

Содержание

Аннотация	2
Введение	3
1 Литературный обзор	5
1.1 Обзор методов обнаружения разладки в случайных процессах . .	5
1.1.1 Тест Неймана–Пирсона	6
1.1.2 Тест Вальда	7
1.1.3 Сравнение методов Неймана–Пирсона и Вальда для Винеров- ского процесса	9
1.2 Концептуальная модель компьютерной имитации транспортных потоков на двухполосных автомобильных дорогах	10
2 Математическая модель	21
2.1 Концептуальная постановка задачи	21
2.2 Формальная постановка задачи	21
2.3 Математическая постановка задачи	23
3 Программная реализация	26
3.1 Модуль обнаружения разладки	27
3.2 Модуль взаимодействия с системой ВАДС	27
3.3 Интегральный модуль	28
3.4 Статистические испытания	29
Заключение	34
Список использованных источников	35
Приложение А	37
Исходный код модуля обнаружения разладки	37

Аннотация

В данной дипломной работе описываются некоторые современные методы обнаружения разладки в случайных процессах. Метод Неймана-Пирсона и последовательный тест Вальда. Последний из которых используется в разработанной программе в целях наискорейшего обнаружения смены интенсивности движения с заданной точностью. А так же система ВАДС, использующаяся в программе для определения пропускной способности перекрёстка.

Основным результатом проведённой работы является программный продукт, обладающий простым и интуитивно понятным интерфейсом, который позволяет в реальном времени отслеживать изменение пропускной способности перекрёстка и информировать пользователя о возможности возникновения затора.

Введение

Невозможно представить современную жизнь при отсутствии дорожного транспорта. Сегодня у многих жителей России есть свой личный автомобиль, тогда как отыскать человека, никогда в жизни не пользовавшегося общественным транспортом довольно сложная задача. Это безусловно удобно. Но с развитием этой отрасли, неизбежно и появление издержек. Машин становится всё больше, плотность проживания людей в результате урбанизации увеличивается. Эти тенденции приводят к увеличению загруженности транспортных систем. Встаёт вопрос о том, чтобы в автоматическом режиме обнаруживать слабые места в транспортной сети и распределять нагрузку с наиболее загруженных участков.

Для решения этих проблем и были придуманы интеллектуальные транспортные системы. ИТС — это интеллектуальная система, использующая инновационные разработки в моделировании транспортных систем и регулировании транспортных потоков, предоставляющая конечным потребителям большую информативность и безопасность, а также качественно повышающая уровень взаимодействия участников движения по сравнению с обычными транспортными системами. Особенность ИТС относительно других видов наблюдения за транспортными потоками или их моделирования заключается в том, что ИТС даёт возможность в автоматическом режиме сигнализировать о наступлении событий, ведущих к возникновению на дороге проблемных ситуаций.

ИТС включает в себя информационные и коммуникационные технологии в сфере автотранспорта (включая инфраструктуру, транспортные средства, участников системы, а также дорожно-транспортное регулирование), и имеющую наряду с этим возможность взаимодействия с другими видами транспорта.[2]

Одной из важнейших задач интеллектуальных транспортных систем является заблаговременное обнаружение опасности возникновения заторов на пересечениях транспортных потоков, вызванных ограниченной пропускной способностью этих участков. Наискорейшее обнаружение опасности затора даёт возможность соответствующим дорожным службам успеть подготовиться к этой проблеме и принять какие-либо меры для её предотвращения. Пропускная способность перекрёстка зависит от множества факторов, таких как состав движения, угол об-

зора, качество покрытия. Но основным параметром является интенсивность движения на встречном и перекрёстных направлениях.

Для измерения интенсивности движения на участке автодороги современные технические средства позволяют определять интервалы времени, отделяющие два ближайших автомобиля. Тогда значение мгновенной интенсивности движения на данном участке является отношением количества машин, проехавших за данный промежуток к его длительности. Вычисляемые данным способом интенсивности составляют случайный процесс.

Целью работы является разработка программного комплекса, позволяющего с заданной точностью определить, сменилась ли интенсивность движения, насколько она сменилась, и появилась ли в новой ситуации опасность возникновения затора.

Работа состоит из трёх основных разделов. В первом рассмотрены различные подходы к обнаружению разладки в случайных процессах и их сравнение, а так же описана модель имитации транспортных потоков на двухполосных автомобильных дорогах, позволяющая вычислять величину пропускной способности перекрёстка. Во втором разделе описывается концептуальная, формальная и математическая постановки задачи и её решение.

1 Литературный обзор

1.1 Обзор методов обнаружения разладки в случайных процессах

В современном мире методы теории принятия решений в условиях неопределённости используются довольно широко. Есть несколько классических книг, посвящённых теории принятия решений, нацеленных на решение конкретных задач общего интереса, возникающих при динамическом анализе (в режиме реального времени) статистических данных, получаемых, например, в финансовой инженерии, в теории обнаружения сигналов на фоне помех и т.д. В первую очередь стоит отметить такие монографии как [3] и [4].

Из русскоязычных авторов наибольший вклад в изучение этой теории осуществил Альберт Николаевич Ширяев. В частности, вопрос об оптимальной остановке Винеровского процесса был подробно рассмотрен в книге [1]. В данной работе автор рассматривает методы обнаружения нежелательных внедрений, основанного на Anomaly Detection Systems.

Обычно, внедрение в сети происходит в неизвестный заранее момент времени θ и сопровождается изменением вероятностно-статистических свойств некоторых характеристик наблюдаемого процесса (например, количества отправленных и принятых сервером пакетов). Поэтому естественно возникает идея математически сформулировать задачу обнаружения атаки как задачу скорейшего обнаружения момента θ появления разладки в ходе наблюдаемого процесса.

Основным аппаратом решения таких задач является последовательный анализ принятия решений. Такие задачи удобно формулировать как задачи об оптимальной остановке. Методы их решения в значительной мере опираются на современный аппарат теории случайных процессов, стохастического исчисления, теории мартингалов, нелинейной фильтрации и т.д.

Для начала рассмотрим различные способы решения задачи о разладке, описанные в современной литературе.

1.1.1 Тест Неймана–Пирсона

Предположим, что наблюдаемые данные описываются числовой последовательностью

$$x_1, x_2, \dots, x_n, \dots$$

являющейся результатом наблюдений над независимыми одинаково распределёнными случайными величинами

$$\xi_1, \xi_2, \dots, \xi_n, \dots$$

Далее мы будем считать, что ξ_k — это одномерные случайные величины, и предполагать, что одномерные функции распределения $F_\theta = F_\theta(x) (= P_\theta(\xi_n \leq x))$ имеют (при любых $n \geq 1$) плотности $f_\theta(x)$:

$$dF_\theta(x) = f_\theta(x)dx$$

Независимость и одинаковая распределённость означают, что плотность $p_\theta(x_1, x_2, \dots, x_n)$ совместного распределения $F_\theta(x_1, x_2, \dots, x_n) = P_\theta(\xi_1 \leq x_1, \xi_2 \leq x_2, \dots, \xi_n \leq x_n)$ имеет следующий вид:

$$p_\theta(x_1, x_2, \dots, x_n) = f_\theta(x_1)f_\theta(x_2)\dots f_\theta(x_n)$$

Одной из ключевых статистик (для каждого $n \geq 1$) будет статистика

$$L_n = \frac{f_0(x_1)f_0(x_2)\dots f_0(x_n)}{f_\infty(x_1)f_\infty(x_2)\dots f_\infty(x_n)}$$

Исключительная роль этих статистик L_n , $n \geq 1$, называемых отношениями правдоподобия, проявляется в задаче различения двух гипотез H_0 и H_∞ (по N наблюдениям) о том, какую именно плотность, $f_0(x)$ или $f_\infty(x)$, имеют наблюдаемые случайные величины $\xi_1, \xi_2, \dots, \xi_N$. Решение этой задачи даётся так называемой леммой Неймана–Пирсона (см. [5], гл 3).

На основе данной статистики для обнаружения разладки строится тест Неймана–Пирсона, который предписывает принимать гипотезу H_0 , состоящую в том, что разладка имеет место, если $Z_m \geq h$ и H_1 , состоящую в том, что разладки нет, если $Z_m \leq h$. Где Z_m это процесс вида

$$Z_t = \log L_t(X)$$

А $L_t(X)$ в свою очередь является отношением правдоподобия

$$L_t(X) = \frac{p_0}{p_\infty}(Xt)$$

1.1.2 Тест Вальда

Другой тест, описанный в [6], строится следующим образом.

Пусть имеется числовая последовательность x_1, x_2, \dots с $x_i \in \mathbb{R}$

Будем считать, что заданы две вероятностные меры P_0 и P_∞ . Через $f_\theta(x)$ будем обозначать плотность распределения

$$P_\theta(\xi \in B) = \int_B f_\theta(x) \mu(dx), \theta = \{0, \infty\}$$

Предположим, что шаг за шагом мы получаем данные x_1, x_2, \dots , являющиеся наблюдениями над случайными величинами ξ_1, ξ_2, \dots . Мы хотим различить две гипотезы H_0 и H_∞ — о том, какое действует распределение, P_0 или P_∞ , — используя последовательные тесты, определяемые следующим образом.

Каждый последовательный тест δ определяется парой (τ, ϕ) , где

- 1) $\tau = \tau(x)$ — марковский момент (или момент остановки), принимающий значения $0, 1, \dots, \infty$
- 2) $\phi = \phi(x)$ — функция со значениями в $[0, 1]$, порождённая величинами x_1, x_2, \dots, x_τ .

Момент τ интерпретируется как момент прекращения наблюдений с последующим принятием решения $\phi = \phi(x)$, интерпретируемого как вероятность принятия гипотезы H_0 , когда наблюдениями являются x_1, x_2, \dots, x_τ .

Обозначим вероятность ошибки первого и второго рода как $\alpha(\phi)$ и $\beta(\phi)$ соответственно.

Тогда тест $\delta^* = (\tau^*, \phi^*)$ является оптимальным в классе $\Delta(\alpha, \beta)$, где

$$\Delta(\alpha, \beta) = \{\delta = (\tau, \phi) : E_\infty \tau < \infty, E_0 \tau < \infty \text{ и } \alpha(\phi) \leq \alpha, \beta(\phi) \leq \beta\}$$

если одновременно

$$E_{\infty}\tau^* \leq E_{\infty}\tau$$

$$E_0\tau^* \leq E_0\tau$$

для любого другого теста $\delta = (\tau, \phi) \in \Delta(\alpha, \beta)$

А. Вальд установил, что такой оптимальный тест действительно существует и описывается следующей леммой.

Лемма 1. Пусть $\delta = (\tau, \phi)$ — последовательный тест с

$$E_{\infty}\tau \geq \frac{\omega(\alpha, \beta)}{\rho_{\infty}}$$

и

$$E_0\tau \geq \frac{\omega(\beta, \alpha)}{\rho_{\infty}}$$

где

$$\omega(x, y) = x \log \frac{x}{1-y} + (1-x) \log \frac{1-x}{y}$$

А ρ_{∞} (ρ_0) — расстояния Кульбака–Лейблера между гипотезами H_{∞} и H_0 (H_0 и H_{∞}). Которые определяются по следующим формулам

$$\rho_{\infty} = \int_X \ln \frac{f_{\infty}}{f_0} [f_{\infty} - f_0] dx$$

$$\rho_0 = \int_X \ln \frac{f_0}{f_{\infty}} [f_0 - f_{\infty}] dx$$

Такой оптимальный тест описан в [1] и выглядит следующим образом.

Пусть α и β — ошибки первого и второго рода. А $X_t = x_1 + x_2 + \dots + x_t$.

Момент остановки определяется так:

$$\tau_{A,B} = \inf\{n \geq 1 : Z_n \geq B \text{ или } Z_n \leq A\}$$

где

$$A = \log \frac{\beta}{1-\alpha}$$

$$B = \log \frac{1-\beta}{\alpha}$$

$$A \leq 0 \leq B$$

и

$$Z_n = \log L_n(x) = \sum_{k=1}^n \log \frac{f_0(x_k)}{f_{\infty}(x_k)}$$

1.1.3 Сравнение методов Неймана–Пирсона и Вальда для Винеровского процесса

Подробный анализ сравнения рассмотренных методов был проведён Ширяевым А. в работе [1]. Приведём здесь некоторые выдержки из данного анализа.

Будем считать, что $B = (B_t)_{t \geq 0}$ — Винеровский процесс. Процесс B_t задан координатным образом:

$$B_t(x) = x_t.$$

Относительно наблюдаемого процесса $X = (X_t)_{t \geq 0}$ есть две гипотезы

$$H_0 : \quad X_t = \mu_0 t + B_t, \quad t \leq T$$

и

$$H_\infty : \quad X_t = \mu_\infty t + B_t, \quad t \leq T$$

где $\mu_0 \neq \mu_\infty$

Среднее время обнаружения разладки для метода Неймана–Пирсона определяется формулой

$$T_0(m, h; t) = \int R_0(m, h; y, t) p_\infty(0, 0; t, y) dy$$

Где $R_0(m, h; y, t)$ — среднее время обнаружения разладки, появившейся в момент $t \in [0, m]$. А $p_\infty(0, 0; t, y)$ — плотность распределения величины Z_t в предположении отсутствия разладки и при условии $Z_0 = 0$.

И в приложении к обнаружению разладки в Броуновском движении при $\alpha = \beta \rightarrow 0$ оно будет иметь такую асимптотику:

$$T = \frac{(C_{1-\alpha} - C_\beta)^2}{\left(\frac{\mu_0 - \mu_\infty}{\sigma}\right)^2}$$

где C_γ — γ -квантиль нормального распределения.

Для теста Вальда среднее время обнаружения разладки, как уже упоминалось выше определяется формулами

$$E_0 \tau = \frac{\omega(\beta, \alpha)}{\rho}$$

и

$$E_\infty \tau = \frac{\omega(\alpha, \beta)}{\rho}$$

И при условии, что $\alpha = \beta \rightarrow 0$ то для обнаружения разладки методом Вальда, нужно в 4 раза меньше наблюдений, чем методом Неймана–Пирсона:

$$\frac{E_{\infty}\tau}{T} = \frac{2\omega(\alpha, \beta)}{(C_{\alpha} + C_{\beta})^2} = \frac{1}{4}$$

Поэтому для решения поставленной задачи в данной работе будет использоваться последовательный тест Вальда.

Для определения степени загруженности перекрёстка используется модель компьютерной имитации транспортных потоков на двухполосных автомобильных дорогах.

1.2 Концептуальная модель компьютерной имитации транспортных потоков на двухполосных автомобильных дорогах

Для описания движения автомобилей по автомобильным дорогам, городским улицам и автомагистралям разработаны многочисленные математические модели [8], [9], [11], [12], [13], [14], [15], [16], [18], [19], [20] и др.

Разработка математической модели транспортного потока или математическое описание его характеристик является одним из важнейших и ответственных этапов при решении задач количественной и качественной оценки функционирования системы ВАДС, поскольку именно на этом этапе решается вопрос, насколько точно будут учтены те характеристики транспортного процесса и количественные связи между ними, которые необходимы для решения конкретных поставленных задач.

Строгого определения понятия сложной системы не существует. Основными отличительными признаками сложных систем являются:

- наличие большого количества взаимосвязанных и взаимодействующих элементов;
- сложность функций, выполняемой системой и направленной на достижение заданной цели функционирования;
- возможность разбиения системы на подсистемы, цели функционирования которых подчинены общей цели функционирования системы;
- наличие управления, разветвленной информационной сети и интенсивных потоков информации;

- наличие взаимодействия с внешней средой и функционирование в условиях воздействия случайных факторов [10].

Дорожное движение обладает всеми перечисленными свойствами. Естественно рассматривать его в виде сложной системы и обозначать „водитель — автомобиль — дорога — окружающая среда“ (ВАДС) [19].

Совершенствование электронно-вычислительной техники наряду с достижениями теории сложных систем привело к возникновению и успешному развитию нового направления в области исследования сложных систем — машинной имитации. По определению Т. Нейлора, машинная имитация — это численный метод проведения компьютерных экспериментов с математическими моделями, описывающими поведение сложной системы в течение продолжительных периодов времени [17]. Главным преимуществом метода машинной имитации является возможность проведения экспериментов не с реальной системой, а с ее математической моделью, реализованной на компьютере; при этом не существует ограничений по проведению экспериментов с целью изучения любых характеристик движения автомобиля [19].

Членом – корр. АН СССР Н.П. Бусленко и его школой предложена унифицированная абстрактная схема агрегата, которая позволяет единообразно описывать все элементы сложной системы (дискретные, непрерывные, детерминистические, стохастические) [10].

Общий принцип работы предлагаемой имитационной модели соответствует теории имитационного моделирования транспортных потоков, разработанной одним из авторов данной монографии [13]. Здесь для моделирования сложной системы ВАДС используется схема кусочно-непрерывного агрегата, с математической точки зрения представляющий собой условный марковский процесс с кусочно-непрерывными траекториями в пространстве переменной размерности. Конкретная реализация имитационной модели в виде микроописания отдельных блоков, а также полностью отлаженного программного комплекса, предназначенного для оценки степени опасности дорожного движения, пропускной способности, потерь времени, других характеристик транспортного потока на заданном участке дороги осуществлена другим автором [8] на базе языков программирования FORTRAN и Visual C++.

Агрегат характеризуется множествами моментов времени T , состояний в каждый момент времени Z , входных X и выходных Y сигналов. Состояние агрегата в момент $t \in T$ обозначается $z(t) \in Z$, входные и выходные сигналы соответственно — $x(t) \in X$, и $y(t) \in Y$.

Состояние агрегата в момент $(t + 0)$ обозначим $z(t + 0)$. Предполагается, что из состояния $z(t)$ в состояние $z(t + 0)$ агрегат приходит за малый интервал времени. Переход агрегата из состояния $z(t_1)$ в $z(t_2)$, $t_2 > t_1$, определяется динамическими свойствами самого агрегата и входными сигналами.

Во множестве состояний Z выделяется такое подмножество $Z^{(W)}$, что если в момент t' $z(t')$ достигает $Z^{(W)}$, то агрегат скачкообразно изменяет свое состояние. Пусть эти изменения описываются оператором W :

$$z(t' + 0) = W[t', z(t')] \quad (1)$$

В случае воздействия входного сигнала x_n поведение модели описывается оператором V . Тогда состояние $z(t_n + 0)$, где t_n — момент поступления в агрегат входного сигнала x_n , $t_n \in T$,

$$z(t_n + 0) = V[t_n, z(t_n), x(t_n)] \quad (2)$$

Если интервал (t_n, t_{n+1}) не содержит ни одного момента поступления сигналов, то для $t_n \in (t_n, t_{n+1}]$ состояние агрегата определяется оператором

$$z(t) = U[t, t_n, z(t_n + 0)] \quad (3)$$

Совокупность операторов W , V и U рассматривается как оператор переходов агрегата в новое состояние.

Во множестве состояний Z выделяется подмножество $Z^{(Y)}$ ($Z^{(W)}$ и $Z^{(Y)}$ могут пересекаться) такое, что если $z(t^*)$ достигает $Z^{(Y)}$, то выдается выходной сигнал, который определяется оператором выходов

$$y = G[t^*, z(t^*)] \quad (4)$$

Упорядоченная совокупность рассмотренных множеств T , X , Z , $Z^{(W)}$, $Z^{(Y)}$, Y и случайных операторов W , V , U , G полностью задает агрегат как динамическую систему. Определенного рода упорядоченная совокупность конечного числа агрегатов называется агрегативной системой [10].

Агрегативные системы в качестве математической модели транспортных потоков в России использовались, начиная с 1980-х годов, в исследованиях В.В. Сильянова, В.М. Еремина, В.Г. Крбашяна, Р.С. Картанбаева, М.С. Талаева, А.И. Должикова, О.И. Тонконоженкова, С.П. Крысина и других. В этих исследованиях рассмотрены различные проблемы проектирования автомобильных дорог и безопасности дорожного движения.

Объектом данного исследования является дорожное движение на различных фрагментах сети двухполосных автомобильных дорог. Вся сеть дорог представлена в виде графа, который состоит из узлов и ребер. Узлами являются перекрестки (примыкания) вместе с подходами к нему. Длина каждого из подходов определяется расстоянием влияния перекрестка и составляет 150 — 200 м. Ребрами являются двухполосные перегоны. Математической моделью каждого элемента графа, т.е. узла или ребра является кусочно-непрерывный агрегат. Очевидно, что если входные и выходные контакты соответствующих агрегатов соединить каналами связи, то можно получить агрегативную систему, которая соответствует любому заданному подмножеству дорожной сети. При этом количество моделируемых элементов дорожной сети ограничивается только техническими параметрами применяемой вычислительной техники (оперативная память, быстродействие и т.д.). Таким образом, моделируемая система ВАДС состоит из двух типов агрегатов: «Узел» и «Ребро».

Ниже, если особо не будет оговорено, будем рассматривать наиболее общий случай кусочно-непрерывного агрегата «Узел» — перекресток с четырьмя подходами (рис. 1.1).

Продольные оси всех подходов к перекрестку пересекаются в точке О. Углы между ними $\alpha_1, \alpha_2, \alpha_3, \alpha_4$, могут принимать любые значения, в соответствии с конфигурацией конкретного перекрестка. Обычно $\alpha_1 = \alpha_3$, и $\alpha_2 = \alpha_4$. Выберем неподвижную прямоугольную систему координат ХОУ с центром в точке О. Пусть координатная ось ОХ направлена вдоль продольной оси одного из подходов к перекрестку (1-й подход). Весь перекресток разбит на 9 подсистем:

- $L = 1, 3, 5, 7$ — полосы движения, подходящие к перекрестку;
- $L = 2, 4, 6, 8$ — полосы движения, отходящие от перекрестка;
- $L = 9$ — перекресток.

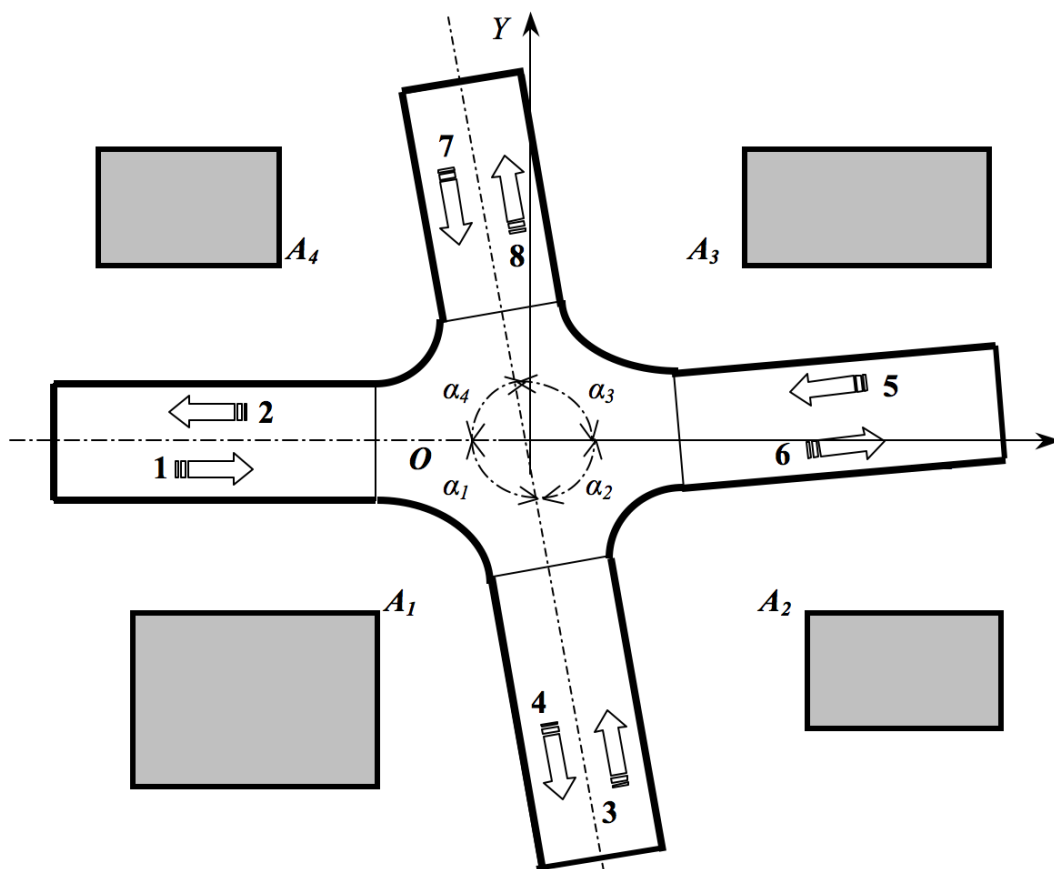


Рисунок 1.1 — Схема моделируемого нерегулируемого перекрестка

Каждая из полос движения $L = 1 \div 8$ в свою очередь может быть разбита на несколько участков, которые отличаются друг от друга геометрическими параметрами и/или средствами организации движения (элементарный участок). Перекресток ($L = 9$) состоит из 12 элементарных участков (рис. 1.2). Под элементарным участком на перекрестке понимается участок полосы движения, соединяющий конец каждого из четырех нечетных полос движения с началом каждой из четных полос движения (кроме сопряженной четной полосы движения).

Состояние агрегата Z описывается постоянными параметрами системы и переменными координатами. К постоянным относятся:

- 1) Параметры каждого элементарного участка дороги:
 - длина элементарного участка дороги по продольной оси;
 - расстояние видимости до встречного автомобиля в плане;
 - расстояние видимости до встречного автомобиля в продольном профиле;
 - коэффициент сцепления шины с поверхностью дороги;

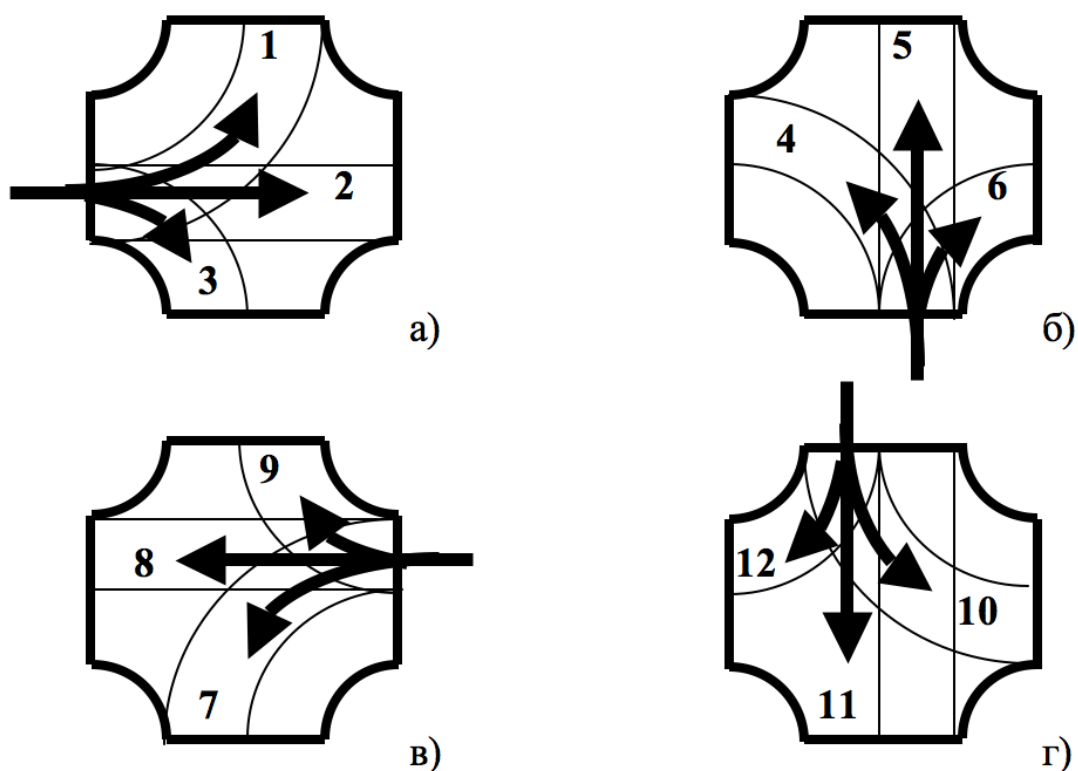


Рисунок 1.2 — Нумерация элементарных участков на перекрестке

- продольный уклон участка дороги;
- поперечный уклон участка дороги;
- ширина проезжей части;
- ширина обочины;
- радиус кривизны в плане;
- радиус вертикальной кривой в плане;
- координата начала продольной оси участка дороги по оси ОХ;
- координата начала продольной оси участка дороги по оси ОУ;
- направление полосы движения;
- параметры уравнения продольной оси участка дороги в аналитическом виде;
- ограничение скорости движения легковых автомобилей;
- ограничение скорости движения средних грузовых автомобилей;
- ограничение скорости движения тяжелых грузовых автомобилей;
- ограничение скорости движения автобусов;

- ровность дорожного покрытия;
- ровность дорожного покрытия;
- обгон запрещен;
- обгон грузовым автомобилям запрещен;
- коэффициент сопротивления качению при скорости движения автомобиля 20 км/ч;
- угловой коэффициент скорости движения для определения коэффициента сопротивления качению.

2) Объекты окружающей среды, ограничивающие видимость.

3) Транспортные потоки, задающиеся следующими параметрами:

- интенсивность в начале полосы движения;
- распределение интенсивности по направлениям движения (направо, прямо, налево);
- состав потока на каждой полосе движения. Транспортный поток в модели состоит из свыше 20 наиболее типичных отечественных и зарубежных марок автомобилей, которые сгруппированы по 4 типам: легковые, средние грузовые, тяжелые грузовые, автобусы.

4) Тягово-динамические и технико-эксплуатационные характеристики автомобилей всех марок:

- передаточное число k -й передачи коробки перемен передач (КПП) автомобиля;
- передаточные числа главной передачи;
- передаточное число делителя КПП;
- габаритная длина автомобиля;
- габаритная ширина автомобиля;
- габаритная высота автомобиля;
- база;
- колея;
- передний свес;

- расстояние от переднего бампера до глаз водителя по продольной оси;
- коэффициент обтекаемости;
- собственная масса;
- нагрузка на переднюю ось;
- нагрузка на заднюю ось;
- полная масса;
- нагрузка на переднюю ось при полной массе;
- нагрузка на заднюю ось при полной массе;
- параметры для определения коэффициента учета вращающихся масс;
- максимальный крутящий момент двигателя;
- угловая скорость коленчатого вала, соответствующая максимальному крутящему моменту;
- максимальная мощность двигателя;
- угловая скорость коленчатого вала, соответствующая максимальной мощности двигателя;
- время срабатывания тормозного привода;
- время запаздывания тормозного привода и времени нарастания замедления (при торможении);
- передаточное число рулевого механизма;
- коэффициент полезного действия трансмиссии;
- коэффициент коррекции двигателя;
- свободный радиус колеса автомобиля;
- динамический радиус колеса;
- радиус поворота по следу внешнего переднего колеса;
- радиус поворота по следу внешней, габаритной точки.

5) Закон распределения коэффициента использования грузоподъемности (загрузки) автомобилей.

К переменным координатам состояния агрегата относятся:

- 1) Количество автомобилей на полосе движения и на перекрестке в текущий момент времени.
- 2) Координаты всех моделируемых автомобилей, зависящие от следующих параметров:
 - пройденный путь;
 - координата автомобиля по продольной оси дороги;
 - расстояние автомобиля от продольной оси в поперечном направлении (принимает положительное значение, если автомобиль расположен левее от продольной оси и отрицательные значения, если правее);
 - скорость движения;
 - тангенциальное ускорение;
 - нормальное ускорение;
 - время, прошедшее с момента формирования автомобиля;
 - курсовой угол автомобиля в системе координат XOY [рис. 2.1];
 - угол наклона;
 - крен автомобиля;
 - угол поворота управляемых колес;
 - угловая скорость поворота рулевого колеса;
 - угловая скорость поворота управляемых колес;
 - координаты автомобиля, соответственно по оси OX и OY ;
 - скорость движения, соответственно по оси OX и OY ;
 - ускорение автомобиля, соответственно по оси OX и OY ;
 - угол поворота управляемых колес, при котором водитель прекращает поворот рулевого колеса на 1-ом и 7-ом этапе маневра обгона (или на 1-ом этапе перестроения);
 - то же самое, но на 4-ом и 10-ом этапе обгона [рис. 2.11];
 - угловая скорость поворота управляемых колес при обгоне;
 - длина прямолинейного участка на 3-м и 9-ом этапе обгона;
 - тип водителя по скорости свободного движения;
 - время реакции водителя на изменение режима движения лидера;

- время реакции водителя при появлении помехи на перекрестке;
 - время, оставшееся до истечения времени реакции водителя на изменение режима движения лидера;
 - то же самое, но после наступления другого события;
 - коэффициент использования грузоподъемности;
 - угловая скорость поворота управляемых колес на 3–м этапе поворота;
- Под координатами автомобиля понимаются координаты середины задней оси автомобиля.
- номер элементарного участка дороги или перекрестка, на котором находится автомобиль в данный момент времени;
 - марка автомобиля;
 - тип автомобиля;
 - состояния в процессе обгона (обгоняющий, обычный, обгоняемый);
 - количество обгоняемых автомобилей, находящихся впереди;
 - номер автомобиля в пачке;
 - тип передачи КПП;
 - направление дальнейшего движения после перекрестка (налево, прямо, направо);
 - состояние сигналов поворота (включен сигнал левого поворота, выключен, включен сигнал правого поворота);
 - состояние стоп-сигналов (включены, выключены);
 - следование за лидером [(см. п. 2.3, ситуации 1–4)];
 - номер ближайшего встречного автомобиля;
 - номер ближайшего встречного автомобиля, который находится в зоне видимости;
 - номер автомобиля – лидера, за которым следует данный автомобиль;
 - номер этапа в процессе поворота (всего 3 этапа);
 - номер этапа в процессе обгона (всего 11 этапов);
 - индекс приоритетов (помеха справа, помеха слева, помехи отсутствуют);

- тип водителя по реакции на ограничение скорости движения (нарушает правила дорожного движения (ПДД) или не нарушает ПДД);
- тип водителя по реакции на дорожный знак «Обгон запрещен» (нарушает требования дорожного знака или не нарушает);
- тип водителя по дистанции следования («осторожный», «обычный», «рискованный»);
- тип водителя по оценке обстановки около перекрестка (изменение скорости движения заранее до безопасной величины, изменение скорости движения только при появлении в зоне видимости помехи справа);
- тип водителя по включению сигналов поворота (включает или не включает);
- номер ситуации, после наступления которого, водитель принял решение изменить режим движения;
- положение, куда попадет автомобиль после перекрестка
- наличие или отсутствие встречного автомобиля на своей полосе движения.

3) Приоритет проезда через перекресток.

В зависимости от необходимого объема выходной информации задается время моделирования $t_{\text{мод}}$, которое в дальнейшем по мере работы модели уменьшается до нуля. Вместо истечения времени моделирования предусмотрены также другие правила остановки модели: прохождение через заданный участок дороги определенного количества автомобилей; накопление определенного количества статистической информации и т.д.

Более подробно данная модель описана в [7]

2 Математическая модель

2.1 Концептуальная постановка задачи

Каждый элемент улично–дорожной сети (УДС) характеризуется своей пропускной способностью. Пропускная способность отдельных участков УДС определяется пропускной способностью узкого места. Узкими местами являются пересечения и примыкания автомобильных дорог. Одной из основных задач интеллектуальных транспортных систем (ИТС) является предсказание появления заторов на участках УДС, которые образуются в случае превышения пропускной способности элементов УДС (в нашем случае — пересечение автомобильных дорог). Чем раньше и надёжнее мы сможем оценить интенсивность движения на подходе к узкому месту, тем эффективнее будут действия ИТС по предотвращению заторов. Таким образом основной задачей данной работы является определение достоверной интенсивности движения на всех возможных подходах к узкому месту с возможностью минимизации времени фиксирования разладки.

Поскольку формирование транспортного потока на подходах к узкому месту являются случайными процессами, то в качестве математического аппарата используются вероятностно статистические методы в теории принятия решений.

Поскольку число пересечений автомобильных дорог различных типов достаточно велико, то в качестве узкого места в данной работе было выбрано нерегулируемое, равнозначное пересечение двухполосных дорог в одном уровне. Такого типа пересечения являются наиболее сложными в описании их функционирования по пропуску транспортных потоков. Перейдём к формализации постановки задачи.

2.2 Формальная постановка задачи

Интенсивностью движения по одной полосе называется число автомобилей, проходящих в единицу времени через заданный створ полосы движения. Пропускной способностью полосы движения называется максимальное число автомобилей, проходящих в единицу времени через конечный створ полосы движения. Пропускная способность полосы движения зависит от многих факторов: геометрических

характеристик полосы, состояние её покрытия, видимость, состав транспортного потока и так далее.

Пропускную способность пересечения определить не так то просто. На рисунке 1.2 представлены возможные маршруты движения транспортных потоков на нерегулируемом равнозначном двухполосном пересечении двух дорог. Мы видим, что всего таких маршрутов 12. При этом каждый из них в принципе может быть независим от всех остальных, поэтому не совсем понятно, что в таком случае считать пропускной способностью пересечения. Пропускная способность определяется по работам [7], [11], [14]. Исходя из этих работ, пропускной способностью пересечения является многомерная функция, зависящая от 11 переменных. Данная функция строится по результатам компьютерных экспериментов с имитационными моделями транспортных потоков, движущихся по пересечению. Компьютерные эксперименты для определения указанной функции заключаются в следующем. Фиксируются значения интенсивности движения в одиннадцати маршрутах, а для двенадцатого маршрута проводятся серии экспериментов для увеличивающихся значений интенсивности движения по нему до тех пор, пока не будет достигнута пропускная способность данного маршрута. Для получения указанной функции требуется проведение сотен тысяч и более компьютерных экспериментов с соответствующем семейством имитационных моделей. Такие эксперименты были проведены для рассматриваемого в данной работе пересечения и их результаты используются в работе в качестве исходных данных.

Предполагается, что фрагмент рассматриваемой ИТС состоит из пунктов наблюдения за четырьмя интенсивностями движения на подходе к пересечению, которые могут находиться на значительном расстоянии от пересечения (2 — 4 километра и более). Задача ИТС состоит в определении точного значения интенсивности движения на подходах и проведении прогноза возможности образования заторов на пересечении.

Рассмотрим в отдельности один подход к пересечению. Пусть в заданном его створе происходит сбор информации о проходящих автомобилях.

Предполагается, что в пункте наблюдения (створ полосы движения) случайным образом появляются автомобили. Случайные интервалы между появлением двух соседних автомобилей могут быть распределены по разным законам (Пуассона, Эрланга, Гаусса). Пункт наблюдения должен проводить оценку ин-

тенсивности движения за конкретный промежуток времени (5 минут, 10, и т.д.) или после появления определённого числа автомобилей (50, 100 и т.д.). Значение интенсивности движения есть случайная величина. Задача ИТС заключается в определении времени наступления и величины разладки с заданной точностью. Эта задача решается в пункте 4 данной работы.

Собрав такого рода информацию на всех подходах к пересечению определяется вероятность появления заторов, т.е. превышения пропускной способности пересечения, о чём говорилось выше.

Желательно, чтобы система также давала предупреждения о значениях интенсивности близкой к пропускной способности.

2.3 Математическая постановка задачи

Пусть имеется пересечение двухполосных дорог. На каждом из направлений установлены датчики, которые снимают время Δt между двумя проезжающими подряд автомобилями. По прошествии некоторого времени можно установить значение мгновенной интенсивности движения x_n на данном направлении за прошедший отрезок времени. В данной работе мгновенная интенсивность вычисляется по прошествии 5 минут. Если отбросить первый интервал, и прибавить к рассматриваемому набору ещё несколько промежутков времени, чтобы сумма снова была более 5 минут, то можно вычислить следующую интенсивность x_{n+1} . Получающаяся таким образом последовательность представляет случайный процесс, который обозначим x . Тогда

$$x = \{x_1, x_2, \dots, x_n\}$$

Задачей математической модели является построение теста, позволяющего определить с заданной величиной ошибок первого и второго рода (α и β) момент разладки, когда сменились параметры процесса. В данной работе нас интересует смена математического ожидания, т.е. смена интенсивности движения. Математическое ожидание до появления разладки будем обозначать μ_0 , а после — μ_∞ .

Как уже было сказано выше в работе будет использоваться последовательный тест Вальда:

$$\delta = (\tau_{A,B}, d_{A,B})$$

где

$$\tau_{A,B} = \inf\{t \geq 0 : Z_t \notin (A,B)\}$$

$$d_{A,B} = \begin{cases} 1, & \text{если } Z_{\tau_{A,B}} \geq B \\ 0, & \text{если } Z_{\tau_{A,B}} \leq A \end{cases}, \quad A \leq 0 \leq B$$

$$A = \log \frac{\beta}{1-\alpha}, \quad B = \log \frac{1-\beta}{\alpha}$$

и процесс $(Z_t)_{t \geq 0}$ построен следующим образом

$$Z_t = \log L_n(x) = X_t \frac{\mu_0 - \mu_\infty}{\sigma^2} - \frac{1}{2\sigma^2}(\mu_0^2 - \mu_\infty^2)t.$$

Решение $d_{A,B} = 0$ сопровождается принятием гипотезы H_0 (разладка отсутствует); если $d_{A,B} = 1$, то принимается гипотеза H_∞ о наличии разладки.

После принятия решения наблюдения не прерываются, а продолжают. В случае, когда процесс Z_t достиг уровня A , то принимается решение, что разладки нет, происходит возврат процесса в нуль и наблюдение продолжается. Когда же значение процесса достигает B , то выносятся решение о разладке, т.е. об окончании наблюдения (рис. 2.1).

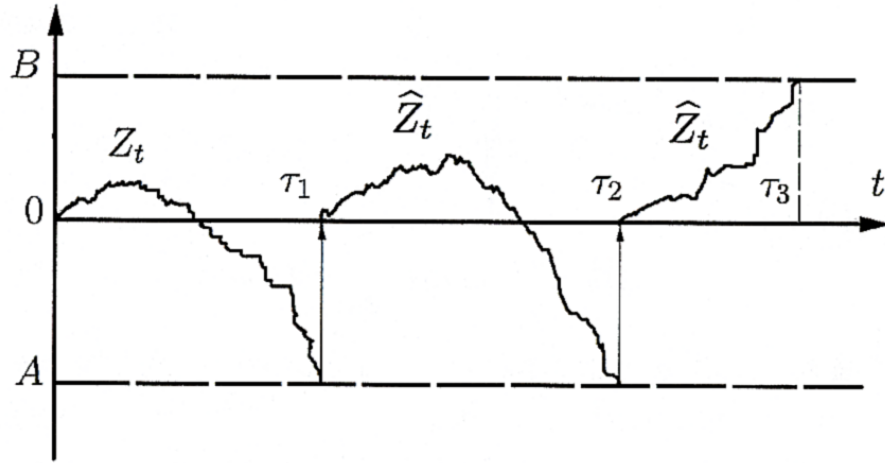


Рисунок 2.1 — Схема процедуры выявления разладки

Остаётся найти величину разладки. Т.е. разницу $\mu_0 - \mu_\infty$. После обнаружения наличия разладки, для определения её величины, мы будем постепенно с минимальным шагом увеличивать или уменьшать величину μ_∞ в зависимости

от того, в какую сторону изменилось значение интенсивности, и заново пересчитывать значение процесса Z_t , пока оно не станет удовлетворять величине, при которой тест не выдаёт решение о наличии разладки.

Т.к. интенсивность движения высчитывается за 5 минут, а во входных данных задаётся промежутком, равным одному часу, то имеет смысл взять за минимальный шаг изменения интенсивности равное $\frac{60}{5} = 12$.

Таким образом мы имеем всю необходимую теорию для программной реализации.

3 Программная реализация

Программный комплекс состоит из нескольких модулей(рис. 3.1):

- модуль, отвечающий за обнаружение разладки. На вход ему поступают данные с моделируемого датчика, на выходе у него величина разладки (WaldDetection);
- модуль, отвечающий за взаимодействия с системой ВАДС. Вход — текущая интенсивность по всем направлениям. Выход — пропускная способность перекрёстка(PrSp);
- интегральный модуль, который отвечает за графический интерфейс и объединение предыдущих двух частей(GUI).

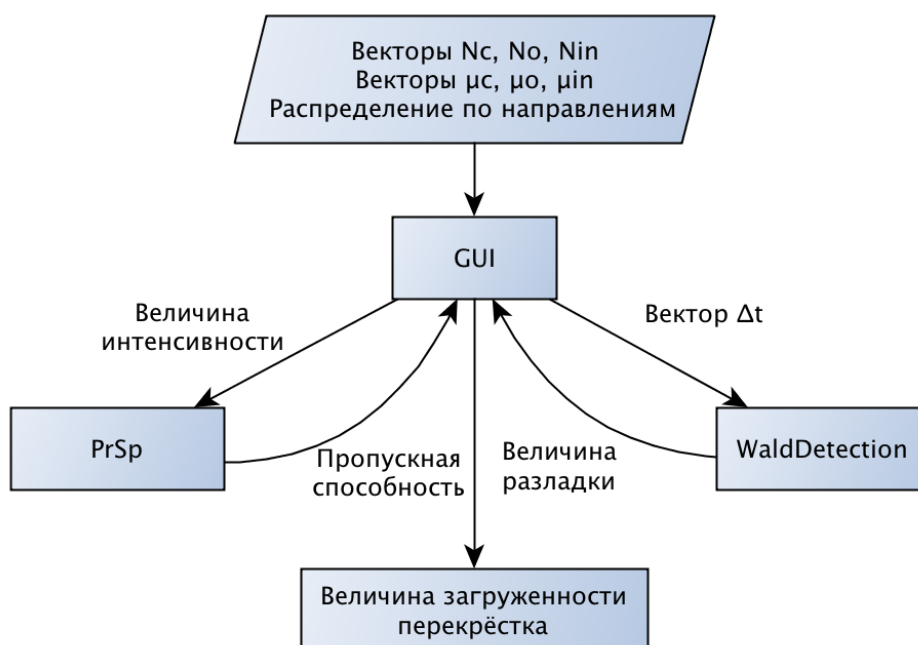


Рисунок 3.1 — Структура программного комплекса

Для программной реализации данного продукта в качестве языка программирования был выбран C++, главными плюсами которого являются скорость выполнения программы, широкое распространение и удобство использования. Для написания графической части программы используется кросс-платформенный инструментарий разработки ПО Qt. Главным преимуществом, определившим выбор именно этого средства для разработки комплекса, является кросс-платформенность создаваемого приложения, при достаточной функциональности.

Разработка ведётся в среде «Qt Creator», которая включает в себя редактор кода, справку, графические средства и возможность отладки приложений. «Qt Creator» может использовать GCC или Microsoft VC++ в качестве компилятора и GDB в качестве отладчика. Для Windows версий библиотека комплектуется компилятором, заголовочными и объектными файлами MinGW.

Рассмотрим по отдельности каждую из частей программы.

3.1 Модуль обнаружения разладки

Данный модуль представляет из себя класс, который устроен следующим образом. При инициализации необходимо передать в конструктор класса данные, которые в течение всего времени жизни экземпляра класса останутся неизменными. Такими параметрами для нашей задачи являются значения ошибок первого и второго рода и величина среднеквадратического отклонения.

Основная математическая часть программы сосредоточена в процедуре `add_point`. Здесь на основе поступающих значений Δt высчитывается мгновенная интенсивность движения и, используя тест Вальда, определяется момент появления разладки.

3.2 Модуль взаимодействия с системой ВАДС

Задачей класса `PrSp` является на основе полученной из программного комплекса ВАДС базы данных установить пропускную способность перекрёстка. Эти данные можно представить, как восьмимерную функцию $f(x_1, x_2, \dots, x_8)$, заданную дискретно. Параметрами функции выступают: интенсивность движения на встречном направлении, интенсивность движения на перекрёстных направлениях и шесть параметров, отвечающих за распределение потока автомобилей по направлениям поворота. Значением этой функции является пропускная способность перекрёстка на данном направлении.

Ввиду того, что в рамках одного эксперимента считается, что распределение потока автомобилей по направлениям поворота не меняется, то можно рассматривать дискретно заданную функцию f как функцию двух переменных. Тогда для того, чтобы получить значение функции для любых значений параметров

(а не только тех, которые есть в базе данных ВАДС), можно применить билинейную интерполяцию.

В модуле **PrSp** данные из системы ВАДС для ускорения поиска хранятся в виде хэш-таблицы, где ключём является закодированное значение всех возможных вариантов входных данных. Кодирование данных довольно просто. Каждое уникальное значение параметра записывается как цифра в n -мерной системе исчисления, где n — максимальное количество возможных вариантов значения любого из параметров.

3.3 Интегральный модуль

Интегральный модуль объединяет и согласует между собой работу двух предыдущих модулей (см. рис. 3.1). Здесь на основе введенных данных на каждом направлении случайным образом генерируются значения Δt и передаются на вход модулю **WaldDetection**. В случае отсутствия разладки, **WaldDetection** возвращает нуль, а в случае обнаружения разладки — её величину.

Изменение интенсивности движения на одном из направлений, говорит о том, что изменилась пропускная способность перекрёстка и возникает необходимость её вычислить. Для этого в модуль **PrSp** передаются величины интенсивностей по всем направлениям. Полученное значение пропускной способности сравнивается с интенсивностью движения на входном направлении перекрёстка и вычисляется величина загруженности перекрёстка. Визуально степень загруженности отображается цветом квадрата, расположенного на графической части программы.

Одной из задач разработки программного продукта является реализация простого и удобного пользовательского интерфейса (рис. 3.2), который позволяет работникам, не обладающим высокой квалификацией использовать данную систему. Исходя из этих соображений, графический интерфейс разделён на три основных блока: блок входной информации, графическое отображение происходящего процесса и таблица с выходными данными, по которой человек может понять, какова была история развития процесса, а так же увидеть точные цифры времени изменения интенсивности движения и новые значения пропускной способности.

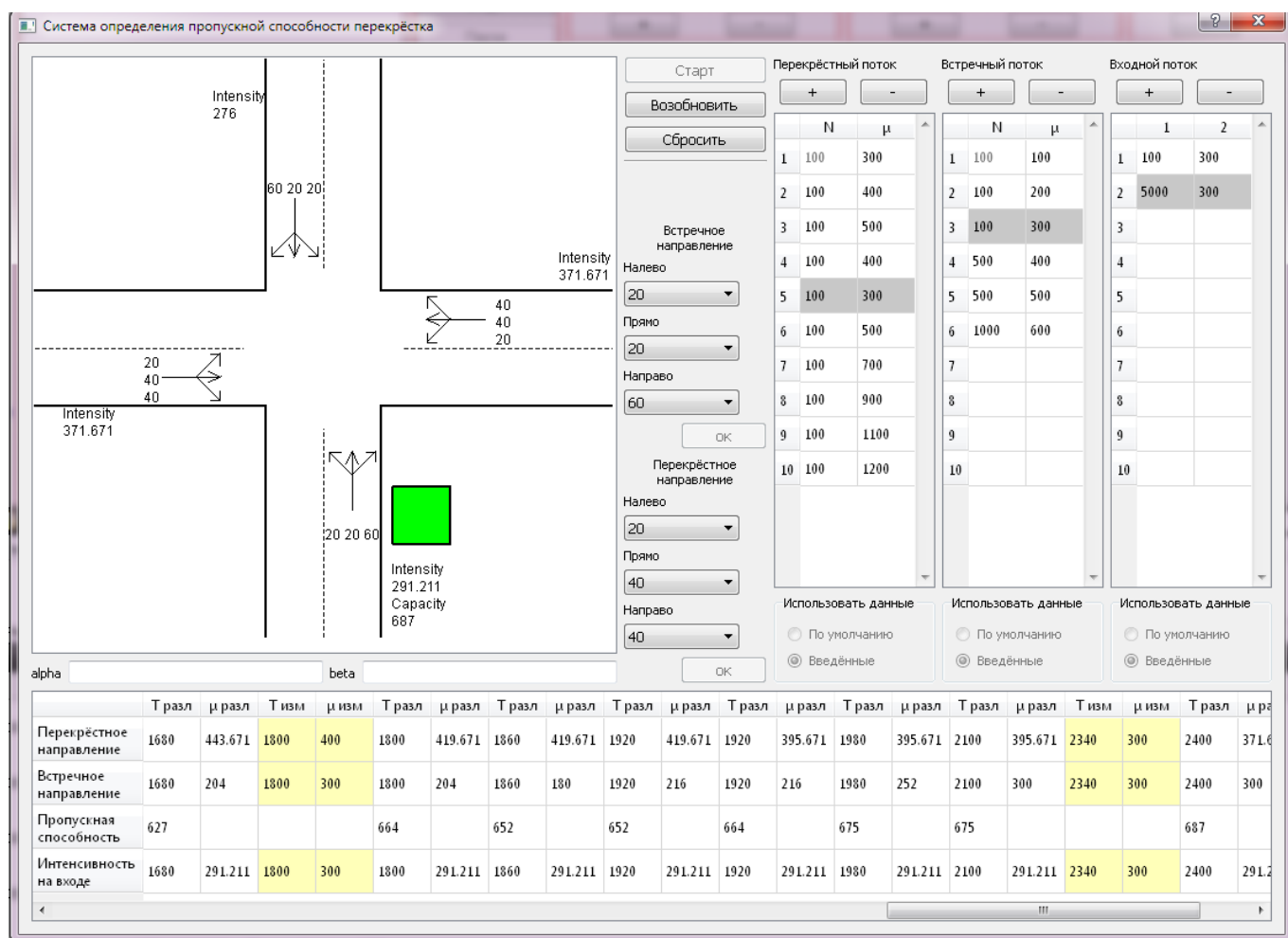


Рисунок 3.2 — Внешний вид системы определения пропускной способности перекрёстка

3.4 Статистические испытания

Самым важным модулем в данном программном продукте является модуль обнаружения разладки. Чтобы проверить правильность его работы, проведём достаточное число испытаний, позволяющих статистически установить точность обнаружения разладки. Каждая процедура обнаружения разладки выглядит как показано на рисунках 3.3 и 3.4. Здесь момент смены подаваемой на вход интенсивности с 250 авт./час до 350 авт./час происходит на 298-й минуте(рис. 3.3), а обнаружение разладки происходит когда процесс Z_t пересекает границу A на 328-й минуте(рис. 3.4).

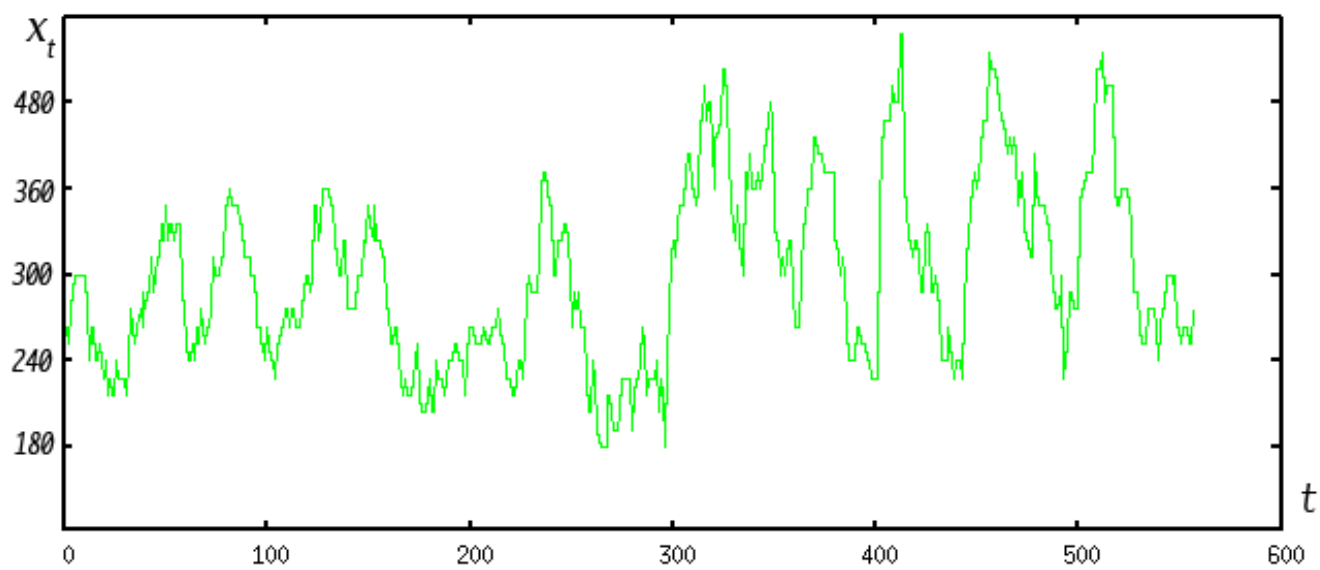


Рисунок 3.3 — Изменение мгновенных интенсивностей со временем

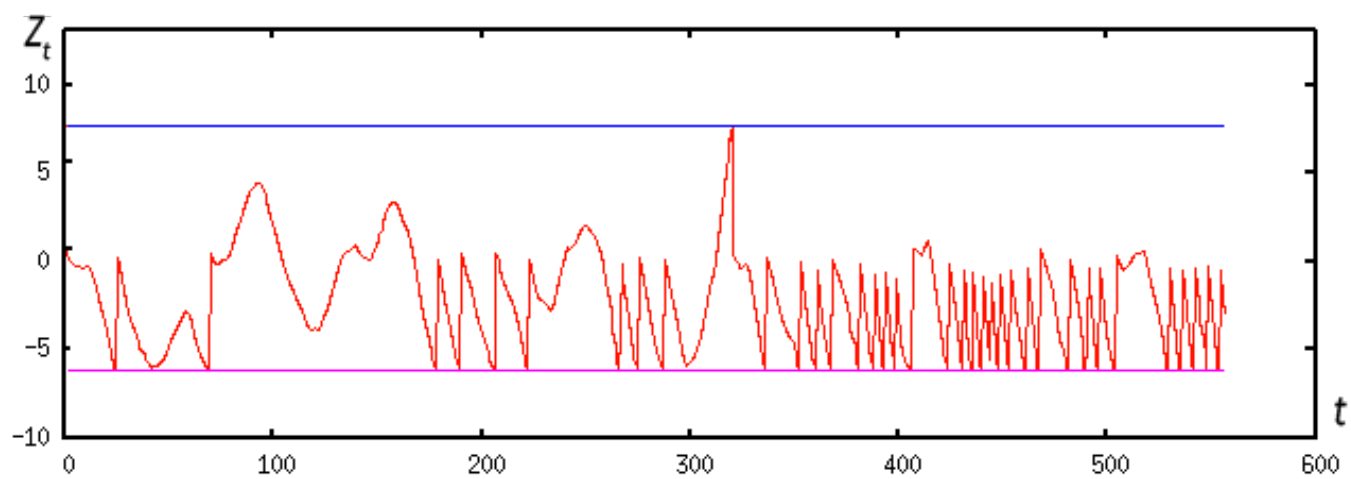


Рисунок 3.4 — Процесс Z_t обнаружения разладки

Проведём десять тысяч подобных испытаний, чтобы установить процент верного определения разладки. Результаты испытаний видны в таблице 3.1, состоящей из пяти столбцов:

- 1) величина изменения интенсивности в отношении к среднеквадратическому отношению — $\frac{\mu_{\infty}-\mu_0}{\sigma}$;
- 2) количество правильных обнаружений разладки — R ;
- 3) количество преждевременных срабатываний — W ;
- 4) количество не обнаруженных разладок за промежуток, в течение которого прошло 100 автомобилей — N ;
- 5) среднее время обнаружения разладки — $\tau - t$.

$\frac{\mu_{\infty}-\mu_0}{\sigma}$	R	W	N	$E(\tau - t)$
0.1	469	74	457	69
0.2	620	76	304	70
0.3	835	51	114	68
0.4	926	39	35	67
0.5	979	14	7	73
0.6	992	3	5	74
0.7	998	0	2	70
0.8	1000	0	0	71
0.9	1000	0	0	70
1.0	1000	0	0	67

Таблица 3.1 — Результаты испытаний

В ходе экспериментов был доказан важный для развития современных интеллектуальных транспортных систем факт, состоящий в том, что пропускная способность перекрёстка зависит далеко не только от величин интенсивности на подходе к нему, но и от распределения направлений поворота машин на перекрёстке. Для подтверждения этого, была проведена серия компьютерных экспериментов на моделях системы ВАДС, результаты которых приведены на рис. 3.5. Здесь ярко-красным цветом отмечено изменение пропускной способности в зависимости от интенсивности движения для распределения «(20, 60, 20), (20, 60, 20)» (Т.е. по

встречному и перекрёстным направлениям, преобладает направление движения прямо), а зелёным — для распределения «(20, 20, 60), (20, 20, 60)». При этом скорость падения пропускной способности перекрёстка увеличивается до полутора раз, в зависимости от разного распределения направлений движения машин.

Проведённые испытания с программным комплексом для данных с рисунка 3.5 показали невозможность определения заторов с помощью традиционного оборудования.

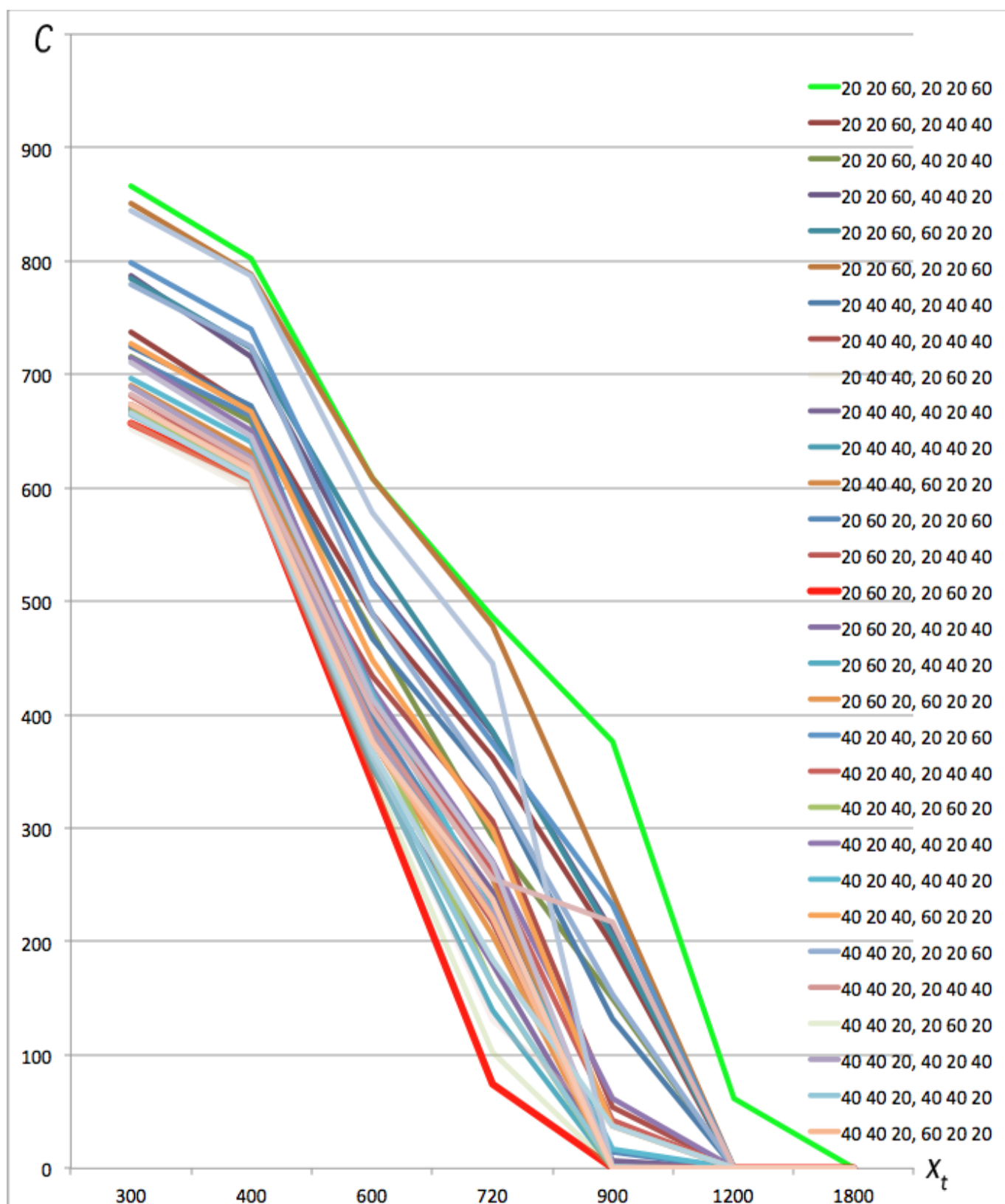


Рисунок 3.5 — Зависимость падения пропускной способности C от распределения направления поворота

Заключение

Рассматриваемая тема является актуальной, поскольку развитие и внедрение интеллектуальных транспортных систем признано во всём мире и в нашей стране главным направлением совершенствования автомобильно–дорожных систем, особенно в современных условиях быстрого роста интенсивности движения. Основными результатами работы являются следующие:

- 1) Для обнаружения разладки в процессе наблюдения за потоком машин реализован последовательный тест Вальда, который показал высокую эффективность.
- 2) В среде программирования Qt на языке C++ реализован программный комплекс, обладающий удобным и интуитивно понятным интерфейсом, который позволяет работникам не обладающим высокой квалификацией использовать данную систему. Основной задачей, которую решает комплекс, является оповещение пользователя о возможном появлении затора на нерегулируемом перекрёстке.
- 3) Полученные результаты показывают широкие возможности применения подхода для решения автотранспортных задач. Показано, что используемое традиционное оборудование ИТС принципиально не решает ряд практически важных задач.

Список использованных источников

- 1 Ширяев А.Н. Вероятностно–статистические методы в теории принятия решений М.: МЦНМО, 2011.
- 2 DIRECTIVE 2010/40/EU OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on the framework for the deployment of Intelligent Transport Systems in the field of road transport and for interfaces with other modes of transport. 7 July 2010.
- 3 Блекуэлл Д., Гиршик М.А. Теория игр и статистических решений М.: ИЛ, 1958. Пер. с англ.: D. Blackwell, M.A. Girshick. Theory of games and statistical decisions, 1954.
- 4 Карлин С. Математические методы в теории игра, программировании и экономике. М.: Мир, 1964. Пер. с англ.: S. Karlin Mathematical methods and theory in games, programming, and economics. London–Paris:Pergamon Press, 1959.
- 5 Леман Э. Проверка статистических гипотез. М.: Наука, 1964.
- 6 Айвазян С.А., Енюков И.С., Мешалкин Л.Д. Прикладная статистика: Основы моделирования и первичная обработка данных. М.: Финансы и статистика, 1983.
- 7 Бадалян А.М., Ерёмин В.М. Компьютерное моделирование конфликтных ситуаций для оценки уровня безопасности движения на двухполосных автомобильных дорогах. Научная монография. М.: ИКФ «Каталог». 2007. 240с.
- 8 Бадалян А.М. Имитационная модель транспортных потоков на перегонах двухполосных городских улиц. Рукопись депонирована в справочно–информационном фонде ЦБНТИ концерна "Росавтодор" N 228 ад. Реферат рукописи опубликован в библиографическом указателе "Депонированные научные работы" N 2, 1992.
- 9 Бадалян А.М. и др. Моделирование энергоэкологических характеристик транспортных потоков в городах по материалам дистанционного мониторинга. Известия РАН. ТиСУ. 2002. N4. С. 160–176.
- 10 Бусленко Н.П. Моделирование сложных систем. М.: Наука, 1978 г. 339с.
- 11 Гаврилов А.А. Моделирование дорожного движения. М.: Транспорт, 1980. 189с.

- 12 Дрю Д.Т. Теория транспортных потоков и управление ими. М.: Транспорт, 1972. 423с.
- 13 Еремин В.М. Теория имитационного моделирования транспортных потоков в проектировании дорог и организации движения. Повышение транспортных качеств автомобильных дорог и безопасности движения. Сб. науч. трудов. МАДИ . М.: 1986. с. 3–14.
- 14 Еремин В.М., Крбашян В.Г. Имитационное моделирование движения потоков автомобилей в зоне примыкания горных дорог для оценки безопасности движения. Сб. науч. трудов МАДИ. М.: 1984. с. 102–108.
- 15 Иносэ Х., Хамада Т.М. Управление дорожным движением. Пер. с англ. М.: Машиностроение, 1983. 248с.
- 16 Клинковштейн Г.И., Афанасьев М.Б. Организация дорожного движения: Учеб. для вузов. 4–е изд., перераб. и доп. М.: Транспорт, 1997. 231с.
- 17 Нейлор Т. Машинные имитационные эксперименты с моделями экономических систем. М.: Мир, 1975. 500с.
- 18 Сильянов В.В. Теория транспортных потоков в проектировании дорог и организации движения. М.: Транспорт, 1977. 303с.
- 19 Сильянов В.В., Еремин В.М., Муравьева Л.И. Имитационное моделирование транспортных потоков в проектировании дорог. М.: МАДИ. 1981. 119с.
- 20 Хейт Ф. Математическая теория транспортных потоков. М.: Мир, 1966. 187с.

Приложение А

Исходный код модуля обнаружения разладки

```
1 #include <vector>
2 #include <fstream>
3 #include <iostream>
4 #include <string.h>
5 #include <math.h>
6 #include "wald_detection.h"
7
8 double WaldDetection::get_mu(){
9     return mu_0;
10 }
11
12 WaldDetection::WaldDetection(double alpha, double beta, double mu_0, double * standard_dts, int size, double s){
13     int i;
14     _T = 300;
15     _A = log(beta/(1-alpha));
16     _B = log((1-beta)/alpha);
17     mytime = 0;
18     car_count = 0;
19     index = 0;
20     j = 0;
21     count1 = 0;
22     sum1 = 0;
23     count2 = 0;
24     sum2 = 0;
25     sigma = s;
26     for (i = 0; i < size; i++){
27         dts[index] = standard_dts[i];
28         index = (index + 1) % 300;
```

```

29         mytime += standard_dts[i];
30         car_count += 1;
31         if (mytime > _T){
32             sum1 += car_count - 1;
33             count1 += 1;
34             sum2 += car_count - 1;
35             count2 += 1;
36             while (mytime > _T){
37                 mytime -= dts[j];
38                 j = (j + 1) % 300;
39                 car_count--;
40             }
41         }
42     }
43     mu_0 = sum1 / count1;
44 }
45
46 double WaldDetection::add_point(double point){
47     double mu_inf_greater = mu_0 + 1;
48     double mu_inf_smaller = mu_0 - 1;
49     int fl = 0;
50     int cc;
51     int _T = 300;
52     double _Zt, _Zt_s;
53     mytime += point;
54     car_count += 1;
55     dts[index] = point;
56     index = (index + 1) % 300;
57     if (mytime > _T){
58         sum1 += car_count - 1;
59         count1 += 1;
60         sum2 += car_count - 1;
61         count2 += 1;

```

```

62      _Zt = sum1 * ((mu_0 - mu_inf_greater)/(sigma
*sigma)) - (1/(2*sigma*sigma))*(mu_0*mu_0 - mu_inf_greater*m
u_inf_greater)*count1;
63      _Zt_s = sum2 * ((mu_0 - mu_inf_smaller)/(sig
ma*sigma)) - (1/(2*sigma*sigma))*(mu_0*mu_0 - mu_inf_smaller
*mu_inf_smaller)*count2;
64      if (_Zt_s >= _B){
65          sum2 = 0;
66          count2 = 0;
67      }else{
68          while (_Zt_s <= _A && mu_inf_smaller > 1
){
69              fl = 1;
70              mu_inf_smaller -= 1;
71              _Zt_s = sum2 * ((mu_0 - mu_inf_small
er)/(sigma*sigma)) - (1/(2*sigma*sigma))*(mu_0*mu_0 - mu_inf
_smaller*mu_inf_smaller)*count2;
72          }
73          if (fl == 1) {
74              sum2 = 0;
75              count2 = 0;
76              mu_0 = mu_inf_smaller;
77              cc = car_count;
78              while (mytime > _T){
79                  mytime -= dts[j];
80                  j = (j + 1) % 300;
81                  car_count--;
82              }
83              return cc - 1;
84          }
85      }
86      if (_Zt >= _B){
87          sum1 = 0;

```

```

88             count1 = 0;
89         }else{
90             while (_Zt <= _A && mu_inf_greater <= 36
00){
91                 fl = 1;
92                 mu_inf_greater += 1;
93                 _Zt = sum1 * ((mu_0 - mu_inf_greater
)/(sigma*sigma)) - (1/(2*sigma*sigma))*(mu_0*mu_0 - mu_inf_g
reater*mu_inf_greater)*count1;
94             }
95
96             if (fl == 1) {
97                 sum1 = 0;
98                 count1 = 0;
99                 mu_0 = mu_inf_greater;
100                cc = car_count;
101                while (mytime > _T){
102                    mytime -= dts[j];
103                    j = (j + 1) % 300;
104                    car_count--;
105                }
106                return cc - 1;
107            }
108        }
109        while (mytime > _T){
110            mytime -= dts[j];
111            j = (j + 1) % 300;
112            car_count--;
113        }
114    }
115    return 0;
116 }

```