

Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Институт прикладной математики и компьютерных наук
Кафедра прикладной математики

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК

Руководитель ООП

д-р техн. наук, профессор

 А.М. Горцев

« 03 » июне 2020 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

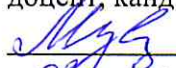
ВЕРОЯТНОСТНЫЕ ХАРАКТЕРИСТИКИ ПОТОКА СОБЫТИЙ С
ПРОДЛЕВАЮЩИМСЯ МЕРТВЫМ ВРЕМЕНЕМ СПЕЦИАЛЬНОГО ТИПА

по основной образовательной программе подготовки бакалавров
«Прикладная математика и информатика»
направления подготовки 01.03.02 Прикладная математика и информатика

Карлина Дарья Павловна

Руководитель ВКР

доцент, канд. техн. наук

 М.Е. Завгородняя

« 02 » июне 2020 г.

Автор работы

студент группы № _____

 Д.П. Карлина

Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Институт прикладной математики и компьютерных наук
Кафедра прикладной математики

УТВЕРЖДАЮ
Руководитель ООП
д-р техн.наук, профессор
_____ А.М.Горцев

« 17 » декабря 2019 г.

ЗАДАНИЕ

по подготовке ВКР бакалавра студентке Карлиной Дарье Павловне группы №1161

1. Тема ВКР

«Вероятностные характеристики потока событий с продлевающимся мертвым временем специального типа»

2. Срок сдачи студентом выполненной ВКР:

- а) на кафедре 03.06.2020
б) в ГЭК 11.06.2020

3. Исходные данные к работе.

Цель и задача этой работы состоит в том, чтобы найти вероятностные характеристики интервала между соседними событиями в наблюдаемом потоке и общего периода мертвого времени такие, как преобразование Лапласа для плотности распределения вероятностей в общем случае и плотности распределения вероятностей в частных случаях.

Для достижения этой цели были поставлены и решены следующие задачи:

- Изучена литература по данной тематике
- Получен вид преобразования Лапласа для плотности распределения вероятностей наблюдаемого потока в общем случае и плотности распределения вероятностей в частных случаях
- Построена имитационная модель простейшего потока в условиях мертвого времени специального типа
- Проведены статистические эксперименты

Объектом исследования является простейший поток событий в условиях случайного мертвого времени специального типа. Методами исследования являются методы теории вероятностей, теории случайных процессов, теории массового обслуживания, методы имитационного моделирования и численные методы.

Методы оценки достоверности результатов: проверка полученного вида плотности распределения для частных случаев критерием χ^2 .

4. Краткое содержание работы.

1. Вывод преобразования Лапласа для плотности распределения вероятностей;
2. Имитационное моделирование простейшего потока в условиях мертвого времени специального типа;

3. Проведение статистических экспериментов на основании имитационной модели.

5. Указать предприятие, организацию, по заданию которого выполняется работа:
Кафедра прикладной математики Национального исследовательского Томского государственного университета (НИ ТГУ).

6. Дата выдачи задания: «17» декабря 2019г.

Научный руководитель ВКР

доцент, канд. техн. наук

доцент каф. МА ИИ ТГУ

Задание принял к исполнению

М.Е. Завгородняя

подпись

М.Е.Завгородняя

17.12.19

дата, подпись студента

РЕФЕРАТ

Карлина Дарья Павловна

«Вероятностные характеристики потока событий с продлевающимся мертвым временем специального типа»

Цель работы: необходимо вывести вероятностные характеристики потока событий с продлевающимся мертвым временем специального типа такие, как преобразования Лапласа интервала между соседними событиями и преобразования Лапласа общего периода ненаблюдаемости.

Результаты работы: для потока с продлевающимся мертвым временем специального типа выведены преобразования Лапласа интервала между соседними событиями и преобразования Лапласа общего периода ненаблюдаемости, для случая экспоненциально распределенного периода мертвого времени получен явный вид плотности интервала между соседними событиями явный вид плотности общего периода ненаблюдаемости.

Практическая ценность: результаты, которые были получены, могут применяться при обработке данных в различных физических экспериментах, в задачах разработки и исследования систем массового обслуживания (информационно-вычислительные сети, торговые сети, сети связи), которые функционируют, как поток событий с продлевающимся мертвым временем специального типа.

Ключевые слова: простейший поток событий, мертвое время, преобразование Лапласа, имитационное моделирование.

Выпускная квалификационная работа бакалавра представлена на 43 листах, содержит 4 раздела, включает в себя 8 рисунков, 8 таблиц, 1 приложение.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	6
1 ПОСТАНОВКА ЗАДАЧИ.....	9
2 ВЕРОЯТНОСТНЫЕ ХАРАКТЕРИСТИКИ ПОТОКА СОБЫТИЙ С ПРОДЛЕВАЮЩИМСЯ МЕРТВЫМ ВРЕМЕНЕМ СПЕЦИАЛЬНОГО ТИПА.....	11
2.1 Преобразование Лапласа интервала между событиями в наблюдаемом потоке.....	11
2.2 Преобразование Лапласа длительности общего периода ненаблюдаемости.....	13
3 ЧАСТНЫЕ СЛУЧАИ.....	14
3.1 Длительность мертвого времени, распределенная по экспоненциальному закону.....	16
3.2 Длительность мертвого времени - константа.....	17
4 ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ПРОСТЕЙШЕГО ПОТОКА СОБЫТИЙ С ПРОДЛЕВАЮЩИМСЯ МЕРТВЫМ ВРЕМЕНЕМ СПЕЦИАЛЬНОГО ТИПА.....	19
4.1 Моделирование непрерывной случайной величины методом обратных функций.....	21
4.2 Имитационное моделирование простейшего потока событий при наличии мертвого времени, имеющего экспоненциальное распределение.....	22
4.3 Проверка имитационной модели на адекватность.....	25
ЗАКЛЮЧЕНИЕ.....	29
ЛИТЕРАТУРА.....	30

ВВЕДЕНИЕ

Актуальность работы.

В современном мире все чаще возникают проблемы решения задач, связанных со случайностями и изучения детерминированных величин недостаточно для их решения. Поэтому сегодня широко используются методы теории вероятностей, теории случайных процессов и теории массового обслуживания. Одним из популярных направлений в этой области является изучение обслуживания потоков случайных событий. Методы, рассматриваемые здесь, находят свое применение в системах массового обслуживания, вычислительных сетях, сетях связи и др.

Основателем теории массового обслуживания является датский математик, инженер и статистик А.К. Эрланг. Большой вклад в разработку общей теории массового обслуживания был внесен А.Я. Хинчиным выдающимся советским математиком и ученым. Им же было предложено понятие теории массового обслуживания. Основы математической теории массового обслуживания заложены в трудах А.Я. Хинчина, Б.В. Гнеденко, И.Н. Коваленко. Чаще всего в зарубежных изданиях встречается понятие «теория очередей».

В большинстве случаев, рассматриваются математические модели потоков событий, когда все события потока доступны для наблюдения. Однако на практике встречаются ситуации, когда наступление события влечет за собой период ненаблюдаемости, который обусловлен тем, что регистрирующий прибор, обрабатывая текущее событие, вводит систему в мертвое время, в течение которого следующие поступившие события теряются. При этом, такие приборы могут быть с продлевающимся мертвым временем и с непродлевающимся.

Задачи по оценке параметров в условиях непродлевающегося мертвого времени фиксированной длины решены: в [4, 5] для простейшего потока событий, в [7] для синхронного альтернирующего потока, в [8, 9, 11] для

асинхронного альтернирующего потока, в [13] для асинхронного потока, в [14, 15] для синхронного потока, в [17, 19] для полусинхронного потока, в [23, 24] для обобщенного полусинхронного потока, были также рассмотрены случаи для МАР-потока, для модулированного синхронного потока, для модулированного обобщенного полусинхронного потока, для модулированного МАР-потока. Задачи по оценке параметров в условиях продлевающегося мертвого времени фиксированной длины решены для пуассоновского потока, для асинхронного альтернирующего потока, для синхронного потока, для полусинхронного потока.

В данной работе рассматривается случай с продлевающимся мертвым временем специального типа [1]. Исчерпывающей характеристикой потока однородных событий является функция распределения или плотность распределения длины интервала между соседними событиями. Однако на практике не всегда удается найти эти характеристики и, зачастую, приходится использовать для описания потока другие. В данной работе рассматриваются такие характеристики наблюдаемого потока, как преобразование Лапласа длительности интервала между соседними событиями и преобразование Лапласа общего периода ненаблюдаемости. В некоторых случаях, с помощью обратного преобразования Лапласа, можно получить явный вид плотности распределения вероятностей.

Цель работы.

По окончании наблюдения потока, необходимо проверить гипотезу о вероятностных характеристиках наблюдаемого потока, полученных в ходе исследования.

Методы исследования.

При выполнении диссертационной работы использовались методы теории вероятностей, теории случайных процессов, теории массового обслуживания, численные методы, а также методы имитационного моделирования.

Научная новизна.

Научная новизна данной работы в рассмотрении и нахождении характеристик однородного потока событий с продлевающимся мертвым временем специального типа.

1 ПОСТАНОВКА ЗАДАЧИ

Рассматривается однородный поток событий, с заданным распределением вероятностей интервала между соседними событиями. Наступление каждого события формирует после себя период мертвого времени случайной длительности θ_i также с заданным распределением вероятностей, в течение которого остальные события недоступны для наблюдения. При этом, эти события вызывают также мертвое время, с теми же вероятностными характеристиками, что и предыдущий период ненаблюдаемости. Особенность данного вида мертвого времени в том, что текущий интервал мертвого времени заканчивается (обрубается), когда достигается следующее событие потока и наблюдаемым становится то событие, которое наступит после окончания последнего периода ненаблюдаемости. Таким образом создается период общего мертвого времени. По окончании общего периода мертвого времени первое наступившее событие снова влечет за собой период ненаблюдаемости и т.д. вариант возникновения такой ситуации приведен на рисунке 1.

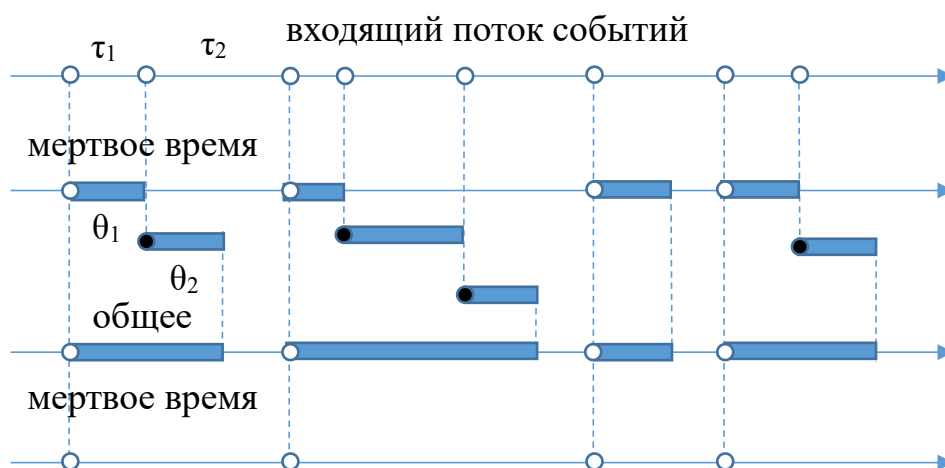


Рис 1. Формирование наблюдаемого потока событий

Обозначим $\tau_i, i=1,2,\dots$ - длительность i -го интервала между соседними событиями входящего потока с заданной плотностью или функцией распределения вероятностей, через $\theta_i, i=1,2,\dots$ - длительность i -го периода мертвого времени с заданным законом распределения вероятностей.

Необходимо найти такие вероятностные характеристики наблюдаемого потока, как преобразование Лапласа интервала между соседними событиями и преобразование Лапласа общего периода ненаблюдаемости, а также рассмотреть частные случаи для конкретных законов распределения.

2 ВЕРОЯТНОСТНЫЕ ХАРАКТЕРИСТИКИ ПОТОКА СОБЫТИЙ С ПРОДЛЕВАЮЩИМСЯ МЕРТВЫМ ВРЕМЕНЕМ СПЕЦИАЛЬНОГО ТИПА.

2.1 Преобразование Лапласа интервала между соседними событиями в наблюдаемом потоке

В качестве вероятностной характеристики интервала между соседними событиями в наблюдаемом потоке будем рассматривать преобразование Лапласа.

Обозначим $\tau_i, i=1,2,\dots$ - длительность i -го интервала между соседними событиями входящего потока с заданной функцией распределения или плотностью распределения вероятностей, $\theta_i, i=1,2,\dots$ - продолжительность i -го периода мертвого времени также с заданной плотностью или функцией распределения вероятностей.

Введем две случайные величины:

$$\beta_i = \tau_i \cdot \chi(\tau_i > \theta_i), \quad \gamma_i = \tau_i \cdot \chi(\tau_i \leq \theta_i), \quad i \geq 1, \quad (1)$$

где $\chi(A)$ - идентификатор множества A и функции:

$$F_\beta(x) = P(0 < \tau_i < x, \tau_i \geq \theta_i), \quad f_\beta(x) = F'_\beta(x), \quad (2)$$

$$F_\gamma(x) = P(0 < \tau_i < x, \tau_i < \theta_i), \quad f_\gamma(x) = F'_\gamma(x), \quad i \geq 1.$$

Обозначим через момент v :

$$v = \min\{i \geq 1 : \gamma_i = 0\}. \quad (3)$$

Тогда длительность интервала между соседними событиями в наблюдаемом потоке τ , учитывая (1) - (3), определяется следующим образом:

$$\tau = \sum_{i=1}^{v-1} \gamma_i + \beta_v. \quad (4)$$

Теорема 1. Преобразование Лапласа $\Phi_\tau(s)$ для плотности распределения вероятностей $f_\tau(x)$ интервала τ между соседними событиями в наблюдаемом потоке из (4) имеет вид

$$\Phi_\tau(x) = \frac{\Phi_\beta(s)}{1 - \Phi_\gamma(s)}, \quad (5)$$

где $\Phi_\beta(s)$ и $\Phi_\gamma(s)$ - преобразования Лапласа для функций $f_\beta(x)$ и $f_\gamma(x)$ соответственно.

$$\Phi_\beta(s) = \int_0^\infty f_\beta(x) e^{-sx} dx \quad (6)$$

$$\Phi_\gamma(s) = \int_0^\infty f_\gamma(x) e^{-sx} dx \quad (7)$$

Доказательство. Обозначим $F_n(x)$ через функцию

$$F_n(x) = P(\gamma_1 + \dots + \gamma_n < x, \gamma_1 > 0, \dots, \gamma_n > 0), f_n(x) = F'_n(x). \quad (8)$$

Тогда

$$f_n(x) = \int_0^x f_{n-1}(u) f_\gamma(x-u) du, \quad n \geq 2. \quad (9)$$

Найдем распределение τ :

$$\begin{aligned} P(\tau < x) &= P(0 < \beta_1 < x) + P(\gamma_1 + \beta_2 < x, \gamma_1 > 0, \beta_2 > 0) + P(\gamma_1 + \gamma_2 + \beta_3 < x, \gamma_1 > 0, \gamma_2 > 0, \beta_3 > 0) + \dots = \\ &= \int_0^x f_\beta(t) dt + \sum_{n=1}^\infty \int_0^x f_n(t) dt \int_0^{x-1} f_\beta(s) ds = \int_0^x f_\beta(t) dt + \int_0^x g(t) dt \int_0^{x-1} f_\beta(s) ds, \end{aligned} \quad (10)$$

где

$$g(t) = \sum_{n=1}^\infty f_n(t). \quad (11)$$

Продифференцируем (10) и получим, что плотность распределения вероятностей $f_\tau(x)$ интервала τ определяется соотношением:

$$f_\tau(x) = f_\beta(x) + \int_0^x g(t) f_\beta(x-t) dt, \quad x > 0 \quad (12)$$

Суммируя (9) по $n = \overline{2, \infty}$ и учитывая соотношение (11), получим уравнение для нахождения $g(t)$:

$$g(x) = f_\gamma(x) + \int_0^x g(t) f_\gamma(x-t) dt. \quad (12)$$

Применяя преобразование Лапласа для (12), получаем уравнение:

$$G(s) = \Phi_\gamma(s) + G(s)\Phi_\gamma(s), \quad (13)$$

разрешая которое относительно $G(s)$, получим

$$G(s) = \frac{\Phi_\gamma(s)}{1 - \Phi_\gamma(s)}. \quad (14)$$

Применяя преобразование Лапласа для плотности распределения вероятностей $f_\tau(x)$ из (12) и учитывая (14), находим

$$\Phi_\tau(s) = \Phi_\beta(s) + G(s)\Phi_\beta(s) = \frac{\Phi_\beta(s)}{1 - \Phi_\gamma(s)}. \quad (15)$$

Так мы получили формулу (5).

Теорема 1 доказана.

2.2. Преобразование Лапласа длительности общего периода ненаблюдаемости

Введем две вспомогательные случайные величины:

$$\tilde{\beta}_i = \theta_i \cdot \chi(\tau_i > \theta_i), \quad \gamma_i = \tau_i \cdot \chi(\tau_i \leq \theta_i), \quad i \geq 1, \quad (16)$$

где $\chi(A)$ - идентификатор множества A и функции:

$$F_{\tilde{\beta}}(x) = P(0 < \tau_i < x, \tau_i \geq \theta_i), \quad f_{\tilde{\beta}}(x) = F'_{\tilde{\beta}}(x), \quad (17)$$

$$F_\gamma(x) = P(0 < \tau_i < x, \tau_i < \theta_i), \quad f_\gamma(x) = F'_\gamma(x), \quad i \geq 1. \quad (18)$$

Обозначим через момент v :

$$v = \min \{i \geq 1 : \gamma_i = 0\}.$$

Тогда длительность интервала между соседними событиями в наблюдаемом потоке τ , определяется так:

$$\theta = \sum_{i=1}^{v-1} \gamma_i + \tilde{\beta}_v. \quad (19)$$

Теорема 2. Преобразование Лапласа $\Phi_\theta(s)$ для плотности распределения вероятностей $f_\theta(x)$ общего периода ненаблюдаемости θ из (25) имеет вид

$$\Phi_\theta(s) = \frac{\Phi_{\tilde{\beta}}(s)}{1 - \Phi_\gamma(s)}, \quad (20)$$

где $\Phi_{\tilde{\beta}}(s)$ и $\Phi_\gamma(s)$ - преобразования Лапласа для функций $f_{\tilde{\beta}}(x)$ и $f_\gamma(x)$.

$$\Phi_{\tilde{\beta}}(s) = \int_0^{\infty} f_{\tilde{\beta}}(x) e^{-sx} dx \quad (21)$$

$$\Phi_{\gamma}(s) = \int_0^{\infty} f_{\gamma}(x) e^{-sx} dx \quad (22)$$

Доказательство. Обозначим $F_n(x)$ через функцию

$$F_n(x) = P(\gamma_1 + \dots + \gamma_n < x, \gamma_1 > 0, \dots, \gamma_n > 0), f_n(x) = F'_n(x). \quad (23)$$

Тогда

$$f_n(x) = \int_0^x f_{n-1}(u) f_{\gamma}(x-u) du, \quad n \geq 2. \quad (24)$$

Найдем распределение τ :

$$\begin{aligned} P(\tau < x) &= P(0 < \tilde{\beta}_1 < x) + P(\gamma_1 + \tilde{\beta}_2 < x, \gamma_1 > 0, \tilde{\beta}_2 > 0) + P(\gamma_1 + \gamma_2 + \tilde{\beta}_3 < x, \gamma_1 > 0, \gamma_2 > 0, \tilde{\beta}_3 > 0) + \dots = \\ &= \int_0^x f_{\tilde{\beta}}(t) dt + \sum_{n=1}^{\infty} \int_0^x f_n(t) dt \int_0^{x-1} f_{\tilde{\beta}}(s) ds = \int_0^x f_{\tilde{\beta}}(t) dt + \int_0^x g(t) dt \int_0^{x-1} f_{\tilde{\beta}}(s) ds, \end{aligned} \quad (25)$$

где

$$g(t) = \sum_{n=1}^{\infty} f_n(t). \quad (26)$$

Дифференцируя равенство (25), получим, что плотность распределения вероятностей $f_{\tau}(x)$ интервала τ определяется соотношением:

$$f_{\tau}(x) = f_{\tilde{\beta}}(x) + \int_0^x g(t) f_{\tilde{\beta}}(x-t) dt, \quad x > 0 \quad (27)$$

Суммируя (24) по $n = \overline{2, \infty}$ и учитывая соотношение (26), получим уравнение для нахождения $g(t)$:

$$g(x) = f_{\gamma}(x) + \int_0^x g(t) f_{\gamma}(x-t) dt. \quad (28)$$

Применяя преобразование Лапласа для (28), получаем уравнение:

$$G(s) = \Phi_{\gamma}(s) + G(s)\Phi_{\gamma}(s), \quad (29)$$

разрешая которое относительно $G(s)$, получим

$$G(s) = \frac{\Phi_{\gamma}(s)}{1 - \Phi_{\gamma}(s)}. \quad (30)$$

Применяя преобразование Лапласа для плотности распределения вероятностей $f_\tau(x)$ из (27) и учитывая (30), находим

$$\Phi_\tau(s) = \Phi_{\tilde{\beta}}(s) + G(s)\Phi_{\tilde{\beta}}(s) = \frac{\Phi_{\tilde{\beta}}(s)}{1 - \Phi_\gamma(s)}. \quad (31)$$

Полученная формула (31) полностью совпадает с формулой (20).
Теорема 2 доказана.

3 ЧАСТНЫЕ СЛУЧАИ

3.1 Длительность мертвого времени, распределенная по экспоненциальному закону

Пусть τ_i и θ_i независимые экспоненциально распределенные случайные величины с параметрами λ и μ соответственно.

Получим вид функции $F_\beta(x)$ из (2) для этого конкретного случая

$$F_\beta(x) = \int_0^\infty \lambda e^{-\lambda t} dt \int_t^\infty \mu e^{-\mu s} ds = \int_0^x \lambda e^{-\lambda t} (1 - e^{-\mu t}) dt, \quad (32)$$

тогда

$$f_\beta(x) = F'_\beta(x) = \lambda(e^{-\lambda x} - e^{-(\lambda+\mu)x}), \quad x > 0. \quad (33)$$

Аналогично найдем вид функции $F_\gamma(x)$ из (2)

$$F_\gamma(x) = \int_0^x \lambda e^{-\lambda t} dt \int_t^\infty \mu e^{-\mu s} ds = \int_0^x \lambda e^{-\lambda t} e^{-\mu t} dt \quad (34)$$

тогда

$$f_\gamma(x) = F'_\gamma(x) = \lambda e^{-(\lambda+\mu)x}, \quad x > 0. \quad (35)$$

Получим преобразование Лапласа $\Phi_\beta(s)$, $\Phi_\gamma(s)$ для функций $f_\beta(x)$ и $f_\gamma(x)$ соответственно из (33) и (35) в виде:

$$\Phi_\beta(s) = \lambda \left(\frac{1}{\lambda + s} - \frac{1}{\lambda + \mu + s} \right) = \frac{\lambda \mu}{(\lambda + s)(\lambda + \mu + s)}, \quad (36)$$

$$\Phi_\gamma(s) = \frac{\lambda}{\lambda + \mu + s}. \quad (37)$$

Используя результат (5) теоремы 1 и формулы (36)-(37), находим преобразование Лапласа $\Phi_\tau(s)$ для плотности распределения вероятностей $f_\tau(x)$

$$\Phi_\tau(s) = \frac{\Phi_\beta(s)}{1 - \Phi_\gamma(s)} = \frac{\lambda \mu}{(\lambda + s)(\mu + s)} = \frac{\lambda \mu}{(\lambda - \mu)} \left(\frac{1}{\mu + s} - \frac{1}{\lambda + s} \right), \quad \lambda \neq \mu \quad (38)$$

С помощью обратного преобразования Лапласа получим явный вид плотности распределения вероятностей интервала между соседними событиями в наблюдаемом потоке

$$f_{\tau}(s) = \frac{\lambda\mu}{\lambda - \mu} (e^{-\mu x} - e^{-\lambda x}), \quad x > 0. \quad (39)$$

Теперь получим явный вид плотности распределения вероятностей общего периода ненаблюдаемости.

Получим вид функции $F_{\tilde{\beta}}(x)$ из (16) для этого конкретного случая:

$$F_{\tilde{\beta}}(x) = \int_0^x \mu e^{-\mu t} dt \int_t^{\infty} \lambda e^{-\lambda s} ds = \int_0^x \mu e^{-(\lambda+\mu)t} dt, \quad (40)$$

тогда

$$f_{\tilde{\beta}}(x) = F'_{\tilde{\beta}}(x) = \mu e^{-(\lambda+\mu)x}, \quad x > 0. \quad (41)$$

Для функции $f_{\gamma}(x)$ справедлива формула (35).

Получим преобразование Лапласа $\Phi_{\tilde{\beta}}(s)$ для функции $f_{\tilde{\beta}}(x)$ из (41) в виде:

$$\Phi_{\tilde{\beta}}(s) = \frac{\mu}{\lambda + \mu + s}. \quad (42)$$

Преобразование Лапласа для функции $f_{\gamma}(x)$ из (35) имеет вид

$$\Phi_{\gamma}(s) = \frac{\lambda}{\lambda + \mu + s} \quad (43)$$

Используя результат (20) теоремы 2 и формулы (42), (43), находим преобразование Лапласа $\Phi_{\theta}(s)$ для плотности распределения вероятностей $f_{\theta}(x)$

$$\Phi_{\theta}(s) = \frac{\Phi_{\tilde{\beta}}(s)}{1 - \Phi_{\gamma}(s)} = \frac{\mu}{\mu + s}, \quad (44)$$

С помощью обратного преобразования Лапласа для формулы (44) получим явный вид плотности распределения вероятностей $f_{\theta}(x)$ общего периода ненаблюдаемости θ :

$$f_{\theta}(x) = \mu e^{-\mu x}, \quad x > 0. \quad (45)$$

3.2 Длительность мертвого времени - константа

Пусть τ_i независимые экспоненциально распределенные случайные величины с параметрами λ , а $\theta_i \equiv T$ где $T = const$.

Тогда функции из (2) для этого конкретного случая имеют вид:

$$f_{\beta}(x) = \begin{cases} \lambda e^{-\lambda x}, & x > T, \\ 0, & x \leq T, \end{cases} \quad (46)$$

$$f_{\gamma}(x) = \begin{cases} 0, & x \leq T, \\ \lambda e^{-\lambda x}, & x > T, \end{cases} \quad (47)$$

Получим преобразование Лапласа $\Phi_{\beta}(s)$, $\Phi_{\gamma}(s)$ для функций $f_{\beta}(x)$ и $f_{\gamma}(x)$ соответственно из (46)-(47) в виде:

$$\Phi_{\beta}(s) = \frac{\lambda}{\lambda + s} e^{-(\lambda+s)T}, \quad \Phi_{\gamma}(s) = \frac{\lambda}{\lambda + s} (1 - e^{-(\lambda+s)T}). \quad (48)$$

Используя результат (5) теоремы 1 и формулы (48), находим преобразование Лапласа $\Phi_{\tau}(s)$ для плотности распределения вероятностей $f_{\tau}(x)$

$$\Phi_{\tau}(s) = \frac{\Phi_{\beta}(s)}{1 - \Phi_{\gamma}(s)} = \frac{\frac{\lambda}{\lambda + s} e^{-(\lambda+s)T}}{1 - \frac{\lambda}{\lambda + s} (1 - e^{-(\lambda+s)T})}. \quad (49)$$

4 ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ПРОСТЕЙШЕГО ПОТОКА СОБЫТИЙ С ПРОДЛЕВАЮЩИМСЯ МЕРТВЫМ ВРЕМЕНЕМ СПЕЦИАЛЬНОГО ТИПА

Метод имитационного моделирования является мощным аппаратом для проектирования и анализа сложных систем и функционирования процессов, протекающих в этих системах. Этот метод стал актуален с появлением ЭВМ, когда появилась возможность работать с имеющимися системами, в то время, как экспериментировать с реальными объектами было недоступно. Имитационное моделирование — это частный случай математического моделирования. Существует класс объектов, для которых по различным причинам не разработаны аналитические модели, создание аналитической модели принципиально невозможно, не разработаны методы решения полученной модели либо решения неустойчивы. В этом случае аналитическая модель заменяется имитатором или имитационной моделью.

Идея имитационного моделирования очень проста, но при этом очень привлекательна. С его помощью можно экспериментировать с уже имеющимися системами или системами, которые используются тогда, когда сделать это на реальных объектах, чаще всего, невозможно.

Имитационное моделирование как метод решения нетривиальных задач сложилось в середине XX в., в связи с появлением ЭВМ. Воспринятое поначалу как своего рода численный метод, предназначенный для решения нетривиальных задач, как «младший брат» аналитического моделирования, оно со временем стало основным, а иногда единственным удобным методом при исследовании и синтезе сложных процессов и систем.

Известно, что правильный естественный эксперимент максимально информативен. Конкуренцию ему составляет эксперимент с компьютерной имитационной моделью. Не говоря о том, что естественный эксперимент порой невозможен или лишен смысла, эксперимент с имитационной моделью может быть исполнен гораздо быстрее и дешевле естественного,

при это быть более информативным. Это и является причиной столь стремительного и повсеместного внедрения имитационного моделирования.

Проблемы реального мира имитационное моделирование решает безопасно, разумно и просто. Данный метод удобен для анализа: он нагляден, понятен и прост для проверки. В различных областях бизнеса, науки имитационное моделирование находит оптимальные решения и дает четкие представления о сложных системах.

Имитационная модель – универсальное средство исследования сложных систем, которое представляет собой алгоритмическое описание поведения отдельных частей системы, отображающих последовательность событий, возникающих в моделируемой системе. Если статистическое моделирование выполняется с использованием имитационной модели, то такое моделирование называется имитационным.

Есть 4 подхода к имитационному моделированию:

- Агентное моделирование. Цель агентных моделей — получить представления о правилах, общей динамике системы, исходя из предположений об индивидуальном поведении её отдельных активных объектов и взаимодействии этих объектов. Агент — некая сущность, которая обладает активностью, автономным поведением, принимающая решения в соответствии с некоторым набором правил, взаимодействующая с окружением, а также самостоятельно изменяющаяся;
- Дискретно - событийное моделирование. Это подход к моделированию, который «отходит» от непрерывной природы событий и рассматривает лишь основные события моделируемой системы. Удобно применять для моделирования производственных процессов. Оно сильнее всего развито и имеет огромную сферу приложений — от логистики и систем массового обслуживания до транспортных и производственных систем;
- Системная динамика. Такой вид моделирования больше всех других помогает понять суть происходящего выявления причинно-следственных связей между объектами и явлениями. На основании системной

динамики строятся модели бизнес-процессов, модели производства, динамики популяции, развития эпидемии;

- Монте-Карло симуляция – задачи решаются численно с помощью моделирования случайных величин.

4.1 Моделирование непрерывной случайной величины методом обратных функций

Пусть α - случайная величина, равномерно распределенная на интервале $(0,1)$, а ξ - искомая случайная величина с функцией распределения $F(x)$, необходимо найти функцию вида $\xi = \varphi(x)$, чтобы генерировать случайную величину ξ .

Пусть сначала $\varphi(x) \uparrow$, тогда имеем:

$$F(x) = P\{\xi < x\} = P\{\varphi(\alpha) < x\} = P\{\alpha < \varphi^{(-1)}(x)\} = \varphi^{(-1)}(x) \quad (50)$$

Поэтому, $\varphi^{(-1)}(x) = F(x)$, $\varphi(x) = F^{-1}(x)$ и $\xi = F^{-1}(\alpha)$.

Пусть $\varphi(x) \downarrow$, тогда имеем:

$$F(x) = P\{\xi < x\} = P\{\varphi(\alpha) < x\} = P\{\alpha > \varphi^{(-1)}(x)\} = 1 - \varphi^{(-1)}(x) \quad (51)$$

При, $x = \varphi(\alpha)$, $\varphi^{(-1)}(x) = \alpha$ $F(x) = 1 - \alpha$ и $x = F^{-1}(1 - \alpha)$,

тогда $\xi = \varphi(\alpha) = F^{-1}(1 - \alpha)$. В силу того, что случайная величина $1 - \alpha$ равномерно распределена на $(0,1)$, формулы, полученные для монотонно возрастающей и монотонно убывающей $\varphi(\alpha)$ равнозначны.

Используя данное выражение, выведем формулу для случайной величины, распределенной по экспоненциальному закону, ее плотность распределения имеет вид:

$\xi \sim p(x) = \lambda e^{-\lambda x}$ при $\lambda > 0$, тогда получим:

$$F(\xi) = \int_0^{\xi} p(x) dx = - \int_0^{\xi} \lambda e^{-\lambda x} d(-\lambda x) = e^{-\lambda \xi} + 1 \quad (52)$$

Учитывая то, что $F(x) = \gamma$ следует, что $e^{-\lambda \xi} + 1 = \gamma$. После вычислений, с учетом (34567654) получаем:

$$\xi = -\frac{1}{\lambda} \ln(1 - \gamma) . \quad (53)$$

Так как рассматриваемый входящий поток является простейшим, то данное выражение будет использовано для дальнейшего моделирования, так как длина интервала между соседними событиями в простейшем потоке имеет экспоненциальное распределение.

4.2 Имитационное моделирование простейшего потока событий при наличии мертвого времени, имеющего экспоненциальное распределение

Примеры работы программы при разных исходных данных представлены на рисунках 2 – 8. В таблицах 1 – 8 представлены результаты программной реализации алгоритма имитационного моделирования простейшего потока событий в условиях продлевающегося мертвого времени, имеющего экспоненциальное распределение.



Рис2. Входные параметры: $\lambda = 2$, $\mu = 3$

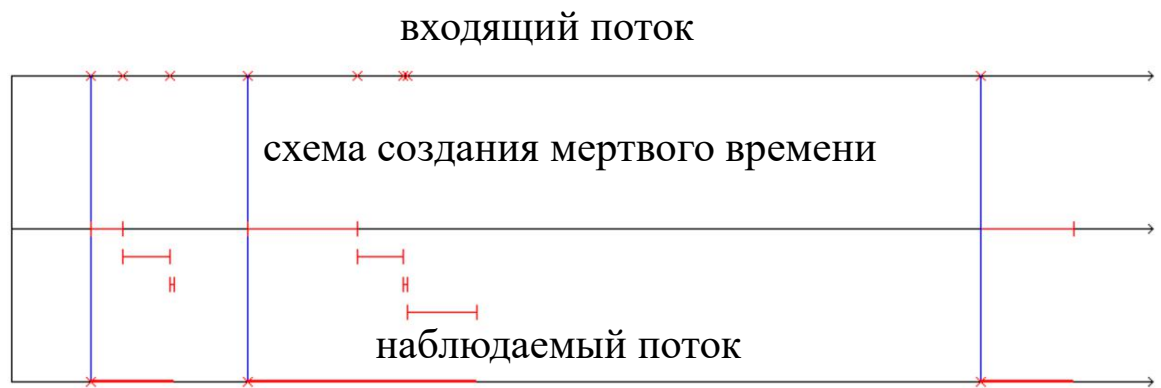


Рис3. Входные параметры: $\lambda = 2$, $\mu = 1$



Рис4. Входные параметры: $\lambda = 3$, $\mu = 3$



Рис5. Входные параметры: $\lambda = 1$, $T = 0,5$



Рис6. Входные параметры: $\lambda = 2$, $\mu = 8$

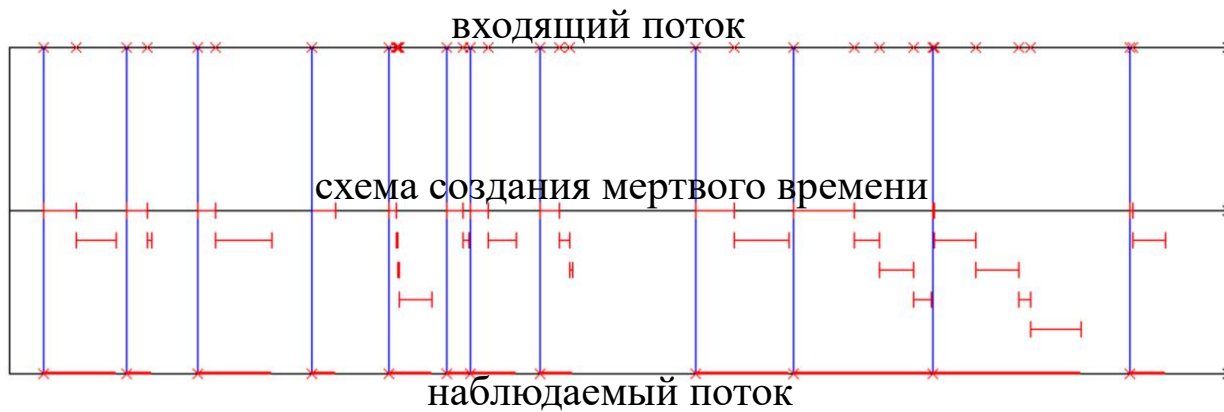


Рис 7. Входные параметры: $\lambda = 4$, $\mu = 3$

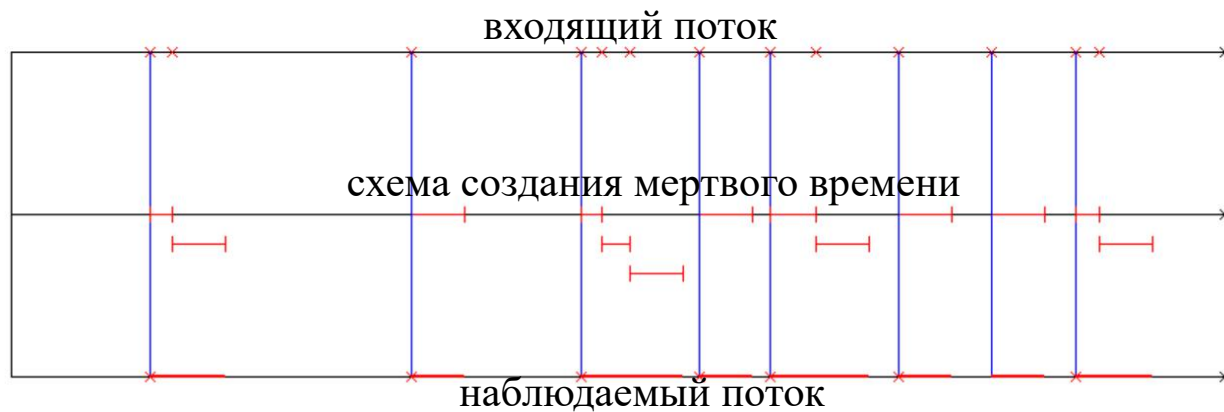


Рис 8. Входные параметры: $\lambda = 2$, $T = 0,3$

4.3 Проверка имитационной модели на адекватность

В общей теории проверки статистических гипотез хи-квадраткритерии занимают важное место. Эти критерии используются для решения различных типовых задач проверки статистических гипотез (включая задачи согласия, однородности, независимости, случайности). Критерий согласия Пирсона или критерий согласия хи-квадрат - это непараметрический метод, позволяющий оценить значимость различий между фактическим (выявленным в результате исследования) количеством исходов или качественных характеристик выборки, попадающих в каждую категорию, и теоретическим количеством, которое можно ожидать в изучаемых группах при справедливости нулевой гипотезы. Выражаясь проще, метод позволяет оценить статистическую значимость различий двух или нескольких относительных показателей (частот, долей). Наиболее часто употребляемый критерий для проверки гипотезы о принадлежности наблюдаемой выборки объёмом n некоторому теоретическому закону распределения. Критерий хи-квадрат для анализа таблиц сопряженности был разработан и предложен в 1900 году английским математиком, статистиком, биологом и философом, основателем математической статистики и одним из основоположников биометрики Карлом Пирсоном (1857-1936).

В первом частном случае был получен явный вид плотности распределения f_{τ} случайной величины τ (интервал между соседними событиями в наблюдаемом потоке) для простейшего потока с экспоненциально распределенным мертвым временем.

В ходе исследования была построена имитационная модель рассматриваемого потока и проведены эксперименты с различными значениями входящих параметров. Была выдвинута гипотеза H_0 , о том, что полученная в ходе экспериментов случайная величина τ имеет распределение f_{τ} . Адекватность построенной модели была проверена с помощью критерия χ^2 .

$$\chi^2 = \sum_{i=1}^k \frac{(n_i - np_i)^2}{np_i}, \quad (54)$$

где k - число разбиений отрезка, n - объем выборки, n_i - количество попаданий случайной величины в i -ый отрезок, p_i - вероятность попадания случайной величины в i -ый отрезок.

Ниже приведены таблицы с разными входящими параметрами λ и μ при различных начальных объемах выборки. При заданном уровне значимости $\alpha = 0.1$ и числе степеней свободы $k - 1 = 9$, критическое значение $\Delta = 2.08$. Все значения статистики не превосходят критического значения 2.08, значит, построенная имитационная модель адекватна заявленному распределению.

N	200	500	1000	1500
χ^2	0,0203108	0,00886018	0,00437406	0,00428756

Таблица 1 – Результат программной реализации имитационного моделирования с параметрами $\lambda=2, \mu = 3$.

N	200	500	1000	1500
χ^2	0,049842	0,00236228	0,00121999	0,000428756

Таблица 2 – Результат программной реализации имитационного моделирования с параметрами $\lambda=1, \mu = 4$.

N	200	500	1000	1500
χ^2	0,146374	0,0974528	0,0638369	0,0052890

Таблица 3 – Результат программной реализации имитационного моделирования с параметрами $\lambda=5, \mu = 2$.

N	200	500	1000	1500
χ^2	0,04738936	0,037648240	0,00121999	0,00193643

Таблица 4 – Результат программной реализации имитационного моделирования с параметрами $\lambda=4$, $\mu = 1$.

N	200	500	1000	1500
χ^2	0,0692302	0,0258184	0,0015724	0,00096272

Таблица 5 – Результат программной реализации имитационного моделирования с параметрами $\lambda=3$, $\mu = 2$.

N	200	500	1000	1500
χ^2	0,1532826	0,0783522	0,00935343	0,00473863

Таблица 6 – Результат программной реализации имитационного моделирования с параметрами $\lambda=8$, $\mu = 3$.

N	200	500	1000	1500
χ^2	0,0725282	0,00236228	0,00021969	0,000528774

Таблица 7 – Результат программной реализации имитационного моделирования с параметрами $\lambda=4$, $\mu = 7$.

N	200	500	1000	1500
χ^2	0,0793497	0,05263842	0,00628352	0,00097253

Таблица 8 – Результат программной реализации имитационного моделирования с параметрами $\lambda=7$, $\mu = 4$.

По результатам статистического эксперимента можно сделать вывод о том, что при увеличении изначального объема выборки, статистика хи-квадрат улучшает свои показатели.

ЗАКЛЮЧЕНИЕ

Таким образом, в данной работе был рассмотрен поток однородных событий, находящегося в условиях продлевающегося мертвого времени специального типа, были получены такие вероятностные характеристики данного потока событий, как

- преобразование Лапласа интервала между соседними событиями в наблюдаемом потоке;
- преобразование Лапласа общего периода ненаблюдаемости.

Также, для конкретных видов распределений вероятностей длительности мертвого времени для простейшего потока были получены обе характеристики. Была построена имитационная модель простейшего потока событий с экспоненциально распределенным мертвым временем и с константным мертвым временем; проведены статистические эксперименты и на их основе, пользуясь полученными в ходе исследования результатами, проверена адекватность построенной модели с помощью критерия хи-квадрат для случая, когда мертвое время распределено экспоненциально.

Полученные результаты можно использовать в различных задачах, связанных с системами массового обслуживания, вычислительными сетями, сетями связи и др.

ЛИТЕРАТУРА

1. Завгородняя М.Е. Вероятностные характеристики потока событий с продлевающимся мертвым временем специального типа // Вестник Томского государственного университета. 2018. № 43. С. 232–239.
2. Курочкин С.С. Многомерные статистические анализаторы. М. : Атомиздат, 1968. 446 с.
3. Апанасович В.В., Коляда А.А., Чернявский А.Ф. Статистический анализ случайных потоков в физическом эксперименте. Минск : Университетское, 1988. 256 с.
4. Gortsev A.M., Klimov I.S. Estimation of intensity of the Poisson stream of events for condition under which it is partially unobservable // Telecommunications and Radio Engineering. 1992. V. 47, No. 1. P. 33–38.
5. Gortsev A.M., Klimov I.S. Estimation of the parameters of an alternating Poisson stream of events // Telecommunications and Radio Engineering. 1993. V. 48, No. 10. P. 40–45.
6. Gortsev A.M., Klimov I.S. Estimation of the non-observability period and intensity of Poisson event flow // Radiotekhnika. 1996. No. 2. P. 8–11.
7. Горцев А.М., Нежелская Л.А. Оценивание длительности мертвого времени и параметров синхронного альтернирующего потока событий // Вестник Томского государственного университета. 2003. № 6. С. 232–239.
8. Горцев А.М., Ниссенбаум О.В. Оценивание длительности мертвого времени и параметров асинхронного альтернирующего потока событий при непродлеваемом мертвом времени // Известия высших учебных заведений. Физика. 2005. № 10. С. 35–49.
9. Горцев А.М., Ниссенбаум О.В. Оценивание длительности мертвого времени и параметров асинхронного альтернирующего потока событий с иницированием лишнего события // Вестник Томского государственного университета. 2004. № 284. С. 137–145.

10. Gortsev A.M., Parshina M.E. Estimation of parameters of an alternate stream of events in “dead” time conditions // Russian Physics Journal. 1999. V. 42, No. 4. P. 373–378.
11. Горцев А.М., Завгородняя М.Е. Оценка параметров альтернирующего потока событий при условии его частичной наблюдаемости // Оптика атмосферы и океана. 1997. Т. 10, № 3. С. 273–280.
12. Bushlanov I.V., Gortsev A.M., Nezhelskaya L.A. Estimating parameters of the synchronous twofold-stochastic flow of events // Automation and Remote Control. 2008. V. 69, No. 9. P. 1517–1533.
13. Горцев А.М., Нежелская Л.А. Оценивание длительности «мертвого времени» и интенсивностей синхронного дважды стохастического потока событий // Радиотехника. 2004. № 10. С. 8–16.
14. Vasileva L.A., Gortsev A.M. Estimation of the dead time of an asynchronous double stochastic flow of events under incomplete observability // Automation and Remote Control. 2003. V. 64, No.12. P. 1890–1898.
15. Vasileva L.A., Gortsev A.M. Estimation of parameters of a double-stochastic flow of events under conditions of its incomplete observability // Automation and Remote Control. 2002. V. 63, No. 3. P. 511–515.
16. Горцев А.М., Куснатдинов Р.Т. Оценивание состояний МС-потока событий при его частичной наблюдаемости // Известия высших учебных заведений. Физика. 1998. № 4. С. 22–30.
17. Gortsev A.M., Nezhelskaya L.A. Estimation of the dead-time period and parameters of a semi-synchronous double-stochastic stream of events // Measurement Techniques. 2003. V. 46, No. 6. P. 536–545.
18. Gortsev A.M., Nezhelskaya L.A. Semisynchronous double stochastic flow of events when the dead time is prolonged // Vychislitel'nye tekhnologii. 2008. V. 13, No. 1. P. 31–41.
19. Нежелская Л.А. Оптимальное оценивание состояний полусинхронного потока событий в условиях его частичной наблюдаемости // Вестник Томского государственного университета. 2000. № 269. С. 95–98.

20. Горцев А.М., Леонова М.А., Нежелская Л.А. Совместная плотность вероятностей длительности интервалов обобщенного асинхронного потока событий при непродлеваемом мертвом времени // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2012. № 4 (21). С. 14–25.
21. Леонова М.А., Нежелская Л.А. Оценка максимального правдоподобия длительности мертвого времени в обобщенном асинхронном потоке событий // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2013. № 2 (23). С. 54–63.
22. Горцев А.М., Леонова М.А., Нежелская Л.А. Сравнение МП- и ММ-оценок длительности мертвого времени в обобщенном асинхронном потоке событий // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2013. № 4 (25). С. 32–42.
23. Горцев А.М., Калягин А.А., Нежелская Л.А. Оценка максимального правдоподобия длительности мертвого времени в обобщенном полусинхронном потоке // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2015. № 1 (30). С. 27–37. 39
24. Горцев А.М., Калягин А.А., Нежелская Л.А. Совместная плотность вероятностей длительности интервалов обобщенного полусинхронного потока событий при непродлеваемом мертвом времени // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2014. № 2 (27). С. 19–29.
25. Gortsev A.M., Solov'ev A.A. Joint probability density of interarrival interval of a flow of physical events with unextendable dead time period // Russian Physics Journal. 2014. V. 57, No. 7. P. 973–983.

Приложение А Программная реализация имитационного моделирования простейшего потока простейшего потока событий при наличии продлевающегося мертвого времени специального типа, имеющего экспоненциальное распределение

```
#include <iostream>
#include <cmath>
#include <stdlib.h>
#include <algorithm>
using namespace std;

double Prost(double r, int la)
{
    double g = (-1 / (double)la)*log(1-r);
    return g;
}
double comp1(const void * a, const void * b)
{
    return (*(double*)a - *(double*)b);
}
double F(double x, double l1, double l2)
{
    double ba = l1 - l2;
    double ea = exp(-x * l1);
    double eb = exp(-x * l2);
    double eab = exp(-x * l1 - x * l2);
    return 1 - ((l1 * eb - l2 * ea) / ba);
}
double F1(double x, double l1)
{
    return 1 - exp(-l1*x);
}

void main()
{
    int n = 1000;
    int flag = 1;
    double l1 = 1;
    double l2 = 4;
    double *prostMas = new double[n];
    prostMas[0] = (double)(rand()) / RAND_MAX;
    double *dead1 = new double[n];
    double *dead2 = new double[n];
    cout << "vhodyashchiy potok:" << endl;
    cout << prostMas[0] << endl;
    for (int i = 1; i < n; i++)//входящий поток
```

```

{
    double r = (double)(rand()) / RAND_MAX;
    if (r != 0 && r != 1)
    {
        double g = Prost(r, l1);
        prostMas[i] = prostMas[i - 1] + g;
        cout << prostMas[i] << endl;
    }
}
cout << endl;
cout << "mertvoe vremya:" << endl;
if (flag == 1)//мервое время- exp
{
    for (int i = 0; i < n; i++)
    {
        double r = ((double)rand() / (RAND_MAX));
        if (r != 0 && r != 1)
        {
            dead1[i] = Prost(r, l2);
            cout << dead1[i] << endl;
        }
    }
}

int dc = 0;
double *np = new double[n];//наблюдаемый поток
np[0] = prostMas[0];
for (int i = 0; i < n; i++)
    dead2[i] = 0;
for (int i = 0; i < n - 1; i++)//продленное мертвое время
{
    if (prostMas[i + 1] - prostMas[i] > dead1[i])//если мв вмещает между
событиями
    {
        dead2[dc] += dead1[i];
        dc++;
        np[dc] = prostMas[i + 1];
    }
    else//если мв выходит за рамки
    {
        dead2[dc] += prostMas[i + 1] - prostMas[i];
    }
}

if (prostMas[n - 1] - prostMas[n - 2] > dead1[n - 2])

```

```

{
    //dc++;
    dead2[dc] = dead1[n - 1];
}

cout << endl;
double *dead3 = new double[dc];
cout << "prodlennoe mertvoe vremya:" << endl;
for (int i = 0; i < dc; i++)
{
    dead3[i] = dead2[i];
    cout << dead2[i] << endl;
}
cout << endl;
cout << "nabludaemyi potok:" << endl;
for (int i = 0; i < dc; i++)
    cout << np[i] << endl;

cout << endl;
double *np1 = new double[dc-1]; //интервалы между событиями в
наблюдаемом потоке

cout << "intervaly mezhdu sobyitiyami v nablyudaемом potoke:" << endl;
for (int i = 0; i < dc - 1; i++)
{
    np1[i] = np[i + 1] - np[i];
}

double min = np1[0], max = np1[0];

for (int i = 0; i < dc - 1; i++)
{
    if (np1[i] > max)
        max = np1[i];
    if (np1[i] < min)
        min = np1[i];
}

double temp;
for (int i = 0; i < dc - 1; i++)
{
    for (int j = 0; j < dc - i - 1; j++) {
        if (np1[j] > np1[j + 1]) {

```

```

        // меняем элементы местами
        temp = np1[j];
        np1[j] = np1[j + 1];
        np1[j + 1] = temp;
    }
}
}
double dlina = max - min;
int k = 10;
double r = dlina / k; // k=k
int *ni = new int[k]; // количество попаданий
for (int i = 0; i < k; i++)
    ni[i] = 0;
for (int i = 0; i < k; i++)
    for (int j = 0; j < dc - 1; j++)
        if (np1[j] >= min + r * i && np1[j] <= min + r * (i + 1))
            ni[i]++;

cout << "kolichество popadaniy v interval:" << endl;
cout << endl;
for (int i = 0; i < k; i++)
    cout << ni[i] << endl;
double f11 = 0;
double hi2 = 0;
double f2, f1, f;

for (int i = 0; i < k; i++)
{
    f1 = F(min + r * (i + 1), l1, l2);
    f2 = F(min + r * (i), l1, l2);
    f = f1 - f2;
    double nf = (dc - 1) * f;
    hi2 += pow((ni[i] - nf), 2) / (nf);
}
cout << hi2;
system("pause");
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

using System.Windows.Forms;

namespace WindowsFormsApp1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        static double Prost(double r, int la)
        {
            double g = (-1 / (double)la) * Math.Log(r);
            return g;
        }
        static double meaning()
        {
            Random rnd = new Random(Guid.NewGuid().GetHashCode());
            return ((double)rnd.NextDouble()) / ((double)rnd.NextDouble() /
uint.MaxValue + 1);
        }
        static double F(double x, double l1, double l2)
        {
            double ba = l2 - l1;
            double ea = Math.Exp(x * l1);
            double eb = Math.Exp(x * l2);
            double eab = Math.Exp(-x * l1 - x * l2);
            return (eab * (l1 * ea + eb * (ba*ea-l2))) / (ba);

//(Math.Exp(-l2*x-l1*x)*(l1*Math.Exp(l1*x))+Math.Exp(l2*x)*(-l2
Math.Exp(l1*x)*(l2-l1))) / (l2 - l1);
        }
        private void button1_Click(object sender, EventArgs e)
        {
            //строит  $x^2/4 + y^2 = 1$ 
            //Graphics g = e.Graphics;
            Graphics g = panel1.CreateGraphics();
            System.Drawing.Pen pen1;
            pen1 = new System.Drawing.Pen(System.Drawing.Color.Red,2);
            System.Drawing.Pen pen2;
            pen2 = new System.Drawing.Pen(System.Drawing.Color.Black);
            g.TranslateTransform(10, 50);
            panel1.Width = 850;
            int dlina = panel1.Width;

```

```

g.DrawLine(Pens.Black, 0, 0, 0, 220); //вертикальная
g.DrawLine(Pens.Black, 0, 0, Width-50, 0);
g.DrawLine(Pens.Black, Width - 53, -3, Width - 50, 0);
g.DrawLine(Pens.Black, Width - 53, 3, Width - 50, 0);

g.DrawLine(Pens.Black, 0, 110, Width-50, 110);
g.DrawLine(Pens.Black, Width - 53, 107, Width - 50, 110);
g.DrawLine(Pens.Black, Width - 53, 113, Width - 50, 110);

g.DrawLine(Pens.Black, 0, 220, Width-50, 220);
g.DrawLine(Pens.Black, Width - 53, 217, Width - 50, 220);
g.DrawLine(Pens.Black, Width - 53, 223, Width - 50, 220);

//int dlina = panel1.Size.Width -60;
g.DrawLine(Pens.Black, Width - 53, -3, Width - 50, 0);
g.DrawLine(Pens.Black, Width - 53, 3, Width - 50, 0);

double la1 = 2;
double la2 = 3;

double t1 = 0;
//double t2 = 0;
int count = 0;
double mash = dlina / 7;
double rasst1 = 0;
double T = 0.5*mash;
int [] sob =new int[500]; //массив событий
int [] times = new int[500]; //массив мертвых времен
int [] itog = new int[500];
int m = 0;

g.TranslateTransform(10, 0);
while (t1 < dlina)
{
    double r1 = meaning();
    if (r1 != 0 && r1 != 1)
    {
        double A = -(1 / la1) * Math.Log(r1);
        A *= mash;
        rasst1 = t1;
        t1 += A;
        sob[count] = (int)t1;
        rasst1 = t1 - rasst1;
        count++; //число событий
    }
}

```

```

    }
}
for (int i = 0; i < count - 1; i++)
{
    g.DrawLine(Pens.Red, -3 + sob[i], -3, 3 + sob[i], 3);
    g.DrawLine(Pens.Red, -3 + sob[i], 3, 3 + sob[i], -3);
}

//int k = 0;

for (int j = 0; j < count; j++)//массив мертвых времен T
    times[j] = (int)T;

//for (int j = 0; j < count; j++)//массив мертвых времен exp
//{
//    double r2 = meaning();
//    double A = -(1 / la2) * Math.Log(r2);
//    A *= mash;
//    times[j] = (int)A;
//}

g.DrawLine(Pens.Red, -3 + sob[0], -3 + 220, 3 + sob[0], 3 + 220);
g.DrawLine(Pens.Red, -3 + sob[0], 3 + 220, 3 + sob[0], -3 + 220);
g.DrawLine(Pens.Blue, sob[0], 0, sob[0], 220);

for (int i = 0; i < count-1;)
{
    if(sob[i] + times[i] >= sob[i + 1])//если мв не влезает
    {
        g.DrawLine(Pens.Red, -3 + sob[i], -3 + 220, 3 + sob[i], 3
+ 220);
        g.DrawLine(Pens.Red, -3 + sob[i], 3 + 220, 3 + sob[i], -3
+ 220);
        int d = 0;
        while ((sob[i] + times[i] >= sob[i + 1]) & sob[i+1] != 0)
        {
            g.DrawLine(Pens.Red, sob[i], 110 + 20*d, sob[i + 1],
110 + 20 * d);
            g.DrawLine(Pens.Red, sob[i], 105 + 20 * d, sob[i],
115 + 20 * d);
        }
    }
}

```

```

+ 1], 115 + 20 * d);
        g.DrawLine(Pens.Red, sob[i + 1], 105 + 20 * d, sob[i
        itog[m] += sob[i + 1] - sob[i];

        g.DrawLine(pen1, sob[i], 220, sob[i + 1], 220);//
        i++;
        d++;
    }
    g.DrawLine(pen1, sob[i], 220, sob[i] + times[i], 220);//
    g.DrawLine(Pens.Red, sob[i], 110 + 20 * d, sob[i] +
times[i], 110 + 20 * d);
    g.DrawLine(Pens.Red, sob[i], 105 + 20 * d, sob[i], 115 +
20 * d);
    g.DrawLine(Pens.Red, sob[i] + times[i], 105 + 20 * d,
sob[i] + times[i], 115 + 20 * d);
    //g.DrawEllipse(pen2, sob[i + 1], 219, 4, 4);
    g.DrawLine(Pens.Red, -3 + sob[i + 1], -3 + 220, 3 + sob[i
+ 1], 3 + 220);
    g.DrawLine(Pens.Red, -3 + sob[i + 1], 3 + 220, 3 + sob[i
+ 1], -3 + 220);

    g.DrawLine(Pens.Blue, sob[i + 1], 0, sob[i + 1], 220);
    itog[m] += sob[i + 1] - sob[i];
    m++;
    i++;
    d = 0;
}
else
{
    g.DrawLine(Pens.Red, sob[i], 110, sob[i] + times[i], 110);
    g.DrawLine(Pens.Red, sob[i], 105, sob[i], 115);
    g.DrawLine(Pens.Red, sob[i] + times[i], 105, sob[i] +
times[i], 115);
    //g.DrawLine(Pens.Red, -3 + sob[i], -3 + 220, 3 + sob[i],
3 + 220);
    //g.DrawLine(Pens.Red, -3 + sob[i], 3 + 220, 3 + sob[i],
-3 + 220);

    g.DrawLine(Pens.Blue, sob[i], 0, sob[i], 220);
    g.DrawLine(Pens.Blue, sob[i + 1], 0, sob[i + 1], 220);
    g.DrawLine(pen1, sob[i], 220, sob[i] + times[i], 220);//
    itog[i] = times[i];
    i++;
    m++;
}
}
la1 = 2;

```

```

la2 = 3;
int n = 500;
double [] prostMas = new double[n];
prostMas[0] = 0;
for (int i = 1; i < n; i++)//ВХОДЯЩИЙ ПОТОК
{
    double r = meaning();
    if (r != 0 && r != 1)
    {
        double g1 = Prost(r, (int)la1);
        prostMas[i] = prostMas[i - 1] + g1;
    }
}
double[] dead1 = new double[n];
double[] dead2 = new double[n];
for (int i = 1; i < n; i++)//мертвое время
{
    double r = meaning();
    if (r != 0 && r != 1)
    {
        double g1 = Prost(r, (int)la2);
        dead1[i] = g1;
    }
}
int dc = 0;
double[] np = new double[n];
np[0] = 0;
for(int i = 0; i < n; i++)
    dead2[i] = 0;
for (int i = 0; i < n - 1; i++)//продленное мертвое время
{
    if (prostMas[i + 1] - prostMas[i] > dead1[i])//если мв
    вмещает между событиями
    {
        dead2[dc] += dead1[i];
        dc++;
        np[dc] = prostMas[i + 1];
    }
    else//если мв выходит за рамки
    {
        dead2[dc] += prostMas[i + 1] - prostMas[i];
    }
}
if (prostMas[n - 1] - prostMas[n - 2] > dead1[n - 2])
{

```

```

        dc++;
        dead2[dc - 1] = dead1[n - 1];
    }
    double[] dead3 = new double[dc - 1]; // массив мертвых
времен(окончательный)
    for (int i = 0; i < dc - 1; i++)
        dead3[i] = dead2[i];

    double[] np1 = new double[dc - 2];
    for (int i = 0; i < dc - 2; i++)
        np1[i] = np[i + 1] - np[i]; // массив интервалов между
событиями в набл потоке
    for (int i = 0; i < dc - 1; i++)
        listBox1.Items.Add(Convert.ToString(np1[i]));

    Array.Sort(np1);
    double minmax = np1.Max() - np1.Min();
    double k = (minmax / 10); // длина каждого отрезка
    int[] kol = new int[10];
    for (int i = 0; i < 10; i++)
        kol[i] = 0; // количество попаданий

    int ko = 0;
    double[] np2 = new double[10]; // значения на границах отрезка
    for (int i = 1; i < 9; i++)
    {
        np2[i] = np1[kol[i] + ko];
        ko += kol[i];
    }
    for (int i = 0; i < 10; i++)
        for (int j = 0; j < dc - 2; j++)
            if (np1[j] >= np1.Min() + k * i && np1[j] < np1.Min() + k
* (i+1))
                kol[i]++;
    double hi2 = 0;

    for (int i = 0; i < 10; i++)
    {
        double f1 = F(np1.Min() + k * (i+1), la1, la2);
        double f2 = F(np1.Min() + k * i, la1, la2);
        double F1 = f1 - f2;
        hi2 += ((kol[i] / n - F1) * (kol[i] / n - F1)) / F1;
    }
    //hi2 = hi2 * n;
    listBox1.Items.Add(Convert.ToString(hi2));

```

}
}
}
}

Отчет о проверке на заимствования №1



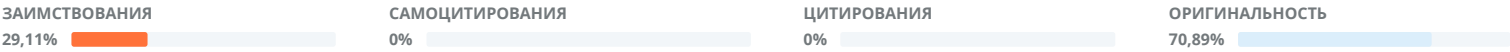
Автор: dashakarlin@gmail.com / ID: 5932358
Проверяющий: dashakarlin@gmail.com / ID: 5932358
Отчет предоставлен сервисом «Антиплагиат»- <http://users.antiplagiat.ru>

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 70
Начало загрузки: 10.06.2020 11:42:54
Длительность загрузки: 00:00:02
Имя исходного файла: вкр.pdf
Название документа: вкр
Размер текста: 1 кБ
Символов в тексте: 46363
Слов в тексте: 5981
Число предложений: 451

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Последний готовый отчет (ред.)
Начало проверки: 10.06.2020 11:42:57
Длительность проверки: 00:01:24
Комментарии: не указано
Модули поиска: Модуль поиска Интернет



Заимствования — доля всех найденных текстовых пересечений, за исключением тех, которые система отнесла к цитированиям, по отношению к общему объему документа.
Самоцитирования — доля фрагментов текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа, по отношению к общему объему документа.
Цитирования — доля текстовых пересечений, которые не являются авторскими, но система посчитала их использование корректным, по отношению к общему объему документа. Сюда относятся оформленные по ГОСТу цитаты; общепотребительные выражения; фрагменты текста, найденные в источниках из коллекций нормативно-правовой документации.
Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.
Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.
Оригинальность — доля фрагментов текста проверяемого документа, не обнаруженных ни в одном источнике, по которым шла проверка, по отношению к общему объему документа.
Заимствования, самоцитирования, цитирования и оригинальность являются отдельными показателями и в сумме дают 100%, что соответствует всему тексту проверяемого документа.
Обращаем Ваше внимание, что система находит текстовые пересечения проверяемого документа с проиндексированными в системе текстовыми источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности заимствований или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

№	Доля в отчете	Доля в тексте	Источник	Ссылка	Актуален на	Модуль поиска	Блоков в отчете	Блоков в тексте
[01]	16,12%	23,75%	Скачать полнотекстовую ве...	http://journals.tsu.ru	18 Сен 2018	Модуль поиска Интернет	56	79
[02]	0%	23,75%	Скачать полнотекстовую ве...	http://journals.tsu.ru	13 Сен 2019	Модуль поиска Интернет	0	79
[03]	3,02%	20,27%	Вероятностные характерист...	http://journals.tsu.ru	24 Окт 2019	Модуль поиска Интернет	13	83

Еще источников: 17
Еще заимствований: 9,97%