \ 0.3 \ 0.1 \ 0.25; \ 0.4 \ 0.1 \ 0.2 \ 0.3; \dots \\ 0.5 \ 0.2 \ 0.1 \ 0.2; \ 0.35 \ 0.2 \ 0.2 \ 0.25].$$

Стационарные вероятности вложенной дискретной марковской цепи получены как решение системы линейных алгебраических уравнений вида (3)

$$a = [0.3829 \ 0.2166 \ 0.147 \ 0.2535].$$

Стационарные вероятности рассматриваемого полумарковского процесса, рассчитанные по формуле (14), следующие:

$$p = [0.5 \ 0.188 \ 0.105 \ 0.207].$$

Далее проведем имитационное моделирование двумерного полумарковского процесса.

Матрицы с исходными данными для имитационной модели имеют вид

$$vf = [1 \ 2 \ 3 \ 4; \ 2 \ 2 \ 3 \ 3; \ 3 \ 3 \ 4 \ 4; \ 4 \ 4 \ 1 \ 1];$$

$$param1 = [7.5 \ 2 \ 1.5 \ 2; \ 4 \ 2.5 \ 2.1 \ 2.4; \ 3 \ 1.5 \ 3 \ 2.5; \dots$$

$$2 \ 3 \ 1.4 \ 2];$$

$$param2 = [0 \ 0 \ 2 \ 3; \ 0 \ 0 \ 1.5 \ 2.2; \ 2 \ 2 \ 1.8 \ 1.2; \ 2 \ 1.6 \ 0 \ 0].$$

Результаты, полученные с помощью имитационной (psostmods) модели при числе переходов $N = 50000$, следующие:

$$psostmods = [0.4982 \ 0.187 \ 0.1054 \ 0.2094].$$

Близость значений стационарных вероятностей подтверждает корректность подхода при построении имитационной модели.

Стационарные вероятности полумарковского процесса не зависят от вида законов распределения для времени пребывания в состояниях, а определяются только средними значениями времени [4]. Поэтому при имитационном моделировании будут получены те же результаты, если все плотности вероятностей задать экспоненциальными для переходов $i - j$, а в качестве значения параметра для каждой плотности вероятности использовать среднее время пребывания в i -ом состоянии до перехода в j -е состояние T_{ij} (значения из предпоследнего столбца таблицы 3).

Для данных таблицы 3 матрица ff будет иметь следующий вид:

$$ff = [7.5 \ 2.507 \ 1.329 \ 6; \ 5.013 \ 3.133 \ 1.896 \ 2.125; \dots$$

$$2.659 \ 1.329 \ 5.4 \ 3; \ 4 \ 4.8 \ 1.4 \ 2];$$

Результаты, полученные с помощью имитационной (psostmods) модели при числе переходов $N = 50000$, следующие:

$$psostmods = [0.5039 \ 0.1861 \ 0.1043 \ 0.2057].$$

Здесь имеется очевидное неудобство для подготовки исходных данных для имитационной модели, связанное с необходимостью вычислений T_{ij} .

Если закон распределения для переходов из состояний является экспоненциальным и *не зависит* от направления перехода, то в системе протекает марковский процесс [3], являющийся частным случаем полумарковского процесса.

Для марковского процесса в строки матрицы ff должны быть записаны одинаковые значения.

Стационарные вероятности марковского процесса определяются по формуле [3]

$$p_j = \frac{a_j}{\lambda_j \sum_{i=1}^n \frac{a_i}{\lambda_i}}, \quad j = 1, 2, \dots, n, \quad (16)$$

где: $a_j, j = 1, 2, \dots, n$ – стационарные вероятности для вложенной дискретной марковской цепи; $\lambda_j, j = 1, 2, \dots, n$ – интенсивности переходов из состояний.

Выполним расчеты для следующих исходных данных, приведенных в [3]:

$$Q = \begin{pmatrix} 0 & 0.1 & 0.5 & 0.4 \\ 0.2 & 0.1 & 0.2 & 0.5 \\ 0.3 & 0.2 & 0.2 & 0.3 \\ 0.1 & 0.3 & 0.6 & 0 \end{pmatrix}; \quad \lambda = \begin{pmatrix} 0.5 \\ 1 \\ 0.75 \\ 2 \end{pmatrix}.$$

Стационарные вероятности вложенной дискретной марковской цепи, полученные при решении системы линейных алгебраических уравнений вида (3), следующие:

$$a = [0.1740 \ 0.1909 \ 0.3616 \ 0.2735].$$

Стационарные вероятности марковского процесса, рассчитанные по формуле (16), равны

$$p = [0.3006 \ 0.1649 \ 0.4164 \ 0.1181].$$

Результаты, полученные с помощью имитационной (`psostmods`) модели при числе переходов $N = 50000$, следующие:

$$psostmods = [0.3003 \ 0.1652 \ 0.4166 \ 0.1180].$$

Для имитационной модели матрица `ff` имеет вид

$$ff = [2 \ 2 \ 2 \ 2; \ 1 \ 1 \ 1 \ 1; \ 1.333 \ 1.333 \ 1.333 \ 1.333; \dots \\ 0.5 \ 0.5 \ 0.5 \ 0.5].$$

В каждой строке этой матрицы записаны значения, обратные интенсивностям переходов из состояний.

Листинг имитационной модели полумарковских процессов со сменой состояний дискретной марковской цепью приведен в приложении 2.

Имитационное моделирование полумарковских

процессов с вложенными непрерывными марковскими цепями

Рассмотрим имитационное моделирование функционирования систем, в которых состояния изменяются мгновенно, но при этом случайные моменты смены состояний задаются экспоненциальными плотностями вероятностей. Кроме того, система после перехода в некоторое состояние остается в нем в течение некоторого случайного времени с произвольным законом распределения, зависящим от направления перехода.

Модель такой системы представляет собой полумарковский процесс [3, 4].

В этом двумерном процессе переход в новое состояние осуществляется вложенной непрерывной марковской цепью, а случайное время пребывания системы в состоянии до перехода в новое состояние определяется плотностью вероятности, вид и параметры которой зависят от направления перехода.

В качестве исходных данных для модели должны использоваться четыре матрицы, если ограничиться двухпараметрическими плотностями вероятностей.

Одна матрица должна определять параметры плотностей вероятностей для вложенной непрерывной марковской цепи, а три других определять соответственно виды распределений и параметры плотностей вероятностей для времени пребывания системы в состояниях до перехода в другие состояния.

Процедура выбора состояния, в которое должна перейти система из текущего состояния, следующая:

- генерируются случайные числа в соответствии с плотностями вероятностей для всех потоков событий, переводящих систему из текущего состояния в другие возможные состояния;
- выбирается минимальное случайное число, соответствующее наступлению ближайшего события, и фиксируется номер состояния, в которое должна перейти система.

Далее в зависимости от того, в какое состояние должна перейти система, формируется с учетом соответствующей плотности вероятности случайное значение времени пребывания системы в текущем состоянии.

Основная часть имитационной модели имеет вид

```
for i=1:N, %Цикл по количеству смен состояний
    for m=1:n, sl(m)=0;
        if(ff(ktek,m)~=0), sl(m)=random('exp',ff(ktek,m)); end
    end
    %Определение наиболее раннего события
    mint=nval;
    for m=1:n,
        if((m ~= ktek) && (sl(m)< mint) && (sl(m)~=0)),
            mint=sl(m);knew=m; %knew - номер состояния,
                %в которое должна перейти система
        end
    end
    %Формирование значения времени пребывания системы
    %в состоянии до перехода
    if (vf(ktek,knew)==1)
        ssl=random('exp',param1(ktek,knew));
    end
    if (vf(ktek,knew)==2)
        ssl=random('rayl',param1(ktek,knew));
    end
    if (vf(ktek,knew)==3)
        ssl=random('wbl',param1(ktek,knew),...
            param2(ktek,knew));
    end
    if (vf(ktek,knew)==4)
        ssl=random('gam',param1(ktek,knew),...
            param2(ktek,knew));
    end
    Ttek=ssl; %Время пребывания системы в состоянии
                %до перехода
    Tsost(ktek)=Tsost(ktek)+Ttek; %Суммарное время пребывания
                % системы в состоянии
    ktek=knew;%ktek - номер текущего состояния системы
end
```

В приведенном коде имитационной модели предусмотрено применение для времени пребывания системы в состоянии четырех наиболее часто используемых распределений: экспоненциального (*exp*), Рэля (*rayl*), Вейбулла (*wbl*), Гамма-распределения (*gam*) (приложение 1).

В матрице νf указывается код вида плотности вероятности для каждого варианта перехода из состояния, а в матрицах param1 и param2 – параметры плотностей вероятностей.

Оценки вероятностей состояний $P_i, i = 1, 2, \dots, n$ в установившемся режиме вычисляются как отношения времени пребывания системы в конкретных состояниях ко времени пребывания системы во всех состояниях при достаточно большом количестве испытаний модели.

В общем случае смена состояний может задаваться плотностями вероятностей, отличающимися от экспоненциальных.

В качестве исходных данных для такой модели должны использоваться шесть матриц, если ограничиться двухпараметрическими плотностями вероятностей.

Три матрицы должны определять параметры плотностей вероятностей для переходов в новые состояния, а три других определять соответственно виды распределений и параметры плотностей вероятностей для времен пребывания системы в состояниях до перехода в другие состояния.

Листинг такого варианта модели приведен в приложении 3.

В модели реализован цикл по количеству переходов N и сглаживание оценок стационарных вероятностей с помощью рекуррентного усреднения.

Имитационную модель можно считать корректной, поскольку она объединяет решения, корректность которых уже проверена.

Выводы

Таким образом, с применением системы Matlab разработаны имитационные модели сложных систем с дискретными состояниями для разных видов случайных процессов, протекающих в системах, позволяющие получить оценки стационарных вероятностей состояний систем.

Разработанные имитационные модели обладают преимуществами в использовании по сравнению с аналитическими моделями, обеспечиваемыми возможностью параметрической настройки моделей для каждой конкретной системы.

В моделях обеспечена сходимости оценок вероятностей состояний при использовании их рекуррентного усреднения в сеансах моделирования с нарастающим числом реализаций модели.

Обеспечена визуализация результатов моделирования, позволяющая принимать решения об изменении числа сеансов моделирования.

Корректность имитационных моделей подтверждена путем сравнения результатов имитационного моделирования и аналитических методов расчета.

Кроме того, разработанные модели позволяют исследовать влияние параметров систем на значения вероятностей состояний.

Дополнительная возможность имитационных моделей – протоколирование процесса смены состояний за счет возможностей, предоставляемых Matlab.

Разработанные модели могут использоваться как для проведения практических расчетов при исследовании сложных систем с дискретными состояниями, так и для проверки корректности аналитических методов расчета.

Приложение 1

Вид и параметры плотности вероятности для некоторых распределений в Matlab

Таблица П1 – Вид и параметры распределений

Наименование, параметры распределения, обозначения в Matlab	Формулы для плотности вероятности $f(x)$, функции распределения $F(x)$ и математического ожидания m_x
<p>Экспоненциальное, параметр m – среднее значение В Matlab – exp или Exponential</p>	$f(x) = \begin{cases} \frac{1}{m} e^{-\frac{x}{m}} & 0 \leq x < \infty; \\ 0, & x < 0 \end{cases}$ $F(x) = 1 - e^{-x/m}, m_x = m$
<p>Рэля, $b > 0$ – параметр масштабирования В Matlab – rayl или Rayleigh</p>	$f(x) = \begin{cases} \frac{x}{b^2} e^{-\frac{x^2}{2b^2}}, & x > 0, \\ 0, & x \leq 0 \end{cases}$ $F(x) = 1 - e^{-\frac{x^2}{2b^2}}, m_x = b \sqrt{\frac{\pi}{2}} = 1,25331b$
<p>Вейбулла, $a > 0$ – параметр масштабирования, $b > 0$ – параметр формы с параметрами a и b. В Matlab – wbl или Weibull</p>	$f(x) = \begin{cases} \frac{b}{a} \left(\frac{x}{a}\right)^{b-1} e^{-\left(\frac{x}{a}\right)^b}, & x > 0; \\ 0, & x \leq 0 \end{cases}$ $F(x) = 1 - e^{-\left(\frac{x}{a}\right)^b}, m_x = a\Gamma(1+1/b),$ <p>$\Gamma(\cdot)$ – гамма-функция, gamma – в Matlab</p>
<p>Гамма-распределение, $a > 0$ – параметр формы, $b \geq 0$ – параметр масштабирования В Matlab – gam или Gamma</p>	$f(x) = \begin{cases} \frac{1}{b^a \Gamma(a)} x^{a-1} e^{-\frac{x}{b}} & x > 0; \\ 0, & x \leq 0 \end{cases}$ $F(x) = \frac{\Gamma(a, x/b)}{b^a \Gamma(a)}, m_x = ab,$ <p>$\Gamma(\cdot)$ – гамма-функция, gamma – в Matlab</p>

Примечание. Распределение Рэля является частным случаем распределения Вейбулла. Распределение Рэля с параметром b эквивалентно распределению Вейбулла с параметрами $A = \sqrt{2}b$ и $B = 2$.

Приложение 2

Имитационная модель марковского и полумарковского процессов с дискретным временем смены состояний

```
%Имитационная модель марковского и полумарковского процессов
% со сменой состояний дискретной марковской цепью
%N - количество переходов
%NN - количество сеансов моделирования
NN=20
n=4; %Количество состояний
upr=1 %upr=0 - плотности вероятностей для времен пребывания
      % системы в состояниях экспоненциальные; upr=1- плотности
      % вероятностей для времен пребывания системы в состояниях
      % произвольные, включая экспоненциальные
%Вероятности состояний в начальный момент времени
p=[0.3 0.2 0.4 0.1];
%Матрица переходных вероятностей
q=[0.35 0.3 0.1 0.25; 0.4 0.1 0.2 0.3; 0.5 0.2 0.1 0.2;...
   0.35 0.2 0.2 0.25];
%Средние значения для времени пребывания системы в состоянии до
%перехода
ff=[7.5 2.507 1.329 6; 5.013 3.133 1.896 2.125;...
   2.659 1.329 5.4 3; 4 4.8 1.4 2];
vf=[1 2 3 4; 2 2 3 3; 3 3 4 4; 4 4 1 1];%Код вида плотности
% вероятностей для времени пребывания системы в состоянии
% до перехода
param1=[7.5 2 1.5 2; 4 2.5 2.1 2.4; 3 1.5 3 2.5; 2 3 1.4 2];
%Первые параметры плотности вероятностей для времени пребывания
%системы в состоянии до перехода
param2=[0 0 2 3; 0 0 1.5 2.2; 2 2 1.8 1.2; 2 1.6 1.4 2];%Вторые
% параметры плотности вероятностей для времени пребывания системы
%в состоянии до перехода
%Формирование матрицы вероятностных интервалов
for i=1:n,sper=0;
    for j=1:n,
        for k=1:j,sper=sper+q(i,k);end
        mvi(i,j)=sper; sper=0;
    end,
end
%Определение состояния в начальный момент времени
s=p(1); k=1;
gg=rand; %Получение случайного числа, равномерно распределенного
%в (0;1)
while (gg>s),k=k+1;s=s+p(k);end %k - номер начального состояния
ktek=k; %ktek - номер текущего состояния
for ii=1:NN,
    N=ii*1000;
    for i=1:n,Tsost(i)=0; end %Tsost(i) - время пребывания
        %системы в i-ом состоянии
    %Моделирование процесса смен состояний, определение времени
% пребывания системы в состояниях до переходов, определение
```

```
%вероятностей состояний
for i=1:N,
    %Определение номера нового состояния
    sl=rand;k=1;
    while(sl>mvi(ktek,k)),k=k+1; end,
    knew=k;%knew - номер нового состояния системы
    if(upr==0),
        Ttek=random('exp',ff(ktek,knew));%Время пребывания
        %системы в состоянии до перехода для марковского процесса
    end
    if(upr==1),
        %Формирование значения времени пребывания системы
        %в состоянии до перехода для полумарковского процесса
        if (vf(ktek,knew)==1)
            ssl=random('exp',param1(ktek,knew));
        end
        if (vf(ktek,knew)==2)
            ssl=random('rayl',param1(ktek,knew));
        end
        if (vf(ktek,knew)==3)
            ssl=random('wbl',param1(ktek,knew),param2(ktek,knew));
        end
        if (vf(ktek,knew)==4)
            ssl=random('gam',param1(ktek,knew),param2(ktek,knew));
        end
        Ttek=ssl; %Время пребывания системы в состоянии
        %до перехода
    end
    Tsost(ktek)=Tsost(ktek)+Ttek; %Суммарное время пребывания
    %системы в состоянии
    ktek=knew; %ktek - номер текущего состояния системы
end
N
%Определение времени пребывания системы во всех состояниях
sumps=0;
for j=1:n,sumps=sumps+Tsost(j); end
%Определение вероятностей пребывания системы в каждом
%из состояний
for j=1:n,psostmod(ii,j)=Tsost(j)/sumps; end
% Сглаживание вероятностей состояний с помощью рекуррентного
%усреднения
if(ii==1),
    for(j=1:1:n),x(j)=psostmod(ii,j);end
    for j=1:n,xx(ii,j)=x(j); end
end
if(ii~=1),
    for j=1:n,x(j)=x(j)+(psostmod(ii,j)-x(j))/ii; end
    for j=1:n,xx(ii,j)=x(j); end
end
end
for(j=1:1:n),psostmods(j)=psostmod(NN,j);end
psostmods
%Формирование графика значений вероятностей пребывания системы
```

```
%в каждом из состояний, полученных при имитац. моделировании
%при разных количествах реализаций модели (смен состояний)
ii=1:1:NN;
plot(ii*1000,xx(ii,1),'r',ii*1000,xx(ii,2),'g',...
      ii*1000,xx(ii,3),'b',ii*1000,xx(ii,4),'k')
xlabel('Количество реализаций модели')
ylabel('Вероятности состояний')
```

Приложение 3

Имитационная модель полумарковского процесса с непрерывным временем смены состояний

```
%Имитационная модель процесса со сменой состояний потоком событий,
%описываемом непрерывной марковской цепью, и полумарковским
%потоком событий
%N - количество смен состояний
%NN - количество "прогонов" модели
NN=20
%Количество состояний
n=4;
upr=0 %upr=0 - смена состояний осуществляется непрерывной
% марковской цепью; upr=1 - смена состояний осуществляется
%полумарковским потоком событий
%Вероятности состояний в начальный момент времени
p=[0.3 0.2 0.4 0.1];
%Средние значения времени между сменами состояний
ff=[0 4.0 0 3.9; 0 0 1.5 2.5; 0 0 0 2.5; 1.4 0 0 0];
vf=[0 1 0 1; 0 0 1 1; 0 0 0 1; 3 0 0 0];%Код вида плотности
%вероятностей для времени пребывания системы в состоянии
%до перехода
param1=[0 2 0 2; 0 0 4 3; 0 0 0 3; 5.642 0 0 0];%Первые
%параметры плотности вероятностей для времени пребывания системы
%в состоянии до перехода
param2=[0 0 0 0; 0 0 0 0; 0 0 0 0; 2 0 0 0];%Вторые параметры
%плотности вероятностей для времени пребывания системы
%в состоянии после перехода
vf1=[0 1 0 1; 0 0 1 1; 0 0 0 1; 3 0 0 0];%Код вида плотности
%вероятности для времени смены состояний
param11=[0 4 0 3.9; 0 0 1.5 2.5; 0 0 0 2.5; 5.642 0 0 0];%Первые
%параметры плотности вероятности для времени смены состояний
%param11=[0 2 0 0.4; 0 0 2 0.4; 0 0 0 0.4; 2.4 0 0 0];
param21=[0 0 0 0; 0 0 0 0; 0 0 0 0; 2 0 0 0];%Вторые параметры
%плотности вероятности для времени смены состояний
nval=1e6; %Нач. значение для поиска минимума
%Определение состояния в начальный момент времени
s=p(1); k=1;
gg=rand; %Получение случайного числа, равномерно распределенного
%в (0;1)
while (s<gg),k=k+1;s=s+p(k);end %k - номер начального состояния
%Моделирование процессов в системе (смен состояний, пребывания
%системы в состоянии после перехода в это состояние)
%i определение вероятностей состояний системы
```

```
ktek=k; %ktek -текущее состояние
for ii=1:NN,
    N=ii*1000;
    for i=1:n,Tsost(i)=0; end %Tsost(i) - время пребывания
        % системы в i-ом состоянии
    for i=1:N, %Цикл по количеству смен состояний
        %Формирование моментов смен состояний
        if(upr==0),
            for m=1:n,sl(m)=0;
                if(ff(ktek,m)~=0), sl(m)=random('exp',ff(ktek,m)); end
            end
        end
        if(upr==1),
            for m=1:n, sl(m)=0;
                if((vf1(ktek,m)~=0) && (vf1(ktek,m)==1))
                    sl(m)=random('exp',param11(ktek,m));
                end
                if((vf1(ktek,m)~=0) && (vf1(ktek,m)==2))
                    sl(m)=random('rayl',param11(ktek,m));
                end
                if((vf1(ktek,m)~=0) && (vf1(ktek,m)==3))
                    sl(m)=random('wbl',param11(ktek,m),param21(ktek,m));
                end
                if((vf1(ktek,m)~=0) && (vf1(ktek,m)==4))
                    sl(m)=random('gam',param11(ktek,m),param21(ktek,m));
                end
            end
        end
    end
    %Определение наиболее раннего события
    mint=nval;
    for m=1:n,
        if((m ~= ktek) && (sl(m)< mint) && (sl(m)~=0)),
            mint=sl(m);knew=m; %knew - номер состояния,
                %в которое должна перейти система
        end
    end
    %Формирование значения времени пребывания системы
    %в состоянии до перехода
    if(vf(ktek,knew)==1)
        ssl=random('exp',param1(ktek,knew));
    end
    if(vf(ktek,knew)==2)
        ssl=random('rayl',param1(ktek,knew));
    end
    if (vf(ktek,knew)==3)
        ssl=random('wbl',param1(ktek,knew),param2(ktek,knew));
    end
    if (vf(ktek,knew)==4)
        ssl=random('gam',param1(ktek,knew),param2(ktek,knew));
    end
    Ttek=ssl; %Время пребывания системы в состоянии
        %до перехода
    Tsost(ktek)=Tsost(ktek)+Ttek; %Суммарное время пребывания
```

```
% СИСТЕМЫ В СОСТОЯНИИ
ktek=knew;%ktek - номер текущего состояния системы
end
N
%Определение времени пребывания системы во всех состояниях
sumps=0;
for j=1:n,sumps=sumps+Tsost(j); end
%Определение вероятностей пребывания системы в каждом
%из состояний
for j=1:n,psostmod(ii,j)=Tsost(j)/sumps; end
% Сглаживание вероятностей состояний с помощью
%рекуррентного усреднения
if(ii==1),
    for(j=1:1:n),x(j)=psostmod(ii,j);end
    for j=1:n,xx(ii,j)=x(j); end
end
if(ii~=1),
    for j=1:n,x(j)=x(j)+(psostmod(ii,j)-x(j))/ii; end
    for j=1:n,xx(ii,j)=x(j); end
end
end
for(j=1:1:n),psostmods(j)=psostmod(NN,j);end
psostmods
%Формирование графика значений вероятностей пребывания системы
%в каждом из состояний, полученных при имитац. моделировании
%при разных количествах реализаций модели (смен состояний)
ii=1:1:NN;
plot(ii*1000,xx(ii,1),'r',ii*1000,xx(ii,2),'g',...
    ii*1000,xx(ii,3),'b',ii*1000,xx(ii,4),'k')
xlabel('Количество реализаций модели')
ylabel('Вероятности состояний')
```

Литература

1. Андреещев И. А., Будников С. А. Полумарковская модель оценки надежности функционирования автоматизированной системы управления силами и средствами радиоэлектронной борьбы // Вестник военно-воздушной академии. 2019. № 2 (32). С. 283-288.
2. Борисевич А. В., Дякин Н. В. Полумарковская модель для оценки показателей надежности источника бесперебойного питания дата-центра // Современные научные исследования и инновации. 2015. № 8. Ч. 1 [Электронный ресурс]. URL: <http://web.snauka.ru/issues/2015/08/57039> (дата обращения: 12.01.2018).
3. Андронов А. М., Копытов Е. А., Гринглаз Л. Я. Теория вероятностей и математическая статистика: Учебник для вузов. – СПб.: Питер, 2004. – 461 с.
4. Тихонов В. И., Миронов М. А. Марковские процессы. – М.: Советское радио, 1977. – 488 с.
5. Коваленко И. Н., Москатов Г. К., Барзилович Е. Ю. Полумарковские модели в задачах проектирования систем управления летательными аппаратами. – М.: Машиностроение, 1973. – 176 с.

6. Xie W., Hong Y., Trivedi K. Analysis of a two-level software rejuvenation policy // *Reliability Engineering and System Safety*. 2005. v. 87. P. 13-22.

7. Гуляев А. К. Визуальное моделирование в среде Matlab. – СПб.: Питер, 2000. – 432 с.

8. Afanas'evskij L. B., Gorin A. N., Chursin M. A. Metod imitacionnogo modelirovanija kak sredstvo analiza stohasticheskikh setevykh grafikov [Method of simulation modeling as a means of analyzing stochastic network graphs]. *Vestnik of Voronezh State University. Series: Systems Analysis and Information Technologies*, 2015, no. 2, pp. 54-59. (in Russian).

9. Afanas'evskij L. B., Gorin A. N., Fadin A. G. Imitational modeling of event-driven systems in MatLab + Simulink. *Informatika: problemy, metodologiya, tekhnologii: materialy XVIII Mezhdunarodnoj nauchno-metodicheskoy konferencii* [Information: Problems, Methodology, Technologies. Materials of the XVIII International scientific methodical conference]. Voronezh, Voronezh State University, 2018, vol. 2, pp. 46-50 (in Russian).

10. Вентцель Е. С., Овчаров Л. А. Теория случайных процессов и ее инженерные приложения – М.: Высшая школа, 2000. – 384 с.

11. Фадин А. Г. Моделирование радиоэлектронных систем на ЭВМ. – Воронеж: ВИРЭ, 2000. – 493 с.

References

1. Andreeshhev I. A., Budnikov S. A. Polumarkovskaja model' ocenki nadezhnosti funkcionirovanija avtomatizirovannoj sistemy upravlenija silami i sredstvami radiojelektronnoj bor'by [The semi-Markov model for assessing the reliability of the automated control system for forces and electronic warfare weapons]. *Vestnik voenno-vozdushnoj akademii*, 2019, no. 2 (32), pp. 283-288 (in Russian).

2. Borisevich A. V., Djakin N. V. Polumarkovskaja model' dlja ocenki pokazatelej nadezhnosti istochnika besperebojnogo pitanija data-centra [The semi-Markov model for assessing the reliability indicators of the uninterruptible power supply of the data center]. *Sovremennye nauchnye issledovanija i innovacii*, 2015, no. 8. Available at: <http://web.snauka.ru/issues/2015/08/57039> (accessed 12 January 2018) (in Russian).

3. Andronov A. M., Kopytov E. A., Gringlaz L. Ja. *Teorija verojatnostej i matematicheskaja statistika* [Probability theory and mathematical statistics]. Saint Petersburg, 2004. 461 p. (in Russian).

4. Tihonov V. I., Mironov M. A. *Markovskie processy* [Markov processes]. Moscow, Sovetskoe radio, 1977, 488 p. (in Russian).

5. Kovalenko I. N., Moskatov G. K., Barzilovich E. Ju. *Polumarkovskie modeli v zadachah proektirovanija sistem upravlenija letatel'nymi apparatami* [Semi-Markov models in the problems of designing aircraft control systems]. Moscow, Mashinostroenie, 1973, 176 p. (in Russian).

6. Xie W., Hong Y., Trivedi K. Analysis of a two-level software rejuvenation policy. *Reliability Engineering and System Safety*, 2005, vol. 87, pp. 13-22.

7. Gul'tjaev A. K. *Vizual'noe modelirovanie v srede Matlab* [Visual modeling in Matlab environment]. Saint Petersburg, Piter, 2000, 432 p. (in Russian).

8. Afanas'evskij L. B., Gorin A. N., Chursin M. A. Metod imitacionnogo modelirovaniya kak sredstvo analiza stohasticheskikh setevykh grafikov [Method of simulation modeling as a means of analyzing stochastic network graphs]. *Vestnik of Voronezh State University. Series: Systems Analysis and Information Technologies*, 2015, no. 2, pp. 54-59 (in Russian).

9. Afanas'evskij L. B., Gorin A. N., Fadin A. G. Imitational modeling of event-driven systems in MatLab + Simulink. *Informatika: problemy, metodologiya, tekhnologii: materialy XVIII Mezhdunarodnoj nauchno-metodicheskoy konferencii* [Information: Problems, Methodology, Technologies. Materials of the XVIII International scientific methodical conference]. Voronezh, Voronezh State University, 2018, vol. 2, pp. 46-50 (in Russian).

10. Ventcel' E. S., Ovcharov L. A. *Teorija sluchajnykh processov i ee inzhenernye prilozheniya* [The Theory of Random Processes and its Engineering Applications]. Moscow, 2000, 384 p. (in Russian).

11. Fadin A. G. *Modelirovanie radioelektronnykh sistem na JeVM* [Simulation of radio electronic systems on a computer]. Voronezh, Military Institute of Radioelectronics, 2000, 493 p. (in Russian).

Статья поступила 11 февраля 2019 г.

Информация об авторах

Горин Александр Николаевич – кандидат технических наук. Старший преподаватель кафедры автоматизированных систем управления. Военный учебно-научный центр Военно-воздушных сил «Военно-воздушная академия им. проф. Н.Е. Жуковского и Ю.А. Гагарина» (г. Воронеж). Область научных интересов: моделирование сложных систем. E-mail: algorin.algoral@mail.ru

Будников Сергей Алексеевич – доктор технических наук, доцент. Начальник кафедры автоматизированных систем управления. Военный учебно-научный центр Военно-воздушных сил «Военно-воздушная академия им. проф. Н.Е. Жуковского и Ю.А. Гагарина» (г. Воронеж). Область научных интересов: защита информации, моделирование сложных систем. E-mail: buser@bk.ru

Адрес: 394064, Россия, г. Воронеж, ул. Старых Большевиков, 54а.

Use of Matlab Software Environment for Simulation Modeling of Complex Military Systems

A. N. Gorin, S. A. Budnikov

Problem statement: *mathematical models of the functioning of complex military systems, built on the block-modular principle, determine the sequence of changing discrete states that characterize partial or total loss of functional capabilities of the systems. The semi-Markov models are an adequate form of describing such systems, and the probabilities of a system in each of the given states in a stationary mode are usual-*

ly used as performance indicators. The problem of calculating the state probabilities for the stationary mode of the semi-Markov process is rather complicated, it is reduced to multi-stage calculations and is inconvenient for programming. **Objective:** simplification of the procedure of obtaining estimates for the stationary probabilities of the states for the systems described by semi-Markov models. **Methods used:** simulation methods for semi-Markov systems, the formation of event flows with different distribution laws in Matlab models, initiating transitions from current states to other states and determining residence times in states, obtaining empirical estimates of state probabilities. **Novelty:** the usage of structural elements of semi-Markov models in simulation models, the usage of recurrent averaging to obtain "point-like" estimates of the probabilities of states. **Result:** Matlab simulation models, easily customizable for the study of systems with an arbitrary number of discrete states and various random processes occurring in the systems. **Practical significance:** the developed simulation models make it possible to obtain estimates of the probabilities of states quite simply by specifying the structural parameters of the system, the types and parameters of distributions for event flows occurring in real systems. Possible ways of using probability estimates is to help in making decisions about the compliance of systems with their requirements and checking the correctness of analytical models.

Keywords: simulation modeling, semi-Markov models, stationary state probabilities.

Information about Authors

Alexander Nikolaevich Gorin – Ph.D of Engineering Sciences. Senior Lecturer at the Department of Automated Control Systems. Military Educational and Scientific Center of the Air Force "N.E. Zhukovsky and Y.A. Gagarin Air Force Academy" (Voronezh). Research interests: modeling of complex systems. E-mail: algorin.algoral@mail.ru

Sergey Alekseevich Budnikov – Advanced Doctor, Docent. Head of the Department of automated control systems. Military Educational and Scientific Center of the Air Force "N.E. Zhukovsky and Y.A. Gagarin Air Force Academy" (Voronezh). Research interests: information security, modeling of complex systems. E-mail: buser@bk.ru

Address: Russia, 394064, Voronezh, ulica Staryh Bol'shevikov, 54a.