

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

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК

Зав.каф. прикладной информатики
д-р.физ.-мат.н., профессор
_____ И.О. Фамилия
« ____ » _____ 2021 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

ИССЛЕДОВАНИЕ ДВУМЕРНОГО ВЫХОДЯЩЕГО ПОТОКА
МАРКОВСКОЙ МОДЕЛИ УЗЛА ОБРАБОТКИ ЗАПРОСОВ С
ПОВТОРНЫМИ ОБРАЩЕНИЯМИ И ВЫЗЫВАЕМЫМИ ЗАЯВКАМИ

по основной образовательной программе подготовки бакалавров
направление подготовки
090303 - Прикладная информатика

Благинин Алексей Леонидович

Руководитель ВКР
доц. каф. прикладной информатики
_____ И.Л. Лапатын
« ____ » _____ 2021 г.

Автор работы
студент группы № 931704
_____ А.Л. Благинин
« ____ » _____ 2021 г.

ОГЛАВЛЕНИЕ

Введение	3
1 Исследование систем с простейшим входящим потоком	6
1.1 RQ-система с суммарным выходящим потоком	6
1.1.1 Уравнения Колмогорова	7
1.1.2 Метод асимптотического анализа	10
1.1.3 Переход к явному распределению вероятности	13
1.2 RQ-система с двумерным выходящим потоком	14
1.2.1 Уравнения Колмогорова	14
1.2.2 Метод асимптотического анализа	15
1.2.3 Переход к явному распределению вероятности	18
1.2.4 Коэффициент корреляции	19
2 Исследование систем с ММРР-потоком	20
2.1 RQ-система с входящим ММРР-потоком и двумерным выходящим потоком	20
2.1.1 Уравнения Колмогорова	20
2.1.2 Метод асимптотического анализа	22
2.1.3 Переход к явному распределению вероятности	27
3 Имитационное моделирование	28
3.1 Объектная модель предметной области	28
3.2 Процесс моделирования	34
3.3 Функционирование элементов системы массового обслуживания	36
3.4 Интерфейс и работа программы	40
3.4.1 Моделирование в реальном времени	41
3.4.2 Ускоренное моделирование	43
3.4.3 Работа с главным графиком	44
3.4.4 Экспорт результатов моделирования	47
3.5 Особенности реализации	49
3.5.1 Генерация псевдослучайных чисел	49
3.5.2 Отображение статуса при ускоренном моделировании	50
4 Численные эксперименты	52
4.1 Проверка стабильности имитационной модели	52
4.2 Сравнение распределений вероятностей с эмпирическим распределением	53

ВВЕДЕНИЕ

Во многих жизненных ситуациях человек сталкивается с необходимостью получить доступ к различным распределенным ресурсам. Ресурсы могут быть разного вида и происхождения — касса в магазине, запись к врачу и многое другое. Помимо повседневной деятельности, такая необходимость в распределенных ресурсах возникает и в других более специализированных сферах жизнедеятельности и науке — протоколы множественного сетевого доступа, автоматизированное производство. Во всех данных ситуациях при доступе к ресурсу существует очередь, которая может быть упорядоченной, либо неупорядоченной.

Теория массового обслуживания занимается изучением и математическим моделированием различных способов организации доступа к распределенному ресурсу [1]. Своим возникновением теория массового обслуживания обязана А.К. Эрлангу, который получил ряд базовых формул для исследования этой области, решая практические задачи оптимизации систем телефонной связи [2]. Терминология, используемая в теории массового обслуживания также взяла свое начало в практических задачах Эрланга — заявка, прибор, требование, буфер и другие.

Классическими моделями массового обслуживания являются модели с очередью и модели с отказами. В моделях с очередью заявка (требование), которая обратилась к прибору (ресурсу) и застала его занятым, встает в очередь для ожидания обслуживания. В моделях с отказами заявка при невозможности получить обслуживание теряется. Существует достаточно много модификаций и комбинаций таких моделей. Во второй половине XX века происходит бурное развитие телекоммуникационных сетей, что приводит к необходимости моделировать и проектировать сети передачи данных различной структуры и назначения. Для этого стали применяться модели систем массового обслуживания с повторными вызовами. В системах такого типа заявка, которая не может получить доступ к ресурсу повторяет попытку захвата ресурса через случайную задержку, а не встает в очередь. В таком случае, в рамках теории массового обслуживания, говорят, что заявка находится на орбите. В англоязычной литературе такие модели называются *retrial queue (RQ)*, большой обзор работ по исследованию которых приведен в монографии [3]. Основные принципы функционирования моделей систем с повторными вызовами изложены в [4, 5]. Такие модели массового обслуживания возникают при стохастическом моделировании многих протоколов связи и локальных сетей. В частности, данные модели использовались при проектировании алгоритмов доступа и устранения заторов транспортных и канальных уровней модели OSI [6, 7, 8], так как позволяют моделировать поведение системы при различных дисциплинах обслуживания и параметрами.

Помимо повторных вызовов, существуют ситуации (например, сценарий *call-центра*), когда обслуживающие единицы имеют возможность делать исходящие запросы

на обслуживание в тот период времени, когда они находятся в простое. Эта модель организации известна как парная коммутация. В [9] получены интегральные формулы для частичных производящих функций и явные выражения для ожидаемого значения некоторых характеристик производительности системы с повторными вызовами и двухсторонней связью (парной коммутацией) в предположении, что длительности входящих и исходящих вызовов соответствуют одинаковому распределению времени обслуживания. Однако на практике это предположение носит ограничительный характер, поскольку разные типы клиентов обычно демонстрируют разное поведение и, следовательно, у них должны быть разные потребности в обслуживании. В [10] использовали метод анализа среднего значения для получения некоторых ожидаемых значений, связанных со временем ожидания в очереди на повторное обращение с двусторонней связью и различным распределением времени обслуживания входящих и исходящих вызовов.

Особенность RQ-систем с повторными вызовами заключается в том, что в них представлены разные типы заявок, что порождает множество новых дисциплин обслуживания, что, в свою очередь, является мощным инструментом при проектировании и оптимизации систем с множественным случайным доступом к ресурсу.

Также, в современных телекоммуникационных сетях возникают точечные процессы с изменяющейся скоростью поступления заявок. Для моделирования таких процессов в рамках теории массового обслуживания используется процесс Пуассона с марковской модуляцией (ММРР) [11, 12], потому что он имеет механизм для учета временной неоднородности скорости поступления заявок, но дает аналитически поддающиеся обработке результаты организации очередей [13]. По этой причине ММРР широко используется в исследованиях сети Интернет, в частности, при помощи ММРР в [14] была построена модель трафика, которая точно аппроксимирует LRD (Long Range Dependence) характеристики трассировок интернет-трафика. С использованием понятий сеансов и потоков, предлагаемая модель ММРР имитирует реальное иерархическое поведение процесса генерации пакетов пользователями сети Интернет. Благодаря своей иерархической структуре предлагаемая модель одновременно простой и интуитивно понятный: он позволяет генерировать трафик с желаемыми характеристиками, легко устанавливая несколько входных параметров, которые имеют четкий физический смысл. Результаты доказывают, что поведение трафика в очередях, генерируемое моделью ММРР согласуется с моделью, созданной реальными следами пакетов, собранных на граничном маршрутизаторе при различных сценариях и нагрузке.

Несмотря на большое количество исследований в рассматриваемой области, наименее изученной составляющей RQ-систем является выходящий поток заявок, иначе, — заявки, покидающие систему по завершении обслуживания. Сведения о выходящем потоке крайне важны, так как в целом характеризуют работу системы и являются показательными в задачах, связанных с оптимизацией обслуживания. Усложняют анализ выходящего потока наличие разных типов заявок — объем обслуженных заявок одного

типа напрямую влияет на другой. При различных параметрах обслуживания результат работы системы и степень зависимости выходящих потоков заявок могут существенно разниться.

В данной работе будут рассматриваться несколько моделей RQ-систем: с суммарных выходящим потоком и двумерным, а так же модель системы, интенсивность поступления заявок в котором периодически меняется. Целью данной работы является исследование корреляции в двумерном выходящем потоке модели узла обработки запросов с повторными обращениями, вызываемыми заявками и разными моделями входящего потока обращений. В рамках данной цели поставлены следующие задачи:

1. Построить математические модели функционирования узла обработки запросов с повторными обращениями, вызываемыми заявками и различными входящими потоками обращений.
2. Для предложенных моделей с помощью метода асимптотического анализа получить аналитические формулы аппроксимации двумерной характеристической функции числа обслуженных заявок входящего потока и обслуженных вызываемых требований.
3. На основании полученных аппроксимаций характеристической функции реализовать алгоритм вычисления распределения вероятностей, коэффициента корреляции и других числовых характеристик двумерного случайного процесса числа обслуженных заявок входящего потока и обслуженных вызываемых требований
4. Разработать имитационную модель предложенных математических моделей узлов обработки запросов.
5. Реализовать предложенную имитационную модель математических моделей узлов обработки запросов.
6. С помощью имитационной модели сделать оценку области применимости асимптотических результатов.
7. Провести численный анализ коэффициента корреляции компонент двумерного выходящего потока предложенных математических моделей узла обработки запросов.

1 Исследование систем с простейшим входящим потоком

В этом разделе предлагаются к рассмотрению системы с повторными вызовами и вызываемыми заявками, заявки в которые приходят посредством простейшего (Пуассоновского) потока.

Рассмотрим общий вид RQ-системы с простейшим входящим потоком:

На вход поступает простейший поток заявок с интенсивностью λ . Заявка входящего потока, поступая в систему и обнаруживая прибор свободным, занимает его. Прибор, в свою очередь, начинает обслуживание в течение случайного времени, распределенного экспоненциально с параметром μ_1 . Если же при поступлении в систему заявка обнаруживает прибор занятым, она мгновенно уходит на орбиту и осуществляет там случайную задержку в течение экспоненциально распределенного времени с параметром σ . В свободное от обслуживания заявок с входящего потока время прибор сам вызывает заявки с интенсивностью α и обслуживает их в течение экспоненциально-распределенного времени с параметром μ_2 .

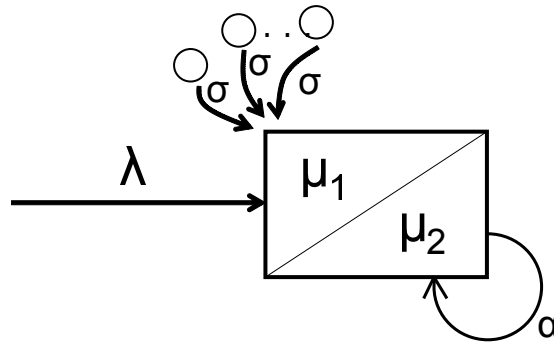


Рисунок 1 — Общий вид RQ-системы с простейшим входящим потоком

Введем следующие обозначения: $i(t)$ — число заявок на орбите в момент времени t , $k(t)$ — состояние прибора: 0 — прибор свободен, 1 — прибор занят обслуживанием заявки входящего потока, 2 — прибор занят обслуживанием вызванной заявки.

Поскольку, целью исследования является характеристика работы выходящего потока системы, будет рассмотрено два варианта предложенной модели — с суммарным и двумерным выходящими потоками.

1.1 RQ-система с суммарным выходящим потоком

Суммарный выходящий поток RQ-системы подразумевает, что приходящие заявки и заявки, вызываемые прибором самостоятельно, являются однородными, поэтому результат работы системы рассматривается как совокупность обслуженных заявок входящего потока и вызванных.

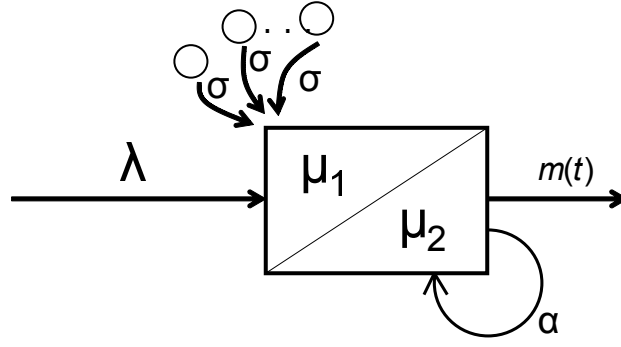


Рисунок 2 — Модель RQ-системы с суммарным выходящим потоком

Введем обозначение: $m(t)$ — число обслуженных заявок в момент времени t .

1.1.1 Уравнения Колмогорова

Итак, мы имеем три характеристики, определяющие результат функционирования системы за некоторое время t : состояние прибора — $k(t)$, количество заявок на орбите — $i(t)$ и количество обслуженных заявок — $m(t)$, что можно представить в виде трехмерного Марковского процесса

$$\{k(t), i(t), m(t)\}$$

Заметим, что именно такая комбинация характеристик будет являться Марковским процессом, так как даёт достаточно информации о том, какое состояние система примет в следующий момент времени. Для этого необходимо знать, в каком состоянии прибор был в предшествующий момент времени, и какое количество заявок находилось в источнике повторных вызовов. Следующее состояние, которое прибор может принять, зависит от состояния, в котором он находился прежде, то есть, каждое из трех состояний $k(t)$ принимается прибором с некоторыми вероятностями. Введем их в рассмотрение

$$P\{k(t) = 0, i(t) = i, m(t) = m\} = P_0(i, m, t)$$

$$P\{k(t) = 1, i(t) = i, m(t) = m\} = P_1(i, m, t)$$

$$P\{k(t) = 2, i(t) = i, m(t) = m\} = P_2(i, m, t)$$

Теперь, необходимо определить вероятность перехода в каждое из трех состояний. Для этого обратимся к модели и рассмотрим $k(t) = 0$, то есть, состояние, в котором прибор свободен. Чтобы в следующий момент времени $t + \Delta t$ прибор был свободен необходимо выполнение одного из следующих условий:

1. Прибор был свободным в момент времени t и к моменту времени $t + \Delta t$ прибор не вызвал заявку, не пришла заявка с входящего потока и не было обращений заявок с орбиты.

2. На момент времени $t + \Delta t$ прибор закончил обслуживание заявки с входящего потока.
3. На момент времени $t + \Delta t$ прибор закончил обслуживание вызываемой заявки.

Таким образом, применяя формулу полной вероятности, вероятность того, что прибор окажется свободен в момент времени $t + \Delta t$ равна сумме вероятностей наступления вышеперечисленных условий:

$$P_0(i, m, t + \Delta t) = P_0(i, m, t)(1 - \alpha\Delta t)(1 - i\sigma\Delta t)(1 - \lambda\Delta t) + \\ + P_1(i, m - 1, t)\mu_1\Delta t + P_2(i, m - 1, t)\mu_2\Delta t$$

Аналогично для принятия состояния, в котором прибор занят, необходимо выполнение одного из следующих условий:

1. Прибор был занят в момент времени t , и к моменту времени $t + \Delta t$ прибор не закончил обслуживание заявки, и не пришла заявка с входящего потока.
2. Прибор был свободен, и пришла заявка с входящего потока.
3. Прибор был свободен, и повторно обратилась заявка с орбиты.
4. Прибор был занят, и пришла заявка с входящего потока, сразу переместившись на орбиту.

В результате получим следующее равенство:

$$P_1(i, m, t + \Delta t) = P_1(i, m, t)(1 - \lambda\Delta t)(1 - \mu_1\lambda\Delta t) + P_0(i, m, t)\lambda\Delta t + \\ + P_0(i + 1, m, t)(i + 1)\sigma\Delta t + P_1(i - 1, m, t)\lambda\Delta t + o(t)$$

Вероятность принятия прибором состояния обслуживания вызываемой заявки является суммой следующих вероятностей:

1. Прибор обслуживал вызываемую заявку, и к моменту времени $t + \Delta t$ прибор не закончил обслуживание вызываемой заявки, и не пришла заявка с входящего потока.
2. Прибор обслуживал вызываемую заявку, и пришла заявка с входящего потока. Поскольку прибор занят, она ушла на орбиту.
3. Прибор был свободен, и он вызвал заявку.

Получим равенство:

$$P_2(i, m, t + \Delta t) = P_2(i, m, t)(1 - \alpha\Delta t)(1 - \lambda\Delta t)(1 - \mu_2\Delta t) + \\ + P_2(i - 1, m, t)\lambda\Delta t + P_0(i, m, t)\alpha\Delta t + o(t)$$

Запишем получившуюся систему уравнений

$$\begin{aligned}
P_0(i, m, t + \Delta t) &= P_0(i, m, t)(1 - \alpha\Delta t)(1 - i\sigma\Delta t)(1 - \lambda\Delta t) + \\
&\quad + P_1(i, m - 1, t)\mu_1\Delta t + P_2(i, m - 1, t)\mu_2\Delta t, \\
P_1(i, m, t + \Delta t) &= P_1(i, m, t)(1 - \lambda\Delta t)(1 - \mu_1\lambda\Delta t) + P_0(i, m, t)\lambda\Delta t + \\
&\quad + P_0(i + 1, m, t)(i + 1)\sigma\Delta t + P_1(i - 1, m, t)\lambda\Delta t + o(t), \\
P_2(i, m, t + \Delta t) &= P_2(i, m, t)(1 - \alpha\Delta t)(1 - \lambda\Delta t)(1 - \mu_2\Delta t) + \\
&\quad + P_2(i - 1, m, t)\lambda\Delta t + P_0(i, m, t)\alpha\Delta t + o(t).
\end{aligned}$$

Так как почти все слагаемые содержат Δt , разделим систему на Δt и сделаем предельный переход $\Delta t \rightarrow 0$

$$\begin{aligned}
\frac{\partial P_0(i, m, t)}{\partial t} &= -(\lambda + i\sigma + \alpha)P_0(i, m, t) + P_1(i, m - 1, t)\mu_1 + P_2(i, m - 1, t)\mu_2, \\
\frac{\partial P_1(i, m, t)}{\partial t} &= -(\lambda + \mu_1)P_1(i, m, t) + (i + 1)\sigma P_0(i + 1, m, t) + \lambda P_0(i, m, t), \\
\frac{\partial P_2(i, m, t)}{\partial t} &= -(\lambda + \mu_2)P_2(i, m, t) + \lambda P_2(i - 1, m, t) + \alpha P_0(i, m, t).
\end{aligned} \tag{1}$$

Полученная система уравнений – система дифференциальных уравнений Колмогорова, где в левой части каждого уравнения находится производная вероятности состояния рассматриваемого процесса, а в правой – сумма произведений вероятностей состояний, из которых прибор может принять это состояние, на интенсивности соответствующих потоков заявок. Решением данной системы будут являться вероятности всех состояний прибора в виде функций времени. Таким образом, задача сводится к решению данной системы дифференциальных уравнений. Решить данную систему аналитически не получится, так как это система бесконечного числа дифференциальных конечно-разностных уравнений с переменными коэффициентами. Для того, чтобы перейти к конечному числу уравнений, введем частные характеристические функции, обозначив $j = \sqrt{-1}$,

$$H_k(u_1, u, t) = \sum_{i=0}^{\infty} \sum_{m=0}^{\infty} e^{ju_1 i} e^{jum} P_k(i, m, t).$$

Тогда перепишем систему (1) в виде

$$\begin{aligned}
\frac{\partial H_0(u_1, u, t)}{\partial t} &= -(\lambda + \alpha)H_0(u_1, u, t) + j\sigma \frac{\partial H_0(u_1, u, t)}{\partial u_1} + \\
&\quad + \mu_1 e^{ju} H_1(u_1, u, t) + \mu_2 e^{ju} H_2(u_1, u, t), \\
\frac{\partial H_1(u_1, u, t)}{\partial t} &= -(\lambda + \mu_1)H_1(u_1, u, t) - j\sigma e^{-ju_1} \frac{\partial H_0(u_1, u, t)}{\partial u_1} + \\
&\quad + \lambda H_0(u_1, u, t) + \lambda e^{ju_1} H_1(u_1, u, t), \\
\frac{\partial H_2(u_1, u, t)}{\partial t} &= -(\lambda + \mu_2)H_2(u_1, u, t) + \lambda e^{ju_1} H_2(u_1, u, t) + \\
&\quad + \alpha H_0(u_1, u, t).
\end{aligned} \tag{2}$$

Таким образом, мы получили ровно три дифференциальных уравнения в частных производных с переменными коэффициентами.

1.1.2 Метод асимптотического анализа

Полученную систему дифференциальных уравнений в частных производных (2) будем решать методом асимптотического анализа [15] в предельном условии большой задержки заявок на орбите ($\sigma \rightarrow 0$). Обозначим $\epsilon = \sigma$, $u_1 = \epsilon w$, $F_k(w, u, t, \epsilon) = H_k(u_1, u, t)$, тогда система запишется в виде

$$\begin{aligned}\frac{\partial F_0(w, u, t, \epsilon)}{\partial t} &= -(\lambda + \alpha)F_0(w, u, t, \epsilon) + j\frac{\partial F_0(w, u, t, \epsilon)}{\partial w} + \\ &\quad + \mu_1 e^{ju} F_1(w, u, t, \epsilon) + \mu_2 e^{ju} F_2(w, u, t, \epsilon), \\ \frac{\partial F_1(w, u, t, \epsilon)}{\partial t} &= -(\lambda + \mu_1)F_1(w, u, t, \epsilon) - j e^{-j\epsilon w} \frac{\partial F_0(w, u, t, \epsilon)}{\partial w} + \\ &\quad + \lambda F_0(w, u, t, \epsilon) + \lambda e^{j\epsilon w} F_1(w, u, t, \epsilon), \\ \frac{\partial F_2(w, u, t, \epsilon)}{\partial t} &= -(\lambda + \mu_2)F_2(w, u, t, \epsilon) + \lambda e^{j\epsilon w} F_2(w, u, t, \epsilon) + \\ &\quad + \alpha F_0(w, u, t, \epsilon).\end{aligned}\tag{3}$$

Заметим, что с помощью введенных функций можно записать характеристическую функцию процесса $m(t)$ в следующем виде

$$M\{\exp(jum(t))\} = \sum_{k=0}^2 H_k(0, u_1, u_2, t) = \sum_{k=0}^2 F_k(0, u_1, u_2, t, \epsilon).$$

Теорема 1. *Асимптотическое приближение характеристической функции числа обслуженных заявок за некоторое время t имеет вид*

$$\begin{aligned}\mathbf{F}(u, t) &= \lim_{\sigma \rightarrow 0} M\{\exp(jum(t))\} = \\ &= \lim_{\epsilon \rightarrow 0} \sum_{k=0}^2 F_k(0, u, t, \epsilon) = \mathbf{R} \cdot \exp\{G(u)t\} \cdot \mathbf{E}\end{aligned}$$

где

$$\mathbf{G}(u) = \begin{bmatrix} -(\lambda + \alpha + \kappa) & \mu_1 e^{ju} & \mu_2 e^{ju} \\ \kappa + \lambda & -\mu_1 & 0 \\ \alpha & 0 & -\mu_2 \end{bmatrix}^T,$$

вектор-строка $\mathbf{R} = \{R_0, R_1, R_2\}$ — стационарное распределение вероятности состояния прибора

$$\mathbf{R} = \left\{ \frac{\mu_2(\mu_1 - \lambda)}{\mu_1(\mu_2 - \alpha)}, \frac{\lambda}{\mu_1}, \frac{\alpha(\mu_1 - \lambda)}{\mu_1(\mu_2 + \alpha)} \right\},$$

κ — нормированное среднее число заявок на орбите

$$\kappa = \frac{\lambda(\lambda\mu_2 + \alpha\mu_1)}{\mu_2(\mu_1 - \lambda)},$$

а \mathbf{E} — единичный вектор-столбец соответствующей размерности.

Доказательство. Делая предельный переход $\lim_{\epsilon \rightarrow 0} F_k(w, u, t, \epsilon) = F_k(w, u, t)$ в полученной системе (3), система уравнений будет записана в виде

$$\begin{aligned}\frac{\partial F_0(w, u, t)}{\partial t} &= -(\lambda + \alpha)F_0(w, u, t) + j\frac{\partial F_0(w, u, t)}{\partial w} + \\ &\quad + \mu_1 e^{ju} F_1(w, u, t) + \mu_2 e^{ju} F_2(w, u, t), \\ \frac{\partial F_1(w, u, t)}{\partial t} &= -(\lambda + \mu_1)F_1(w, u, t) - j\frac{\partial F_0(w, u, t)}{\partial w} + \\ &\quad + \lambda F_0(w, u, t) + \lambda F_1(w, u, t), \\ \frac{\partial F_2(w, u, t)}{\partial t} &= -(\lambda + \mu_2)F_2(w, u, t) + \lambda F_2(w, u, t) + \\ &\quad + \alpha F_0(w, u, t).\end{aligned}\tag{4}$$

Решение системы (4) будет получено в следующей форме

$$F_k(w, u, t) = \Phi(w)F_k(u, t).\tag{5}$$

$\Phi(w)$ — асимптотическое приближение характеристической функции числа заявок на орбите при условии большой задержки на орбите.

Подставив (5) в систему (4) и разделив обе части уравнений на $\Phi(w)$, получим

$$\begin{aligned}\frac{\partial F_0(u, t)}{\partial t} &= -(\lambda + \alpha)F_0(u, t) + j\frac{\Phi'(w)}{\Phi(w)}F_0(u, t) + \\ &\quad + \mu_1 e^{ju} F_1(u, t) + \mu_2 e^{ju} F_2(u, t), \\ \frac{\partial F_1(u, t)}{\partial t} &= -(\lambda + \mu_1)F_1(u, t) - j\frac{\Phi'(w)}{\Phi(w)}F_0(u, t) + \\ &\quad + \lambda F_0(u, t) + \lambda F_1(u, t), \\ \frac{\partial F_2(u, t)}{\partial t} &= -(\lambda + \mu_2)F_2(u, t) + \lambda F_2(u, t) + \\ &\quad + \alpha F_0(u, t).\end{aligned}\tag{6}$$

Заметим, что w содержится только в отношении $\frac{\Phi'(w)}{\Phi(w)}$, а остальные слагаемые и левые части уравнений не зависят от w . Это означает, что $\Phi(w)$ имеет вид экспоненты. Учитывая, что $\Phi(w)$ имеет смысл асимптотического приближения характеристической функции числа заявок на орбите, мы можем конкретизировать вид данной функции

$$\frac{\Phi'(w)}{\Phi(w)} = \frac{e^{j\kappa w} j\kappa}{e^{j\kappa w}},\tag{7}$$

где κ — нормированное среднее число заявок на орбите, которое было получено в [15] и имеет вид

$$\kappa = \frac{\lambda(\lambda\mu_2 + \alpha\mu_1)}{\mu_2(\mu_1 - \lambda)}.$$

Исходя из этого, система (6) примет следующий вид

$$\begin{aligned}\frac{\partial F_0(u, t)}{\partial t} &= -(\lambda + \alpha + \kappa)F_0(u, t) + \mu_1 e^{ju} F_1(u, t) + \mu_2 e^{ju} F_2(u, t), \\ \frac{\partial F_1(u, t)}{\partial t} &= (\lambda + \kappa)F_0(u, t) - \mu_1 F_1(u, t) + 0F_2(u, t), \\ \frac{\partial F_2(u, t)}{\partial t} &= \alpha F_0(u, t) + 0F_1(u, t) - \mu_2 F_2(u, t).\end{aligned}\tag{8}$$

Введем следующие обозначения

$$\begin{aligned}\mathbf{F}(u, t) &= \{F_0(u, t), F_1(u, t), F_2(u, t)\}, \\ \mathbf{G}(u) &= \begin{bmatrix} -(\lambda + \alpha + \kappa) & \mu_1 e^{ju} & \mu_2 e^{ju} \\ \kappa + \lambda & -\mu_1 & 0 \\ \alpha & 0 & -\mu_2 \end{bmatrix}^T,\end{aligned}$$

$\mathbf{G}(u)$ — транспонированная матрица коэффициентов системы (6). Тогда получим следующее матричное уравнение

$$\frac{\partial \mathbf{F}(u, t)}{\partial t} = \mathbf{F}(u, t) \mathbf{G}(u),$$

общее решение которого имеет вид

$$\mathbf{F}(u, t) = \mathbf{C} e^{\mathbf{G}(u)t}.\tag{9}$$

Для того, чтобы получить единственное решение, которое соответствует поведению рассматриваемой системы, примем в рассмотрение начальное условие

$$\mathbf{F}(u, 0) = \mathbf{R},\tag{10}$$

где вектор–строка \mathbf{R} — стационарное распределение вероятности состояния прибора, то есть процесса $k(t)$, которое имеет форму [15]

$$\mathbf{R} = \left\{ \frac{\mu_2(\mu_1 - \lambda)}{\mu_1(\mu_2 - \alpha)}, \frac{\lambda}{\mu_1}, \frac{\alpha(\mu_1 - \lambda)}{\mu_1(\mu_2 + \alpha)} \right\}.$$

Описав начальное условие, мы можем перейти к решению задачи Коши (9,10).

Поскольку нас интересует распределение вероятностей количества заявок в выходном процессе, необходимо найти маргинальное распределение. Для этого суммируем компоненты вектор–строки $\mathbf{F}(u, t)$ по k и умножаем результат на единичный вектор–столбец \mathbf{E} . Получим

$$\mathbf{F}(u, t) \mathbf{E} = \mathbf{R} e^{\mathbf{G}(u)t} \mathbf{E}.\tag{11}$$

Эта формула позволяет найти асимптотическое приближение характеристической функции количества заявок, обслуженных системой к некоторому моменту времени t . Другими словами, формула (11) является решением рассматриваемой системы. \square

1.1.3 Переход к явному распределению вероятности

Полученная характеристическая функция (11) так же, как и распределение вероятности полностью описывает процесс $m(t)$, однако делает это в неявном виде. Поэтому, для использования полученной формулы для вычислений необходимо получить из нее распределение вероятности. Но прежде заметим, что в полученной формуле (11) содержится матричная экспонента, вычислить которую в ее исходном виде не получится. Для преобразования экспоненты применим преобразование подобия [16], которое выглядит следующим образом

$$\mathbf{G}(u) = \mathbf{T}(u)\mathbf{GJ}(u)\mathbf{T}(u)^{-1},$$

где $\mathbf{T}(u)$ – матрица собственных векторов матрицы $\mathbf{G}(u)$, а $\mathbf{GJ}(u)$ – диагональная матрица, содержащая собственные числа матрицы $\mathbf{G}(u)$. Данное преобразование справедливо для любой степени m некоторой матрицы \mathbf{A}^m , из чего следует, что оно так же справедливо для матричной экспоненты [17]:

$$e^{\mathbf{G}(u)t} = \mathbf{T}(u) \cdot \begin{bmatrix} e^{t\Lambda_1(u)} & 0 & 0 \\ 0 & e^{t\Lambda_2(u)} & 0 \\ 0 & 0 & e^{t\Lambda_3(u)} \end{bmatrix} \cdot \mathbf{T}(u)^{-1},$$

где Λ_n – собственное число матрицы $\mathbf{G}(u)$. Тогда распределение примет следующий вид

$$\mathbf{F}(u, t) = \mathbf{R} \cdot \mathbf{T}(u) \cdot \begin{bmatrix} e^{t\Lambda_1(u)} & 0 & 0 \\ 0 & e^{t\Lambda_2(u)} & 0 \\ 0 & 0 & e^{t\Lambda_3(u)} \end{bmatrix} \cdot \mathbf{T}(u)^{-1} \cdot \mathbf{E}.$$

Для того, чтобы получить явное распределение числа обслуженных заявок, воспользуемся свойством характеристической функции, из которого следует, что распределение всегда восстанавливается по характеристической функции. Для обращения функции применим обратное преобразование Фурье для дискретных случайных величин

$$P(m, t) = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{-i \cdot u \cdot m} \mathbf{F}(u, t) du. \quad (12)$$

Полученное распределение характеризует вероятность обслуживания m заявок к моменту времени t в рассматриваемой системе.

1.2 RQ–система с двумерным выходящим потоком

В этом разделе будет рассмотрена модель RQ–системы, выходящий поток заявок которой, является двумерным. Другими словами, в системе обслуживаются два типа заявок — пришедшее извне и вызванные. Исходя из этого добавим к имеющимся обозначениям следующие: $m_1(t)$ — число обслуженных заявок входящего потока к моменту времени t , $m_2(t)$ — обслуженных вызванных заявок к моменту времени t .

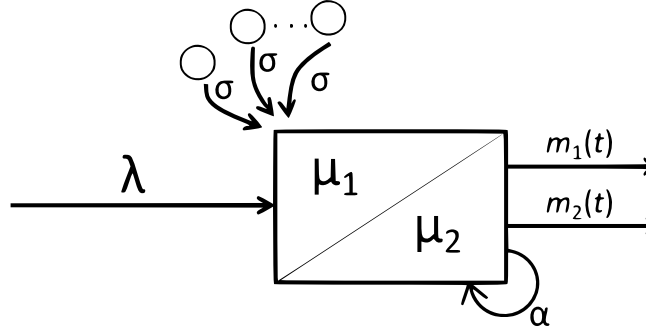


Рисунок 3 — Модель RQ–системы с двумерным выходящим потоком

Функционирование данной системы аналогично рассмотренной ранее, поэтому процесс ее решения будет схож.

1.2.1 Уравнения Колмогорова

В данном случае, мы имеем четыре характеристики, определяющие результат функционирования системы за некоторое время $i(t)$: состояние прибора — $k(t)$, количество заявок на орбите — $i(t)$, количество обслуженных заявок входящего потока — $m_1(t)$, количество обслуженных вызванных заявок — $m_2(t)$, что можно представить в виде четырех–мерного Марковского процесса

$$\{k(t), i(t), m_1(t), m_2(t)\}$$

Аналогично системе с суммарным выходящим потоком, следующее состояние зависит от предыдущего, вероятность принятия прибором одного из трех состояний задается, как

$$P\{k(t) = 0, i(t) = i, m_1(t) = m_1, m_2(t) = m_2\} = P_0(i, m_1, m_2, t)$$

$$P\{k(t) = 1, i(t) = i, m_1(t) = m_1, m_2(t) = m_2\} = P_1(i, m_1, m_2, t)$$

$$P\{k(t) = 2, i(t) = i, m_1(t) = m_1, m_2(t) = m_2\} = P_2(i, m_1, m_2, t)$$

Запишем систему уравнений Колмогорова, составленную на основе введенных вероятностей перехода

$$\begin{aligned}
\frac{\partial P_0(i, m_1, m_2, t)}{\partial t} &= -(\lambda + i\sigma + \alpha)P_0(i, m_1, m_2, t) + P_1(i, m_1 - 1, m_2, t)\mu_1 + \\
&\quad + P_2(i, m_1, m_2 - 1, t)\mu_2, \\
\frac{\partial P_1(i, m_1, m_2, t)}{\partial t} &= -(\lambda + \mu_1)P_1(i, m_1, m_2, t) + (i + 1)\sigma P_0(i + 1, m_1, m_2, t) + \\
&\quad + \lambda P_0(i, m_1, m_2, t), \\
\frac{\partial P_2(i, m_1, m_2, t)}{\partial t} &= -(\lambda + \mu_2)P_2(i, m_1, m_2, t) + \lambda P_2(i - 1, m_1, m_2, t) + \\
&\quad + \alpha P_0(i, m_1, m_2, t).
\end{aligned} \tag{13}$$

Введем частные характеристические функции, обозначив $j = \sqrt{-1}$,

$$H_k(u, u_1, u_2, t) = \sum_{i=0}^{\infty} \sum_{m_1=0}^{\infty} \sum_{m_2=0}^{\infty} e^{ju_i} e^{ju_1 m_1} e^{ju_2 m_2} P_k(i, m_1, m_2, t).$$

Тогда перепишем систему (13) в виде

$$\begin{aligned}
\frac{\partial H_0(u, u_1, u_2, t)}{\partial t} &= -(\lambda + \alpha)H_0(u, u_1, u_2, t) + j\sigma \frac{\partial H_0(u, u_1, u_2, t)}{\partial u} + \\
&\quad + \mu_1 e^{ju_1} H_1(u, u_1, u_2, t) + \mu_2 e^{ju_2} H_2(u, u_1, u_2, t), \\
\frac{\partial H_1(u, u_1, u_2, t)}{\partial t} &= -(\lambda + \mu_1)H_1(u, u_1, u_2, t) - j\sigma e^{-ju} \frac{\partial H_0(u, u_1, u_2, t)}{\partial u} + \\
&\quad + \lambda H_0(u, u_1, u_2, t) + \lambda e^{ju} H_1(u, u_1, u_2, t), \\
\frac{\partial H_2(u, u_1, u_2, t)}{\partial t} &= -(\lambda + \mu_2)H_2(u, u_1, u_2, t) + \lambda e^{ju} H_2(u, u_1, u_2, t) + \\
&\quad + \alpha H_0(u, u_1, u_2, t).
\end{aligned} \tag{14}$$

Бесконечное количество уравнений сведено к трем.

1.2.2 Метод асимптотического анализа

Полученную систему дифференциальных уравнений в частных производных (14) будем решать методом асимптотического анализа в предельном условии большой задержки заявок на орбите ($\sigma \rightarrow 0$).

Обозначим $\epsilon = \sigma, u = \epsilon w, F_k(w, u_1, u_2, t, \epsilon) = H_k(u, u_1, u_2, t)$, тогда система запи-

шется в виде

$$\begin{aligned}
\frac{\partial F_0(w, u_1, u_2, t, \epsilon)}{\partial t} &= -(\lambda + \alpha)F_0(w, u_1, u_2, t, \epsilon) + j\frac{\partial F_0(w, u_1, u_2, t, \epsilon)}{\partial w} + \\
&\quad + \mu_1 e^{ju_1} F_1(w, u_1, u_2, t, \epsilon) + \mu_2 e^{ju_2} F_2(w, u_1, u_2, t, \epsilon), \\
\frac{\partial F_1(w, u_1, u_2, t, \epsilon)}{\partial t} &= -(\lambda + \mu_1)F_1(w, u_1, u_2, t, \epsilon) - j e^{-j\epsilon w} \frac{\partial F_0(w, u_1, u_2, t, \epsilon)}{\partial w} + \\
&\quad + \lambda F_0(w, u_1, u_2, t, \epsilon) + \lambda e^{j\epsilon w} F_1(w, u_1, u_2, t, \epsilon), \\
\frac{\partial F_2(w, u_1, u_2, t, \epsilon)}{\partial t} &= -(\lambda + \mu_2)F_2(w, u_1, u_2, t, \epsilon) + \lambda e^{j\epsilon w} F_2(w, u_1, u_2, t, \epsilon) + \\
&\quad + \alpha F_0(w, u_1, u_2, t, \epsilon).
\end{aligned} \tag{15}$$

Затем, что используя условие согласованности многомерных распределений, характеристическая функция процессов $m_1(t)$ и $m_2(t)$ будет записана в следующем виде с введенными функциями

$$M\{\exp(ju_1 m_1(t)) \exp(ju_2 m_2(t))\} = \sum_{k=0}^2 H_k(0, u_1, u_2, t) = \sum_{k=0}^2 F_k(0, u_1, u_2, t, \epsilon).$$

Теорема 2. Асимптотическое приближение двумерной характеристической функции числа обслуженных заявок входящего потока и числа обслуженных вызванных заявок за некоторое время t имеет вид

$$\begin{aligned}
\mathbf{F}(u_1, u_2, t) &= \lim_{\epsilon \rightarrow 0} M\{\exp(ju_1 m_1(t)) \exp(ju_2 m_2(t))\} = \\
&= \lim_{\epsilon \rightarrow 0} \sum_{k=0}^2 F_k(0, u_1, u_2, t, \epsilon) = \mathbf{R} \cdot \exp\{G(u_1, u_2)t\} \cdot \mathbf{E}
\end{aligned}$$

где

$$\mathbf{G}(u_1, u_2) = \begin{bmatrix} -(\lambda + \alpha + \kappa) & \mu_1 e^{ju_1} & \mu_2 e^{ju_2} \\ \kappa + \lambda & -\mu_1 & 0 \\ \alpha & 0 & -\mu_2 \end{bmatrix}^T,$$

вектор-строка $\mathbf{R} = \{R_0, R_1, R_2\}$ — стационарное распределение вероятности состояния прибора

$$\mathbf{R} = \left\{ \frac{\mu_2(\mu_1 - \lambda)}{\mu_1(\mu_2 - \alpha)}, \frac{\lambda}{\mu_1}, \frac{\alpha(\mu_1 - \lambda)}{\mu_1(\mu_2 + \alpha)} \right\},$$

κ — нормированное среднее число заявок на орбите

$$\kappa = \frac{\lambda(\lambda\mu_2 + \alpha\mu_1)}{\mu_2(\mu_1 - \lambda)},$$

а \mathbf{E} — единичный вектор-столбец соответствующей размерности.

Доказательство. Делая предельный переход $\lim_{\epsilon \rightarrow 0} F_k(w, u_1, u_2, t, \epsilon) = F_k(w, u_1, u_2, t)$ в

полученной системе (15), система уравнений будет записана в виде

$$\begin{aligned}
\frac{\partial F_0(w, u_1, u_2, t)}{\partial t} &= -(\lambda + \alpha)F_0(w, u_1, u_2, t) + j\frac{\partial F_0(w, u_1, u_2, t)}{\partial w} + \\
&\quad + \mu_1 e^{ju_1} F_1(w, u_1, u_2, t) + \mu_2 e^{ju_2} F_2(w, u_1, u_2, t), \\
\frac{\partial F_1(w, u_1, u_2, t)}{\partial t} &= -(\lambda + \mu_1)F_1(w, u_1, u_2, t) - j\frac{\partial F_0(w, u_1, u_2, t)}{\partial w} + \\
&\quad + \lambda F_0(w, u_1, u_2, t) + \lambda F_1(w, u_1, u_2, t), \\
\frac{\partial F_2(w, u_1, u_2, t)}{\partial t} &= -(\lambda + \mu_2)F_2(w, u_1, u_2, t) + \lambda F_2(w, u_1, u_2, t) + \\
&\quad + \alpha F_0(w, u_1, u_2, t).
\end{aligned} \tag{16}$$

Решение системы (16) будет получено в следующей форме

$$F_k(w, u_1, u_2, t) = \Phi(w)F_k(u_1, u_2, t). \tag{17}$$

$\Phi(w)$ — асимптотическое приближение характеристической функции числа заявок на орбите при условии большой задержки на орбите.

Подставив (17) в систему (16) и разделив обе части уравнений на $\Phi(w)$, получим

$$\begin{aligned}
\frac{\partial F_0(u_1, u_2, t)}{\partial t} &= -(\lambda + \alpha)F_0(u_1, u_2, t) + j\frac{\Phi'(w)}{\Phi(w)}F_0(u_1, u_2, t) + \\
&\quad + \mu_1 e^{ju_1} F_1(u_1, u_2, t) + \mu_2 e^{ju_2} F_2(u_1, u_2, t), \\
\frac{\partial F_1(u_1, u_2, t)}{\partial t} &= -(\lambda + \mu_1)F_1(u_1, u_2, t) - j\frac{\Phi'(w)}{\Phi(w)}F_0(u_1, u_2, t) + \\
&\quad + \lambda F_0(u_1, u_2, t) + \lambda F_1(u_1, u_2, t), \\
\frac{\partial F_2(u_1, u_2, t)}{\partial t} &= -(\lambda + \mu_2)F_2(u_1, u_2, t) + \lambda F_2(u_1, u_2, t) + \\
&\quad + \alpha F_0(u_1, u_2, t).
\end{aligned} \tag{18}$$

Ранее, мы конкретизировали вид $\Phi(w)$ в (7), что позволяет нам привести систему (18) к следующему виду

$$\begin{aligned}
\frac{\partial F_0(u_1, u_2, t)}{\partial t} &= -(\lambda + \alpha + \kappa)F_0(u_1, u_2, t) + \\
&\quad + \mu_1 e^{ju_1} F_1(u_1, u_2, t) + \mu_2 e^{ju_2} F_2(u_1, u_2, t), \\
\frac{\partial F_1(u_1, u_2, t)}{\partial t} &= (\lambda + \kappa)F_0(u_1, u_2, t) - \mu_1 F_1(u_1, u_2, t) + \\
&\quad + 0F_2(u_1, u_2, t), \\
\frac{\partial F_2(u_1, u_2, t)}{\partial t} &= \alpha F_0(u_1, u_2, t) + 0F_1(u_1, u_2, t) - \\
&\quad - \mu_2 F_2(u_1, u_2, t).
\end{aligned} \tag{19}$$

Введем следующие обозначения

$$\mathbf{F}(u_1, u_2, t) = \{F_0(u_1, u_2, t), F_1(u_1, u_2, t), F_2(u_1, u_2, t)\},$$

$$\mathbf{G}(u_1, u_2) = \begin{bmatrix} -(\lambda + \alpha + \kappa) & \mu_1 e^{ju_1} & \mu_2 e^{ju_2} \\ \kappa + \lambda & -\mu_1 & 0 \\ \alpha & 0 & -\mu_2 \end{bmatrix}^T,$$

$\mathbf{G}(u_1, u_2)$ — транспонированная матрица коэффициентов системы (19). Тогда получим следующее матричное уравнение

$$\frac{\partial \mathbf{F}(u_1, u_2, t)}{\partial t} = \mathbf{F}(u_1, u_2, t) \mathbf{G}(u_1, u_2),$$

общее решение которого имеет вид

$$\mathbf{F}(u_1, u_2, t) = \mathbf{C} e^{\mathbf{G}(u_1, u_2)t}. \quad (20)$$

Для получения единственного решения примем в рассмотрение начальное условие

$$\mathbf{F}(u_1, u_2, 0) = \mathbf{R}, \quad (21)$$

где вектор–строка \mathbf{R} — стационарное распределение вероятности состояния прибора, то есть процесса $k(t)$, которое имеет форму [15]

$$\mathbf{R} = \left\{ \frac{\mu_2(\mu_1 - \lambda)}{\mu_1(\mu_2 - \alpha)}, \frac{\lambda}{\mu_1}, \frac{\alpha(\mu_1 - \lambda)}{\mu_1(\mu_2 + \alpha)} \right\}.$$

Описав начальное условие, переходим к решению задачи Коши (20, 21).

Чтобы получить маргинальное распределение вероятностей числа обслуженных заявок, суммируем компоненты вектор–строки $\mathbf{F}(u_1, u_2, t)$ по k и умножаем результат на единичный вектор–столбец \mathbf{E} . Получим

$$\mathbf{F}(u_1, u_2, t) \mathbf{E} = \mathbf{R} e^{\mathbf{G}(u_1, u_2)t} \mathbf{E}. \quad (22)$$

Формула (22) является решением рассматриваемой системы. \square

1.2.3 Переход к явному распределению вероятности

Характеристическая функция (22) описывает процессы $m_1(t)$ и $m_2(t)$, однако делает это в неявной форме. Для использования полученной формулы для расчетов, получим явное распределение вероятности. Формула (22), так же как и (11) содержит матричную экспоненту, к которой мы применим преобразование подобия [16]

$$\mathbf{G}(u_1, u_2) = \mathbf{T}(u_1, u_2) \mathbf{GJ}(u_1, u_2) \mathbf{T}(u_1, u_2)^{-1},$$

где $\mathbf{T}(u_1, u_2)$ — матрицы собственных векторов матрицы $\mathbf{G}(u_1, u_2)$, а $\mathbf{GJ}(u_1, u_2)$ — диагональная матрица, содержащая собственные числа $\mathbf{G}(u_1, u_2)$. Данное преобразование справедливо для любой степени m некоторой матрицы \mathbf{A}^m , из чего следует, что оно так же справедливо для матричной экспоненты [17]

$$e^{\mathbf{G}(u_1, u_2)t} = \mathbf{T}(u_1, u_2) \cdot \begin{bmatrix} e^{t\Lambda_1(u_1, u_2)} & 0 & 0 \\ 0 & e^{t\Lambda_2(u_1, u_2)} & 0 \\ 0 & 0 & e^{t\Lambda_3(u_1, u_2)} \end{bmatrix} \cdot \mathbf{T}(u_1, u_2)^{-1},$$

где Λ_n — собственное число матрицы $\mathbf{G}(u_1, u_2)$. Тогда распределение примет следующий вид

$$\mathbf{F}(u_1, u_2, t) = \mathbf{R} \cdot \mathbf{T}(u_1, u_2) \cdot \begin{bmatrix} e^{t\Lambda_1(u_1, u_2)} & 0 & 0 \\ 0 & e^{t\Lambda_2(u_1, u_2)} & 0 \\ 0 & 0 & e^{t\Lambda_3(u_1, u_2)} \end{bmatrix} \cdot \mathbf{T}(u_1, u_2)^{-1} \cdot \mathbf{E}.$$

Для восстановления распределения используем обратное преобразование Фурье для дискретных величин

$$P(m_1, m_2, t) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} e^{-i \cdot u_1 \cdot m_1} e^{-i \cdot u_2 \cdot m_2} \mathbf{F}(u_1, u_2, t) du_1 du_2. \quad (23)$$

Полученная формула характеризует вероятность обслуживания m_1 входящих заявок и m_2 вызванных заявок к моменту времени t в рассматриваемой системе.

1.2.4 Коэффициент корреляции

Полученное асимптотическое приближение характеристической функции (22) позволяет нам подробнее изучить выходящие потоки рассматриваемой системы, а именно — найти корреляционную зависимость случайных процессов $m_1(t)$ и $m_2(t)$. Рассмотрим нахождение коэффициента корреляции, который будет зависеть от параметра t

$$r(t) = \frac{\text{cov}(m_1(t), m_2(t))}{\sqrt{D(m_1(t))} \sqrt{D(m_2(t))}}.$$

Воспользуемся свойством характеристической функции о существовании ее n -ой производной, соответствующей n -му начальному моменту случайной величины. Тогда ковариация и дисперсия будут вычисляться следующим образом

$$\begin{aligned} \text{cov}(m_1(t), m_2(t)) &= M\{m_1(t)m_2(t)\} - M\{m_1(t)\}M\{m_2(t)\} = \frac{1}{j^2} \frac{\partial^2}{\partial u_1 \partial u_2} \mathbf{F}(u_1, u_2, t) \Big|_{\substack{u_1=0 \\ u_2=0}} - \\ &\quad - \frac{1}{j^2} \frac{\partial}{\partial u_1} \mathbf{F}(u_1, u_2, t) \Big|_{\substack{u_1=0 \\ u_2=0}} \frac{\partial}{\partial u_2} \mathbf{F}(u_1, u_2, t) \Big|_{\substack{u_1=0 \\ u_2=0}}, \\ D\{m_1(t)\} &= M^2\{m_1(t)\} - (M\{m_1(t)\})^2 = \frac{1}{j^2} \frac{\partial^2}{\partial u_1^2} \mathbf{F}(u_1, u_2, t) \Big|_{\substack{u_1=0 \\ u_2=0}} - \\ &\quad - \left(\frac{1}{j^2} \frac{\partial}{\partial u_1} \mathbf{F}(u_1, u_2, t) \Big|_{\substack{u_1=0 \\ u_2=0}} \right)^2, \\ D\{m_2(t)\} &= M^2\{m_2(t)\} - (M\{m_2(t)\})^2 = \frac{1}{j^2} \frac{\partial^2}{\partial u_2^2} \mathbf{F}(u_1, u_2, t) \Big|_{\substack{u_1=0 \\ u_2=0}} - \\ &\quad - \left(\frac{1}{j^2} \frac{\partial}{\partial u_2} \mathbf{F}(u_1, u_2, t) \Big|_{\substack{u_1=0 \\ u_2=0}} \right)^2. \end{aligned}$$

Полученные формулы позволяют использовать нам численно исследовать поведение системы при разных параметрах.

2 Исследование систем с ММРР–потокком

В этом разделе предлагается к рассмотрению система обслуживания, в которой в качестве источника входящих заявок выступает ММРР–поток.

ММРР (Markov Modulated Poisson Process) представляет собой поток заявок, управляемый непрерывной цепью Маркова. Изменение состояния влечет за собой изменение интенсивности прихода заявок. Данный тип входящего потока широко используется для моделирования телекоммуникационных сетей, поскольку он точно описывает изменяющуюся во времени скорость поступления заявок и фиксирует некоторые важные корреляции между временами прибытия, оставаясь при этом поддающимся анализу [18].

Для того, чтобы описать изменение состояние ММРР–потока во времени используются две матрицы.

Матрица инфинитезимальных характеристик Q . Величина q_{ij} определяет интенсивность перехода процесса из i -ого в j -ое состояние, а величина $-q_{ii}$ — интенсивность выхода из состояния i . Матрица Q имеет свойство $\sum_j q_{ij} = 0$

$$Q = \begin{bmatrix} q_{11} & \dots & q_{1n} \\ \vdots & \ddots & \\ q_{n1} & & q_{nn} \end{bmatrix}$$

Диагональная матрица Λ задаёт интенсивность поступления заявок для каждого из состояний процесса.

$$\Lambda = \begin{bmatrix} \lambda_{11} & \dots & 0 \\ \vdots & \lambda_{22} & \\ & & \ddots \\ 0 & \dots & \lambda_{nn} \end{bmatrix}$$

2.1 RQ–система с входящим ММРР–потокком и двумерным входящим потокком

В случае RQ–системы с ММРР–потокком, к описанной ранее общей модели добавляется так же процесс $n(t)$ — состояние входящего потока в момент времени t . Тогда, результат функционирования системы будет описываться пяти–мерным Марковским процессом

$$\{k(t), n(t), i(t), m_1(t), m_2(t)\}$$

2.1.1 Уравнения Колмогорова

Исходя из этого описанного Марковского процесса, вероятности того, что прибор будет находиться в одном из трех состояний k , на орбите будет i заявок, m_1 заявок

входящего потока и m_2 вызванных заявок будет обслужено, а управляющая ММРР-поток цепь Маркова будет находиться в состоянии n , будут иметь вид

$$\begin{aligned} P\{k(t) = 0, n(t) = n, i(t) = i, m_1(t) = m_1, m_2(t) = m_2\} &= P_0(n, i, m_1, m_2, t) \\ P\{k(t) = 1, n(t) = n, i(t) = i, m_1(t) = m_1, m_2(t) = m_2\} &= P_1(n, i, m_1, m_2, t) \\ P\{k(t) = 2, n(t) = n, i(t) = i, m_1(t) = m_1, m_2(t) = m_2\} &= P_2(n, i, m_1, m_2, t) \end{aligned}$$

На основе данных вероятностей запишем систему уравнений Колмогорова с учетом текущего состояния входящего потока

$$\begin{aligned} \frac{\partial P_0(n, i, m_1, m_2, t)}{\partial t} &= -(\lambda_n + i\sigma + \alpha)P_0(n, i, m_1, m_2, t) + P_1(n, i, m_1 - 1, m_2, t)\mu_1 + \\ &\quad + P_2(n, i, m_1, m_2 - 1, t)\mu_2 + \sum_{v=1}^N P_0(v, i, m_1, m_2, t)q_{vn}, \\ \frac{\partial P_1(n, i, m_1, m_2, t)}{\partial t} &= -(\lambda_n + \mu_1)P_1(n, i, m_1, m_2, t) + (i + 1)\sigma P_0(n, i + 1, m_1, m_2, t) + \\ &\quad + \lambda_n P_0(i, m_1, m_2, t) + \sum_{v=1}^N P_1(v, i, m_1, m_2, t)q_{vn}, \\ \frac{\partial P_2(n, i, m_1, m_2, t)}{\partial t} &= -(\lambda_n + \mu_2)P_2(n, i, m_1, m_2, t) + \lambda_n P_2(n, i - 1, m_1, m_2, t) + \\ &\quad + \alpha P_0(n, i, m_1, m_2, t) + \sum_{v=1}^N P_2(v, i, m_1, m_2, t)q_{vn}. \end{aligned} \tag{24}$$

Аналогично процедуре решения вышеописанных систем, введем частные характеристические функции, обозначив $j = \sqrt{-1}$

$$H_k(n, u, u_1, u_2, t) = \sum_{i=0}^{\infty} \sum_{m_1=0}^{\infty} \sum_{m_2=0}^{\infty} e^{ju_i} e^{ju_1 m_1} e^{ju_2 m_2} P_k(n, i, m_1, m_2, t).$$

Перепишем систему (24) с учетом введенных частных характеристических функций

$$\begin{aligned} \frac{\partial H_0(n, u, u_1, u_2, t)}{\partial t} &= -(\lambda_n + \alpha)H_0(n, u, u_1, u_2, t) + j\sigma \frac{\partial H_0(n, u, u_1, u_2, t)}{\partial u} + \\ &\quad + \mu_1 e^{ju_1} H_1(n, u, u_1, u_2, t) + \mu_2 e^{ju_2} H_2(n, u, u_1, u_2, t) + \\ &\quad + \sum_{v=1}^N H_0(v, u, u_1, u_2, t)q_{vn}, \\ \frac{\partial H_1(n, u, u_1, u_2, t)}{\partial t} &= -(\lambda_n + \mu_1)H_1(n, u, u_1, u_2, t) - j\sigma e^{-ju} \frac{\partial H_0(n, u, u_1, u_2, t)}{\partial u} + \\ &\quad + \lambda_n H_0(n, u, u_1, u_2, t) + \lambda_n e^{ju} H_1(n, u, u_1, u_2, t) + \\ &\quad + \sum_{v=1}^N H_1(v, u, u_1, u_2, t)q_{vn}, \\ \frac{\partial H_2(n, u, u_1, u_2, t)}{\partial t} &= -(\lambda_n + \mu_2)H_2(n, u, u_1, u_2, t) + \lambda_n e^{ju} H_2(n, u, u_1, u_2, t) + \\ &\quad + \alpha H_0(n, u, u_1, u_2, t) + \sum_{v=1}^N H_2(v, u, u_1, u_2, t)q_{vn}. \end{aligned} \tag{25}$$

Для дальнейшего анализа введем следующие обозначения

$$\mathbf{H}_k(u, u_1, u_2, t) = \{H_k(1, u, u_1, u_2, t), H_k(2, u, u_1, u_2, t), \dots, H_k(N, u, u_1, u_2, t)\},$$

а так же диагональную единичную матрицу \mathbf{I} размера N . Тогда система 25 примет вид

$$\begin{aligned} \frac{\partial \mathbf{H}_0(u, u_1, u_2, t)}{\partial t} &= (\mathbf{Q} - \mathbf{\Lambda} - \alpha \mathbf{I}) \mathbf{H}_0(u, u_1, u_2, t) + \mu_1 e^{ju_1} \mathbf{H}_1(u, u_1, u_2, t) + \\ &\quad + \mu_2 e^{ju_2} \mathbf{H}_2(u, u_1, u_2, t) + j\sigma \frac{\partial \mathbf{H}_0(u, u_1, u_2, t)}{\partial u}, \\ \frac{\partial \mathbf{H}_1(u, u_1, u_2, t)}{\partial t} &= \mathbf{\Lambda} \mathbf{H}_0(u, u_1, u_2, t) + (\mathbf{Q} + (e^{ju} - 1) \mathbf{\Lambda} - \mathbf{I} \mu_1) \mathbf{H}_1(u, u_1, u_2, t) - \\ &\quad - j\sigma e^{-ju} \frac{\partial \mathbf{H}_0(u, u_1, u_2, t)}{\partial u}, \\ \frac{\partial \mathbf{H}_2(u, u_1, u_2, t)}{\partial t} &= \alpha \mathbf{H}_0(u, u_1, u_2, t) + (\mathbf{Q} + (e^{ju} - 1) \mathbf{\Lambda} - \mathbf{I} \mu_2) \mathbf{H}_2(u, u_1, u_2, t). \end{aligned} \quad (26)$$

2.1.2 Метод асимптотического анализа

Полученную систему дифференциальных уравнений (26) будем решать методом асимптотического анализа в предельном условии большой задержки заявок на орбите ($\sigma \rightarrow 0$).

Обозначим $\epsilon = \sigma, u = \epsilon w, \mathbf{F}_k(w, u_1, u_2, t, \epsilon) = \mathbf{H}_k(u, u_1, u_2, t)$, тогда система запишется в виде

$$\begin{aligned} \frac{\partial \mathbf{F}_0(w, u_1, u_2, t, \epsilon)}{\partial t} &= (\mathbf{Q} - \mathbf{\Lambda} - \alpha \mathbf{I}) \mathbf{F}_0(w, u_1, u_2, t, \epsilon) + \mu_1 e^{ju_1} \mathbf{F}_1(w, u_1, u_2, t, \epsilon) + \\ &\quad + \mu_2 e^{ju_2} \mathbf{F}_2(w, u_1, u_2, t, \epsilon) + j \frac{\partial \mathbf{F}_0(w, u_1, u_2, t, \epsilon)}{\partial w}, \\ \frac{\partial \mathbf{F}_1(w, u_1, u_2, t, \epsilon)}{\partial t} &= \mathbf{\Lambda} \mathbf{F}_0(w, u_1, u_2, t, \epsilon) + (\mathbf{Q} + (e^{j\epsilon w} - 1) \mathbf{\Lambda} - \mathbf{I} \mu_1) \mathbf{F}_1(w, u_1, u_2, t, \epsilon) - \\ &\quad - j e^{-j\epsilon w} \frac{\partial \mathbf{F}_0(w, u_1, u_2, t, \epsilon)}{\partial w}, \\ \frac{\partial \mathbf{F}_2(w, u_1, u_2, t, \epsilon)}{\partial t} &= \alpha \mathbf{F}_0(w, u_1, u_2, t, \epsilon) + (\mathbf{Q} + (e^{j\epsilon w} - 1) \mathbf{\Lambda} - \mathbf{I} \mu_2) \mathbf{F}_2(w, u_1, u_2, t, \epsilon). \end{aligned} \quad (27)$$

Решение системы (27) будет сформулировано в последующих теоремах.

Теорема 3. Пусть $i(t)$ — количество заявок на орбите в системе с входящим ММРР-поток и вызываемыми заявками, тогда в стационарном режиме мы получим

$$\lim_{\epsilon \rightarrow 0} \left\{ \sum_{k=0}^2 F_k(w, 0, 0, t, \epsilon) \right\} = \lim_{\sigma \rightarrow 0} M e^{jw\sigma i(t)} = e^{jw\kappa},$$

где κ — положительный корень уравнения

$$\kappa \mathbf{R}_0(\kappa) \mathbf{e} = [\mathbf{R}_1(\kappa) + \mathbf{R}_2(\kappa)] \mathbf{\Lambda} \mathbf{e}.$$

Более того, векторы \mathbf{R}_k определяются следующим образом

$$\begin{cases} \mathbf{R}_0(\kappa) = \mathbf{r}\{\mathbf{I} + [\mathbf{\Lambda} + \kappa\mathbf{I}](\mu_1\mathbf{I} - \mathbf{Q})^{-1} + \alpha(\mu_2\mathbf{I} - \mathbf{Q})^{-1}\}^{-1}, \\ \mathbf{R}_1(\kappa) = \mathbf{R}_0(\kappa)[\mathbf{\Lambda} + \kappa\mathbf{I}](\mu_1\mathbf{I} - \mathbf{Q})^{-1}, \\ \mathbf{R}_2(\kappa) = \alpha\mathbf{R}_0(\kappa)(\mu_2\mathbf{I} - \mathbf{Q})^{-1} \end{cases}$$

Вектор-строка \mathbf{r} — стационарное распределение вероятности процесса управляющей цепи ММРР-потока $n(t)$, которое задается как единственное решение системы $\mathbf{r}\mathbf{Q} = 0, \mathbf{r}\mathbf{e} = 1$.

Доказательство. В системе уравнений (27) мы обозначили $u_1 = u_2 = 0$, тем самым убирая процессы $m_1(t)$ и $m_2(t)$ из рассмотрения. Тогда, мы можем получить систему уравнений уже для трехмерного процесса $\{n(t), k(t), i(t)\}$, рассматривая его в стационарном режиме, что позволяет нам избавиться от производных по времени t . Обозначим

$$\mathbf{F}_k(w, \epsilon) = \lim_{t \rightarrow \infty} \mathbf{F}_k(w, 0, 0, t, \epsilon),$$

для получения следующей системы

$$\begin{aligned} (\mathbf{Q} - \mathbf{\Lambda} - \alpha\mathbf{I})\mathbf{F}_0(w, \epsilon) + \mu_1\mathbf{F}_1(w, \epsilon) + \mu_2\mathbf{F}_2(w, \epsilon) + j\mathbf{F}'_0(w, \epsilon) &= 0, \\ \mathbf{\Lambda}\mathbf{F}_0(w, \epsilon) + (\mathbf{Q} + (e^{j\epsilon w} - 1)\mathbf{\Lambda} - \mathbf{I}\mu_1)\mathbf{F}_1(w, \epsilon) - je^{-j\epsilon w}\mathbf{F}'_0(w, \epsilon) &= 0, \\ \alpha\mathbf{F}_0(w, \epsilon) + (\mathbf{Q} + (e^{j\epsilon w} - 1)\mathbf{\Lambda} - \mathbf{I}\mu_2)\mathbf{F}_2(w, \epsilon) &= 0. \end{aligned} \quad (28)$$

Делая в системе (28) предельный переход $\epsilon \rightarrow 0$, получим

$$\begin{aligned} (\mathbf{Q} - \mathbf{\Lambda} - \alpha\mathbf{I})\mathbf{F}_0(w) + \mu_1\mathbf{F}_1(w) + \mu_2\mathbf{F}_2(w) + j\mathbf{F}'_0(w) &= 0, \\ \mathbf{\Lambda}\mathbf{F}_0(w) + (\mathbf{Q} - \mathbf{I}\mu_1)\mathbf{F}_1(w) - j\mathbf{F}'_0(w) &= 0, \\ \alpha\mathbf{F}_0(w) + (\mathbf{Q} - \mathbf{I}\mu_2)\mathbf{F}_2(w) &= 0. \end{aligned} \quad (29)$$

Решение данной системы будет найдено в следующей форме

$$\mathbf{F}_k = \Phi(w)\mathbf{R}_k, \quad (30)$$

где \mathbf{R}_n — стационарное распределение вероятности прибора, а $\Phi(w)$ — асимптотическое приближение характеристической функции числа заявок на орбите при условии большой задержки на орбите. Подставляя (30) в (29) и деля на $\Phi(w)$, получим

$$\begin{aligned} (\mathbf{Q} - \mathbf{\Lambda} - \alpha\mathbf{I})\mathbf{R}_0 + \mu_1\mathbf{R}_1 + \mu_2\mathbf{R}_2 + j\frac{\Phi'(w)}{\Phi(w)}\mathbf{R}_0 &= 0, \\ \mathbf{\Lambda}\mathbf{R}_0 + (\mathbf{Q} - \mathbf{I}\mu_1)\mathbf{R}_1 - j\frac{\Phi'(w)}{\Phi(w)}\mathbf{R}_0 &= 0, \\ \alpha\mathbf{R}_0 + (\mathbf{Q} - \mathbf{I}\mu_2)\mathbf{R}_2 &= 0. \end{aligned} \quad (31)$$

Вид функции $\Phi(w)$ уже ранее был нами конкретизирован в (7) так, что $j\frac{\Phi'(w)}{\Phi(w)} = -\kappa$. Подставим данное выражение в систему 31, получим

$$\begin{aligned} (\mathbf{Q} - \mathbf{\Lambda} - \alpha\mathbf{I})\mathbf{R}_0 + \mu_1\mathbf{R}_1 + \mu_2\mathbf{R}_2 - \kappa\mathbf{R}_0 &= 0, \\ \mathbf{\Lambda}\mathbf{R}_0 + (\mathbf{Q} - \mathbf{I}\mu_1)\mathbf{R}_1 + \kappa\mathbf{R}_0 &= 0, \\ \alpha\mathbf{R}_0 + (\mathbf{Q} - \mathbf{I}\mu_2)\mathbf{R}_2 &= 0. \end{aligned} \quad (32)$$

Запишем условия нормировки для стационарного распределения вероятностей числа обслуженных заявок прибором

$$\mathbf{R}_0 + \mathbf{R}_1 + \mathbf{R}_2 = \mathbf{r}.$$

Основываясь на этом уравнении, а также на двух последних уравнения системы (32) запишем систему

$$\begin{cases} \mathbf{R}_1 = \mathbf{R}_0[\mathbf{\Lambda} + \kappa\mathbf{I}](\mu_1\mathbf{I} - \mathbf{Q})^{-1}, \\ \mathbf{R}_2 = \alpha\mathbf{R}_0(\mu_2\mathbf{I} - \mathbf{Q})^{-1}, \\ \mathbf{R}_0 + \mathbf{R}_1 + \mathbf{R}_2 = \mathbf{r}. \end{cases} \quad (33)$$

Просуммируем уравнения системы (28)

$$\begin{aligned} & [\mathbf{F}_0(w, \epsilon) + \mathbf{F}_1(w, \epsilon) + \mathbf{F}_2(w, \epsilon)]\mathbf{Q} + \\ & + \mathbf{F}_1(w, \epsilon)(e^{jw\epsilon} - 1)\mathbf{\Lambda} + \mathbf{F}_2(w, \epsilon)(e^{jw\epsilon} - 1)\mathbf{\Lambda} + je^{-jw\epsilon}(e^{jw\epsilon} - 1)\mathbf{F}'_0(w, \epsilon) = 0. \end{aligned}$$

Умножая получившееся уравнения на единичный вектор столбец \mathbf{e} , получим

$$\{\mathbf{F}_1(w, \epsilon) + \mathbf{F}_2(w, \epsilon)\}\mathbf{\Lambda}\mathbf{e} + je^{-jw\epsilon}\mathbf{F}'_0(w, \epsilon)\mathbf{e} = 0$$

Затем, подставим произведение (30) в получившееся уравнение

$$[\mathbf{R}_1 + \mathbf{R}_2]\mathbf{\Lambda}\mathbf{e} + j\frac{\Phi'(w)}{\Phi(w)}\mathbf{R}_0\mathbf{e} = 0$$

и делаем замену

$$[\mathbf{R}_1 + \mathbf{R}_2]\mathbf{\Lambda}\mathbf{e} - \kappa\mathbf{R}_0\mathbf{e} = 0 \quad (34)$$

Из (34) мы можем выразить κ с помощью $\mathbf{R}_0, \mathbf{R}_1$ и \mathbf{R}_2 . Помимо этого, мы можем переписать систему (33) в следующем виде

$$\begin{cases} \mathbf{R}_0(\kappa) = \mathbf{r}\{\mathbf{I} + [\mathbf{\Lambda} + \kappa\mathbf{I}](\mu_1\mathbf{I} - \mathbf{Q})^{-1} + \alpha(\mu_2\mathbf{I} - \mathbf{Q})^{-1}\}^{-1}, \\ \mathbf{R}_1(\kappa) = \mathbf{R}_0(\kappa)[\mathbf{\Lambda} + \kappa\mathbf{I}](\mu_1\mathbf{I} - \mathbf{Q})^{-1}, \\ \mathbf{R}_2(\kappa) = \alpha\mathbf{R}_0(\kappa)(\mu_2\mathbf{I} - \mathbf{Q})^{-1} \end{cases}$$

□

Теорема 3 является вспомогательной, так как основное решение рассматриваемой системы изложено в теореме 4 и нуждается в полученных на данном этапе результатах, а именно — среднее число заявок на орбите при условии их большой задержки κ и стационарное распределение вероятностей состояния прибора \mathbf{R}_k .

Теорема 4. *Асимптотическое приближение двумерной характеристической функции числа обслуженных заявок входящего ММРР-потока и числа обслуженных вызванных заявок за некоторое время t имеет вид*

$$\begin{aligned} \lim_{\sigma \rightarrow 0} M\{\exp(ju_1m_1(t))\exp(ju_2m_2(t))\} &= \lim_{\epsilon \rightarrow 0} \left\{ \sum_{k=0}^2 F_k(0, u_1, u_2, t, \epsilon) \right\} \mathbf{e} = \\ &= \mathbf{R} \cdot \exp\{G(u_1, u_2)t\} \mathbf{e}, \end{aligned}$$

где матрица $\mathbf{G}(u_1, u_2)$ представима в виде

$$\mathbf{G}(u_1, u_2) = \begin{bmatrix} \mathbf{Q} - \mathbf{\Lambda} - (\alpha + \kappa)\mathbf{I} & \mu_1 e^{ju_1} \mathbf{I} & \mu_2 e^{ju_2} \mathbf{I} \\ \mathbf{\Lambda} + \kappa \mathbf{I} & \mathbf{Q} - \mu_1 \mathbf{I} & 0 \\ \alpha \mathbf{I} & 0 & \mathbf{Q} - \mu_2 \mathbf{I} \end{bmatrix}^T,$$

вектор-строка $\mathbf{R} = \{\mathbf{R}_0, \mathbf{R}_1, \mathbf{R}_2\}$ – двумерное стационарное распределение вероятности случайного процесса $\{k(t), n(t)\}$, где, соответственно, \mathbf{R}_k имеет размерность N , κ – нормированное среднее число заявок на орбите, а e и ee – единичные вектор-столбцы размерности N и $N \cdot K$ соответственно.

Доказательство. Делая предельный переход $\lim_{\epsilon \rightarrow 0} \mathbf{F}_k(w, u_1, u_2, t, \epsilon) = \mathbf{F}_k(w, u_1, u_2, t)$ в полученной системе (27), система уравнений будет записана в виде

$$\begin{aligned} \frac{\partial \mathbf{F}_0(w, u_1, u_2, t)}{\partial t} &= (\mathbf{Q} - \mathbf{\Lambda} - \alpha \mathbf{I}) \mathbf{F}_0(w, u_1, u_2, t) + \mu_1 e^{ju_1} \mathbf{F}_1(w, u_1, u_2, t) + \\ &\quad + \mu_2 e^{ju_2} \mathbf{F}_2(w, u_1, u_2, t) + j \frac{\partial \mathbf{F}_0(w, u_1, u_2, t)}{\partial w}, \\ \frac{\partial \mathbf{F}_1(w, u_1, u_2, t)}{\partial t} &= \mathbf{\Lambda} \mathbf{F}_0(w, u_1, u_2, t) + (\mathbf{Q} - \mathbf{I} \mu_1) \mathbf{F}_1(w, u_1, u_2, t) - \\ &\quad - j \frac{\partial \mathbf{F}_0(w, u_1, u_2, t)}{\partial w}, \\ \frac{\partial \mathbf{F}_2(w, u_1, u_2, t)}{\partial t} &= \alpha \mathbf{F}_0(w, u_1, u_2, t) + (\mathbf{Q} - \mathbf{I} \mu_2) \mathbf{F}_2(w, u_1, u_2, t). \end{aligned} \quad (35)$$

Решение системы (35) будет получено в форме

$$\mathbf{F}_k(w, u_1, u_2, t) = \Phi(w) \mathbf{F}_k(u_1, u_2, t). \quad (36)$$

Подставив (36) в систему (35) и разделив обе части уравнений на $\Phi(w)$, получим

$$\begin{aligned} \frac{\partial \mathbf{F}_0(u_1, u_2, t)}{\partial t} &= (\mathbf{Q} - \mathbf{\Lambda} - \alpha \mathbf{I}) \mathbf{F}_0(u_1, u_2, t) + \mu_1 e^{ju_1} \mathbf{F}_1(u_1, u_2, t) + \\ &\quad + \mu_2 e^{ju_2} \mathbf{F}_2(u_1, u_2, t) + j \frac{\Phi'(w)}{\Phi(w)} \mathbf{F}_0(u_1, u_2, t), \\ \frac{\partial \mathbf{F}_1(u_1, u_2, t)}{\partial t} &= \mathbf{\Lambda} \mathbf{F}_0(u_1, u_2, t) + (\mathbf{Q} - \mathbf{I} \mu_1) \mathbf{F}_1(u_1, u_2, t) - \\ &\quad - j \frac{\Phi'(w)}{\Phi(w)} \mathbf{F}_0(u_1, u_2, t), \\ \frac{\partial \mathbf{F}_2(u_1, u_2, t)}{\partial t} &= \alpha \mathbf{F}_0(u_1, u_2, t) + (\mathbf{Q} - \mathbf{I} \mu_2) \mathbf{F}_2(u_1, u_2, t). \end{aligned} \quad (37)$$

Исходя из того, что $\Phi(w)$ имеет смысл асимптотического приближения характеристической функции числа заявок на орбите и имеет вид экспоненты (7), система (37) примет

следующий вид

$$\begin{aligned}
\frac{\partial \mathbf{F}_0(u_1, u_2, t)}{\partial t} &= (\mathbf{Q} - \mathbf{\Lambda} - (\alpha + \kappa)\mathbf{I})\mathbf{F}_0(u_1, u_2, t) + \mu_1 e^{ju_1} \mathbf{F}_1(u_1, u_2, t) + \\
&\quad + \mu_2 e^{ju_2} \mathbf{F}_2(u_1, u_2, t), \\
\frac{\partial \mathbf{F}_1(u_1, u_2, t)}{\partial t} &= (\mathbf{\Lambda} + \kappa\mathbf{I})\mathbf{F}_0(u_1, u_2, t) + (\mathbf{Q} - \mathbf{I}\mu_1)\mathbf{F}_1(u_1, u_2, t) + \\
&\quad + 0\mathbf{F}_2(u_1, u_2, t), \\
\frac{\partial \mathbf{F}_2(u_1, u_2, t)}{\partial t} &= \alpha\mathbf{F}_0(u_1, u_2, t) + 0\mathbf{F}_1(u_1, u_2, t) + (\mathbf{Q} - \mathbf{I}\mu_2)\mathbf{F}_2(u_1, u_2, t).
\end{aligned} \tag{38}$$

Введем следующие обозначения

$$\mathbf{F}\mathbf{F}(u_1, u_2, t) = \{\mathbf{F}_0(u_1, u_2, t), \mathbf{F}_1(u_1, u_2, t), \mathbf{F}_2(u_1, u_2, t)\},$$

$$\mathbf{G}(u_1, u_2) = \begin{bmatrix} \mathbf{Q} - \mathbf{\Lambda} - (\alpha + \kappa)\mathbf{I} & \mu_1 e^{ju_1} \mathbf{I} & \mu_2 e^{ju_2} \mathbf{I} \\ \mathbf{\Lambda} + \kappa\mathbf{I} & \mathbf{Q} - \mu_1 \mathbf{I} & 0 \\ \alpha\mathbf{I} & 0 & \mathbf{Q} - \mu_2 \mathbf{I} \end{bmatrix}^T,$$

$\mathbf{G}(u_1, u_2)$ — транспонированная матрица коэффициентов системы (38). Тогда получим следующее матричное уравнение

$$\frac{\partial \mathbf{F}\mathbf{F}(u_1, u_2, t)}{\partial t} = \mathbf{F}\mathbf{F}(u_1, u_2, t)\mathbf{G}(u_1, u_2),$$

общее решение которого имеет вид

$$\mathbf{F}\mathbf{F}(u_1, u_2, t) = \mathbf{C}e^{\mathbf{G}(u_1, u_2)t}. \tag{39}$$

Для того, чтобы получить единственное решение, соответствующее поведению рассматриваемой системы, примем в рассмотрение начальное условие

$$\mathbf{F}\mathbf{F}(u_1, u_2, 0) = \mathbf{R}, \tag{40}$$

где вектор–строка \mathbf{R} — стационарное распределение вероятности состояния прибора, то есть процесса $k(t)$, полученное в теореме 3. Описав начальное условие, мы можем перейти к решению задачи Коши (39, 40)

$$\mathbf{F}\mathbf{F}(u_1, u_2, t) = \mathbf{R}e^{\mathbf{G}(u_1, u_2)t},$$

Поскольку нас интересует распределение вероятностей количества заявок в выходных процессах, необходимо найти маргинальное распределение. Для этого умножаем компоненты вектор–строки $\mathbf{F}\mathbf{F}(u_1, u_2, t)$ на единичный вектор–столбец \mathbf{e} размера N и правую часть уравнения на единичный вектор столбец $\mathbf{e}\mathbf{e}$ размерности $K \cdot N$. Получим

$$\mathbf{F}\mathbf{F}(u_1, u_2, t)\mathbf{e} = \mathbf{R}e^{\mathbf{G}(u_1, u_2)t}\mathbf{e}\mathbf{e}. \tag{41}$$

Формула (41) является решением рассматриваемой системы. \square

2.1.3 Переход к явному распределению вероятности

Характеристическая функция (41) позволяет нам получить явное распределение вероятностей числа обслуженных заявок процессов $m_1(t)$ и $m_2(t)$. Аналогично ранее рассмотренным (11,22), формула (41) содержит матричную экспоненту, к которой мы применяем преобразование подобия [16]. Процесс разложения матричной экспоненты полностью идентичен процессу в разделе 1.2.3, поэтому будет опущен. Вид восстановленного с помощью обратного преобразования Фурье для дискретных величин распределения имеет следующую форму

$$P(m_1, m_2, t) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} e^{-i \cdot u_1 \cdot m_1} e^{-i \cdot u_2 \cdot m_2} \mathbf{F} \mathbf{F}(u_1, u_2, t) du_2 du_1. \quad (42)$$

Полученная формула характеризует вероятность обслуживания m_1 входящих заявок и m_2 вызванных заявок к моменту времени t в рассматриваемой системе.

3 Имитационное моделирование

В данной работе для анализа полученных решений систем массового обслуживания с повторными вызовами и вызываемыми заявками используется имитационное моделирование. Построив имитационную модель системы, мы получаем возможность провести численное сравнение результатов ее работы с аналитическим решением для оценки его применимости на практике.

Имитационное моделирование систем массового обслуживания сводится к генерированию случайных времен с соответствующим распределением вероятности, по прошествии которого в системе наступит некоторое событие — приход заявки, окончание обслуживания и т.д., что называется дискретно-событийным подходом. Таким образом, в течение моделирования на протяжении некоторого заданного времени модель обрабатывает происходящие в системе события — они регистрируются, и на их основе подсчитываются численные характеристики, такие как распределение вероятности.

Реализованная имитационная модель, как программный продукт, разрабатывалась с возможностью расширения ее функционала для предоставления возможности тонко настраивать параметры моделирования и без препятствий извлекать полученные результаты в виде распределения вероятности для последующего анализа. В этих целях, программа реализована при помощи объектно ориентированного подхода, так как данный подход позволил выстроить логику работу программы на уровне абстракций, что и дает возможность легкого расширения функционала и реализации моделей других систем массового обслуживания.

3.1 Объектная модель предметной области

Для построения объектной модели предметной области программы рассматриваются следующие объекты систем массового обслуживания, используемые в данной работе:

- Заявка — объект, который поступает в систему, перемещается по ней и, в конечном итоге, покидает систему в течение моделирования.
- Обслуживающий прибор — объект, принимающий на вход заявки и обслуживающий их. По окончании обслуживания заявки покидают прибор.
- Орбита — скопление заявок, ожидающих повторного обращения на обслуживающий прибор.
- Входящий поток — объект, генерирующий заявки, которые поступают на прибор.

Поскольку процесс работы модели заключается в перемещении заявок по системе, то в некоторый момент времени моделирования T_{cur} местонахождение каждой заявки и

ее характеристики могут меняться. В таком случае, необходимо отслуживать момент времени, в который заявка изменит свое состояние — T_{shift} . В момент времени, когда $T_{shift} = T_{curr}$, происходит смена состояния заявки, и T_{shift} обновляется объектом системы массового обслуживания, в которой она на данный момент находится. Среди событий, при которых происходит обновление момента смены состояния заявки, выделим следующие:

- Заявка покидает входящий поток — в данном случае в источнике генерируется новая заявка с моментом времени T_{shift} , при наступлении которого, она также покинет входящий поток.
- Заявка успешно поступила на прибор и начала обслуживаться — в этом случае обслуживающий прибор определяет для заявки момент времени T_{shift} , в который ее обслуживание закончится.
- При попытке поступить на прибор заявка застала его занятым обслуживанием другой заявки и отправилась на орбиту. По прибытии на орбиту, новый момент времени T_{shift} заявки определяется как время, при наступлении которого заявка покинет орбиту.

Исходя из вышеописанного, центральными объектами в предметной области являются заявка и событие, наступающее в системе при смене состояния заявки. В предметной области они представлены классом Event, который связан ассоциацией с перечислением EventType, элементы которого соответствуют объектам системы обслуживания, в которых наступило событие. Это позволяет собирать статистику о произошедших событиях в системе.

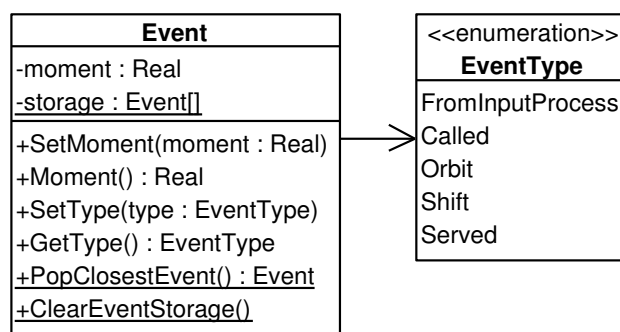


Рисунок 4 — Класс Event

Поле `moment` хранит момент времени T_{shift} до изменения состояния объекта. Для того, чтобы определить, какое событие должно произойти в модели следующим, класс Event содержит статическую коллекцию собственных объектов — `storage`. При создании новых событий они также помещаются и в `storage` для того, чтобы в последствии извлекать следующее по времени событие с помощью метода `PopClosestEvent`.

Для реализации подхода, основанном на постоянном обновлении состоянии заявок, находящихся в системе, предметная область базируется на общем интерфейсе, который реализуют все конкретные элементы рассматриваемых систем массового обслуживания. Его цель заключается в формализации изменения состояния элемента системы с каждым промежутком времени, в течение которого наступило очередное событие. Поскольку, сам класс RQ-системы, как будет показано ниже, реализует данный интерфейс, то возможно использовать ее как входящий поток для другой системы.

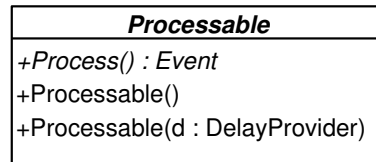


Рисунок 5 — Общий интерфейс Processable для всех элементов системы массового обслуживания

Метод Process унифицирует все изменения в элементе системы массового обслуживания, которые должны произойти в нем при наступлении события. Реализация этого интерфейса не обязательно должна иметь только этот метод, так как для удобства организации логики работы элемента, удобно размещать некоторые алгоритмы в отдельных методах.

Также, не все элементы системы можно полностью описать данным интерфейсом — обслуживающий прибор и орбита, помимо общего метода Process, должны иметь точки входа для заявок. По этой причине для них определены соответствующие интерфейсы, наследуемые от Processable.

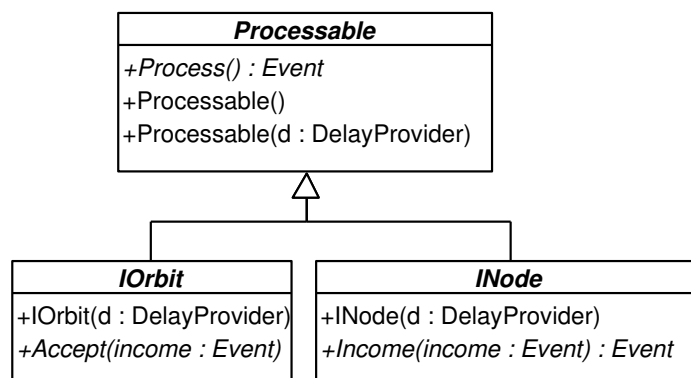


Рисунок 6 — Интерфейсы для обслуживающего прибора (INode) и орбиты (IOrbit), наследуемые от Processable

Метод IOrbit Accept служит для принятия заявок на орбиту, а метод INode

Income — для принятия заявки на прибор для обслуживания. Отличие между данными методами заключается в том, что, если прибор занят обслуживанием другой заявки, то метод Income вернет заявку, которая пытается встать на прибор, в то время как орбита не имеет ограничения по количеству находящихся на ней заявок.

Для вычисления экспоненциальной задержки до наступления событий в модели, в предметную область был добавлен общий интерфейс для объектов, вычисляющих задержку, что позволяет менять способ вычисления задержки в процессе моделирования.

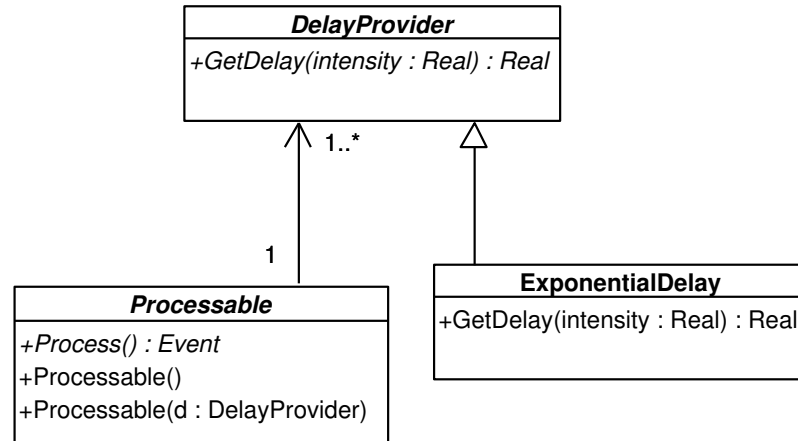


Рисунок 7 — DelayProvider — интерфейс для вычисления задержки, который используют элементы системы массового обслуживания

Для обеспечения среды, которой будет происходить моделирование, а именно — вести подсчет прошедшего времени, взаимодействие с реализацией интерфейса Processable и сбор статистики, введен глобальный объект Environment

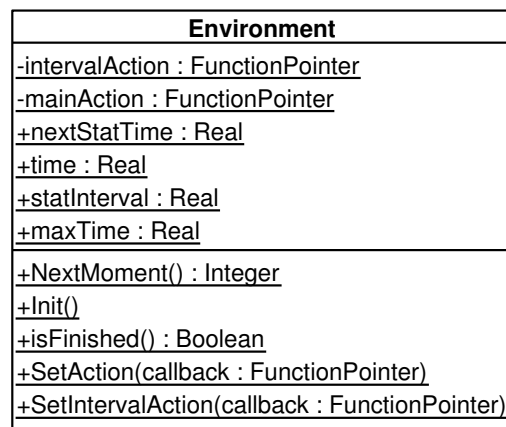
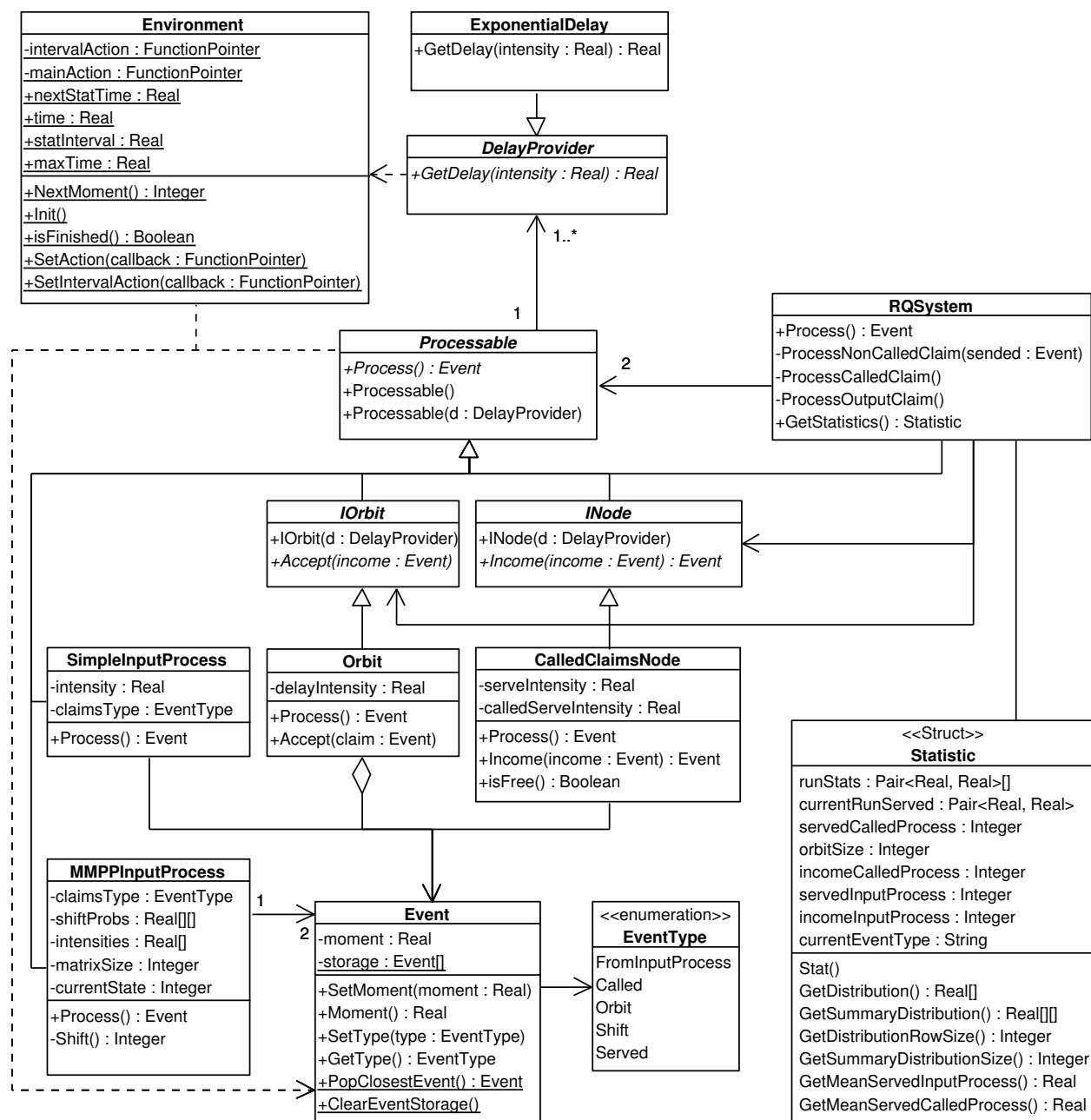


Рисунок 8 — Глобальный объект Environment

Статический класс Environment содержит общую для модели информацию —

время моделирования, время конца моделирования и методы, для управления этими параметрами. Члены класса `Environment` будут описаны в последующих разделах.



- Orbit — реализация источника повторных вызовов, содержащая хранилище заявок (Event), что на диаграмме классов показано агрегацией. Поле `delayIntensity` хранит интенсивность обращений заявок с орбиты, то есть используется для вычисления экспоненциальной задержки соответствующим объектом `delayProvider`.
- CalledClaimsNode — реализация обслуживающего прибора, способная обслуживать два типа заявок — пришедшие извне и вызванные, для чего служат интенсивности `serveIntensity` и `calledServeIntensity` соответственно.
- SimpleInputProcess — реализация простейшего входящего потока, порождающего заявки. Ассоциация с Event показывает, что класс имеет поле с заявкой, которая готовится покинуть источник заявок по истечении задержки.
- MMPPInputProcess — реализация MMPP-потока. Поле `shiftProbs` хранит матрицу интенсивностей переходов между состояниями, а поле `intensities` — интенсивность поступления заявок для каждого состояния. Логика смены состояний находится в приватном методе `Shift`, где используются поле `currentState` — номер текущего состояния, а так же объект Event, в котором хранится момент смены состояния входящего потока, что на диаграмме показано ассоциацией с множителем 2.
- RQSystem представляет агрегирующий класс, где при помощи описанных выше классов строится логика работы системы.
- Структура `Statistic` служит для сбора и подсчета различных характеристик системы в процессе моделирования — одномерного и двумерного распределения, математического ожидания и др.

3.2 Процесс моделирования

В качестве среды для моделирования в реализованной программе выступает глобальный объект `Environment`, представленный на рисунке 8. В процессе моделирования у него вызывается метод `NextMoment`, который переносит систему в следующий момент моделирования, то есть, совершается одна итерация процесса моделирования. Перед началом моделирования вызывается метод `Init`, который подготавливает модель к началу работы. Метод `isFinished` служит для осведомления других участков программы, что моделирование окончено, то есть, что выставленное время моделирования (`maxTime`) равно текущему (`time`).

Для обеспечения гибкости действий, которые модель может совершать в процессе работы, метод `NextMoment` имеет частичную реализацию, в которой вызывается указатель на функцию или лямбда-выражение `mainAction`. Таким образом, поведение модели можно менять в процессе ее работы, что делает класс `Environment` универсальным для многих систем, использующих пошаговое моделирование.

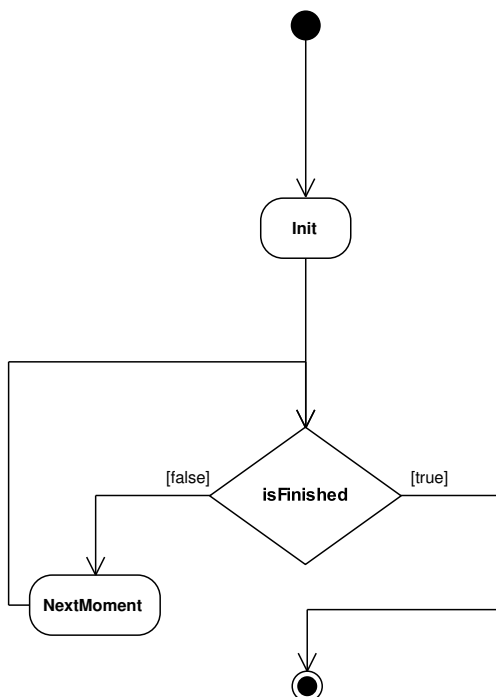


Рисунок 10 — Общий алгоритм процесса моделирования на основе класса `Environment`

Для сбора статистики и построения распределения количества обслуженных заявок, в классе `Environment` предусмотрен механизм интервального сбора данных моделирования. Поле `statInterval` хранит интервал модельного времени, по прошествии которого должен производиться вызов функции или лямбда выражения по указателю `intervalAction`, в котором задается логика сбора данных.

Как видно на рисунке 9, класс `Environment` зависит от класса `Event`, так как в нем содержится статистическая очередь событий `storage`, которые вскоре должны произойти в системе. Наглядно это показано на диаграмме последовательностей:

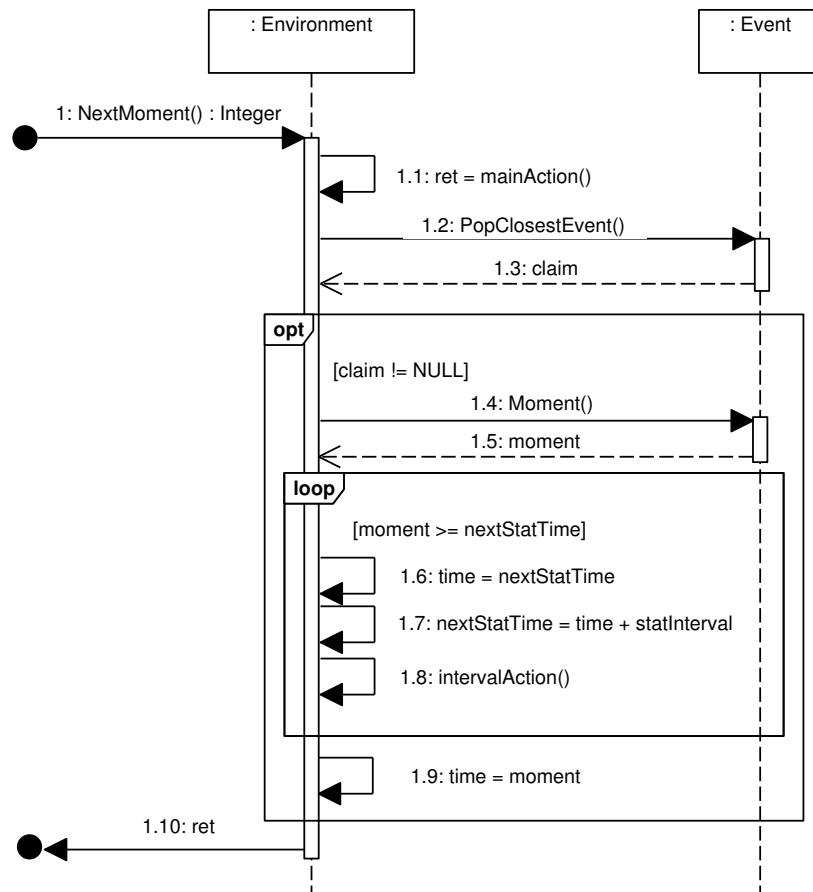


Рисунок 11 — Диаграмма последовательностей метода NextMoment

После выполнения `mainAction` (1.1), из очереди событий извлекается первое предстоящее (1.2). Если такое нашлось, то модельное время перемещается на момент события (1.9). Если это время больше, чем время сбора статистики, то сначала выполняется сбор (1.8), и уже после этого модельное время смещается на момент события. Сбор статистики выполняется в цикле, так как время наступления события может быть больше интервала сбора статистики в несколько раз.

3.3 Функционирование элементов системы массового обслуживания

Функционирование же отдельных элементов системы описывается в методах интерфейса Processable и наследуемых от него INode и IOrbit. Происходящее с обслуживающим прибором во время очередной итерации отражено на диаграммах последовательностей 12 и 13

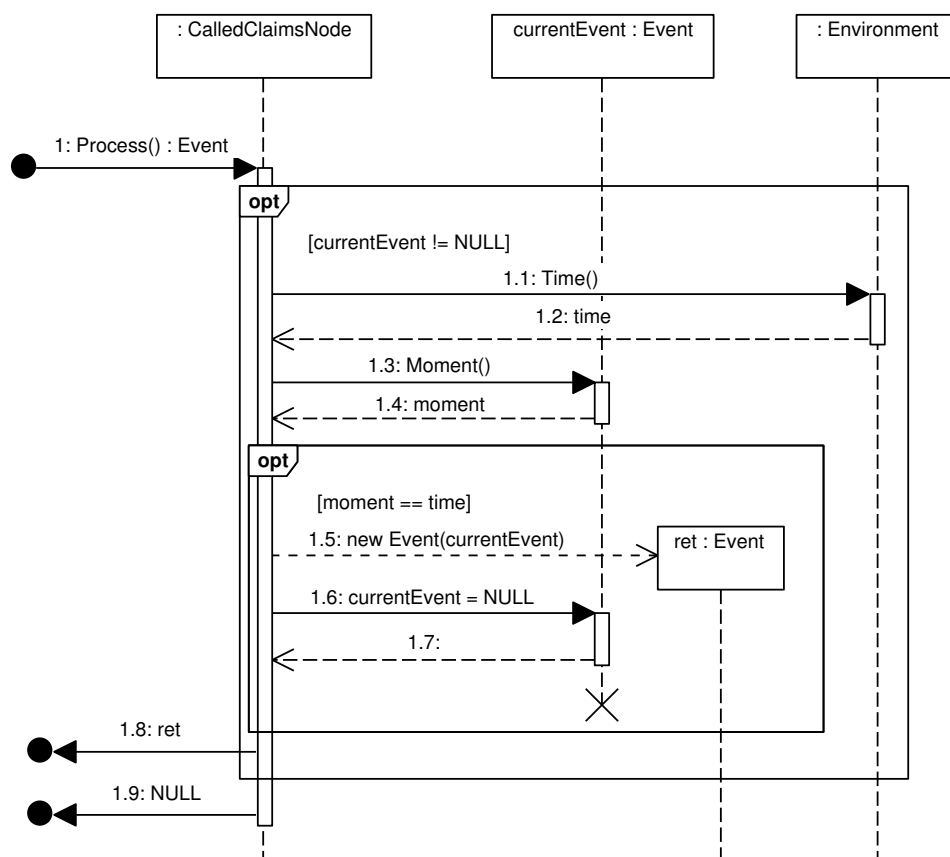


Рисунок 12 — Диаграмма последовательностей метода Process класса CalledClaimsNode

Если прибор на данный момент не обслуживает заявку currentEvent (поле currentEvent на диаграмме 9 обозначено ассоциацией), то возвращается NULL, говорящий о том, что на данной итерации прибор не закончил обслуживание заявки, так как он либо свободен, либо обслуживание еще продолжается. Иначе, производится проверка, совпадает ли текущее модельное время с моментом наступления события. При совпадении обслуженная заявка уходит с прибора (1.5, 1.6, 1.8).

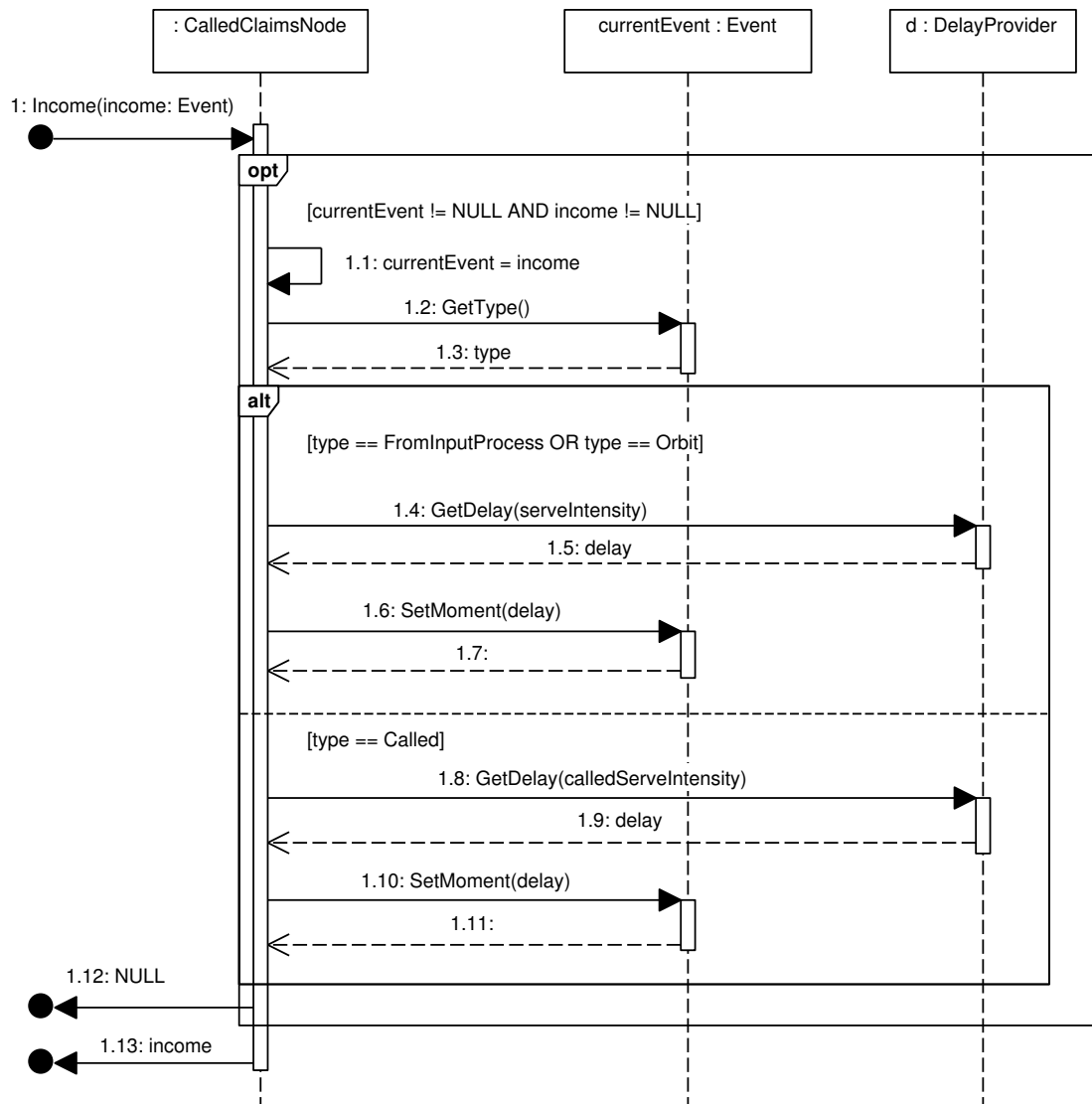


Рисунок 13 — Диаграмма последовательностей метода Income класса CalledClaimsNode

В случае, если пришла заявка, и прибор свободен, пришедшая заявка становится текущей обслуживаемой на приборе (1.1), иначе она возвращается (1.13). В зависимости от типа пришедшей заявки для нее вычисляется время обслуживания — для заявок с входящего потока и орбиты используется `serveIntensity` (1.4, 1.6), а для вызванных используется `calledServeIntensity` (1.8, 1.10).

Подобный алгоритм, заключающийся в сравнении модельного времени с моментом наступления события, реализован и в других элементах системы. В методе `Ассерт` класса `Orbit` также вычисляется задержка для пришедшей заявки. После чего она помещается в коллекцию `claimStorage`, отображенной на диаграмме предметной области (рисунок 9) в качестве агрегации. В методе `Process` (рисунок 14) производится обход данной коллекции со сравнением времени, когда заявка должна покинуть орбиту, и модельным временем. В случае, если нашлась готовая заявка, она извлекается (1.5) из коллекции и возвращается (1.6)

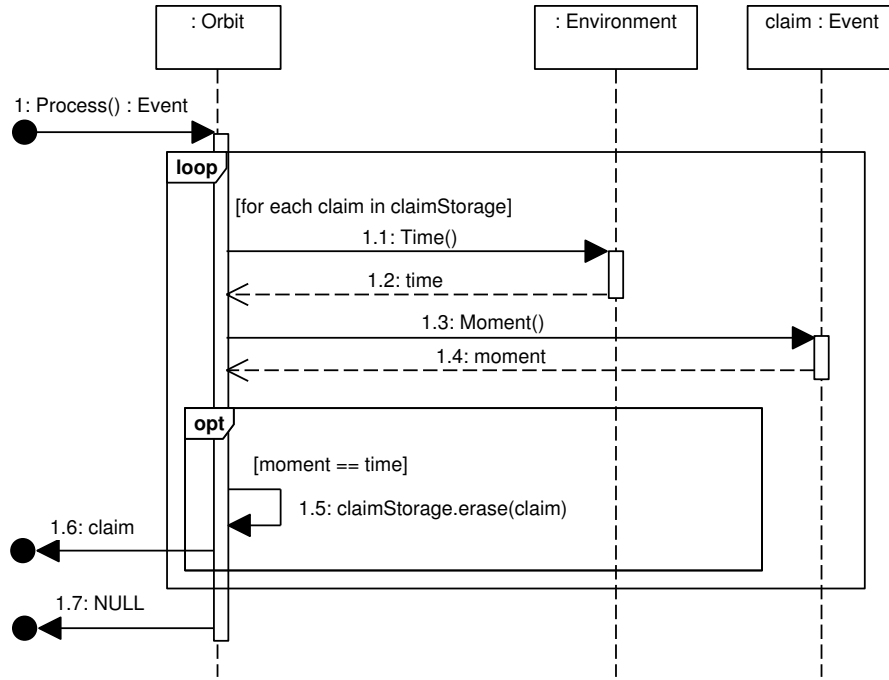


Рисунок 14 — Диаграмма последовательностей метода Process класса Orbit

Тот же подход используется для метода Process классов SimpleInputProcess и MMPPInputProcess. Более подробного рассмотрения требует алгоритм смены состояния управляющей цепи MMPP-потока, реализованного в методе Shift класса MMPPInputProcess.

Как было показано в главе 2, время нахождения в одном из состояний и вероятность перехода в другое для MMPP-потока определяется матрицей инфинитезимальных характеристик Q

$$Q = \begin{bmatrix} q_{11} & \cdots & q_{1n} \\ \vdots & \ddots & \\ q_{n1} & & q_{nn} \end{bmatrix}$$

Интенсивностью, на основе которой вычисляется задержка потока в i -ом состоянии является диагональный элемент матрицы $-q_{i,i}$, а вероятностью перехода из i -ого состояния в j -ое является выражение $\frac{q_{i,j}}{-q_{i,i}}$. В таком случае, при наступлении события смены состояния выполняется следующий алгоритм

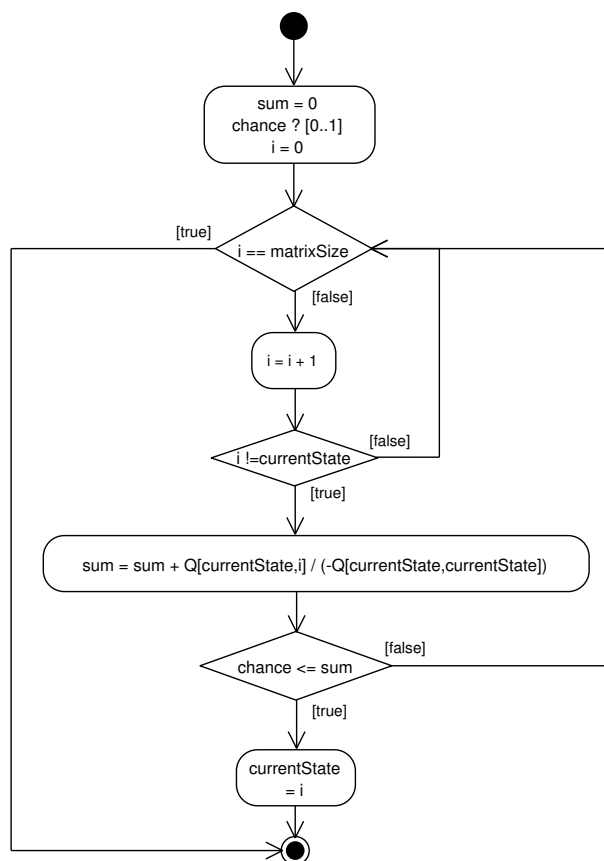


Рисунок 15 — Алгоритм смены состояния ММРР-потока

В начале работы вычисляется случайная величина *chance*, определяется переменная *sum* для суммирования вероятностей по строке матрицы *Q* и счетчик цикла *i*. Цикл идет по строке текущего состояния *currentState*. В случае, если текущий элемент строки недиагональный, к *sum* прибавляется вероятность попадания в состояние *i*, после чего производится проверка, больше ли величина *sum* случайной величины *chance*. Графически это представляется таким образом, что точка *chance* принадлежит отрезку $[0, sum]$, а прибавленная к *sum* на текущей итерации вероятность попадания управляющей цепи в состояние *i* обеспечила принадлежность точки к отрезку, следовательно, прибор принимает состояние *i*. После этого время нахождения управляющей цепи в нем рассчитывается с помощью метода *GetDelay* интерфейса *DelayProvider*.

3.4 Интерфейс и работа программы

Для начала работы с моделью, в первую очередь, необходимо задать параметры моделирования. Среди них:

- Время моделирования (Time limit). Данные единицы являются условными и используются в качестве модельного времени.
- Тип входящего потока (Input Process).
- Интенсивность входящего потока (Simple input). Для ММРР–потока матрица инфинитезимальных характеристик и вектор интенсивностей задается в модельном окне.
- Интенсивность вызова заявок прибором (Calling intensity).
- Интенсивность обслуживания входящих и вызванных заявок (Serving intensity, Called serving intensity)
- Интервал сбора статистики для построения распределения вероятности (Moment T). Так же выражен в единицах модельного времени

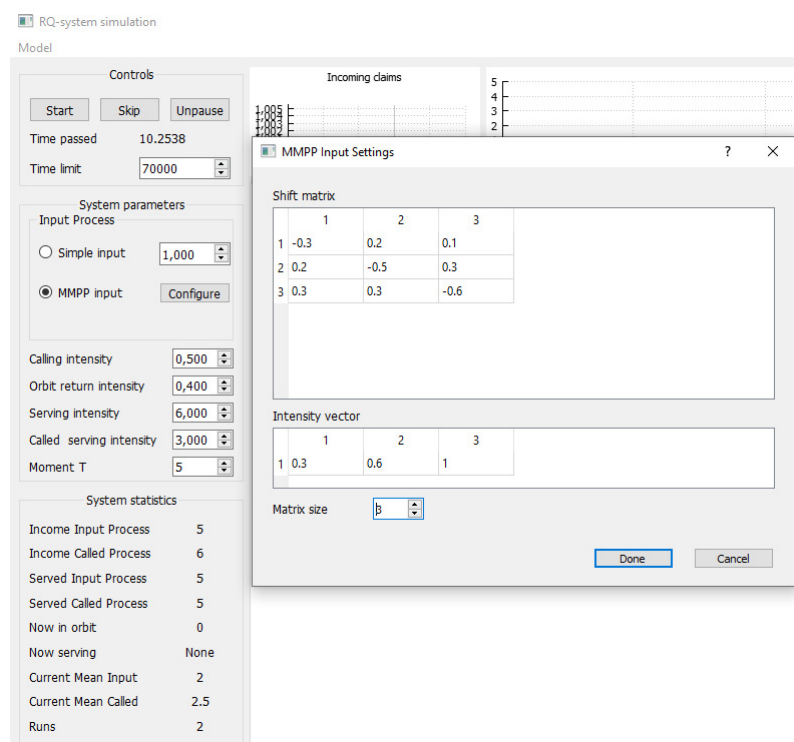


Рисунок 16 — Задание параметров моделирования

Управление процессом моделирования выполняется с помощью специальной области Controls (рисунок 16). Кнопка Start сбрасывает результаты предыдущего запуска модели и начинает моделирование заново с установленными параметрами.

3.4.1 Моделирование в реальном времени

Реализация программы предусматривает два варианта работы с моделью. В первом случае, пользователь имеет возможность в реальном времени наблюдать, как меняется состояние системы с помощью визуализации распределения вероятности числа обслуженных заявок и других характеристик на графиках и в области System statistics и приостанавливать моделирование с помощью кнопки Pause, которая так же служит и для возобновления процесса моделирования. Данный подход реализован с помощью таймера, при каждом тике которого выполняется очередная итерация моделирования. Для настройки длины интервала таймера используется модальное окно конфигурации:

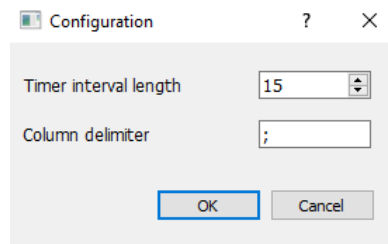


Рисунок 17 — Модальное окно конфигурации

Измененные пользователем настройки сохраняются в файле конфигурации для использования их в последующих запусках программы. Предназначение второго параметра будет описано в следующих разделах.

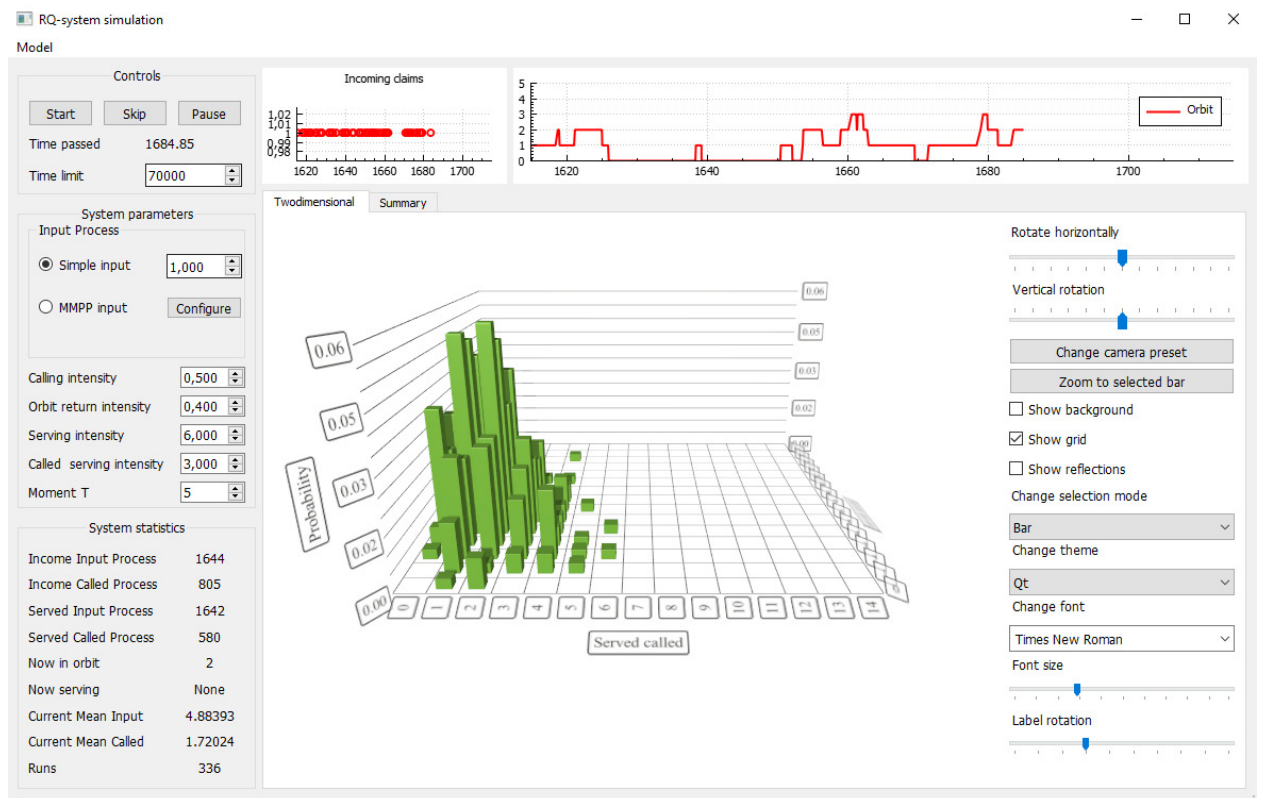


Рисунок 18 — Моделирование работы системы в реальном времени

Как видно на рисунке 18, в момент, когда в системе наступает очередное событие, то есть, ее состояние меняется, производится сбор статистики и ее показ в реальном времени. Основная статистика работы модели показана в области System statistics, которая содержит следующую информацию:

- Количество пришедших заявок входящего потока (Income Input Process).
- Количество заявок, вызванных прибором (Income Called Process).
- Количество обслуженных заявок входящего потока (Served Input Process).
- Количество обслуженных вызванных прибором заявок (Served Called Process).
- Количество заявок на орбите в данный момент (Now in orbit).
- Тип в данный момент обслуживаемой заявки (Now serving).
- Математическое ожидание обслуженных заявок входящего потока (Current Mean Input).
- Математическое ожидание обслуженных вызванных прибором заявок (Current Mean Called).
- Количество сборов статистики, интервал которых задан параметром Moment T (Runs).

Поскольку математическое ожидание вычисляется на основе полученного на данный момент распределения вероятности, обновление полей Current Mean Input и Current Mean Called происходит с интервалом Moment T. Для отображения области System statistics используется ранее упомянутая структура Statistic (рисунок 9), которая служит для накопительного сбора статистики и подсчета распределения и математического ожидания.

Помимо текстового представления, изменение состояния системы отражено на графиках. Левый верхний график отображает моменты поступления заявок входящего потока. По оси X отсчитывается модельное время.



Рисунок 19 — Отображение времени прихода заявок в систему на графике

Правый верхний график отображает изменение количества заявок на орбите в случае моделирования системы с простейшим входящим потоком и историю изменения состояния управляющей цепи ММРР–потока при моделировании системы с ММРР–потоком. По оси X для обоих графиков отсчитывается модельное время. По оси Y количество заявок на орбите, либо состояние управляющей цепи.

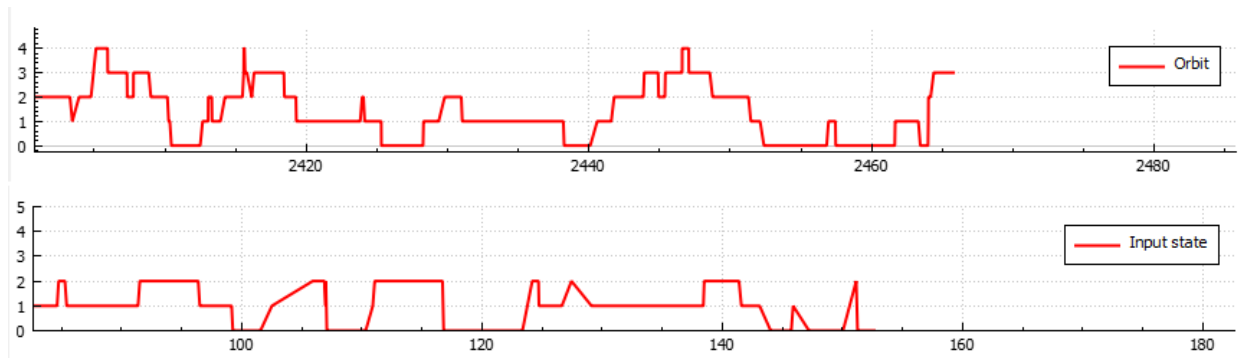


Рисунок 20 — Отображение изменения размера орбиты (верхний) и состояния управляющей цепи (нижний) на графике

Графики на рисунках 19 и 20 имеют опцию масштабирования к выделенной пользователем области.

3.4.2 Ускоренное моделирование

Второй вариант работы с программой предполагает, что пользователю требуется быстро получить результаты моделирования. Для этого служит кнопка Skip в области Controls, при нажатии которой таймер останавливается, и производится моделирование системы до установленного времени в цикле. В таком случае, программа отобразит модальное окно, уведомляющее пользователя, что на данный момент выполняется моделирование, блокируя основное.

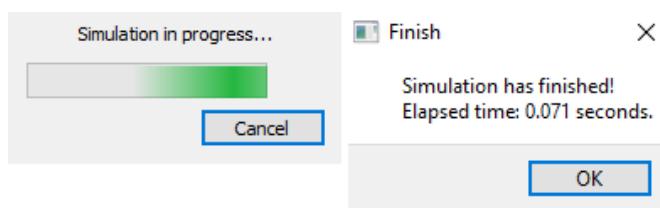


Рисунок 21 — Модальные окна уведомления пользователя о статусе процесса моделирования

По завершении моделирования отображаемая статистика и главный график обновляются в соответствии с актуальными результатами. По нажатии кнопки отмены (Cancel) в модальном окне статуса моделирования цикл закончит свою работу, а таймер возобновит отсчет. Таким образом, программа продолжит моделирование в реальном времени.

3.4.3 Работа с главным графиком

Важнейшую информацию о работе системы отображает главный график, занимающий наибольшую рабочую область окна. На нем отображается текущее двумерное распределение числа обслуженных заявок входящего потока и вызванных заявок в виде трехмерных столбцов. Распределение обновляется в модельный момент времени, кратный параметру Moment T. По осям X и Y отложены количество обслуженных заявок входящего потока вызванных заявок соответственно.

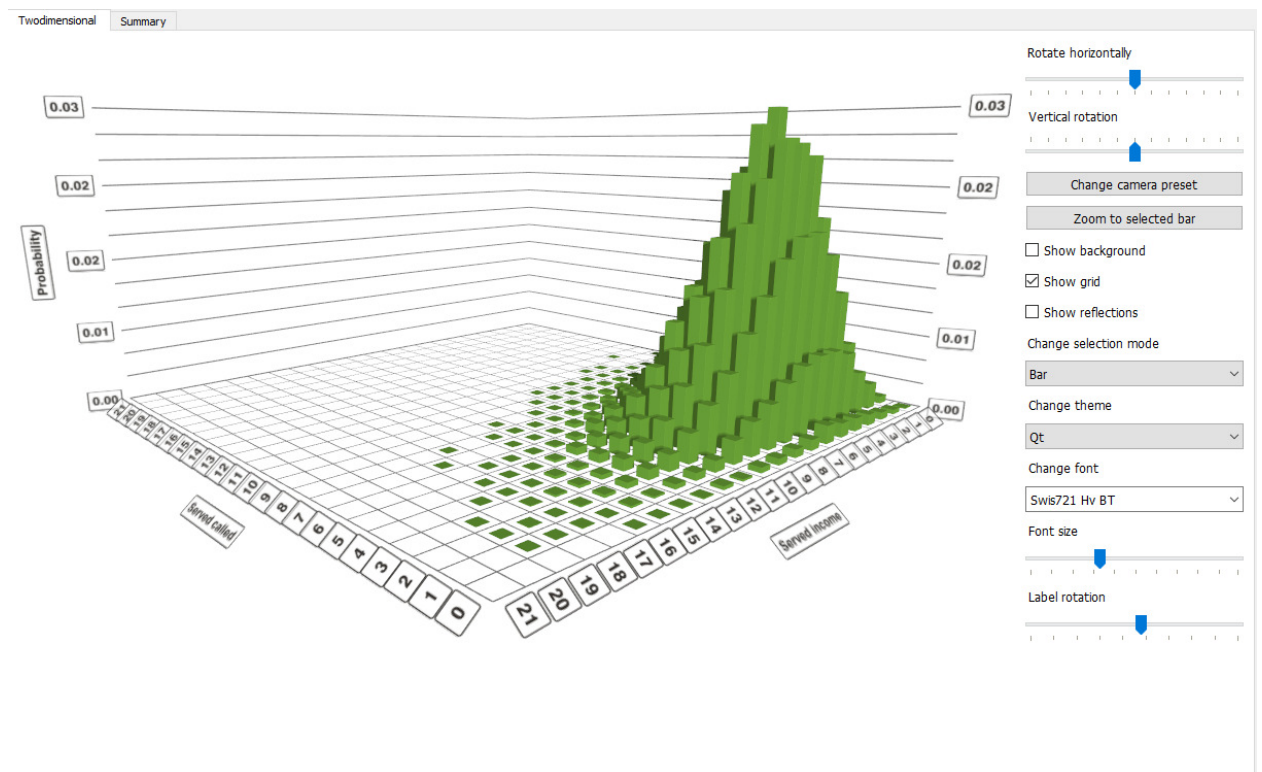


Рисунок 22 — Отображение распределения вероятности на главном графике

Рабочая область графика позволяет проводить предварительный анализ работы системы. В первую очередь график поддерживает масштабирование и перемещение в пространстве отображаемой области с помощью мыши, клавиатуры и слайдеров поворота камеры относительно графика по горизонтали (Rotate horizonatally) и вертикали (Vertical Rotation). Также, пользователь имеет возможность выделить любой столбец на графике для получения о нем подробной информации. Помимо этого, кнопка Zoom to selected bar при нажатии позволяет масштабировать и переносить камеру к выбранному столбцу автоматически.

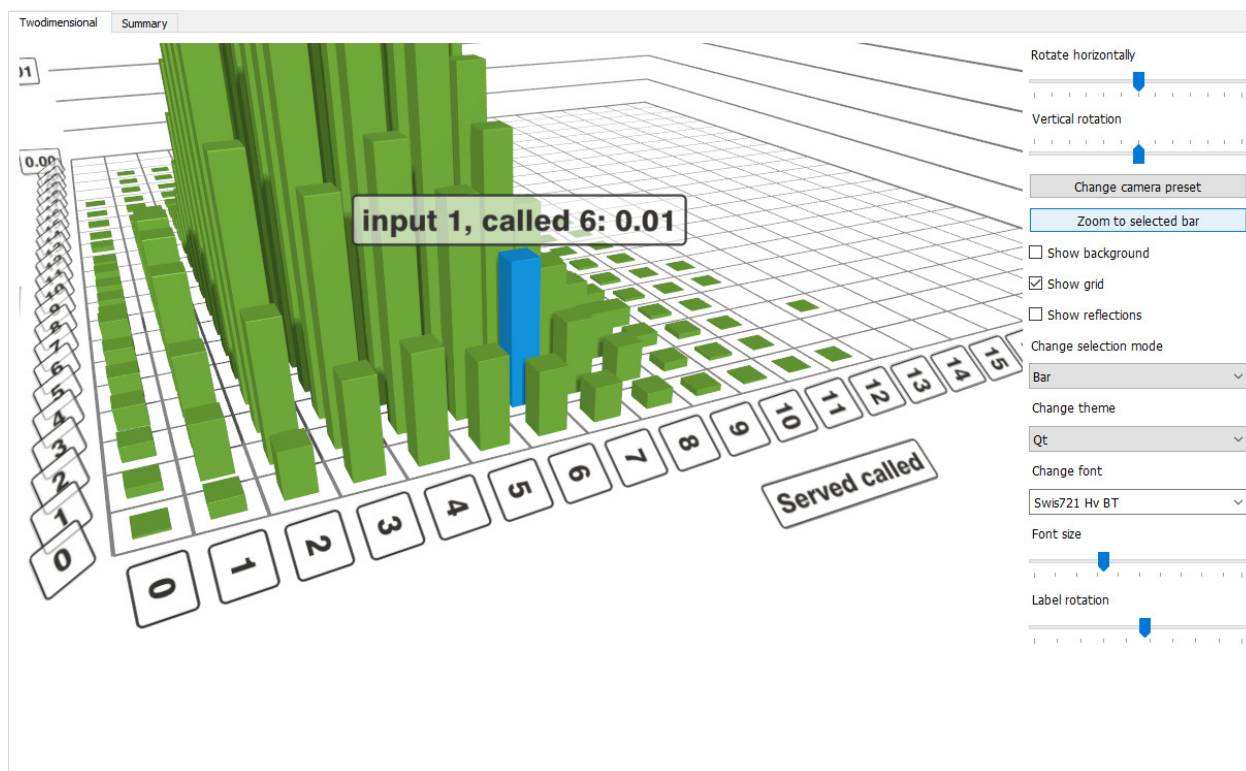


Рисунок 23 — Перенос камеры к выбранному столбцу

Выделение участков графика реализовано в различных вариациях:

- Выделение столбца
- Выделение ряда
- Выделение ряда и столбца, так же, с подробной информацией о выделенном столбце
- Перекрестное выделение рядов
- Перекрестное выделение рядов с выделением столбца
- Срез по ряду
- Срез по ряду с выделением столбца

В отличие от других вариантов выделения, при выборе среза выделенного ряда строится дополнительная область, в которой выбранный ряд отображен в качестве отдельного графика. По нажатии на трехмерное представление графика в левом верхнем углу, выделение сбрасывается.

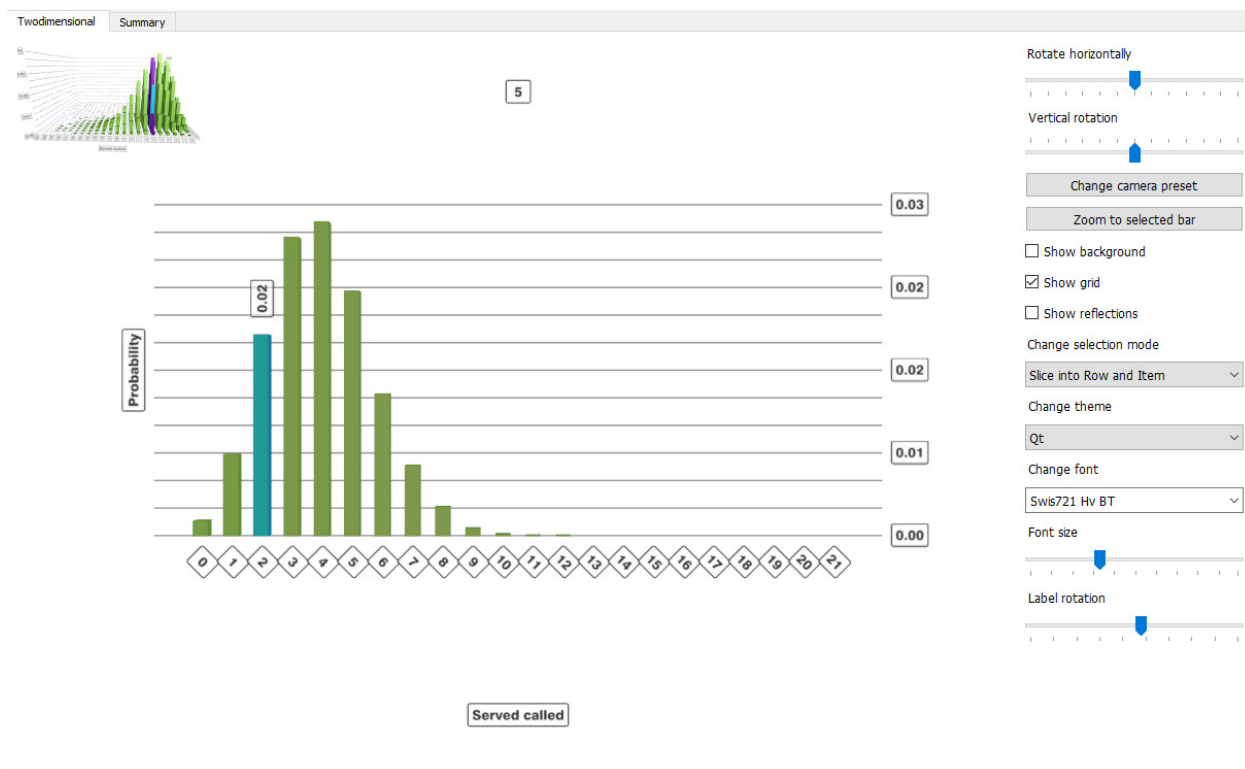


Рисунок 24 — Выделение с построением среза по выбранному ряду и столбцу

Также, для удобной работы с визуальной точки зрения, трехмерный график предусматривает настройку отображения графических элементов. Так, пользователь может изменять видимость фона (Show background), сетки (Show grid), отражений поверхностей (Show reflections), цветовую схему (Change theme), шрифт (Change font), размер шрифта (Font size) и поворот ярлыков (Label Rotation).

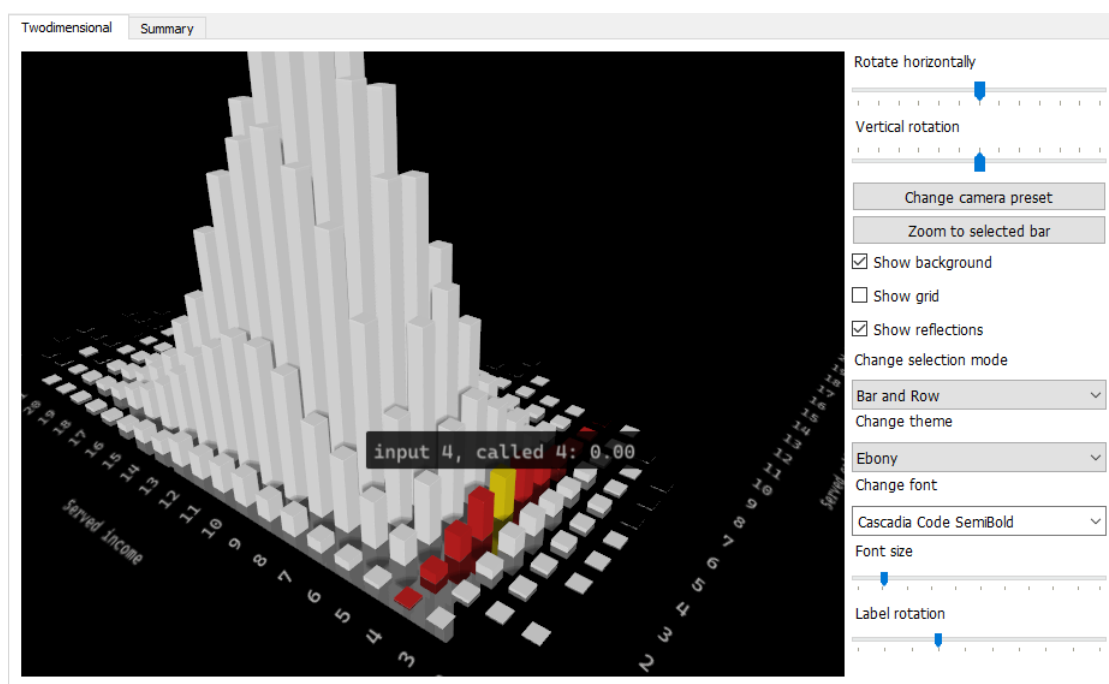


Рисунок 25 — Настройка отображения основного графика

Также, помимо двумерного распределения вероятностей числа обслуженных заявок, пользователь может наблюдать распределение вероятностей суммарного числа обслуженных заявок на вкладке Summary основного графика. Это представление так же, как и другие графики в программе, имеют опцию масштабирования к выделенной области. Переключение между двумя представлениями распределения вероятностей может выполняться пользователем в любой момент работы программы за исключением времени, когда производится ускоренное моделирование.

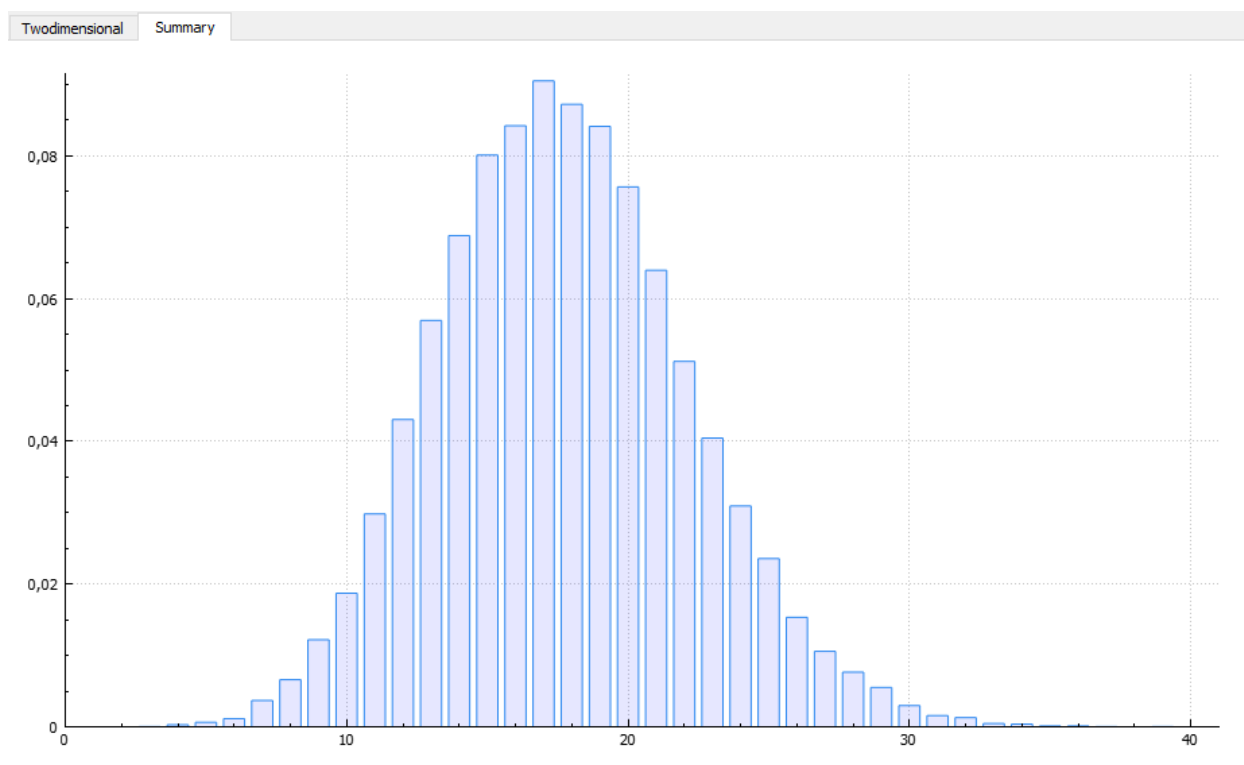


Рисунок 26 — Вкладка основного графика, отображающая распределение вероятностей суммарного числа обслуженных заявок

3.4.4 Экспорт результатов моделирования

В меню Model основного окна имеется опция экспорта в текстовый файл текущего распределения вероятности числа обслуженных заявок в двумерном виде и суммарном.

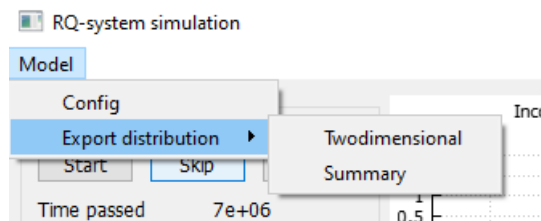


Рисунок 27 — Опции экспорта

При выборе опции Summary распределение будет записано в файл формата CSV,

где каждое значение будет с новой строки. При выборе опции Twodimensional распределение будет записано в виде таблицы с разделителем, установленным пользователем в окне конфигурации (рисунок 17). По умолчанию разделителем является символ ";". Поскольку формат CSV является основным при работе с данными ввиду удобного табличного представления с помощью разделителей, импортировать и использовать полученные при моделировании данные можно в широком спектре программного обеспечения. Для анализа результатов в данной работе используется система компьютерной алгебры Mathcad.

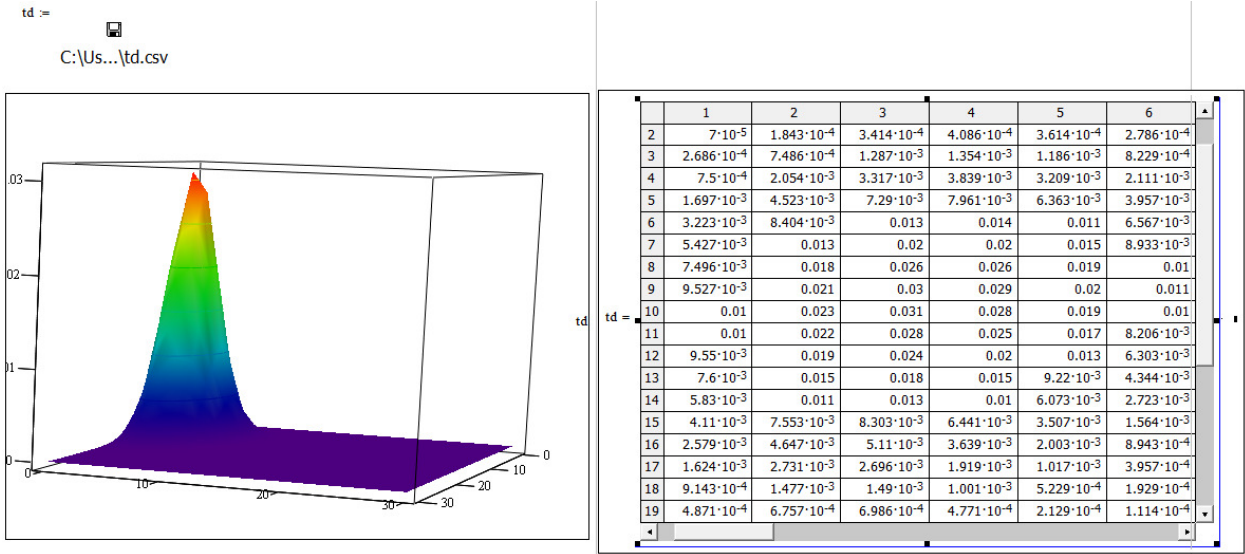


Рисунок 28 — Результаты моделирования, импортированные в среду Mathcad

На рисунке 28 представлен трехмерный график плоскости двумерного распределения вероятности числа обслуженных заявок (слева) и таблица распределения (справа), полученных в результате моделирования.

3.5 Особенности реализации

Реализация предметной области программы выполнена на языке программирования C++ стандарта 2017 года (C++1z). Оболочка программы и графический интерфейс были выполнены на базе фреймворка для разработки кроссплатформенного программного обеспечения QT. Таким образом, представленная программа является кроссплатформенной, то есть, может быть собрана для работы в операционных системах GNU/Linux, Microsoft Windows и macOS. Выбор в сторону разработки на C++ был сделан, в первую очередь, из-за низкоуровневого программирования, которое позволяет писать эффективный код и более точно распоряжаться используемыми ресурсами системы. По этой же причине, C++ отлично подходит для работы с графикой. Также, в C++ поддерживается объектно-ориентированный подход к программированию, что важно для реализации ранее описанной предметной области программы.

Минимальные системные требования для работы программы:

- Процессор с тактовой частотой 1.8 ГГц и выше.
- Оперативная память емкостью 2 Гб и больше.
- Видеоадаптер и монитор VGA с разрешением 1280 x 1024 и выше.
- 50 Мб свободного места на диске
- Клавиатура и мышь

3.5.1 Генерация псевдослучайных чисел

Поскольку генераторы псевдослучайных чисел во многих языках программирования, в том числе, в C++ генерируют новые значения на основе системного времени или такта процессора, возникает проблема повторения последовательности значений при частой генерации, например, в цикле. А поскольку для расчета экспоненциальной задержки каждого события в модели используется генерация случайных чисел, возникновение этой проблемы неизбежно.

Решением проблемы стало использование вихря Мерсенна — генератора псевдослучайных чисел, основывающимся на свойствах простых числе Мерсенна. Данный генератор лишен многих недостатков других генераторов, таких как предсказуемость, малый период и легко выявляемые статистические закономерности.

В C++ данный алгоритм имеет реализацию под названием MT19937 и содержится в стандартной библиотеке, начиная со стандарта C++11. Его период составляет 2^{19937} , чего с излишком хватит для продолжительного процесса моделирования.

В коде использование вихря Мерсенна представлено следующим образом

```
#include <random>
```

```

std::random_device rd;
std::mt19937 gen(rd());
std::uniform_real_distribution<double> distribution(0,1);

double NextDouble()
{
    return distribution(gen) ;
}

```

3.5.2 Отображение статуса при ускоренном моделировании

При ускоренном моделировании возникает проблема отображения статуса моделирования без блокировки основа потока, в котором производится отрисовка графического интерфейса. Проблема заключается в том, что цикл, в котором производится моделирование системы блокирует отрисовку графического интерфейса до окончания его работы, а все объекты графического интерфейса должны находиться в основном потоке программы, что исключает создание параллельного потока для отрисовки. Так, пользователь не может удостовериться, что программа продолжает свою работу, ибо процесс перестает отвечать на сигналы операционной системы.

```

class MainWindow : public QMainWindow
{
    ...
private:
    ...
    QFutureWatcher<void> watcher;
    ...
};

```

```

void MainWindow::skip()
{
    if(Environment::Time()>0 && !Environment::isFinished()){
        ...
        timer->stop();
        skipping = true;
        QFuture<void> future = QtConcurrent::run([this]{
            while (!Environment::isFinished() && skipping){
                Environment::NextMoment();
            }
        });
        watcher.setFuture(future);
        progressDialog.exec();
    }
}

```

Для решения данной проблемы был использован инструмент фреймворка QtConcurrent, предназначенный для параллельного запуска ресурсоёмких задач. Так, основной поток

будет занят отрисовкой статуса моделирования (рисунок 21), а моделирование будет происходить асинхронно, что положительно скажется на быстродействии.

```
connect(&watcher,&QFutureWatcher<void>::finished,this,[=]() {  
    progressDialog.close();  
    updateTime();  
});
```

Для уведомления модального окна об окончании цикла моделирования служит объект `QFutureWatcher`, который осуществляет привязку события завершения работы потока к лямбда-функции, в которой производится закрытие окна статуса и обновление отображаемой статистики.

4 Численные эксперименты

4.1 Проверка стабильности имитационной модели

Для начала проведения численного анализа полученных решений необходимо проверить точность работы реализованной имитационной модели. Для проверки, запустим модель несколько раз с одинаковыми параметрами и для полученных результатов вычислим критерий согласия Колмогорова. Критерий согласия Колмогорова (расстояние Колмогорова) предназначен для проверки гипотезы о том, что некое эмпирическое распределение, в данном случае, распределение, построенное в ходе работы имитационной модели, соответствует предполагаемой модели, которая, в данном случае, так же является результатом работы имитационной модели.

Расстояние Колмогорова вычисляется по следующей формуле

$$\Delta_l = \max_{0 \leq i \leq \infty} \left| \sum_{v=0}^i (P(v) - P^l(v)) \right|, l = 1, 2$$

Для численного анализа в данной работе применяется система компьютерной алгебры Mathcad. В ней были построены графики распределения вероятностей и вычислено расстояния Колмогорова для двух запусков имитационной модели с одинаковыми параметрами системы

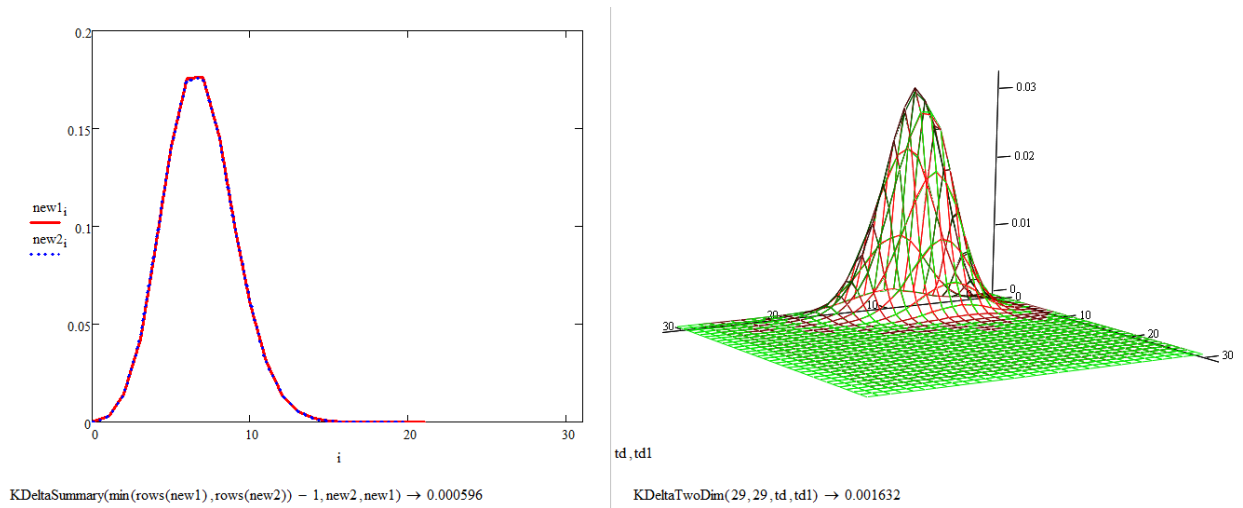


Рисунок 29 — Сравнение двух отдельных запусков имитационной модели с простейшим входящим потоком

На рисунке 29 представлено сравнение двух запусков имитационной модели с параметрами системы: $\lambda = 1, \alpha = 0.5, \sigma = 0.4, \mu_1 = 6, \mu_2, t = 10$. Запуски модели дали практически одинаковые результаты. Вычисленное расстояние Колмогорова составило 0.000596 для суммарного распределения и 0.0016632 для двумерного распределения. Графики плотности распределения вероятностей полностью накладываются друг на друга.

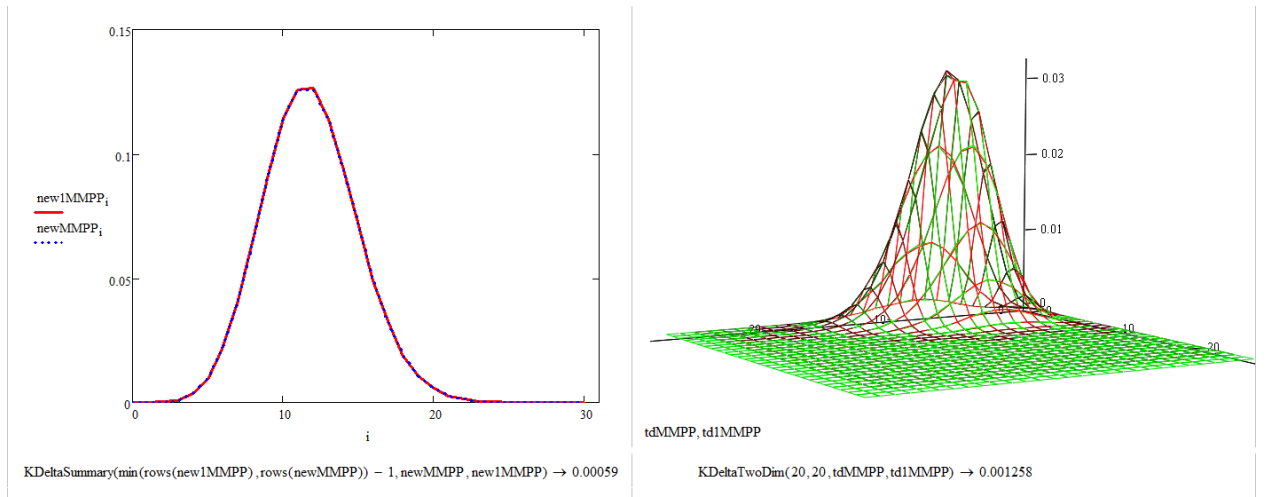


Рисунок 30 — Сравнение двух отдельных запусков имитационной модели с ММРР-поток

Данный эксперимент был проведен и для системы с входящим ММРР-поток. Были заданы следующие параметры: $\alpha = 0.5, \sigma = 0.4, \mu_1 = 6, \mu_2, t = 10$,

$$Q = \begin{bmatrix} -0.3 & 0.1 & 0.1 \\ 0.2 & -0.5 & 0.3 \\ 0.3 & 0.3 & -0.6 \end{bmatrix}$$

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.6 & 0 \\ 0 & 0 & 0.7 \end{bmatrix}.$$

Результаты эксперимента представлены на рисунке 30. Для суммарного распределения расстояние Колмогорова составило 0.00059, для двумерного — 0.001258. Графики плотности распределений также полностью накладываются друг на друга.

Исходя из вышеописанных экспериментов, можно сделать вывод, что реализованная имитационная модель дает стабильные результаты, что подтверждают вычисленные значения расстояния Колмогорова для обоих видов распределений, следовательно, модель можно использовать для анализа полученных решений и оценки их применимости.

4.2 Сравнение распределений вероятностей с эмпирическим распределением

Теперь, когда стабильность имитационной модели подтверждена экспериментально, сравним результаты ее работы с полученным асимптотическим приближением функции распределения вероятностей числа обслуженных заявок для систем, рассмотренных в разделах 1.1, 1.2 и 2.1 при разной интенсивности возврата заявок с орбиты.

Значение этого параметра влияет на точность при сравнении, так как решение систем было получено при асимптотическом условии большой задержки заявок на орбите.

Для того, чтобы проиллюстрировать влияние задержки заявок на орбите на получаемый результат, зададим исходные параметры:

$$\lambda = 2, \alpha = 0.9, \mu_1 = 3.5, \mu_2 = 2.1, t = 15 \quad (43)$$

Теперь, рассчитаем распределения вероятностей числа обслуженных заявок по формуле (12) и запустим имитационную модель, варьируя интенсивность возвращения заявок с орбиты σ

Таблица 1 — Расстояние Колмогорова при различных значениях параметра σ

σ	1	0.6	0.4	0.2	0.1	0.05	0.01
Δ	0.0148	0.0112	0.0096	0.0062	0.0044	0.0023	0.0019

Как видно в таблице 1, при сравнении работы имитационной модели и асимптотических результатов расстояние Колмогорова тем меньше, чем меньше интенсивность возврата заявок с орбиты. Однако, даже при больших значениях σ , таких как 1 и 0.6, мы получаем результаты, где расстояние Колмогорова составляет не больше двух сотых.

Чтобы проверить, наблюдается ли подобная тенденция при большей загруженности системы, установим в ранее описанных параметрах (43) $\lambda = 2.7$. Тогда получим следующий результат

Таблица 2 — Расстояние Колмогорова при различных значениях параметра σ с увеличенной интенсивностью прихода заявок

σ	1	0.6	0.4	0.2	0.1	0.05	0.01
Δ	0.0148	0.0112	0.0096	0.0062	0.0044	0.0023	0.0019

Список литературы

- [1] Назаров Анатолий Андреевич, Терпугов Александр Федорович [et al.]. Теория массового обслуживания: учебное пособие по специальностям 010200 (010501) "Прикладная математика и информатика 061800 (080116) "Математические методы в экономике". — 2010.
- [2] Erlang Agner Krarup. The theory of probabilities and telephone conversations // Nyt. Tidsskr. Mat. Ser. B. — 1909. — Vol. 20. — P. 33–39.
- [3] Artalejo Jesus R. Accessible bibliography on retrial queues: progress in 2000–2009 // Mathematical and computer modelling. — 2010. — Vol. 51, no. 9-10. — P. 1071–1081.

- [4] Artalejo Jesús R., Gómez-Corral Antonio. Retrial Queueing Systems: A Computational Approach. — Springer-Verlag Berlin Heidelberg, 2008.
- [5] Falin Gennadi, Templeton James GC. Retrial queues. — CRC Press, 1997. — Vol. 75.
- [6] Bjørnstad Jon Henrik. Traffic characteristics and queueing theory: implications and applications to web server systems : Master's thesis / Jon Henrik Bjørnstad. — 2006.
- [7] Kritzinger P. A performance model of the OSI communication architecture // IEEE transactions on communications. — 1986. — Vol. 34, no. 6. — P. 554–563.
- [8] Олифер В. Г., Олифер Н. А. Компьютерные сети. Принципы, технологии, протоколы. — Питер, 2010.
- [9] Falin GI. Model of coupled switching in presence of recurrent calls // Engineering Cybernetics. — 1979. — Vol. 17, no. 1. — P. 53–59.
- [10] Artalejo Jesús R, Resing Jacobus Adrianus Cornelis. Mean value analysis of single server retrial queues // Asia-Pacific Journal of Operational Research. — 2010. — Vol. 27, no. 03. — P. 335–345.
- [11] Baiocchi Andrea, Blefari-Melazzi Nicola. Steady-state analysis of the MMPP/G/1/K queue // IEEE transactions on communications. — 1993. — Vol. 41, no. 4. — P. 531–534.
- [12] Lapatin Ivan, Nazarov Anatoly. Asymptotic Analysis of the Output Process in Retrial Queue with Markov-Modulated Poisson Input Under Low Rate of Retrials Condition // International Conference on Distributed Computer and Communication Networks / Springer. — 2019. — P. 315–324.
- [13] Meier-Hellstern Kathleen S. A fitting algorithm for Markov-modulated Poisson processes having two arrival rates // European Journal of Operational Research. — 1987. — Vol. 29, no. 3. — P. 370–377.
- [14] An MMPP-based hierarchical model of Internet traffic / Luca Muscariello [et al.] // 2004 IEEE International Conference on Communications (IEEE Cat. No. 04CH37577) / IEEE. — Vol. 4. — 2004. — P. 2143–2147.
- [15] Asymptotic analysis of Markovian retrial queue with two-way communication under low rate of retrials condition / Anatoly A Nazarov [et al.]. — 2017.
- [16] Bronson Richard. Matrix methods: An introduction. — Gulf Professional Publishing, 1991.
- [17] Егоров И. М. Программирование: Учебное методическое пособие. — Томск: Томский межвузовский центр дистанционного образования, 2006. — P. 79.

- [18] Fischer Wolfgang, Meier-Hellstern Kathleen. The Markov-modulated Poisson process (MMPP) cookbook // Performance evaluation. — 1993. — Vol. 18, no. 2. — P. 149–171.
- [19] MEIER KS. A STATISTICAL PROCEDURE FOR FITTING MARKOV-MODULATED POISSON PROCESSES (APPLIED PROBABILITY). — 1986.
- [20] Kendall David G. Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded Markov chain // The Annals of Mathematical Statistics. — 1953. — P. 338–354.
- [21] Kulkarni Vidyadhar G. On queueing systems with retrials // Journal of Applied Probability. — 1983. — P. 380–389.
- [22] Gharbi Nawel, Dutheillet Claude. An algorithmic approach for analysis of finite-source retrial systems with unreliable servers // Computers & Mathematics with Applications. — 2011. — Vol. 62, no. 6. — P. 2535–2546.
- [23] Burke P.J. The output process of a stationary M/M/s queueing system // The Annals of Mathematical Statistics. — 1968. — Vol. 39, no. 4. — P. 1144–1152.
- [24] Paul Svetlana, Phung-Duc Tuan. Retrial Queueing Model with Two-Way Communication, Unreliable Server and Resume of Interrupted Call for Cognitive Radio Networks // Information Technologies and Mathematical Modelling. Queueing Theory and Applications. — Springer, 2018. — P. 213–224.
- [25] Mirasol Noel M. The output of an M/G/∞ queueing system is poisson // Operations Research. — 1963. — Vol. 11, no. 2. — P. 282–284.
- [26] Daley D.J. Queueing output processes // Advances in Applied Probability. — 1976. — Vol. 8, no. 2. — P. 395–415.