Министерство образования и науки Российской Федерации
Федеральное агентство по образованию
Санкт-Петербургский Государственный Университет
Информационных технологий, механики и оптики

И. А. Коняхин

МЕТОДЫ И СРЕДСТВА СТАТИСТИЧЕСКОГО МОДЕЛИРОВАНИЯ ОЭС

(анализ надежности)

Учебное пособие

Санкт –Петербург
2005

УДК 658.512.011:535.8

Коняхин И.А. Методы и средства статистического моделирования ОЭС (анализ надежности) / Учебное пособие. — СПб: ИТМО, 2005. – с.

Издается по решению Ученого Совета факультета Оптико-информационных систем и технологий от 17 мая 2005 г., протокол № 5.

В пособии изложены алгоритмы и методы статистического моделирования (метод Монте-Карло) на примере анализа надежности сложных технических (в частности, оптико-электронных) систем.

Рассматриваются параметры и характеристики надежности технических систем, методика построения и исследования их статистических моделей в технологии MatLab. Анализируются базовые методики синтеза случайных величин с заданными законами распределения , а также инструментарий Matlab, используемый для моделирования и обработки результатов компьютерного эксперимента.

Пособие предназначено для обучающихся в магистратуре по программе магистерской подготовки - направление 551900 «Оптотехника», дисциплина «Методы и средства статистического моделирования ОЭС», а также для студентов, обучающихся по программе подготовки дипломированных специалистов - специальность 19.07.00 «Оптико-электронные приборы и системы» при изучении дисциплин «Компьютерные технологии проектирования ОЭС», «Специальные вопросы конструирования и технологии ОЭП»

Пособие также используется студентами факультета Вечернего и заочного обучения.

.

ВВЕДЕНИЕ

Метод статистического моделирования (иногда называемый методом Монте-Карло) широко применяют в проектировании оптико-электронных систем (ОЭС) при автоматизации проектных процедур, в экспериментальных исследованиях ОЭС с использованием компьютерных моделей, тестировании надежности, автоматизированном управлении качеством при производстве ОЭС.

Статистическое моделирование представляет собой метод получения с помощью компьютерных моделей статистических данных о процессах, происходящих в изучаемой системе.

Фактически статистическое моделирование сводится к построению для процесса функционирования исследуемой системы S некоторого моделирующего алгоритма, имитирующего функционирование и взаимодействие элементов системы с учетом случайных входных воздействий, изменений параметров элементов, влияние факторов внешней среды E с последующей компьютерной реализацией этого алгоритма.

В результате статистического моделирования системы S получается набор значений искомых величин (например, определяющих надежность проектируемой системы) или функциональных зависимостей, последующая статистическая обработка которых позволяет получить сведения о поведении проектируемого объекта или процесса в произвольные моменты времени, при изменении параметров элементов, в случае возникновения ситуаций отказа. Если количество полученных значений N достаточно велико, то полученные результаты моделирования системы приобретают статистическую устойчивость и с достаточной точностью могут быть приняты в качестве оценок искомых характеристик функционирования или качества системы S.

Можно выделить следующие основные этапы статистического моделирования:

- генерирование N реализации случайной величины Y с требуемой функцией распределения, описывающий влияющий фактор или исследуемое взаимодействие между элементами ОЭС;
- преобразование значений полученной величины, определяемой математической моделью функционирования ОЭС;
- статистическая обработка реализации и получение искомой характеристики функционирования ОЭС.

Рассмотрим реализацию указанного алгоритма статистического моделирования на примере анализа надежности ОЭС.

Модели систем будут выполняться в технологии MatLab.

1. МЕТОДИКА ПОСТРОЕНИЕ МОДЕЛЕЙ ДЛЯ РАСЧЕТА ПАРАМЕТРОВ И ХАРАКТЕРИСТИК НАДЕЖНОСТИ

1.1. Характеристики и параметры надежности невосстанавливаемых элементов и приборов

Под невосстанавливаемыми понимаются приборы и элементы, использование которых прекращается после отказа. Ремонту, восстановлению функционирования такие объекты не подлежат по техническим или экономическим причинам.

Пусть испытывается группа в составе N_0 одинаковых объектов с порядковыми номерами $i = 1...N_0$ см. рис. 1.1. Под испытуемым объектом понимается невосстанавливаемый прибор или его отдельный элемент. Для определенности далее речь пойдет об элементе, однако проведенное рассмотрение справедливо и для невосстанавливаемого прибора в целом, если целью испытаний не является анализ надежности составляющих его элементов.

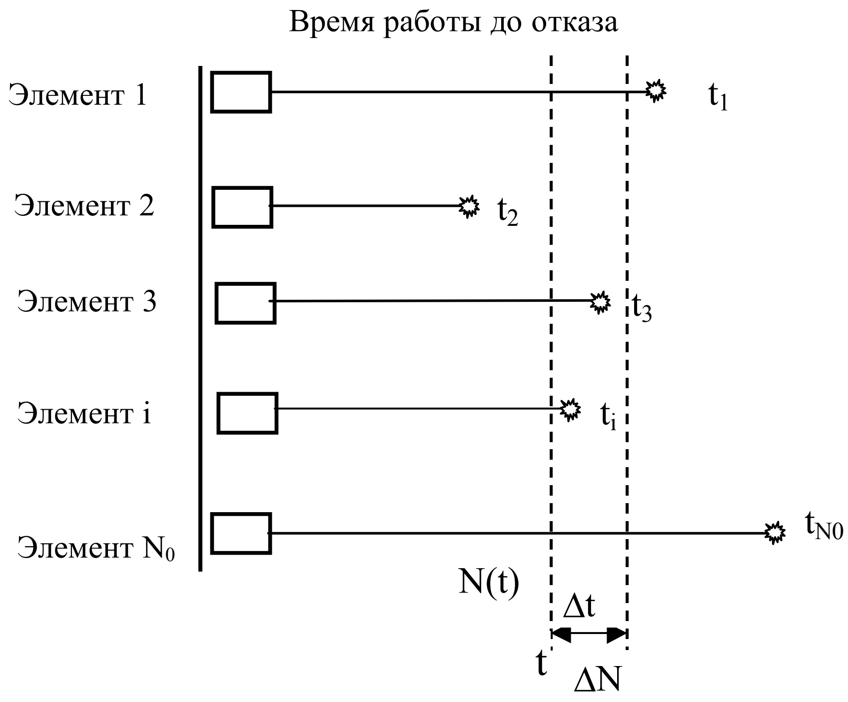

Рис. 1. 1. К расчету параметров надежности

В процессе испытаний элементы отказывают, проработав время t_i, где i - порядковый номер прибора. При достаточно большом количестве N_0 можно время работы t элементов до отказа можно рассматривать как некоторую случайную величину, принимающую значения t_i и характеризующую надежность этого прибора. Тогда характеристики случайной величины t будут определять надежность рассматриваемого элемента.

Рассмотрим функцию $f(t)$ как плотность вероятности времени работы t элемента до отказа.

Тогда вероятность отказа $s(t)$ элемента за период времени 0...t определится интегралом:

$$s(t) = \int_0^t f(\tilde{t})\, d\tilde{t} . \qquad (1.1)$$

По результатам эксперимента приближенное значение функции $s(t)$ определится как :

$$s(t) = \frac{n(t)}{N_0}, \qquad (1.2)$$

где $n(t)$ – количество элементов, отказавших за интервал времени 0...t.

Соответственно, вероятность безотказной работы $P(t)$ в течении времени t определится как:

$$P(t) = \int_t^\infty f(\tilde{t}) d\tilde{t} . \qquad (1.3)$$

По результатам эксперимента приближенное значение функции $P(t)$ определится как:

$$P(t) = \frac{N(t)}{N_0}, \qquad (1.4)$$

где $N(t) = N_0 - n(t)$ есть количество элементов из общего количества N_0, еще работающих на момент времени t – см. рис. 1.1.

Среднее время работы элемента (средняя наработка на отказ) tcp, как математическое ожидание величины t находится из выражения:

$$tcp = \int_0^\infty p(t) dt . \qquad (1.5)$$

Экспериментальное значение времени tcp находится после завершения испытаний (все элементы отказали) как усреднение величин t_i :

$$tcp = \frac{\sum_{i=1}^{N_0} t_i}{N_0}, \qquad (1.6)$$

где t_i – время работы i-го элемента до отказа.

Важной характеристикой надежности является интенсивность отказов. Интенсивностью отказов $\lambda(t)$ называют вероятность отказа элемента в единичный отрезок времени после данного момента t времени при условии, что до данного момента времени отказа не произошло.

Интенсивность отказов $\lambda(t)$ выражается через функции $p(t)$ и $f(t)$:

$$\lambda(t) = \frac{f(t)}{p(t)} . \qquad (1.7)$$

Различным этапам использования объекта соответствует определенный вид функции интенсивности отказов. Типовой вид функции интенсивности отказов изображен на рис. 1.2.

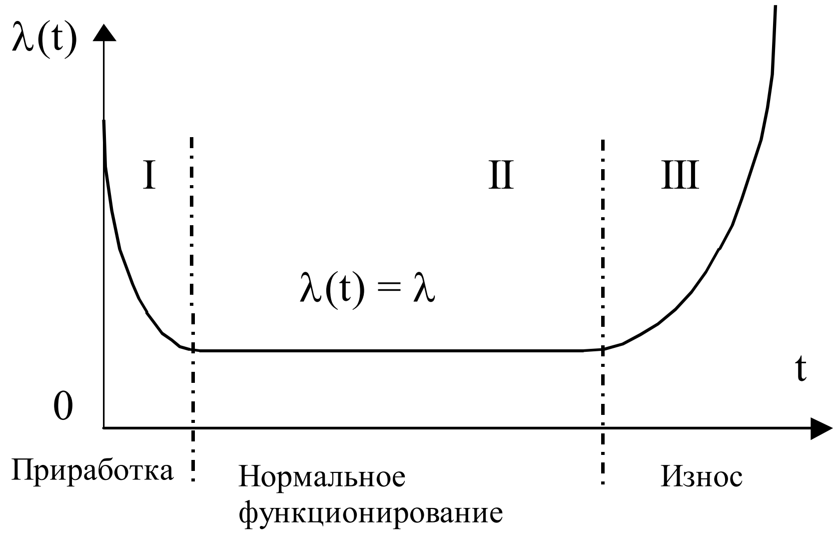

Рис. 1.2. Типовая зависимость интенсивности отказов от времени

В соответствии с рисунком 1.2., при использовании прибора (или элемента) сначала на этапе I – (приработки) интенсивность отказов уменьшается, затем, на этапе II нормального функционирования остается практически постоянной и, наконец, на этапе износа III – интенсивность отказов возрастает.

Экспериментальное значение интенсивности отказов $\lambda(t)$ в момент времени t находится по следующей методике см. рис. 1.1.

1. Выбирается требуемый момент времени t. Пусть на этот момент времени еще работает $N(t)$ элементов из общего количества N_0.
2. От момента времени t отсчитывается малый интервал времени Δt. Определяется количество элементов $N(\Delta t)$, вышедших из строя за интервал времени Δt.
3. Определяется искомое значение интенсивности отказов как:

$$\lambda(t) = \frac{N(\Delta t)}{\Delta t \cdot N(t)}. \tag{1.8}$$

Аналогичным образом по результатам рассматриваемого эксперимента может быть определена характеристика под названием частота отказов $F_0(t)$, численно равная величине плотности вероятности $f(t)$ времени работы элемента до отказа в момент времени t:

$$F_0(t) = \frac{N(\Delta t)}{\Delta t \cdot N_0}, \tag{1.9}$$

где N_0 – общее количество испытуемых элементов.

1.2. Экспериментальное определение параметров и характеристик невосстанавливаемых систем

Пусть объектом испытаний является система из нескольких приборов или отдельный прибор, рассматриваемый как совокупность компонентов. Для общности рассмотрения будем рассматривать объект испытаний как систему, состоящую из N отдельных элементов.

В зависимости от характера влияния отказа отдельного элемента на время работы системы различают последовательное, параллельное и смешанное соединение элементов.

При последовательном соединении N элементов (см. рис. 1.3.) отказ любого i - того из них приводит к отказу всей системы.

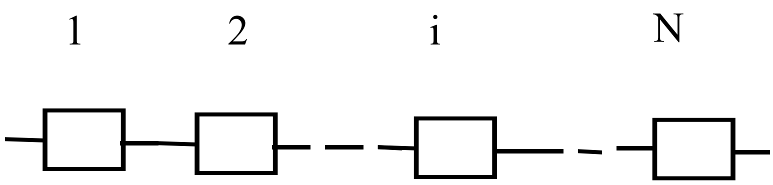

Рис. 1.3. Последовательное соединение элементов

Если рассматривать время работы каждого элемента системы до его отказа как элемент некоторого массива t_i (где i = 1...N), то для однократного эксперимента время работы всей системы при последовательном соединении определится как

$$t_{line} = \min \{t_1, t_2, ... t_i, ... t_N\}. \qquad (1.10)$$

Вероятность безотказной работы и интенсивность отказов системы при последовательном соединении определяется как:

$$P_{line}(t) = P_1(t)\cdot P_2(t)\cdot .. \cdot P_i(t)\cdot ... \cdot P_N(t), \qquad \cdot(1.11)$$

$$\lambda_{line}(t) = \lambda_1(t) + \lambda_2(t) + .. + .. \lambda_i(t) + .. + \lambda_N(t), \qquad (1.12)$$

где $P_i(t)$, $\lambda_i(t)$– вероятности безотказной работы и интенсивности отказов отдельных элементов.

При параллельном соединении N элементов (см. рис. 1.4.) при отказе любого из них функционирование системы будет обеспечиваться другими, еще работающими.

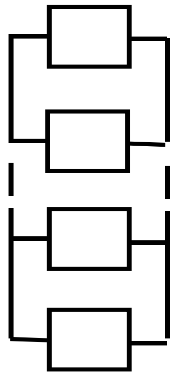

Рис. 1.4. Параллельное соединение элементов

Таким образом, отказ системы произойдет только после отказа последнего работающего элемента.

Если рассматривать время работы каждого элемента системы до его отказа как элемент некоторого массива t_i (где i = 1...N), то время работы всей системы для однократного эксперимента при параллельном соединении будет равно наибольшему времени работы из всех элементов:

$$t_{col} = \max \left\{ t_1, t_2, \ldots t_i, \ldots t_N \right\}. \tag{1.13}$$

Вероятность безотказной работы системы с параллельным соединением элементов определяется выражением:

$$P_{col}(t) = 1 - \prod_{i=1}^{N} (1 - P_i(t)), \tag{1.14}$$

где $P_i(t)$ вероятности безотказной работы отдельных элементов.

При параллельном соединении надежность системы превышает надежность отдельных элементов. Это свойство используется для увеличения надежности отдельных элементов или всей системы в целом путем резервирования – включением параллельно ненадежному элементу точно таких же – резервных.

Из проведенного рассмотрения следует, что при экспериментальном исследовании систему, состоящую из отдельных элементов можно представить как эквивалентный единый компонент с некоторым временем работы до отказа t_{line} или t_{col} в каждом опыте в соответствии с выражениями (1.10),(1.13).

В случае смешанного соединения элементов (см. рис. 1.5), действуя по указанной методике, фрагменты с последовательным соединением можно представить как эквивалентные компоненты с временем работы t_{line_j} где j = 1...N_{line}– количество таких фрагментов (пример на рис. 1.6.)

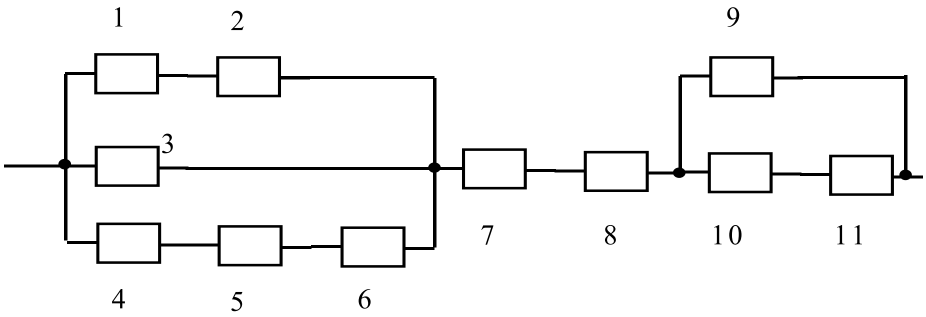

Рис. 1.5. Исходная схема

Соответственно, фрагменты с параллельным соединением заменяются на эквивалентные элементы с временем работы t_{col_k} , где k =1...N_{col} количество таких фрагментов. Пример приведен на рис. 1.7.

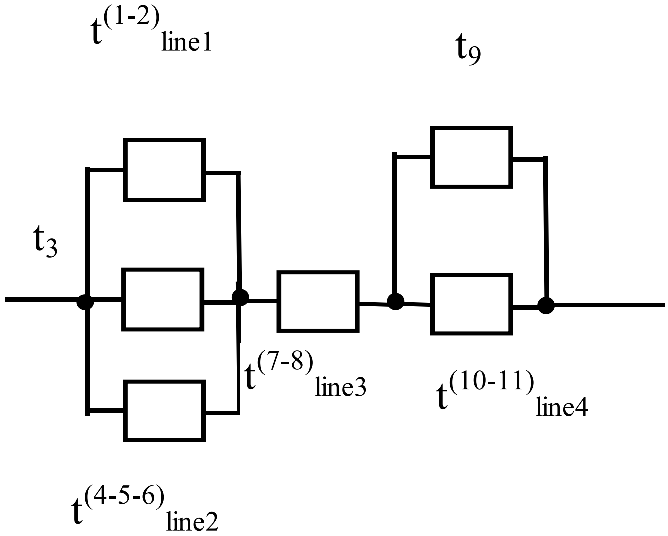

Рис. 1.6. Эквивалентная схема (этап 1)

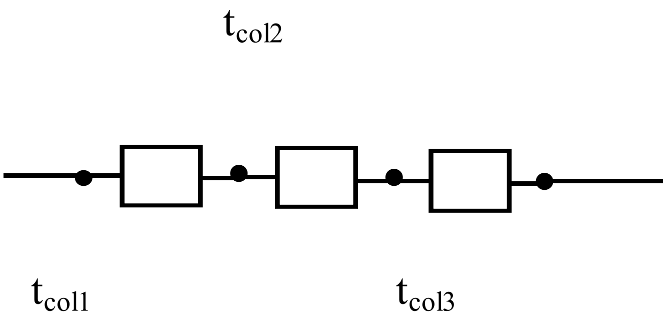

Рис. 1.7. Эквивалентная схема (этап 2)

Далее, в зависимости от соединения полученных эквивалентных элементов они аналогичным образом заменяются на единый вторичный эквивалентный элемент с временем работы до отказа t'_{row} или t'_{col} в каждом опыте.

1.3. Расчет надежности типологически сложных систем

Во многих практических случаях анализа надежности система не может быть представлена как совокупность фрагментов с последовательным или параллельным соединением элементов (см. рис. 1.8). В таких случаях говорят о топологической сложности системы с точки зрения соединения по надежности.

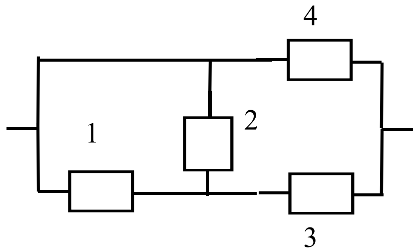

Рис. 1.8. Пример типологически сложного соединения элементов при анализе надежности

Практический пример – система видео- наблюдения, в которой одни телекамеры имеют с некоторыми другими перекрывающиеся поля зрения, а с некоторыми – нет.

Теоретический анализ надежности таких систем сложен и для практических расчетов используются приближенные методики.

В соответствии с методикой, типологически сложная система заменяется на эквивалентные фрагменты с последовательным и параллельным соединением. Далее параметры надежности системы определяются по рассмотренным ранее выражениям.

Эквивалентная замена выполняется методами путей и сечений, базирующихся на понятиях "минимальный путь " и "минимальное сечение".

Минимальным путем называется совокупность работоспособных элементов, которая достаточна для работоспособности системы, причем никакое подмножество этой совокупности таким свойством не обладает.

При этом система в целом функционирует только тогда, когда в ней имеется хотя бы один работоспособный минимальный путь. Минимальный путь работоспособен, если не отказал ни один элемент, ему принадлежащий.

Пусть система имеет R минимальных путей. Каждый путь с номером i (где $i = 1...R$) содержит r_i элементов. В соответствии со свойствами минимальных путей и составленной из них системы, каждый минимальный путь i представляется в виде последовательного соединения r_i элементов с номером j (где $j = 1... r_i$). Сами же минимальные пути в соответствии со своими свойствами, могут рассматриваться как R последовательных цепочек, соединенных параллельно.

В рассматриваемом примере $R = 3$. Минимальный путь R_1 включает элемент 4, соответственно, количество элементов на этом пути $r_1 = 1$, R_2 включает элементы 1,3 и, соответственно, $r_2 = 2$, R_3 – элементы 2,3 и $r_3 = 2$. Заметим, что путь в составе 2,3,4 не является минимальным, поскольку содержит путь в составе элемента 4.

Эквивалентная схема системы принимает вид – см. рис. 1.9.

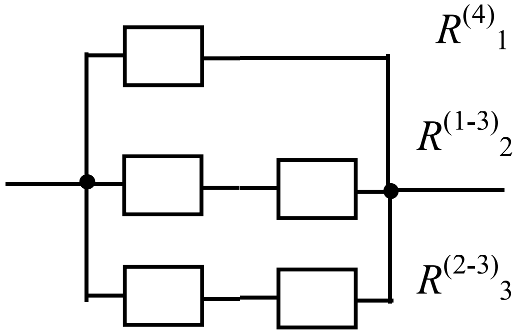

Рис. 1.9. Эквивалентная схема системы по минимальным путям

Вероятность безотказной работы системы на интервале времени от 0 до t определяется выражением:

$$P_r(t) = 1 - \prod_{i=1}^{R} (1 - \prod_{j=1}^{r_i} P_j(t)), \qquad (1.15)$$

где $P_j(t)$ – вероятность безотказной работы j –го элемента на минимальном пути r_i, r_i – также обозначает количество элементов на этом пути.

Минимальным сечением называется совокупность элементов, отказ которых приводит к отказу всей системы, причем никакое подмножество этой совокупности таким свойством не обладает.

Следовательно, система отказывает при отказе ее любого минимального сечения. Минимальное же сечение работоспособно до тех пор, пока на нем не отказал последний элемент, ему принадлежащий.

Пусть система имеет K минимальных сечений. Каждое сечение с номером i (где $i = 1... K$) содержит k_i элементов. В соответствии со свойствами минимальных сечений и составленной из них системы, каждое минимальное сечение i представляется в виде параллельного соединения k_i элементов с номером j (где $j = 1... k_i$). Сами же минимальные сечения в соответствии со своими свойствами, могут рассматриваться как K эквивалентных элементов, соединенных последовательно.

В рассматриваемом примере $K = 2$. Минимальное сечение K_1 включает элементы 1,4,2 соответственно, количество элементов на этом пути $k_1 = 3$, K_2 включает элементы 3,4 и, соответственно, $k_2 = 2$. Сечение в составе 2,3,4 не является минимальным, поскольку содержит сечение в составе элементов 3,4.

Эквивалентная схема системы принимает вид – см. рис. 1.10.

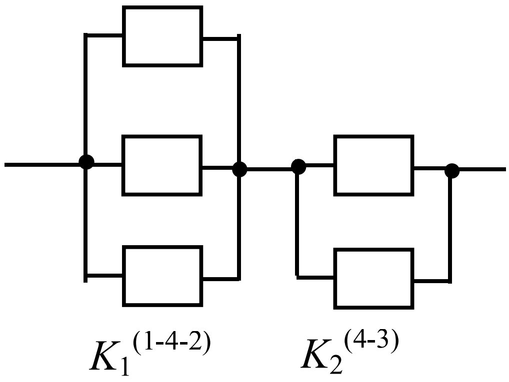

Рис. 1.10. Эквивалентная схема системы по минимальным сечениям

Выражение для вероятности безотказной работы системы в этом случае равно

$$P_K(t) = \prod_{i=1}^{K}(1 - \prod_{j=1}^{k_i}(1 - P_j(t)). \qquad (1.16).$$

Рассмотренный метод позволяет получить две различные оценки характеристики надежности, в данном случае– вероятности безотказной работы. Теоретический анализ показывает, что величина параметра, полученная через минимальные пути дает несколько завышенную, а через минимальные сечения - заниженную оценку надежности системы.

Соответственно, для рассматриваемой вероятности безотказной работы истинная оценка находится в интервале [P_K, P_R].

Как правило, величина получаемого интервала невелика и рассмотренный метод дает достаточную для практики точность оценки вероятности безотказной работы системы.

1.4. Характеристики и параметры надежности восстанавливаемых элементов и приборов

Под восстанавливаемыми понимаются приборы и элементы, обладающие свойством восстановления своей работоспособности после отказа. Для восстанавливаемых объектов характерно неоднократное повторение ситуации отказ-восстановление за весь период их использования.

Восстанавливаемые объекты могут реализовывать мгновенное и немгновенное восстановление.

1.4.1. Параметры надежности объектов с мгновенным восстановлением.

Рассмотрим длительный интервал времени T использования восстанавливаемого объекта – прибора или его элемента. Для определенности, далее речь пойдет об элементе.

При достаточно большом интервале времени T будет иметь место N ситуаций отказ- мгновенное восстановление, причем время работы t_i между (i-1) и i -тым отказами будет отличаться для различных i и может рассматриваться как значение некоторой случайной величина t с функцией плотности вероятности $f(t)$ – см. рис. 1.11.

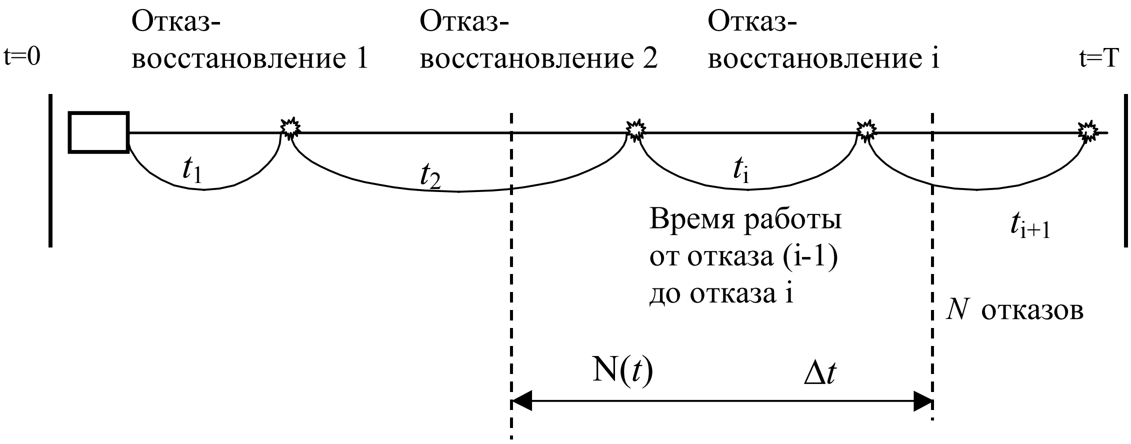

Рис. 1.11. Циклы работы системы с мгновенным восстановлением.

Надежность между отказами определяется ранее рассмотренными параметрами для невосстанавливаемых элементов, в частности, средним временем безотказной работы tcp.

На достаточно больших временных промежутках надежность восстанавливаемой системы определяется так называемым параметром потока отказов ω(t).

Параметр потока отказов определяется как отношение количества отказов $N(t)$ восстанавливаемого объекта за интервал времени Δt к величине этого интервала:

$$\omega(t) = N(t)/\Delta t\ ; \qquad tcp < \Delta t << T. \tag{1.17}$$

Если параметр потока отказов $\omega(t)$ определялся в процессе испытаний сразу нескольких объектов N_0, то полученное количество отказов $N(t)$ нормируется на эту величину и указанное выражение принимает вид:

$$\omega(t) = N(t)/N_0\Delta t\ ;\ tcp << \Delta t << T. \tag{1.18}$$

При определении потока отказов интервал Δt выбирается малым относительно временного интервала T использования объекта, но много больше среднего времени работы между отказами tcp .

1.4.2. Параметры надежности систем с не мгновенным восстановлением.

Рассмотрим длительный интервал времени T использования восстанавливаемого объекта – прибора или его элемента. Для определенности, далее речь пойдет об элементе.

При реализации не мгновенного восстановления после очередного отказа элемента восстановление его нормального функционирования выполняется за некоторое время *tr*.

При достаточно большом интервале времени T будет иметь место N ситуаций отказ-восстановление, причем время работы t_i между (i-1) восстановлением и i – ым отказом будет отличаться для различных i и может рассматриваться как значение некоторой случайной величина t с функцией плотности вероятности $f(t)$ – см. рис. 1.12. Соответственно, время восстановления элемента tr_i между i- тым отказом и i- тым восстановлением может рассматриваться как значение случайной величины tr с плотностью вероятности $g(tr)$.

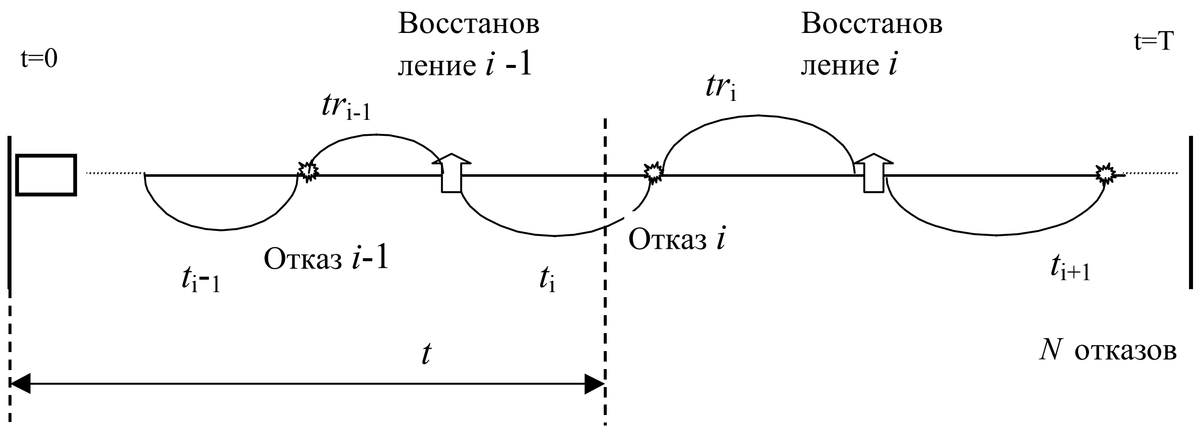

Рис. 1.12. Циклы работы системы с немгновенным восстановлением

Для величины tr, описывающей время восстановления элемента могут быть аналитически и экспериментально получены характеристики надежности: вероятность $Pr(tr)$ восстановления за время tr, интенсивность восстановления $\lambda r(tr)$, $trcp$ - среднее время восстановления. Аналитическое выражение этих характеристик аналогично (1.3)....(1.9) с заменой аргумента t на tr.

Надежность систем с немгновенным восстановлением описывается и другими характеристиками, в частности, функцией готовности $Kr(t)$ – вероятностью того, что система работоспособна (то есть – не в процессе восстановления) в момент времени t, отсчитываемый от начала функционирования (t=0). Можно показать, что при увеличении t функция готовности стремится к пределу Kr, называемому коэффициентом готовности:

$$Kr = \lim_{t \to \infty} (Kr(t)) . \qquad (1.19)$$

При таком определении, коэффициент готовности равен вероятности того, что система работоспособна в произвольный момент времени (не зависимо от начала отсчета временного интервала).

$$Kr(N) = \frac{\sum_{i=1}^{N} t_i}{\sum_{i=1}^{N} t_i + \sum_{i-1}^{N} tr_i} \ . \tag{1.20}$$

Если при экспериментах с элементом зафиксировано N ситуаций отказ-восстановление, то приближенная оценка величины коэффициента готовности $Kr(N)$ определяется из выражения (1.20), где t_i и tr_i, соответственно время работы до и восстановления после отказа с номером i. При увеличении N, величина $Kr(N)$ стремится к величине коэффициента готовности Kr, а ее текущие значения $Kr(N)$ могут трактоваться как значения функции готовности $Kr(t)$ в моменты времени, соответствующие N - той ситуации отказ-восстановление.

1.5. Основные законы распределения времени безотказной работы

В силу того, что процесс возникновения отказов в приборах и элементах носит случайный характер и зависит от многих факторов, для описания распределения случайной величины t - времени работы до отказа используется ряд законов – Вейбулла, экспоненциальный, Релея, нормальный, Пуассона и ряд других.

При моделировании отказов и определении параметров надежности с помощью компьютерных моделей необходимо синтезировать значения случайных величин с этими законами распределения.

Пусть $F(x)$ – функция распределения некоторой случайной величины Ψ. Синтезировать эту случайную величину – значит сформировать последовательность ее значений ψ_i, обладающих свойством: вероятность того, что значение ψ_i будет меньше некоторого значения x, равна $F(x)$, или –

$$P(\psi_i < x) = F(x) \ . \tag{1.21}$$

Известен ряд методов синтеза значений случайной величины.

Рассмотрим общий точный метод «обратной функции», используемый для моделирования случайных величин с неограниченными интервалами изменения значений.

В частности, в соответствии с методом «обратной функции» в случае, если функции распределения $y = F(x)$ имеет обратную функцию вида $x = G(y)$, то значение ψ_i случайной величины синтезируется в результате вычисления выражения:

$$\psi_i = G(\gamma_i), \tag{1.22}$$

где γ_i – значение случайной величины с равномерным распределением в интервале от 0 до 1, подставляемое в это выражение из генератора значений равномерно распределенной случайной величины .

Рассмотрим методику моделирования случайных величин с типовыми законами распределения.

1.5.1. Распределение Релея.

Это распределение достаточно хорошо описывает поведение оптоэлектронных элементов с явно выраженным эффектом старения и износа (завершающий участок области III износа - см. рис. 1.2.). Функция распределения *F(t)* и среднее время безотказной работы *tcp* определяются выражениями :

$$F(t) = 1 - e^{\frac{-t^2}{2 \cdot c^2}} . \qquad (1.23) \qquad\qquad tcp = C\sqrt{\frac{\pi}{2}} , \qquad (1.24)$$

где С – параметр распределения.

По методу обратной функции, моделирование значения t_i времени работы до отказа выполняется по выражению :

$$t_i = \sqrt{-2 * C^2 * \ln(1-\gamma)} , \qquad (1.25)$$

где С определяется по заданному среднему времени безотказной работы *tcp* для данного элемента из выражения (1.24).

1.5.2. Нормальное распределение.

Нормальный закон определяет надежность объектов, для элементов которых характерно наличие некоторого износа с малым разбросом величины износа, причем возможные отказы однородны по качеству. Например, нормальным законом достоверно описывается надежность осветительных систем. Также, время восстановления объекта во многих случаях описывается нормальным законом.

Плотность вероятности времени работы *t* до отказа определяется выражением:

$$f(t) = \frac{1}{\sigma \cdot \sqrt{2 \cdot \pi}} \cdot e^{\{\frac{-(t-M)^2}{2 \cdot \sigma^2}\}} . \qquad (1.26)$$

Основные параметры нормального распределения времени работы *t* элемента – среднее значение *M* = *tcp* и среднее квадратическое отклонение σ. Формально, нормальное распределение определяет вероятность появления и отрицательной величины времени работы *t*. Для достижения адекватного описания надежности с помощью нормального распределения необходимо выполнение условия $\sigma \ll tcp$.

Для синтеза значений нормально распределенной случайной величины используются специальные программные генераторы, формирующие значения так называемой нормированной нормальной величины Xn с нулевым средним значением и единичным средним квадратическим отклонением:

$$M_n = 0, \qquad \sigma_n = 1. \tag{1.27}$$

Моделирование значения t_i времени работы до отказа с использованием нормированной величины Xn при нормальном распределении выполняется по выражению:

$$t_i = tcp + Xn \cdot \sigma , \tag{1.28}$$

где tcp – среднее время безотказной работы элемента, σ – среднее квадратическое отклонение времени t работы элемента до отказа.

1.5.3. Распределение Вейбулла.

Этому распределению достаточно хорошо подчиняется время работы t до отказа в объектах, содержащих большое количество электронных и оптико-электронных элементов, микромодулей, полупроводниковых компонентов.

Функция распределения $F(t)$ времени работы t определяется выражением:

$$F(t) = 1 - e^{-\lambda * t^b} , \tag{1.29}$$

где λ и b - параметры распределения, $\Gamma(1+ 1/b)$ – табулированная полная гамма – функция.

Распределение Вейбулла общего вида описывает надежность объекта на всех трех этапах его функционирования – см. рис. 1.2: на этапе I при $b < 1$, на этапе II при $b = 1$, на этапе III при $b > 1$.

По методу обратной функции, моделирование значения t_i времени работы до отказа выполняется по выражению :

$$t_i = \sqrt[b]{-\frac{1}{\lambda} * \ln(1-\gamma)} , \tag{1.30}$$

где параметр λ определяется из выражения по известной величине tcp как:

$$\lambda = \left(\frac{\Gamma(1+\frac{1}{b})}{tcp} \right)^b . \tag{1.31}$$

Для трех типовых значений $b = 0{,}5$; $b = 1$; b = 1,5 значение функции соответственно равны $\Gamma(3) = 2$; $\Gamma(1) = 1$; $\Gamma(1{,}5) = 0.89$.

1.5.4. Экспоненциальное распределение

Фактически является распределением Вейбулла - см. 1.5.3 при значении параметра $b = 1$. В практических случаях очень часто описывает надежность объектов, что и явилось причиной его отдельного рассмотрения.

Параметры распределения величины t времени работы до отказа :

$$P(t) = e^{-\lambda * t} . \qquad (1.32) \qquad f(t) = \lambda \cdot e^{-\lambda \cdot t} . \qquad (1.33)$$

$$\lambda(t) = \lambda . \qquad (1.34) \qquad tcp = \frac{1}{\lambda} , \qquad (1.35),$$

где λ – параметр распределения, численно равный интенсивности отказов.

Экспоненциальное распределение описывает надежность объекта на этапе II нормальной эксплуатации - см. рис. 1.2 .

Также для систем с немгновенным восстановлением распределение времени tr восстановления распределено экспоненциально при автоматическом поиске и замене отказавшего модуля.

По методу обратной функции, моделирование значения t_i времени работы до отказа (или времени восстановления tr выполняется по выражению :

$$t = \left(\frac{-1}{\lambda} \right) * \ln(1 - \gamma) , \qquad (1.36)$$

где параметр λ и среднее время безотказной работы tcp связаны выражением (1.35).

1.5.5. Логарифмически-нормальное распределение

Логарифмически-нормальное (или логнормальное) распределение определяет надежность объектов при износе в случае лавинообразного характера отказов (по принципу «домино»). В этом случае причины, вызывающие отказ действуют не аддитивно как при других распределениях, а мультипликативно. В результате интенсивность отказов зависит от колебаний влияющих факторов. Функционирование элементов носит неустойчивый характер – наряду с «долгоживущими» элементами встречаются случаи быстрых отказов.

Основные параметры логнормального распределения для времени работы t элемента – математическое ожидание M и среднее квадратическое отклонение σ связаны со средним временем безотказной работы соотношением:

$$tcp = e^{\left(\frac{\sigma^2}{2} + M \right)} . \qquad (1.37)$$

Для синтеза значений логнормально распределенной случайной величины также используется нормированная нормальная случайная величина Xn с нулевым средним значением и единичным средним квадратическим отклонением в соответствии с выражением (1.27).

Моделирование значения t_i времени работы до отказа с использованием нормированной величины Xn при логнормальном распределении выполняется по выражению:

$$t_i = e^{(\sigma \cdot Xn + M)} \qquad (1.38)$$

1.5.6. Равномерное распределение

Равномерное распределение может использоваться для грубого, приближенного описания надежности элементов.

В частности, при поиске неисправности автоматическими системами в ряде случаев время t восстановления может полагаться равномерно распределенным в интервале [t_{min}, t_{max}]:

$$t_{min} < t < t_{max} \,. \tag{1.37}$$

Плотность вероятности времени t работы до отказа (или восстановления) постоянна в указанном интервале – см. рис. 1.13.

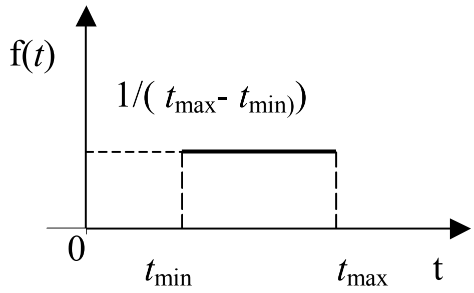

Рис. 1.13. Плотность вероятности при равномерном распределении

Среднее значение tcp (математическое ожидание М) и среднее квадратическое отклонение σ времени t определяются выражениями:

$$tcp = \frac{t_{max} + t_{min}}{2} \,. \tag{1.38}$$

$$\sigma^2 = \frac{(t_{max} - t_{min})^2}{12} \,. \tag{1.39}$$

Моделирование случайной величины t выполняется по выражению

$$t = t_{min} + (t_{max} - t_{min}) \cdot \gamma \,, \tag{1.40}$$

где γ – значение простой случайной величины с равномерным распределением в интервале [0,1].

1.5.7. Распределение Эрланга

Распределением Эрланга описывается время t восстановления системы в случае поиска и устранения причины отказа оператором. Функция распределения *F(t)* и плотность вероятности *f(t)* времени восстановления t определяется выражениями:

$$F(t) = 1 - \left(1 + \frac{2 \cdot t}{tr}\right) \cdot e^{\left(\frac{-2 \cdot t}{tr}\right)}, \tag{1.41}$$

$$f(t) = \frac{4 \cdot t}{tr^2} \cdot e^{\left(\frac{-2 \cdot t}{tr}\right)}, \tag{1.42}$$

где – *tr* среднее время восстановления.

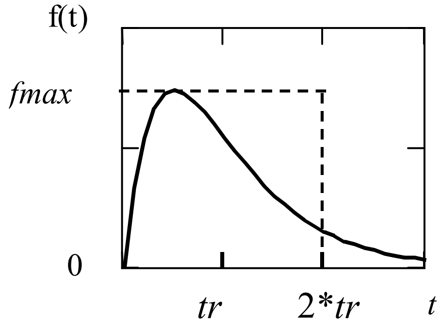

Рис. 1.14. Плотность вероятности распределения времени tr восстановления по Эрлангу

Для функций вида (1.41) отсутствует обратная функция и, следовательно, для их моделирования неприменим метод «обратной функции» и алгоритм (1.22) этого метода.

В этом случае может использоваться общий приближенный метод синтеза значений случайной величины, например, метод Неймана.

Для реализации метода Неймана необходимо, чтобы значения синтезируемой случайной величины находились на ограниченном интервале.

В случае бесконечного распределения следует искусственно определить интервал определения времени *t*. В соответствии с графиком, целесообразно моделирование случайной величины выполнять на интервале $[0,2\cdot tr]$.

Алгоритм моделирования значения случайной величины по методу Неймана имеет циклический характер и включает следующие шаги.

1. Некоторой переменной-ключу, сигнализирующей о том, что значение случайной величины создано, присваивается значение «величина не создана» (например, `flag = 0`).
2. Формируется «цикл по условию» (тело цикла перестает выполняться когда переменная-ключ примет значение «величина создана», например, `flag = 1`).
3. Из программного генератора извлекается значение исходной случайной величины $\gamma 1$ с равномерным распределением в интервале [0, 1].
4. Из программного генератора извлекается другое значение исходной случайной величины $\gamma 2$ с равномерным распределением в интервале [0, 1].
5. Формируются две новых случайных величины с равномерным распределением в интервале $[0, 2\cdot tr]$ и в интервале $[0, f_{\max}]$:

$$\Gamma 1 = 2 * tr * \gamma 1; \qquad \Gamma 2 = f_{max} * \gamma 2, \tag{1.43}$$

где f_{max} - максимальное значение функции $f(t)$.

6. Проверяется условие :

$$Г2 < f(Г1), \tag{1.44}$$

где f(Г1) – значение функции плотности вероятности при t = Г1.

7. Если условие (1.44) выполняется, случайная величина с распределением Эрланга создается как:

$$t = Г1 \tag{1.45}$$

и переменной-ключу присваивается величина «значение создано» (`flag = 1`) что приводит к остановке цикла формирования значения случайной величины.

8. Если условие (1.44) не выполняется, значение переменной-ключа не изменяется и тело цикла – шаги 3…6 алгоритма выполняются вновь.

2. СИСТЕМА MATLAB КАК БАЗОВАЯ ТЕХНОЛОГИЯ СОЗДАНИЯ МОДЕЛЕЙ /4/

2.1. Основные сведения о MATLAB /4/

MatLab (сокращенное **Mat**rix **Lab**oratory) фирмы Math Works Inc. является интерактивной программной средой для численного решения широкого круга научных и инженерных задач.

Базовым расчетным элементом (основной переменной) является матрица, под которой понимается массив как нумерованная одномерная (строка, столбец) или двумерная (собственн28о, матрица) последовательность данных, имеющих единое имя. Скалярные величины рассматриваются, как матрицы размерностью [одна строка, один столбец].

Сотни встроенных функций и функций в сопутствующих пакетах (Tool Box) позволяют решать сложные задачи при минимальном использовании традиционного программирования.

Важным достоинством MatLab является развитая графическая подсистема, позволяющая с помощью простых команд выводить на экран двумерные и трехмерные графики и диаграммы.

2.2. Данные в MATLAB

2.2.1. Создание матрицы как совокупности данных

Как уже отмечалось базисным элементом данных в системе Matlab является прямоугольная матрица, частным случаем которой могут быть одномерный вектор или скаляр, который может рассматриваться как матрица размерностью 1∗1. Универсальность матрицы как главной переменной, над которой выполняются операции системы, определяет разнообразие способов ее задания.

2.2.2. Непосредственное задание матрицы.

При непосредственном задании матрицы используется оператор присваивания, в левой части которого указывается идентификатор (имя) матрицы (должно начинаться с буквы), а справа – в квадратных скобках через пробел указываются элементы матрицы по строкам, строки разделяются символом ";" (см. матрицу `A`). Возможно построчное задание матрицы без использования символа ";" при расположении строк в столбец (см. матрицу `a`):

```
A = [1 2 3; 4 5 6; 7 8 9]
```

```
a = [1 2 3
     4 5 6
     7 8 9]
```

Необходимо отметить, что в MatLab регистр символа является значимым – матрицы A и a – различные объекты.

Аналогично задаются одномерные матрица – строка B = [1 2 3] или - столбец b = [1;4;7] .

В общем случае элементами матрицы при ее вводе могут быть числа, выражения, функции, но выводятся посчитанные численные значения элементов.

2.2.3. Задание матрицы-строки в форме перечисления

Для задания матрицы – строки (другой термин - вектор) при условии, что ее элементы отличаются от соседних на постоянную величину (шаг) может использоваться форма "перечисление":

X=<нач.знач.>:<шаг>:<конеч.знач.>

где X – имя матрицы (вектора), <шаг> - приращение , которое может иметь и отрицательные значения, а также определяться в общем случае вычисляемым выражением.

Если составляющая <шаг> отсутствует, то его значение предполагается равным единице.

Так, перечисление X = 1:5 равносильно X=[1 2 3 4 5].

Еще одна форма генерации вектора определяется функцией linspace:

X = linspace (<нач.знач.>,<конеч.знач.>,<число элементов>)

Например: X=linspace(-pi,pi,4)

дает вектор X= [-3.1416 -1.0472 1.0472 3.1416],

где pi – встроенная переменная, равная числу π.

2.2.4. Монтаж двумерной матрицы из одномерных

При наличии одномерных матриц одинакового размера из них может быть смонтирована двумерная матрица. Например, из матриц - строк ts1 = [1 2 3] , tv1 = [4 5 6] , tp1 = [7 8 9] монтируется матрица A (см. ранее):

A = [ts1;tv1;tp1]

2.2.5. Вычленение фрагмента матрицы

Из любой матрицы размером более двух элементов может быть получена матрица, содержащая прямоугольный фрагмент исходной. Количество и номера вычленяемых элементов задаются как индексные выражения в форме перечисления. Например, C = a(1:2 ; 2:3) вычленяет фрагмент матрицы a :

```
C = [2 3
     5 6]
```

Специальная форма записи `D = a(: ; 2:3)` вычленяет фрагмент, содержащий все строки исходной матрицы, в каждой из которых содержатся элементы столбцов 2 и 3.

2.2.6. Генерация матриц специальными подпрограммами-функциями

Генерация часто используемых специальных матриц может выполняться встроенными подпрограммами-функциями вида Fname(N,M,), аргументом которых являются количество строк N и столбцов M создаваемой матрицы. Например, матрица Z с двумя строками и пятью столбцами, все элементы которой равны нулю, создается как `Z = zeros(2,5)`, а матрица с единичными элементами – как `S = ones(2,5)`.

Можно в качестве аргумента функции указать только одну величину. В этом случае генерируется квадратная матрица. Матрица - строка и матрица-столбец с соответствующими элементами генерируются с указанием двух аргументов: `Z = zeros(1,4)`, `S = ones(7,1)`.

2.3. Элементарные операции с матрицами

2.3.1. Алгебраические операции

Приняты следующие символы алгебраических операций:

"+" – сложение, "-" – вычитание, "*" – умножение, "/" – правое деление, "\" – левое деление, "^" - степень. "Левое деление", фактически означает, что при C=A\B справедливо соотношение B=C*A.

Так как основной конструкцией системы является матрица, то перечисленные операции +,-,*,/,\,^ являются "матричными" и осуществляются по правилам линейной алгебры. Например, если
`A = [1 2 3]` является вектором, а `B = [4;5;6]` является столбцом, то выражение `C = A*B` определяет в результате матричного умножения одноэлементную матрицу `C = 32`.

Но MatLab дает возможность поэлементного выполнения операций, когда операции выполняются не над матрицами по правилам матричной алгебры, а между соответствующими их элементами по обычным правилам.

Признаком таких операций является наличие десятичной точки "." перед знаком операции. Пусть введены два вектора:

`x = [1 2 3]` и `y = [4 5 6]`

Тогда при операции `z = x.*y` будет получено `z = [4 10 18]` ; в результате `z = y./x` получено `z = [4.0000 2.5000 2.00000]`, а для `z = x.^y` соответственно, `z = [1 32 729]`.

Обязательным условием корректности операций поэлементного сложения, вычитания и перемножения является одинаковая размерность матриц.

Сложение и вычитание скаляра из матрицы выполняется по правилам суммирования или вычитания.

Например, `Z = F-2` при `F = [2 3 4]` даст `Z = [0 1 2]`.

При умножении (делении) матрицы на скаляр может использоваться знак "матричного" умножения. В результате все элементы матрицы будут умножены (или разделены) на это число.

Например, `Z = F*7` при `F = [2 3 4]` даст `Z = [14 21 28]`.

2.3.2. Встроенные математические функции.

Применение функции к матрице дает матрицу, каждый элемент которой есть результат применения функции к каждому элементу исходной матрицы. Системы содержит большое число встроенных функций, наиболее употребительными при моделировании являются: abs - абсолютное значение; sqrt - корень квадратный; round - округление до ближайшего целого; rem - остаток от целочисленного деления; тригонометрические функции (sin, cos, tan – тангенс, asin – арксинус, acos - арккосинус, atan - арктангенс), exp – экспонента, log - натуральный логарифм; log10 - десятичный логарифм; bessel - функция Бесселя; gamma - Гамма функция.

Например, для матрицы `b=[1 2 3 4]` применение функции `B = sin(b)` дает массив `B = [0.8415 0.9093 0.1411 -0.7568]`, элементы которого представляют собой значения синусов элементов исходного массива (как от углов в радианах)

2.3.3. Некоторые специальные операции с матрицами

Операция транспонирования матрицы обозначается символом " ' " (апостроф). Например, `B=A'` определяет матрицу B, транспонированную относительно исходной A.

Определитель матрицы A вычисляется как det(A) .

Количество элементов одномерной A матрицы находится с помощью функции length(A). Например, для `A = [2 6 1 9 4 ]`, использование функции `N = length(A)` дает `N = 5`.

Количество строк и столбцов двумерной матрицы B находится с помощью функции size(B). Результатом `C = size(B)` будет матрица C из двух элементов: первый элемент равен количеству строк, а второй – количеству столбцов матрицы B.

2.4. Графические средства (полный перечень см. в /4/)

MatLab предоставляет широкие возможности графического отображения данных как в виде плоских кривых, так и поверхностей в трехмерном пространстве. Рассмотрим простейшие графические команды.

Графическое отображение одномерного массива (вектора) Y выполняется командой:

plot (Y, '<условное обозначение типа линий и цвета>')

Будет построен график зависимости значений элементов массива как функции их порядкового номера.

Если для функции y=f(x) значения аргумента содержатся в массиве X, а функции – в массиве функции Y, то график строится командой:

plot (X,Y,'<условное обозначение типа линий и цвета>')

Возможно построение нескольких графиков в одних осях:

plot (X1,Y1,'< тип линий и цвета 1>' X2,Y2,'< тип линий и цвета 2>',...)

Условное обозначение типа линий и цвета (параметр может отсутствовать):

–цвет — r - красный (по умолчанию), g - зеленый, b - синий, w - белый;

– тип линии (по умолчанию – сплошная), пунктир –, точками :, штрих-пунктир _ ;

– тип знакоместа (альтернатива типу линии) — символом *, символом o , символом +, символом x .

Например `plot (x,y,'og')` - построение линии зеленым "кружком".

Команды для вывода заголовка графика: title ('<текст>'), для вывода текстовых обозначений для координатных осей: xlabel ('текст'), ylabel ('текст').

Чтобы просмотреть график необходимо приостановить вычисления. Для этого используется команда :

```
pause
```

Вычисления продолжаются после нажатия любой клавиши.

Графическое отображение двумерного массива Z выполняется командой :

```
mesh(Z)
```

Будет построена поверхность в трехмерном пространстве как зависимость значений элементов массива (является аппликатой) от двумерной координаты (номер строки, номер столбца) этого элемента (номер строки является абсциссой, а номер столбца – ординатой).

Имеется возможность изменить соотношение размеров фигуры по координатным осям и угол обзора. Для этого используется функция:

```
mesh(Z,S,Q)
```

Здесь S = [sx sy sz] ; значения sx, sy, sz определяют соотношения размеров по координатным осям. Например, задав `S=[1 1 2]`, получим увеличение в два раза масштаба по оси z по сравнению с масштабом по осям x,y.

Массив Q = [az el] содержит два элемента az и el – углы в градусах. Положительное значение az определяет угол поворота фигуры против часовой стрелки относительно вертикальной оси , el поворот фигуры относительно горизонтальной оси.

Дополнительные конфигурационные массивы S и Q можно не применять, используя простую форму команды mesh(Z).

2.5. Некоторые функции анализа и обработки данных

MatLab предлагает ряд функций, позволяющих анализировать и обрабатывать матрицы как таблицы, содержащие некоторые данные.

2.5.1. Функции, сохраняющие размерность исходной матрицы: `sort`, `diff`.

Действует следующее правило: для двумерной матрицы действие функции применяется по- отдельности к столбцам, для одномерной – ко всем элементам.

Сортировка данных *sort*. Матрица

```
A = [7  2  18
     1  11 9
     10 8  3]
```

после сортировки `B = sort(A)` принимает вид:

```
B = [1  2  3
     7  8  9
     10 11 18]
```

Одномерная матрица `a = [13 4 6 10]` после сортировки `b = sort(a)` равна `b = [4 6 10 13]`.

Определение приращения значений элементов `diff`.

Определяется разность между последующими элементами как разность между элементом с номером (i+1) и элементом с номером i .
Так, для двумерной матрицы `A = [ 1 2 3; 4 6 8; 13 16 19]` функция `B = diff(A)` дает результат `B = [ 3 4 5; 9 10 11]`.

Для одномерной `a = [1 4 8 14]` результатом `b = diff(a)` будет `b=[3 4 6]`.

При этом результирующая двумерная матрица имеет на одну строку меньше чем исходная, одномерная матрица – на один элемент меньше исходной.

2.5.2. Функции, изменяющие размерность исходной матрицы `mean`, `std`, `max`, `min`.

При использовании таких функций действует следующее правило: для одномерной матрицы результатом является найденная величина, для двумерной – одномерная матрица, каждый элемент которой представляет величину, являющуюся результатом применения функции к каждому столбцу исходной матрицы.

Среднее значение и среднее квадратическое отклонение *mean,std*
.

```
A =  [1   8   18
       10  2   9
        7   11  3]
```

Для матрицы A
применяя функцию: `B = mean(A)` получаем `B = [6 7 10]` и затем повторно `C = mean(B)` получаем `C = 7.667`.
Также, для `B = std(A)` получаем `B = [3.7417 3.7417 6.1644]` и `C = std(B)` получаем `C =1.1421`.

Наибольшее и наименьшее значение элементов матрицы `max,min`.

Определяется наибольшее значение элементов в каждом столбце (для двумерной матрицы) или наибольший элемент одномерного массива.

Для вышеуказанной матрицы `A` применяя `B = max(A)` получаем `B = [10 11 18]`. Далее, `C=max(B)` дает `C = 18`.

Функции, результат действия которых может быть ассоциирован с номером элемента матрицы, могут использоваться с расширенным синтаксисом вида `[B,N] = max(A)`. В этом случае находится массив `N` с количеством элементов, равных количеству столбцов массива `A`, содержащий номер строки расположения наибольшего элемента в соответствующем столбце `N = [2 3 1]`. Для одномерного массива `[C,M] = max(B)`, переменная `M` содержит номер наибольшего элемента: `M = 3`.
Аналогично действие функции поиска минимального элемента. Для `[B,N] = min(A)` получаем `B = [1.0 2.0 3.0]` и `N = [1 2 3]`, для `[C,M] = min(B)` имеем результат `M = 1` и `C = 1`.

2.6. Управление процессом вычислений (циклы и условные операторы)

2.6.1. Цикл for (цикл с фиксированным количеством повторений)

Представляет собой блочную конструкцию типа:

for <перечисление целых значений индекса цикла >
ОПЕРАТОРЫ ТЕЛА ЦИКЛА
end

Например, матрица `A` с размерностью 3 ∗ 4 формируется как:

```
for i=1:3
    for j=1:4
        A(i,j)=1/(i+j-1)
    end %конец цикла по j%
end %конец цикла по i%
```

2.6.2. Цикл while (с заранее неизвестным коли-

чеством повторений)

Структура оператора:

while *условие*

ОПЕРАТОРЫ ТЕЛА ЦИКЛА

end

Повторение тела цикла производится до тех пор, пока является истинным заданное *условие*.

Условие может быть записано через операции отношения очевидными символами : <, <= ,>, >= и их комбинациями: == означает "равно", ~= означает "не равно". При записи условий также используются логические операции: `&` означает `and`, `I` означает `or`, ~ означает `not`.

Например, для получения значения `teta` равномерно распределенной в интервале (0,1) случайной величины, большего 0,5 может использоваться такой цикл:

```
flag = 0
while flag ==0
    g1 = rand(1,1)
     if g1 >0.5
        teta = g1
        flag = 1
     end %завершение оператора if%
end %завершение цикла%
```

2.6.3. Ветвление `if-else`

Простая форма имеет структуру:

if *условие*

ОПЕРАТОРЫ

end

Операторы в теле ветвления будут выполнены, если справедливо условие в заголовке.

Структура альтернативной формы:

if *условие*

ОПЕРАТОРЫ 1

else

ОПЕРАТОРЫ 2

end

При выполнении *условия* будут исполнены ОПЕРАТОРЫ 1, при невыполнении – ОПЕРАТОРЫ 2.

2.7. Средства проектирования

Основой проектных процедур на системотехническом уровне является ограниченное количество операций автоматизированного проектирования к числу которых относятся дискретное (быстрое) преобразование Фурье, дискретная свертка, синтез случайных величин /1/. Большинство этих функций сосредоточены в библиотеке Signal Processing Toolbox /5/.

2.7.1. Быстрое преобразование Фурье `fft` и `ifft`

Если матрица `G` содержит дискретные отсчеты одномерного сигнала, то его дискретное (быстрое) преобразования Фурье выполняется функцией:

```
S = fft(G)
```

где `S` – матрица комплексных значений дискретного спектра.

Амплитудный спектр находится как `A = abs(S)`, фазовый (в радианах) как `F = angle(S)`.

Обратное дискретное (быстрое) преобразование Фурье выполняется как

```
G = ifft(S)
```

Следует учитывать, что алгоритм быстрого преобразования Фурье выполняется для количества отсчетов сигнала, равного целой степени числа 2 (то есть, 2^L где L – целое). В случае, если это условие не выполняется, Matlab без дополнительного предупреждения добавляет нулевые отсчеты в массив сигнала (до ближайшего 2^L). После выполнения операции возвращается как результат исходное количество отсчетов. Однако, произвольное добавление нулевых отсчетов в исходный сигнал может значительно исказить результат операции, что может привести к методической погрешности в десятки процентов /1/. Таким образом, при использовании Matlab проектировщик должен самостоятельно контролировать указанное условие, предъявляемое к количеству элементов в обрабатываемых массивах.

При спектральном анализе двумерного массива (изображения) `G2` используется соответствующие двумерные функции.

`S2 = fft2(G2)` и `G2 = ifft2(S2)`

2.7.2. Дискретная свертка

Массив сигнала на выходе линейного звена может быть найден с помощью операции дискретной свертки /1/.

Если матрица `G` содержит отсчеты входного сигнала, матрица `H` - отсчеты импульсной характеристики линейного элемента прибора, то матрица `Y` с отсчетами выходного сигнала находится как :

```
Y = conv(G,H)
```

При синтезе оптических элементов массивы сигналов и импульсной характеристики двумерны, что требует использования двумерной функции

```
Y2 = conv2(G2,H2)
```

Количество элементов выходного массива равно сумме количества элементов исходных массивов минус единица.

2.7.3. Синтез значений случайной величины

Случайные величины со сложными функциями распределения синтезируются на основе значений случайной величины с равномерным распределением в интервале [0,1] /1/.

Для генерации двумерной матрицы из `M` строк и `N` столбцов используется функция

```
GAMMA = rand(M,N)
```

Соответственно, двумерная квадратная матрица M*M и одномерная матрица-строка из N элементов генерируются как:

```
GAMMA = rand(M)  gamma = rand(1,N)
```

По умолчанию элементы сгенерированных матриц содержат значения случайной величины с равномерным распределением в интервале [0,1].

Случайные процессы (шумы и фоны) с нормальным распределением и некоторой корреляционной функцией синтезируются на основе нормированной случайной величины с нормальным распределением, нулевым математическим ожиданием и единичной дисперсией /1/.

Генерация такой нормированной нормальной случайной величины выполняется аналогичным образом с использованием функции randn:

```
KCI = randn(M,N)
```

Численные значения входных параметров M и N должны быть определены заранее.

2.8. Функции обмена данными с внешними устройствами; вывод на экран, запись/чтение с диска

2.8.1. Вывод числовых данных на экран

По умолчанию, все рассчитанные величины выводятся на экран. При этом работа MatLab значительно замедляется, что делает актуальной проблему отключения вывода на экран большинства найденных величин. Для этого достаточно поставить знак ";" в конце строки. В случае, если операторами строки находится результирующая искомая величина, знак "двоеточие" ставить не следует – и значение автоматически будет выведено на экран.

Например, для двумерной матрицы `A` (см. ранее) действие операторов:

```
B = mean(A) ;
C= mean(B)
```

приведет к выводу на экран только итогового значения `C = 7.667`.

2.8.2. Запись данных в файл на диске (`save`)

Найденные значения могут быть записаны в файл на диске. Для возможного использования этих данных другими программами следует специально предписать стандартный ASCII - формат записи. Так, оператор:

```
save  Y.prn A -ascii
```

выполняет запись в файл с именем `Y.prn` значений элементов массива `A` с сохранением структуры массива.

2.8.3. Чтение данных из файла на диске (`load`)

Числовые данные могут быть прочитаны из файла на диске и присвоены переменной MatLab.

Соответственно, выполняются две операции. Одна читает данные из файла с указанным именем (полностью), а вторая – присваивает прочтенные данные матрице с указанным идентификатором (в этом случае имя файла приводится без расширения).

Так, операторы:

```
load YG.prn
X = YG;
```

выполняют чтение данных из файла с именем `YG.prn` и присваивают их матрице с именем `X` с сохранением исходной структуры данных.

Данные в файле должны иметь вид строки чисел, разделенных пробелами или таблицы с разделенными столбцами.

2.9. Файлы внешних функций (подпрограммы типа *function*)

Важным инструментом MatLab являются файлы, содержащие подпрограммы типа function.

Такой файл внешней функции имеет имя с расширением .m и начинается с заголовка вида:

function <имя функции>=<имя файла без расширения m>(формальные параметры)

Далее следуют операторы подпрограммы, завершающиеся оператором:

<имя функции> = <вычисляемое выражение>

Результат итогового вычисляемого выражения присваивается имени функции и передается в основную программу.

Вызов подпрограммы из основной программы производится по <u>имени файла</u> с помощью оператора присваивания:

<имя переменной> = <имя файла без расширения >(фактические параметры)

В результате переменная с указанным именем получает значение, рассчитанное подпрограммой-функцией по фактическим параметрам.

Например, подпрограмма для генерации одномерного массива с количеством элементов `N` значений случайной величины с равномерным распределением в интервале [a,b], записанная в файл с именем `randu.m` имеет вид.

```
function y = randu(N,a,b)
    x = rand(1,N);
    y = a + (b-a)*x ;
```

При вызове подпрограммы оператором присваивания основной программы вида:

```
Z = randu(10, -1,1)
```

будет получен массив Z из 10 значений равномерно распределенной в интервале [-1,1] случайной величины.

Переменные, объявленные внутри подпрограмм-функций являются локальными и не сохраняются в рабочей области оперативной памяти после выполнения файла-функции.

Операторы тела подпрограммы могут в свою очередь обращаться к другим подпрограммам, в том числе и рекурсивно - к самим себе.

Фактически пакеты Toolbox расширения MatLab являются библиотеками подпрограмм-функций. Пользователь может создавать собственные файлы для реализации новых функций с учетом специфики решаемых задач, расширяя возможности пакета.

2.10. Работа с MatLab через командные (script) файлы

По умолчанию MatLab предлагает режим работы похожий на работу в технологии MathCAD – с клавиатуры вводятся последовательности предложений (команд, функций, операторов), необходимых для решения поставленной задачи, что приводит к немедленному выводу на экран результатов по каждой команде (если не поставлен разделитель ";") или по последовательности предложений (графики строятся в отдельных окнах).

Для выполнения задач моделирования и проектирования рекомендуется использовать режим, напоминающий работу с языком программирования.

В соответствии с этим режимом операторы модели или проектного задания записываются в отдельный файл с расширением .m и сохраняются на диске в текущем или другом, выбранном пользователем каталоге. Используется встроенный в систему MatLab редактор (окно редактирования) или любой другой, формирующий исходный текст в стандарте ASCII (Far, Windous Commander, Notebook).

Для экспериментов с моделью или выполнения проектного задания в строке командного окна (Command Window) системы MatLab указывается имя этого script файла (без расширения), при этом каталог на диске, в котором находится исполняемый script файл должен быть указан как Рабочий в окне Current Directory.

При появлении в командном окне сообщений об ошибках исполнения следует исправить исходный текст script -файла и повторить его исполнение.

Для исправления исходного текста следует обратиться к пункту меню Файл – Открыть (File – Open) и в раскрывшемся списке выбрать имя файла (при создании файла используется File – New).

В открывшемся Окне Редактирования представлен текст script – файла. После исправления (или создания) текста через пункты Отладка – Запуск (Debug – Run) файл исполняется (при этом предварительно отредактированный текст также записывается в файл на диске).

3. ОПРЕДЕЛЕНИЕ ПАРАМЕТРОВ НАДЕЖНОСТИ ПРИБОРОВ И СИСТЕМ МЕТОДОМ СТАТИСТИЧЕСКИХ ИСПЫТАНИЙ

Проведенный в Разделе 1 анализ параметров и характеристик надежности в совокупности с приведенными в Разделе 2 основными объектами и операторами MatLab позволяет синтезировать модели различных систем и выполнить экспериментальное исследование их надежности.

3.1. Общая методика определения параметров надежности по результатам эксперимента с компьютерной моделью

Определение параметров систем методом статических испытаний включает ряд стандартных этапов.

1. Моделирование функционирования элементов, входящих в состав системы по- отдельности.
2. Формирование из элементов блоков в соответствии с их ролью и соединением.
3. Синтез модели всей системы из отдельных блоков.
4. Эксперименты с моделью системы (многократное «разыгрывание» модели), получение массива выходных сигналов.
5. Обработка полученного массива экспериментальных данных.

Рассмотрим применение указанной методики на типовых примерах.

3.2. Определение параметров надежности невосстанавливаемых приборов и систем

Пусть, например, исследуемый объект является охранной системой и включает две одновременно работающие видеосистемы – явного и скрытого контроля (обозначены как I и II на рис. 3.1.).

Каждая видеосистема в свою очередь состоит из прожектора подсветки (явного контроля – на основе люминесцентных ламп, скрытого контроля – на основе инфракрасных светодиодов), телекамеры и блока пита-

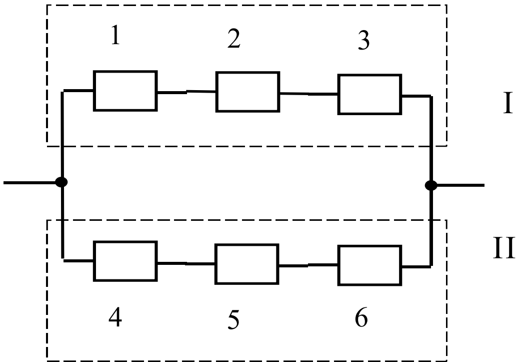

Рис. 3.1. Схема для расчета надежности рассматриваемой системы

ния (элементы 1 и 4, 2 и 5, 3 и 6, соответственно). Эти элементы строго говоря восстанавливаемые, но до первого отказа система может описываться параметрами надежности невосстанавливаемых элементов.

Распределение времени работы каждого элемента до отказа и исходные параметры для моделирования приведены в таблице 3.1.

Таблица 3.1.

N	Распределение времени работы элемента до отказа	Параметры распределения
1	Нормальное	*tср* = 1600 час; σ = 50 час
2	Вейбулла	*tср* = 4000 час; *b* = 0,5
3	Экспоненциальное	*tср* = 6000 час
4	Нормальное	*tср* = 2300 час; σ = 100 час
5	Вейбулла	*tср* = 2000 час; *b* = 1,5
6	Экспоненциальное	*tср* = 6000с

Требуется: определить среднее время безотказной работы *tm* исследуемой системы, а также построить графики функций вероятности безотказной работы *P(t)* в течении времени *t*, интенсивности отказов λ(t) и частоты отказов $F_0(t)$.

Для определения параметров надежности в соответствии с выражениями (1.10)...(1.14) необходимо выполнить эксперимент над некоторым количеством N_0 моделей исследуемой системы. Полагая исследуемые процессы отказа эргодическими, можно заменить эксперимент на статистически адекватный: выполнить N_0 экспериментов, но над одной моделью.

При этом средствами MatLab имеется возможность сразу синтезировать результаты эксперимента над N_0 моделями системы.

Пусть выполняются эксперименты над N_0 = 100 моделями.

1. Выполним моделирование работы элементов первой видеосистемы.

Для синтеза массива времени `tp1` работы прожектора до отказа для всех моделей необходимо предварительно создать массив `X1` из `N0 = 100` значений простой нормально распределенной величины.

Генерация массива – строки значений нормальной случайной величины:

```
X1=randn(1,100) ;
```

По алгоритму (1.28) синтезируется массив `ts1` из 100 значений времени работы прожектора до отказа при среднем времени безотказной работы *tср* = 1600 час.

```
ts1 = X1*50 +1600 ;
```

Аналогичным образом определяются массив `tv1` времени работы телекамеры в соответствии с выражениями (1.30) и (1.31). При этом для *b* = 0,5 значение Г(1+1/b) = 2 и *tср* = 4000 час среднего времени безотказной работы получаем массив значений времени работы телекамеры в 100 моделях системы:

```
b=0.5
X2=rand(1,100) ;
lambda = (2/4000)^b ;
tv1=((- 1/lambda)*log(1 - X2))^(1/b);
```

Определяется массив `tp1` из N_0 = 100 значений времени работы блока питания в соответствии с выражением (1.36) для *tcp* = 6000 час для 100 моделей.

```
X3=rand(1,100) ;
lambda = 1/6000;
tp1 = (- 1/lambda)*log(1 - X3);
```

Из массивов времени работы отдельных элементов монтируется массив экспериментальных данных для всей первой видеосистемы.

```
T1 = [ts1;tv1;tp1];
```

В полученном двумерном массиве будет 100 столбцов по количеству испытуемых моделей и три строки по количеству элементов в видеосистеме. Три числа в столбце соответствуют времени работы до отказа каждого из трех элементов видеосистемы в одной модели..

Массив `t1` из 100 значений времени работы первой видеосистемы в каждой модели определяется по массиву T1 как строка из минимальных времен в каждом столбце (соединение элементов последовательное):

```
t1 = min(T1);
```

2. Выполним моделирование работы элементов второй видеосистемы

Аналогичным образом определяем массив `t2` из 100 значений времени работы второй видеосистемы в каждой испытуемой модели.

```
X4=rand(1,100) ;
ts2 = X4*100 +2300 ;
b=1.5
X5=rand(1,100) ;
lambda = (0.9033/2000)^b ;
tv2 = ((-1/lambda)*log(1 - X5) ).^(1/b);
X6=rand(1,100) ;
lambda = 1/6000;
tp2 = (- 1/lambda)*log(1 - X6) ;
T2 = [ts2;tv2;tp2] ;
t2 = min(T2);
```

3. Выполним моделирование работы всей исследуемой системы.

Из массивов времени работы отдельных видеосистем монтируется массив экспериментальных данных для эквивалентного элемента.

```
T12 = [t1;t2] ;
```

Массив `t12` из 100 значений времени работы всей системы для каждой испытуемой модели определяется по массиву `T12` как строка из максимальных времен в каждом столбце (соединение отдельных видеосистем параллельное):

```
[t12,NN] = max(T12) ;
```

Массив `NN` содержит номер наибольшего элемента в каждом столбце и позволяет проанализировать какая именно видеосистема - первая или вторая отказывала последней в каждой испытуемой модели.

Для контроля моделирования строится график `t12` времени работы системы в зависимости от номера эксперимента и график номеров NN (последний график строится укрупненными точками для большей наглядности).

```
plot(t12)
pause
plot(NN,'o')
pause
```

4. Определение среднего времени безотказной работы системы.

В соответствии с выражением (1.6) находим оценку среднего значения элементов массива:

```
tm = mean(t12)
pause
```

5. Построение графика функции вероятности безотказной работы P(t).

Выполним сортировку значений `t12` времени работы различных моделей до отказа:

```
t = sort(t12);
```

В итоговом массиве–строке результаты эксперимента расположены по порядку возрастания времени работы моделей до отказа. При этом значение времени с номером *i* означает, что на момент времени t_i эксперимента отказало именно *i* испытуемых моделей, а работоспособно, соответственно, N_0-i моделей. Далее, поскольку каждому следующему значению времени t_i массива `t` соответствует отказ следующей по очереди модели, массив `Nt` определяющий количество еще не отказавших моделей представляет собой убывающую последовательность с шагом –1. Для рассматриваемого количества моделей:

```
N0 = 100;
Nt = N0:-1:1;
```

В соответствии с выражением (1.4) находятся численные значения функции P(t) вероятности безотказной работы исследуемой системы:

```
p = Nt/N0;
```

Поскольку массив t определяет моменты времени, в которые работоспособными оказывалось количество моделей, определяемое соответствующими элементами массива `Nt`, график функции P(t) определяется построением, для которого значения массива `t` являются аргументом, а массива `p`- значениями функции.

```
plot(t,p);
pause
```

6. Построение графика функции интенсивности отказов $\lambda(t)$.

В соответствии со структурой упорядоченного массива `t`, за интервал времени $\Delta t = t_i - t_{(i-1)}$ происходит отказ одной испытуемой модели. Следовательно, в выражении (1.8) $\Delta N = 1$. Поэтому для расчета интенсивности отказов по выражению (1.8) создадим массив `dN` единичных значений. Количество элементов в массиве равно N0 – 1.

```
dN = ones(1,N0-1);
```

Определим массив интервалов времени $\Delta t = t_i - t_{(i-1)}$ из N0 – 1 элементов:

```
dt = diff(t);
```

Расчет проводится для первых N_0–1 моментов времени, поэтому формируются два дополнительных массива `tt` и `Ntt`, включающих все элементы массивов `t` и `Nt`, кроме последнего. Предварительно находится количество элементов в массивах `t` или `Nt`:

```
k=length(t);
tt=t(1:k-1);
Ntt=Nt(1:k-1);
```

Далее, по выражению (1.8) определяются значения функции $\lambda(t)$ и строится ее график (как в обычном масштабе так и полулогарифмическом по оси ординат), причем значения аргумента задаются массивом `tt`.

```
lambdaf = (dN./dt)./Ntt;
plot(tt,lambdaf)
pause
semilogy(tt,lambdaf)
pause
```

7. Построение графика функции частоты отказов $F_0(t)$.

Аналогичным образом по выражению (1.9) строится график функции частоты отказов.

```
F = (dN./dt)./N0;
plot(tt,F0)
pause
semilogy(tt,F)
pause
```

Полученные в результате зависимости $P(t),\lambda(t),F_0(t)$ имеют табулированную форму, причем значения аргумента – моменты времени t имеют неравномерный шаг.

При необходимости может выполняться аппроксимация найденных зависимостей. Для этого необходимо их предварительно преобразовать в табличную форму, в которой значениям функции будут соответствовать значения аргумента с постоянным шагом.

3.3. Определение параметров надежности невосстанавливаемых приборов и систем с топологически сложным соединением элементов

Рассмотрим в качестве примера определение параметров надежности системы со схемой по надежности, изображенной на рисунке 1.8. Практической реализацией такой схемы является, например, система измерения дальности в составе основного дальномера 3 и резервного дальномера 4. Резервный дальномер 4 управляется вручную оператором, основной дальномер – действует в автоматическом режиме под управлением компьютера 1. При этом возможно с пульта управления дальномером 4 управлять дальномером 3 через микропроцессорный блок 2. С другой стороны, от компьютера 1 через тот же блок 2 возможна передача команд на дальномер 4.

Таким образом, система имеет сложную топологию с точки зрения анализа надежности.

Как было указано в Разделе 1, в этом случае выполняется замена схемы системы на две эквивалентные – по минимальным путям и минимальным сечениям, после чего выполняется расчет двух оценок параметра надежности по каждой из схем.

Распределение времени работы каждого элемента до отказа и исходные параметры для моделирования приведены в таблице 3.2.

Таблица 3.2.

N	Распределение времени работы элемента до отказа	Параметры распределения
1	Экспоненциальное	tср = 5000 час
2	Экспоненциальное	tср = 3000 час;
3	Экспоненциальное	tср = 2000 час
4	Релея	tср = 3000 час

Определим оценку среднего время безотказной работы tm исследуемой системы по результатам испытаний N_0 = 200 моделей, используя эквивалентную схему по минимальным путям - см. рис. 1.9.

1. Выполним моделирование работы элементов на первом минимальном пути (элемент 4 имеет Релеевское распределение времени работы до отказа). По методике, рассмотренной в предыдущем пункте, начинаем с создания массива исходных значений случайной величины с равномерным распределением.

```
X1=rand(1,200);
```

По выражению (1.25) для *tcp* = 3000 час определяем массив времени `t4` работы до отказа элемента 4, составляющего минимальный путь R1 в каждой из 200 моделей системы.

```
tcp = 3000;
C= tcp/sqrt(2/pi);
t4 = sqrt(-2*C^2*log(1-X1));
```

Определим массив tR1 времени работы всего минимального пути R1 для всех моделей. Поскольку на данном пути только один элемент, искомый массив будет равен массиву t4:

```
tR1=t4;
```

2. Выполним моделирование работы второго и третьего минимального пути.

Синтезируем массивы времени работы элементов 1 и 3, составляющих минимальный путь R2:

```
X2=rand(1,200) ;
lambda = 1/2000;
t3 = (- 1/lambda)*log(1 - X2);
X3=rand(1,200) ;
lambda = 1/5000;
t1 = (- 1/lambda)*log(1 - X3);
```

Из массивов времени работы отдельных элементов монтируется массив экспериментальных `TR2` данных для минимального пути R2 по всем испытуемым моделям.

```
TR2 = [t1;t3] ;
```

Определим массив tR2 итогового времени работы всего минимального пути R2 для всех моделей.

```
tR2=min(TR2);
```

Аналогичным образом синтезируется массив экспериментальных данных `tR3` итогового времени работы всего минимального пути R3 в составе элементов 2 и 3 для всех моделей.

```
X4=rand(1,200) ;
lambda = 1/3000;
t2 = (- 1/lambda)*log(1 - X4);
TR3 = [t2;t3];
tR3=min(TR3);
```

3. Определение массива данных для всей системы.

Из массивов времени работы отдельных минимальных путей монтируется массив экспериментальных данных `T` всей системы.

```
T = [tR1;tR2;tR3];
```

Массив t из 200 значений времени работы всей системы для каждой испытуемой модели определяется по массиву `T` как строка из максимальных времен в каждом столбце (соединение отдельных минимальных путей параллельное):

```
[t,NN] = max(T);
```

Массив `NN` содержит номер наибольшего элемента в каждом столбце и позволяет проанализировать какой именно минимальный путь отказывал последним в каждой испытуемой модели.

Для контроля моделирования строится график `t` времени работы системы в зависимости от номера эксперимента и график номеров `NN` (последний график строится укрупненными точками для большей наглядности).

```
plot(t)
pause
plot(NN,'o')
pause
```

4. Определение среднего времени безотказной работы системы.

В соответствии с выражением (1.6) находим оценку среднего значения массива:

```
tm = mean(t)
pause
```

Построение графиков функций вероятности безотказной работы, интенсивности и частоты отказов выполняется аналогично подразделу 3.2.

Для получения второй оценки параметров надежности необходимо выполнить их расчет для эквивалентной схемы системы по минимальным сечениям в соответствии со схемой на рис. 1.10.

3.4. Определение параметров надежности восстанавливаемых приборов и элементов (с мгновенным восстановлением)

Рассмотрим в качестве примера определение параметров надежности восстанавливаемой системы с последовательным соединением элементов - см. рис. 3.2. Пусть параметры элементов также определяются таблицей 3.2.

Определим параметр потока отказов ω(t) по результатам экспериментов с рассматриваемой моделью системы. В отличие от ранее рассмотренных примеров, в эксперименте участвует одна модель системы. При этом в процессе эксперимента реализуется N= 300 ситуаций отказ - мгновенное восстановление. Для каждой ситуации i фиксируется время t(i) между отказом с номером (i-1) и отказом i. В результате обработки массива t

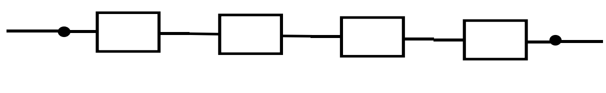

Рис. 3.2. Схема системы с восстановлением

определяются искомые параметры надежности.

1. Для каждого элемента системы определяется время до отказа в одном опыте в виде вектора из 1 элемента (аналогично расчету в подразделе 3.2, но не для 100 или 200, а для одной модели) - соответственно, находятся массивы `t1`,`t2`,`t3`,`t4` времени работы каждого элемента:

```
% первый элемент
   X1=rand(1,1) ;
   lambda1 = 1/5000;
   t1 = (- 1/lambda1)*log(1 - X1);
% второй элемент
   X2=rand(1,1);
   lambda2 = 1/3000;
   t2 = (- 1/lambda2)*log(1 - X2);
% третий элемент
   X3=rand(1,1) ;
   lambda3 = 1/2000;
   t3 = (- 1/lambda)*log(1 - X3);
% четвертый элемент
   X4=rand(1,1);
   tcp = 3000;
   C= tcp/sqrt(2/pi);
   t4 = sqrt(-2*C^2*log(1-X4));
```

2. Запускается цикл эксперимента в виде оператора цикла с переменной цикла `i` и количеством повторений, равным количеству наблюдаемых отказов N = 300, предварительно обнуляется массив результата `t`.

```
t = [];
N=300;
for i = 1:N
```

3. Формируется двумерный массив `t0`, содержащий время работы до отказа всех элементов.

В этом массиве количество строк равно количеству элементов в системе; в каждой строке данные из соответствующего массива `t1,t2,t3,t4`.

```
t0 = [t1;t2;t3;t4];
```

4. Определяется значение времени работы системы tv до следующего отказа как минимальное время работы в массиве `t0`, соответствующему группе из четырех элементов.

Также находится номер `n` элемента, которому соответствует минимальное время работы – фактически номер отказавшего и затем восстановленного элемента.

```
[tv,n] = min(t0);
```

5. Найденное значение `tv` присваивается элементу `t(i)` массива времени работы системы между отказами

```
t(i) = tv;
```

6. Выполняется перенос начала отсчета времени для следующего рабочего цикла как:

```
t1 = t1-tv;
t2 = t2-tv;
t3 = t3-tv;
t4 = t4-tv;
```

Дело в том, что после отказа и последующего восстановления элемента отсчет времени работы до его следующего отказа начинается заново. Следовательно, для всех элементов (кроме уже отказавшего) смоделированное значение времени работы до их отказа следует уменьшить на величину tv промежутка до уже произошедшего отказа.

7. Для отказавшего элемента аналогично пункту 1. генерируется новое значение времени работы.

При этом для всех элементов последовательно выполняется следующая процедура: с помощью оператора if последовательно проверяется, равен ли номер n отказавшего элемента единице. Если да, то отказал первый элемент и для него генерируется новое время работы t1 до следующего отказа.

Далее, проверяется не равен ли номер n отказавшего элемента двум (то есть, не второй ли элемент отказал ?) и так далее для всех элементов системы

```
    if n ==1
         X1=rand(1,1) ;
         t1 = (- 1/lambda1)*log(1 - X1);
    end
         %
    if n ==2
         X2=rand(1,1);
         t2 = (- 1/lambda2)*log(1 - X2);
    end
         %
    if n ==3
         X3=rand(1,1) ;
         lambda3 = 1/2000;
         t3 = (- 1/lambda)*log(1 - X3);
    end
         %
if n ==4
         X4=rand(1,1);
         t4 = sqrt(-2*C^2*log(1-X4));
    end
```

Цикл эксперимента заканчивается

```
end
```

7. Строится график массива времени `t` между отказами

```
plot(t)
pause
```

8. Поскольку за интервал времени, соответствующее каждому элементу массива `t` происходит один отказ, в выражении (1.17) можно принять $N(t) = 1$, $\Delta t = t_i$, где t_i – элемент массива `t`.

Таким образом, массив `Omega` значений параметра потока отказов определяется как массив величин, обратных массиву `t`. Для этого массив `t` возводится в степень (-1).

```
Omega=t.^(-1);
plot(Omega)
pause
semilogy(Omega)
pause
```

9. Для приближенной оценки параметра потока отказов определяются оценка среднего значения и среднего квадратического значения по массиву параметра потока отказов

```
OmegaSred = mean(Omega)
OmegaRms = std(Omega)
```

Аппроксимация полученной в пункте 8 функции параметра потока отказов представляет собой отдельную задачу и здесь не рассматривается.

3.5. Определение параметров надежности восстанавливаемых приборов и элементов (с немгновенным восстановлением)

Рассмотрим в качестве примера определение параметров надежности восстанавливаемой системы со схемой, соответствующей рис. 3.2 при условии немгновенного восстановления. Параметры элементов на этапе работы также определяются таблицей 3.2, параметры этапа восстановления – таблицей 3.3.

Таблица 3.3.

N	Распределение времени восстановления элемента	Параметры распределения
1	Нормальное	tср = 100 час, σ = 10 час
2	Нормальное	tср = 150 час; σ = 15 час
3	Нормальное	tср = 50 час, σ = 5 час
4	Нормальное	tср = 200 час, σ = 20 час

Определим коэффициент готовности Kg и построим график функции готовности Kr(t) по результатам экспериментов с моделью системы. В эксперименте участвует одна модель системы. При этом в процессе эксперимента реализуется N= 400 ситуаций отказ – восстановление. Для каждой ситуации i фиксируется время работы между отказами системы и время восстановления. В результате обработки полученных массивов определяются искомые параметры надежности.

1. Для каждого элемента системы определяется время до отказа в одном опыте в виде вектора из одного элемента. Соответственно, находятся массивы `t1,t2,t3,t4` времени работы каждого элемента:

```
X1=rand(1,1) ;
lambda1 = 1/5000;
t1 = (- 1/lambda1)*log(1 - X1);
X2=rand(1,1);
lambda2 = 1/3000;
t2 = (- 1/lambda2)*log(1 - X2);
X3=rand(1,1);
lambda3 = 1/2000;
t3 = (- 1/lambda)*log(1 - X3);
X4=rand(1,1);
tcp = 3000;
C= tcp/sqrt(2/pi);
t4 = sqrt(-2*C^2*log(1-X4));
```

2. Задаются начальные значения моделируемых величин

Совокупное время t работы системы и совокупное время восстановления tr перед началом эксперимента равны нулю, также обнуляется массив результата:

```
t = 0;
tr = 0;
Kr = [];
```

3. Запускается цикл эксперимента в виде оператора цикла с переменной цикла `i` и количеством повторений, равным количеству наблюдаемых отказов N = 400

```
N=400;
for i = 1:N
```

4. Формируется двумерный массив `t0`, содержащий время работы до отказа всех элементов.

В этом массиве количество строк равно количеству элементов в системе; в каждой строке данные из соответствующего массива `t1`,`t2`,`t3`,`t4`.

```
t0 = [t1;t2;t3;t4];
```

5. Определяется значение времени работы системы `tv` до следующего отказа как минимальное время работы в массиве `t0`, соответствующему группе из четырех элементов. Также находится номер `n` элемента, которому соответствует минимальное время работы – фактически, номер

отказавшего и затем восстановленного элемента.

```
[tv,n] = min(t0);
```

6. Затем текущее время работы t увеличивается на это время tv

```
t = t + tv;
```

7. Выполняется перенос начала отсчета времени следующего рабочего цикла как (подробнее действие этого пункта см. в примере для мгновенного восстановления)

```
t1 = t1-tv;
t2 = t2-tv;
t3 = t3-tv;
t4 = t4-tv;
```

8. Для отказавшего элемента аналогично пункту 1. генерируется новое значение его времени работы и время восстановления tr – как случайная величина с соответствующим распределением (см. табл. 3.2)

При этом для всех элементов последовательно выполняется следующая процедура: с помощью оператора if последовательно проверяется, равен ли номер n отказавшего элемента единице. Если - да, то для первого элемента генерируется новое время работы t1 до следующего отказа и время его восстановления tp . Далее, также проверяется равен ли номер n отказавшего элемента двум (то есть, не второй ли элемент отказал ?) и так далее для всех элементов системы:

```
if n ==1
        X1=rand(1,1) ;
        t1 = (- 1/lambda1)*log(1 - X1);
        X11=randn(1,1);
        tp = X11*10 +100;
    end
        %
    if n ==2
        X2=rand(1,1);
        t2 = (- 1/lambda2)*log(1 - X2);
        X22=randn(1,1);
        tp = X22*15 +150;
end
        %
    if n ==3
        X3=rand(1,1) ;
        lambda3 = 1/2000;
        t3 = (- 1/lambda)*log(1 - X3);
        X33=randn(1,1);
        tp = X33*5 +50;
    end
        %
```

```
if n ==4
        X4=rand(1,1);
        t4 = sqrt(-2*C^2*log(1-X4));
        X44=randn(1,1);
        tp = X44*20 +200;
   end
```

9. Совокупное время восстановления `tr` увеличивается на это время `tp` данного восстановления:

```
tr = tr +  tp;
```

10. Вычисляется текущее значение функции готовности как элемент массива `Kr(i)`

```
Kr(i) = t/(t+tp);
```

Цикл эксперимента заканчивается

```
end
```

11. Строится график функции готовности Kr(t):

```
plot (Kr)
pause
```

12. Определяется значение коэффициента готовности Kg как предела функции готовности Kr(t). В данном случае оценкой коэффициента готовности можно полагать последний элемент массива `Kr`.

```
Kg = Kr(N)
```

Для аппроксимации функции готовности могут использоваться функция polyfit.m, входящая в состав MatLab.

ЗАКЛЮЧЕНИЕ

Рассмотренные в первом и третьем разделе методы построения моделей ориентированы на исследование характеристик надежности ОЭС.

Однако аналогичным образом синтезируются модели для исследования процессов обработки сигналов, влияния шумов и фонов, параметрической чувствительности, а также модели для реализации проектных процедур.

Соответственно, рассмотренные объекты системы MatLab используются не только для построения моделей отказовых ситуаций, но и для параметрического синтеза при компьютерном проектировании.

ЛИТЕРАТУРА

1. Коняхин И.А. Процедуры автоматизированного проектирования ОЭС /Учебное пособие. –СПб: ИТМО, 2000. –59с.

2. Разработка САПР; в 10 кн. (Кн. 8)/Математические методы анализа производительности и надежности САПР: Практическое пособие/ В.Н. Кудрявцев, П.Н. Шкатов //Под. Ред. А.В. Петрова. – М.: Высш. шк., 1990.–144 с. :ил.

3. Леонов А.Н., Дубровский Н.Ф. Основы технической эксплуатации бытовой радиоэлектронной аппаратуры: Учебник для вузов.– М.: Легпромбытиздат, 1991. – 272 с.

4. Потемкин В.Г. Система MATLAB. Справочное пособие. М. – ДИАЛОГ – МИФИ, 1997 – 350 с.

5. Дьяконов В. MATLAB. Обработка сигналов и изображений. Специальный справочник. – СПб.: Питер. 2002. –608 с.: ил.

СОДЕРЖАНИЕ