



РАЗРАБОТКА ИМИТАЦИОННОЙ МОДЕЛИ РАБОТЫ ПЕРЕКРЁСТКА С ВНЕДРЕНИЕМ АДАПТИВНОГО УПРАВЛЕНИЯ СВЕТОФОРАМИ В СРЕДЕ ANYLOGIC

Д. Ю. Марчук

Санкт-Петербургский государственный университет аэрокосмического приборостроения

В данной работе рассматривается разработка имитационной модели функционирования городского перекрёстка в Приморском районе г. Санкт-Петербурга. Был разработан алгоритм работы адаптивного светофорного регулирования и внедрен в среде AnyLogic. Использование адаптивных алгоритмов светофорного регулирования способствует повышению пропускной способности перекрёстка, снижению заторов и улучшению общей транспортной ситуации. Модель позволяет исследовать и оптимизировать процессы управления дорожным движением с учётом динамически изменяющихся транспортных потоков.

Ключевые слова: имитационная модель, перекрёсток, адаптивное светофорное регулирование, AnyLogic, транспортное моделирование, оптимизация трафика, симуляция дорожного движения, автоматизация управления движением.

Для цитирования:

Марчук, Д. Ю. Разработка имитационной модели работы перекрёстка с внедрением адаптивного управления светофорами в среде AnyLogic / Д. Ю. Марчук // Системный анализ и логистика. – 2025. – № 3(46). – с. 8-20. DOI: 10.31799/2077-5687-2025-3-8-20.

THE DESIGN OF AN INTERSECTION SIMULATION MODEL WITH THE IMPLEMENTATION OF ADAPTIVE TRAFFIC LIGHT MANAGEMENT IN THE ANYLOGIC ENVIRONMENT

D. Y. Marchuk

St. Petersburg State University of Aerospace Instrumentation

This article discusses the development of a simulation model of the functioning of an urban intersection in the Primorsky district of St. Petersburg. An algorithm for adaptive traffic light control has been developed and implemented in the AnyLogic environment. The use of adaptive traffic light control algorithms helps to increase the capacity of the intersection, reduce congestion and improve the overall traffic situation. The model allows to investigate and optimize traffic management processes taking into account dynamically changing traffic flows.

Keywords: simulation model, intersection, adaptive traffic light regulation, AnyLogic, transport modeling, traffic optimization, traffic simulation, traffic control automation.

For citation:

Marchuk, D. Y. The design of an intersection simulation model with the implementation of adaptive traffic light management in the AnyLogic environment / D. Y. Marchuk // System analysis and logistics. – 2025. – № 3(46). – с. 8-20. DOI: 10.31799/2077-5687-2025-3-8-20.

Введение

Рост городского населения и увеличение количества транспортных средств приводят к значительному усложнению управления дорожным движением и возникновению пробок на дорогах. Традиционные методы светофорного регулирования с фиксированными циклами часто оказываются неэффективными в условиях динамично меняющихся в течение дня транспортных потоков, что снижает пропускную способность перекрёстков и увеличивает время ожидания участников движения.

Адаптивные системы управления светофорами, способные в реальном времени подстраиваться под текущую ситуацию на дорогах, демонстрируют значительный потенциал для повышения эффективности регулирования движения.

В данной работе рассматривается создание имитационной модели функционирования перекрёстка с адаптивным светофорным регулированием в среде AnyLogic. Цель исследования – разработка модели, учитывающей динамику транспортных потоков и



обеспечивающей оптимальное управление движением для повышения пропускной способности и снижения заторов. Результаты моделирования могут быть использованы для поддержки принятия решений в области городского транспортного планирования и развития интеллектуальных транспортных систем.

1 Сбор исходных данных перекрестка

В качестве объекта для создания микромоделей в AnyLogic был выбран перекрёсток улиц Планерной и Оптиков. Для моделирования данного перекрёстка использовались следующие исходные данные:

- геометрические характеристики перекрёстка;
- приоритетные направления движения на рассматриваемых пересечениях;
- режимы функционирования светофорных устройств и их конфигурация;
- интенсивности входящих транспортных потоков и распределение потоков внутри узла.

Конфигурация перекрестка, включая расположение стоп-линий, пешеходных переходов, ширину полос движения и разделительной полосы, а также разрешённые направления движения, была определена на основе спутникового изображения перекрестка из онлайн-сервиса Яндекс.Карты (см. рисунок 1).



Рис. 1. Снимок моделируемого перекрестка

Для получения остальных исходных данных по данному перекрестку было проведено натурное исследование с применением рекомендаций, изложенных в источниках [1, 2, 3]. В рамках этого исследования осуществлялось наблюдение за транспортными потоками, а также сбор и анализ информации, что позволило получить объективные данные о работе



перекрестка.

Исследование проводилось в рабочий день с 18:00 до 19:00, когда на перекрестке наблюдаются пиковые нагрузки. В этот период фиксируется максимальная интенсивность движения, что создаёт условия для изучения поведения водителей и пешеходов, а также выявления возможных проблем в организации дорожного движения.

Первым этапом натурного исследования стал замер продолжительности каждой фазы переключения светофора. Светофоры с разных сторон перекрестка имеют следующую конфигурацию:

- на северной и восточной сторонах установлены стандартные трёхсекционные светофоры;
- на южной и западной сторонах – трёхсекционные светофоры с дополнительной секцией для поворота направо.

Регулирование движения на перекрестке осуществляется в четыре фазы, которые представлены на рисунке 2.

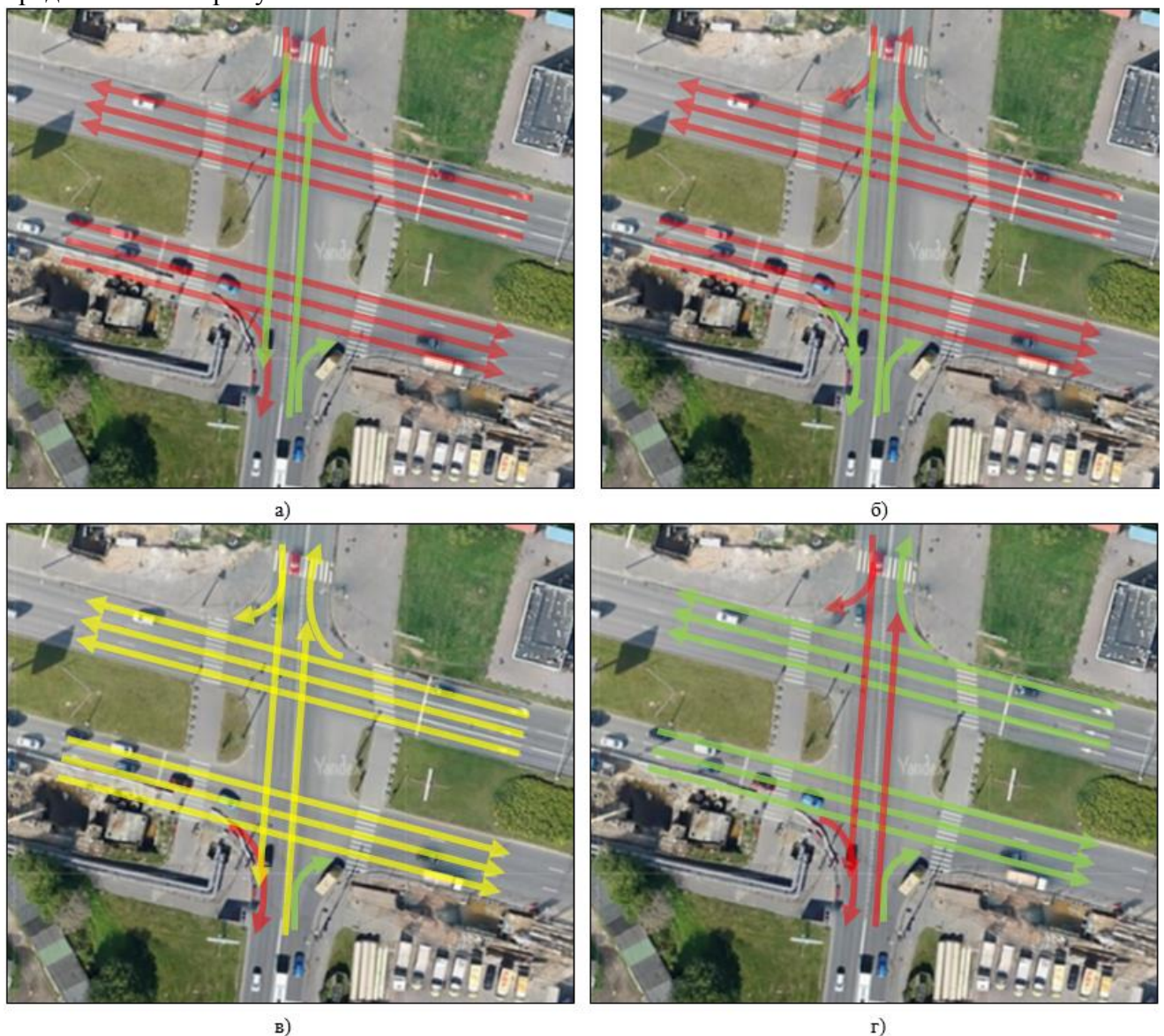


Рис. 2. Световые сигналы светофора для:

- а) – первой фазы,
- б) – второй фазы,
- в) – третьей фазы,
- г) – четвертой фазы



В таблице 1 представлены полученные данные о режиме работы светофора.

Таблица 1 – Режим регулирования светофорных сигналов

	Длительность фазы, сек			
	1 фаза	2 фаза	3 фаза	4 фаза
Участок регулирования	35	15	7	31
Южная сторона				
Южная сторона, доп. секция				
Северная сторона				
Западная сторона				
Западная сторона, доп. секция				
Восточная сторона				

Также были получены данные об интенсивностях движения с каждого направления, а также процент ТС, поворачивающих направо (таблица 2).

Подсчет автомобилей проводился в течение 10 минут для каждого направления.

Таблица 2 – Интенсивность движения в каждом направлении

Участок подсчета	Замер интенсивности в течении 10 минут, кол-во ТС	Интенсивность движения, авт./час.	ТС, поворачивающие направо, %
Южная сторона	311	2799	0,56
Северная сторона	225	2025	0,27
Западная сторона	170	1530	0,55
Восточная сторона	205	1845	0,44

Полученные исходные данные являются ключевыми для разработки модели исследуемого перекрестка, которая будет использоваться для анализа и оптимизации дорожного движения на перекрестке.

2 Разработка микромоделей в Anylogic

Для разработки модели перекрестка на макроуровне был выбран программный пакет Anylogic [4], так как он обладает рядом преимуществ для решения поставленных задач [5,6].

Разработка модели начинается с построения дорожной сети перекрестка. Для этого используются элементы из «Библиотеки дорожного движения»:

- элемент «Дорога» служит для визуального представления дорожной инфраструктуры в модели;
- элемент «Перекресток» предназначен для моделирования точек пересечения дорожных потоков, где транспортные средства могут менять направление движения;
- элемент «Стоп-линия» визуально обозначает место остановки транспортных средств на дороге. Он может быть интегрирован с блоком «Traffic Light» (описан ниже) для управления движением в этой конкретной зоне.

Построение дорожной сети начинается с построения дорог. Здесь задаются количество полос (основного и встречного движения), ширина разделительной полос и присваивается имя для каждой дороги (рисунок 3):

- «roadN»: северный участок улицы Планерной;
- «roadS»: южный участок улицы Планерной;
- «roadW»: западный участок улицы Оптиков;
- «roadE»: восточный участок улицы Оптиков.

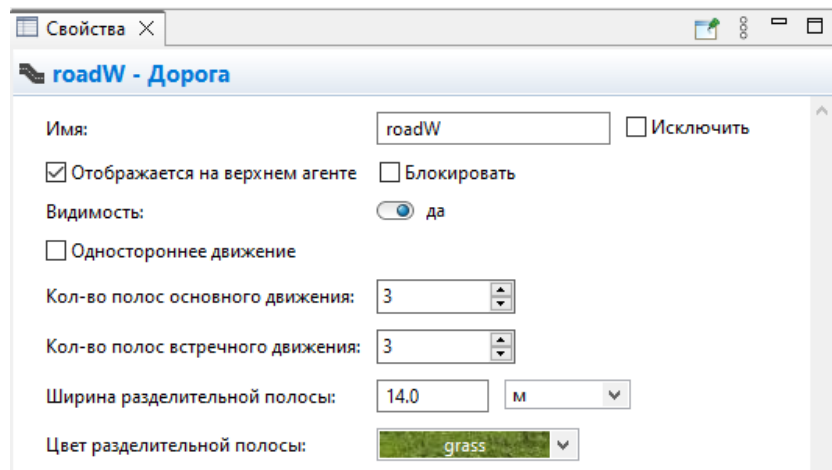


Рис. 3. Контекстное меню свойств элемента «Дорога»

Затем эти дороги соединяются в пересечение с помощью блока «Перекресток». Здесь задаются разрешенные направления движения с помощью «Соединителей полос». Далее размечаются стоп-линии.

В результате получаем каркас перекрестка, по которому будет осуществляться транспортное движение и его дальнейший анализ (рисунок 4).



Рис. 4. Созданный в Anylogic перекресток ул. Планерной и ул. Оптиков

Следующим шагом создадим агентов сети. Агенты дорожного движения представляют собой объекты, которые моделируют поведение различных участников дорожного движения, таких как автомобили, автобусы, грузовики и пешеходы. Здесь же добавим параметр `time()` для расчета среднего времени нахождения в сети каждого агента с помощью команды:

`main_2.времяПроезда.add(time()-time).`

С помощью следующих блоков создадим логику движения агентов по сети:

- «Car Source» создает агентов и является источником движения; здесь задаются интенсивность движения, а также дорога, на которой появляется агент;



- «Car Move To» определяет в какую точку движется автомобиль (остановка, стоп-линия, дорога);
- «Car Dispose» удаляет агентов из сети; здесь на входе в этот блок задана команда для расчета среднего времени в сети;
- «Traffic Light» является блоком светофорного регулирования перекрестка, здесь задаются длительности фаз, а также действия при смене фаз;
- Road Network Descriptor, является блоком управления автомобилями совершающие какое-либо действие: вход/выход из сети, въезд/съезд с дороги, въезд/съезд с перекрестка, смена полосы, а также остановка и возобновление движения.

На рисунке 5 представлен общая логика движения агентов.

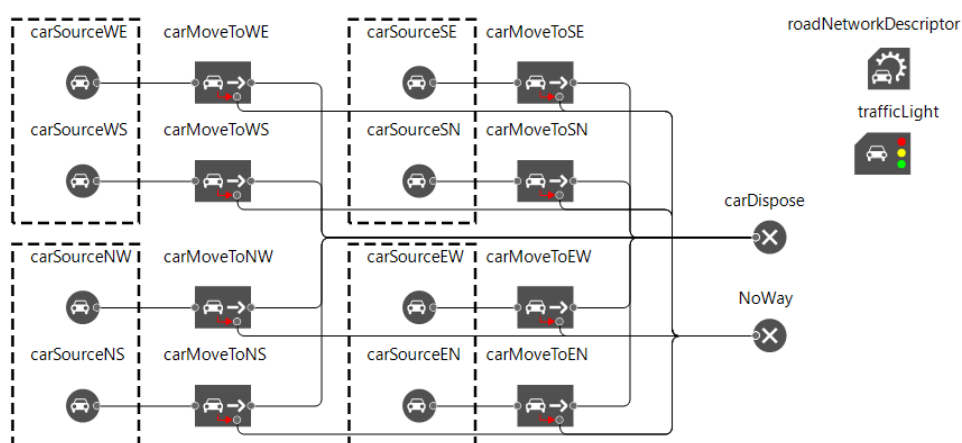


Рис. 5. Общий вид схемы движения агентов

Внесем в блок Traffic Light информацию о длительности фаз для каждого направления движения, полученную в предыдущем пункте (таблица 3.1). Здесь, для удобства, соединители полос названы двумя буквами: первая буква дорога, с которой начинается движение, а вторая буква – куда поворачивает автомобиль (например, ES – автомобиль едет с востока на юг).

На рисунке 6 представлен график работы светофора.

Длительности, сек:		ph1	ph2	7	ph3	7
Соединители полос:						
SN		Green	Green	Yellow	Red	Yellow
SN2		Green	Green	Yellow	Red	Yellow
NS		Green	Green	Yellow	Red	Yellow
NS2		Green	Green	Yellow	Red	Yellow
WS		Red	Green	Yellow	Green	Yellow
EW		Red	Red	Yellow	Green	Yellow
EW2		Red	Red	Yellow	Green	Yellow
EW3		Red	Red	Yellow	Green	Yellow
WE		Red	Red	Yellow	Green	Yellow
WE2		Red	Red	Yellow	Green	Yellow
NW		Green	Green	Yellow	Red	Yellow
EN		Red	Red	Yellow	Green	Yellow
SE		Green	Green	Yellow	Green	Yellow

Рис. 6. График работы светофора



Начальные значения параметров $ph1$, $ph2$ и $ph3$ являются 35, 15, 31 секунд соответственно. Именно эти параметры будут подбираться для достижения наименьшего времени проезда перекрестка.

Рассмотрим подробнее логику сбора информации о каждом автомобиле.

С помощью блока Road Network Descriptor выполняется подсчет количества автомобилей, выполняющих остановку. Этот подсчет необходим для учета очереди перед перекрестком. При остановке выполняется следующая команда:

```
if (car.isCarOn(roadN)) {  
    N++;  
} else if (car.isCarOn(roadS)) {  
    S++;  
} else if (car.isCarOn(roadE)) {  
    E++; // Изменено с W_cars на E_cars  
} else if (car.isCarOn(roadW)) {  
    W++;  
}
```

С помощью операции `car.is.CarOn(road)`, возвращающую булеву переменную (`true` – если суждение верно, `false` – если неверно), выполняется проверка, находится ли остановившийся автомобиль, на заданной дороге. Если да, срабатывает счетчик сложения. В противном случае происходит перебор оставшихся дорог.

При возобновлении движения выполняется такой же перебор, но уже со счетчиком вычитания. Таким образом, на выходе получаются динамические значения, которые меняют свое значение, в зависимости от количества остановленных автомобилей на каждом направлении.

В блоке Traffic Light в окне «При смене фазы» первым шагом проверяется текущая фаза светофора с помощью команды `currentPhaseIndex`.

С помощью нижеприведенной команды, при смене фазы, переменная принимает значение текущего модельного времени, тем самым регистрируя последнюю смену фазы светофора.

```
lastPhaseChangeTS = time();
```

Следующие команды фиксируют в наборе данных, при каждой смене фазы, динамические значения очередей, полученные выше с помощью блока Road Network Descriptor. Этот набор данных необходим для дальнейшей визуализации изменения объема очередей на графике: по оси x – значения времени последней смены фазы, по оси y – значения очереди.

```
N_cars.add(lastPhaseChangeTS,N_);  
S_cars.add(lastPhaseChangeTS,S_);  
E_cars.add(lastPhaseChangeTS,E_);  
W_cars.add(lastPhaseChangeTS,W_);
```

Затем фиксируется максимальное значение очереди на каждом направлении для дальнейшего автоматического расчета длительностей фаз:

1. при переходе в первую фазу фиксируется скопившаяся в пятую очередь на северном и южном направлениях.
2. при переходе в четвертую фазу фиксируется скопившаяся во вторую очередь на западном и восточном направлениях.

`int size_N = N_cars.size();` – получение размера массива очереди.

`if (size_N !=0)` – проверка, не пустой ли массив, для предотвращения ошибки.

`Ocher_N = N_cars.getY(size_N-1);` – получение последнего значения из массива.



Таким образом, за весь цикл работы светофора, мы получаем готовые значения очередей по каждому направлению для дальнейшей оптимизации перекрестка.

Используя полученные выше данные очередей, зададим алгоритм для светофора, который будет динамически в процессе моделирования изменять длительности фаз [7]. В конце каждого цикла алгоритм сравнивает очереди с двух перпендикулярных направлений (например, восточного и северных направлений). В случае, если очередь перед перекрестком с восточного подхода больше очереди со стороны северного подхода, необходимо увеличить длительность третьей фазы, и уменьшить длительность первой фазы, чтобы пропустить больше автомобилей, прибывающие с западного или/и восточного направлений:

$$\begin{cases} l_E > l_N; \\ T_{1i+1} = T_{1i} - T_{sdv}; \\ T_{3i+1} = T_{3i} + T_{sdv}, \end{cases} \quad (3.2)$$

где l_E – очередь перед перекрестком с восточного подхода, l_N – очередь перед перекрестком с северного подхода, T_{1i+1} – длительность следующей фазы, T_{1i} – длительность текущей фазы, T_{sdv} – величина изменения, сдвига фазы.

В обратном случае, если очередь перед перекрестком с северного направления больше очереди с восточного направления, необходимо уменьшить длительность третьей фазы, и увеличить длительность первой фазы:

$$\begin{cases} l_N > l_E; \\ T_{1i+1} = T_{1i} + T_{sdv}; \\ T_{3i+1} = T_{3i} - T_{sdv}, \end{cases} \quad (3.3)$$

Также введем дополнительные ограничения для длительностей каждой фазы:

$$\begin{cases} 10 \leq T_1 \leq 60; \\ 10 \leq T_2 \leq 30; \\ 10 \leq T_3 \leq 60; \end{cases} \quad (3.3)$$

Таким образом получаем, что светофор автоматически итеративно подстраивается под интенсивность автомобилей с каждой стороны, и в соответствии с этим, изменяет длительности фаз для лучшей пропускной способности перекрестка.

3 Результаты работы микромоделей в Anylogic

Для более точного анализа работы адаптивного светофора приблизим ситуацию к более реальной, когда интенсивность движения по каждому направлению меняется в течение дня.

Зададим расписание изменения интенсивностей для каждого направления движения. Интенсивность будет изменяться каждые 20 модельных минут.

На рисунке 7 представлен пример созданного расписания, на рисунке 8 представлено визуальное отображение расписания для каждого направления.



rateSchedWE - Расписание

▼ Данные

Тип: Интенсивность

Единица измерения: в час

Расписание задает: ☒ Интервалы (Начало, Конец) ☐ Моменты времени

Длительность: ☐ Неделя ☐ Дни/Недели ☒ Другая (нет привязки к календарю)

Повторять каждые: мин.

☒ Привязать к мин.

Значение по умолчанию:

☐ Загружается из базы данных

Начало	Конец	Значение
0	20	200.0
20	40	300.0
40	60	400.0
60	80	150.0
80	100	150.0

Рис. 7. Пример созданного расписания изменения интенсивности

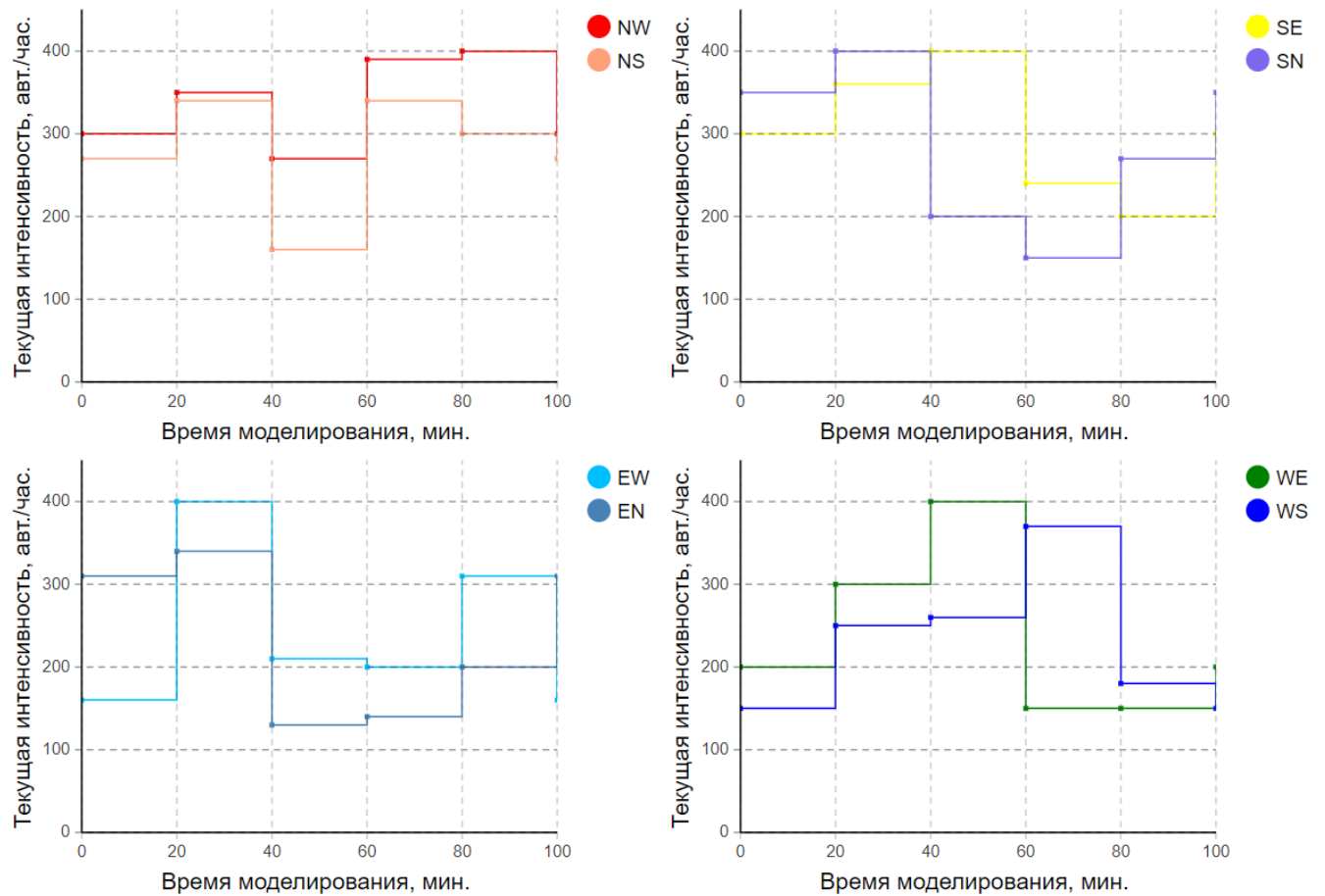


Рис. 8. Визуальное отображение расписания для каждого направления



Присвоим для каждого блока Car Source соответствующее расписание. Запустим моделирование с жестким алгоритмом светофорного регулирования.

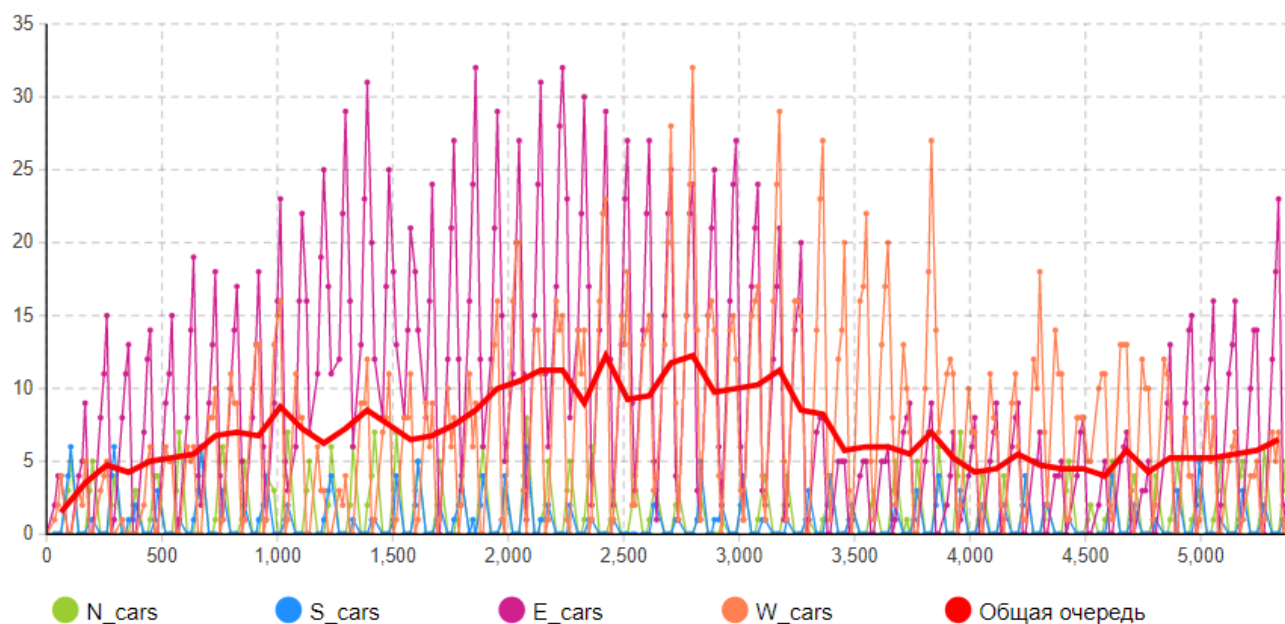


Рис. 9. Объем очередей перед перекрестком со светофором с жестким алгоритмом регулирования

Исходя из результатов, можно сказать, что перекресток в некоторые моменты имеет хорошую пропускную способность (не собирает очередей), но в основном, возникают заторы, в особенности, с восточного подхода (около 20 авт.).

Для минимизации очередей воспользуемся адаптивным алгоритмом светофорного регулирования и сравним результаты.

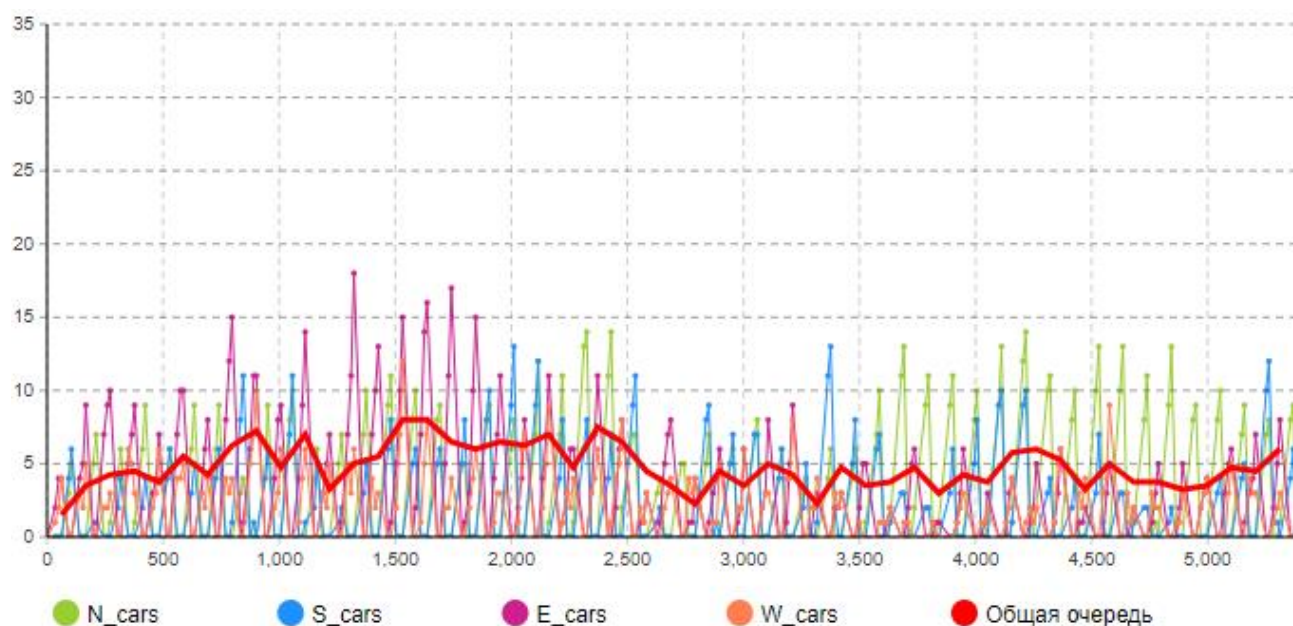


Рис. 10. Количество очередей перед перекрестком со светофором с жестким алгоритмом регулирования



Рисунок 10 показывает, что очереди на каждом подходе снизились, что доказывает, что светофор реагирует на увеличение потока, и адаптирует свои фазы для предотвращения возникновения заторов.

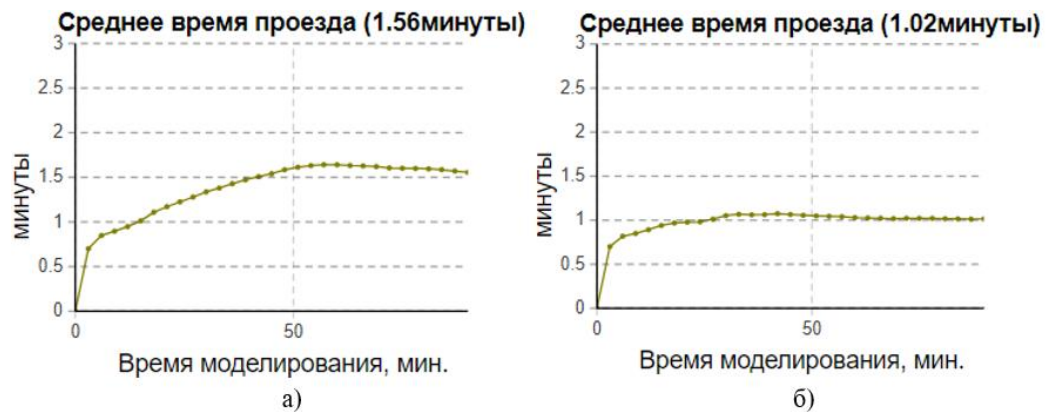


Рис. 11. Среднее время проезда автомобилей через перекресток со светофором с жестким (а) и адаптивным (б) алгоритмами регулирования

На рисунке 11 видно, что время проезда при адаптивном светофорном регулировании уменьшилось и остается неизменным (около 1 сек.), независимо от интенсивности движения.

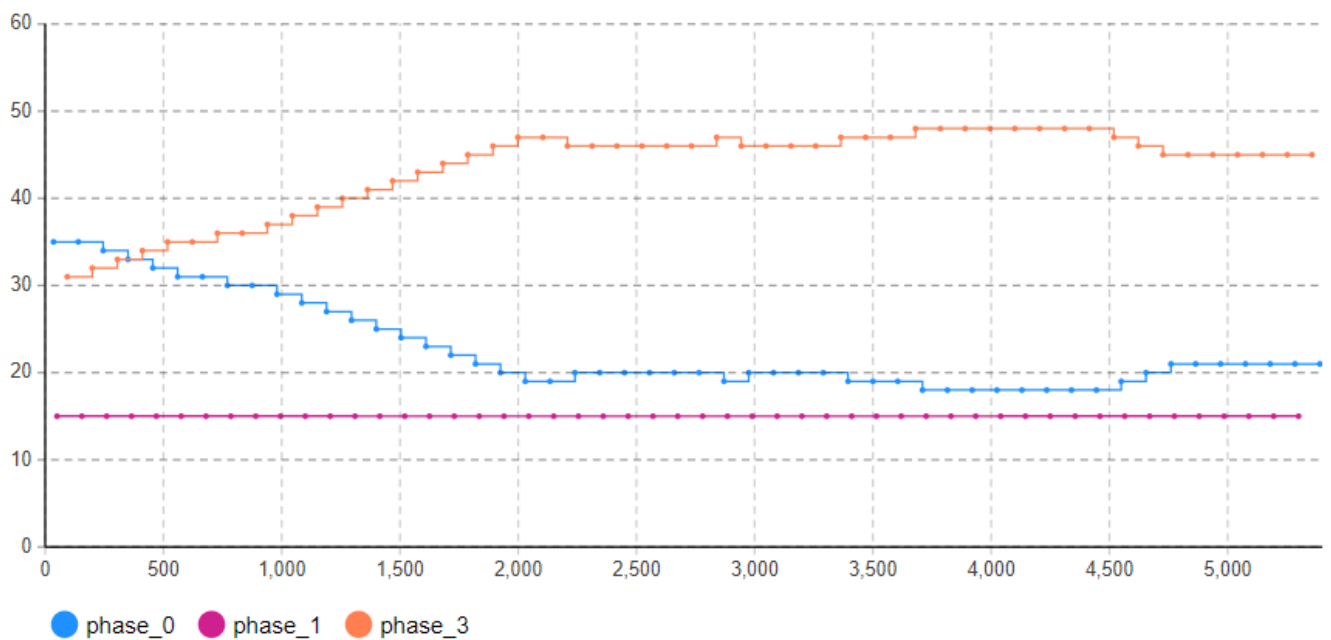


Рис. 12. Динамическое изменение длительностей фаз адаптивного светофора

Таблица 3 – Статистика изменений в работе перекрестка

Алгоритм управления светофором	Общая очередь, авт.	Ср. время проезда, сек
Жесткий	7,04	1,56
Адаптивный	4,81	1,02

Исходя из результатов, приведенных в таблице 3, видно что адаптивный алгоритм светофорного регулирования дает положительный результат: общая очередь снизилась на 39,3%, среднее время проезда уменьшилось на 25,6%.



Эти изменения свидетельствуют о том, что адаптивное регулирование может способствовать более равномерному распределению транспортных потоков, что снижает вероятность возникновения заторов и повышает общую пропускную способность перекрестка. Эти результаты подчеркивают важность внедрения современных технологий в систему управления дорожным движением для достижения оптимальных результатов и улучшения качества городской инфраструктуры.

Заключение

В данной статье была разработана микромодель транспортного движения в Anylogic. Для этого было проведено натурное исследование перекрестка улиц Планерной и Оптиков, на котором в часы пик наблюдаются заторы, с целью подсчета интенсивности движения и фиксации длительностей фаз светофора.

Был разработан и протестирован адаптивный алгоритм светофорного регулирования, который учитывает текущую интенсивность движения и изменяет длительность фаз светофора в зависимости от реальной ситуации на перекрестке. Этот алгоритм позволяет более эффективно распределять время зеленого света между направлениями, что помогает снизить заторы и улучшить пропускную способность перекрестка.

В результате внедрения адаптивного алгоритма светофорного регулирования удалось значительно снизить нагрузку на перекресток. В эксперименте с постоянной интенсивностью движения получилось добиться снижения среднего времени проезда сети с 1,37 до 1,04 секунд (на 24,1%). Также с внесением изменений снизилась средняя очередь перед перекрестком с 7,93 до 5,57 автомобилей (на 29,8%).

Также был проведен эксперимент, приближенный к реальным условиям: интенсивности движения с каждого направления меняются согласно расписанию. Было также получено снижение среднего времени проезда с 1,56 до 1,02 сек. (на 25,6%), и снижение средней очереди с 7,04 до 4,81 авт. (на 39,3 %).

Эти данные подтверждают, что адаптивный алгоритм светофорного регулирования эффективно справляется с изменяющимися условиями дорожного движения и значительно улучшает пропускную способность перекрестка.

СПИСОК ЛИТЕРАТУРЫ

1. *Левашев, А. Г.* Проектирование регулируемых пересечений: учеб. пособие / А. Г. Левашев, А. Ю. Михайлов, И. М. Головных, А. Г. Левашев, А. Ю. Михайлов, И. М. Головных. – Иркутск: Изд-во Иркутского гос. технического ун-та, 2007. – 208 с.
2. Центр дорожной информации РАДАР. Исследование параметров транспортных и пешеходных потоков на участках улично-дорожной сети города Перми в районе перекрестка ул. Героев Хасана — ул. Хлебозаводской — ул. Краснополянской [Электронный ресурс]. – URL: <https://road.perm.ru/issledovanie-parametrov-transportnykh-i-peshexodnykh-potokov-na-uchastkakh-ulichno-dorozhnoi-seti-goroda-permi-v-raione-perekrestka-ul-geroev-khasana-ul-khlebozavodskoi-ul-krasnopolyanskoi/> (дата обращения: 07.05.2025).
3. *Трофимов, А. В.* Методы исследований условий организации дорожного движения / А. В. Трофимов // *Фундаментальные и прикладные исследования: проблемы и результаты.* – 2013. – № 9. – С. 179-184.
4. программный продукт Anylogic. Возможности [Электронный ресурс]. – URL: <https://www.anylogic.ru/use-of-simulation/> (дата обращения: 07.05.2025).
5. *Андронов С. А.* Интеллектуальные транспортные системы / Андронов С. А., Фетисов В. А. — Саратов: Ай Пи Ар Медиа, 2019. — 260 с.
6. *Майоров Н. Н.* Моделирование транспортных процессов / Н. Н. Майоров, В. А. Фетисов. – СПб.: Изд-во ГУАП, 2011. – 163 с.



7. Кретов, А. Ю. Обзор некоторых адаптивных алгоритмов светофорного регулирования перекрестков / А. Ю. Кретов // Известия Тульского государственного университета. Технические науки. – 2013. – № 7-2. – С. 61-67.

ИНФОРМАЦИЯ ОБ АВТОРЕ

Марчук Дмитрий Юрьевич

Магистр

ФГАОУ ВО «Санкт-Петербургский государственный университет аэрокосмического приборостроения»

190000, Россия, Санкт-Петербург, ул. Большая Морская, д. 67, лит. А

E-mail: dima_marchuk_2001@mail.ru

INFORMATION ABOUT THE AUTHOR

Marchuk Dmitrij Yur'evich

Student

Saint-Petersburg State University of Aerospace Instrumentation

67, Bolshaya Morskaya str., Saint-Petersburg, 190000, Russia

E-mail: dima_marchuk_2001@mail.ru

Дата поступления: 04.08.2025

Дата принятия: 21.08.2025