ТЕХНИКАЛЫҚ ҒЫЛЫМДАР
ТЕХНИЧЕСКИЕ НАУКИ
TECHNICAL SCIENCE

CENTRAL ASIAN SCIENTIFIC JOURNAL
ISSN 3135-3061
2026. №4 (32), Том 3. С. 15–25
DOI: https://doi.org/10.5281/zenodo.24528682
Сайт журнала: https://cajournal.kz

УДК 004.94:656.11

СРАВНИТЕЛЬНЫЙ ОБЗОР СИМУЛЯТОРОВ ДОРОЖНОГО ДВИЖЕНИЯ РАЗНЫХ КЛАССОВ ДЛЯ ЗАДАЧ ИНТЕЛЛЕКТУАЛЬНЫХ ТРАНСПОРТНЫХ СИСТЕМ

А.К. Ураз
НАО «Казахский агротехнический исследовательский университет имени С. Сейфуллина», Астана, Казахстан

Ураз Аян Кайроллаулы — магистрант 1 курса образовательной программы 7М06101 «Информационные системы и решения в сфере IT», НАО «Казахский агротехнический исследовательский университет имени С. Сейфуллина», Астана, Казахстан. ORCID: 0009-0001-8955-4397

Научный руководитель: Амед Хайдар Алфи — кандидат философских наук, старший преподаватель кафедры философии, НАО «Казахский агротехнический исследовательский университет имени С. Сейфуллина», Астана, Казахстан.

© А.К. Ураз, 2026

Аннотация: Цель работы — обобщить возможности симуляторов дорожного движения разных классов и предложить рекомендации по их выбору для задач интеллектуальных транспортных систем. На основе обзора публикаций преимущественно 2016–2025 гг. семь открытых и коммерческих симуляторов (SUMO, CityFlow, MATSim, CARLA, MetaDrive, VISSIM, Aimsun) описаны по единой схеме и качественно сопоставлены по девяти критериям; отдельно рассмотрено совместное моделирование и проведён демонстрационный эксперимент в SUMO. Предложено выделять три семейства: симуляторы транспортного потока, агентные симуляторы спроса и трёхмерные симуляторы вождения с моделированием датчиков. Универсального симулятора нет: программы для крупных сетей не формируют изображений камер, трёхмерные ограничены небольшой сценой, а результаты разных программ расходятся при высокой нагрузке и зависят от калибровки; в собственном эксперименте вблизи насыщения среднее ожидание у агента обучения с подкреплением (RL) составило около 213 с против 43–53 с у фиксированного плана и адаптивного регулирования. Для управления светофором по изображениям камер рекомендовано обучать алгоритм в связке SUMO с MetaDrive, а проверять — в связке SUMO с CARLA.

Ключевые слова: имитационное моделирование, микроскопическое моделирование транспортного потока, совместное моделирование, обучение с подкреплением, адаптивное управление светофорами, калибровка модели, SUMO, CARLA

Введение

Нагрузка на улично-дорожные сети растёт и в крупных, и в небольших городах [1]: по данным, приведённым в [2], число легковых автомобилей в Казахстане за 2020–2025 гг. выросло с 3,87 до 6,37 млн. Если сеть и светофорное регулирование организованы недостаточно эффективно, возникают заторы и снижается пропускная способность [3]. По наблюдению автора как водителя и пассажира автобуса, в Астане утром и вечером, особенно с 17 до 20 ч, движение почти не бывает свободным, а машины, вставшие на перекрёстке, усиливают затор. Насколько нам известно, полностью эта проблема не решена ни в одном

городе Казахстана. Один из ответов — интеллектуальные транспортные системы (ИТС), в том числе адаптивное управление светофорами, которое меняет длительность фаз под загрузку перекрёстка [1]; в Казахстане такие решения уже изучают на моделях в Aimsun [2] и AnyLogic [1].

Симуляторы позволяют без риска воспроизводить даже аварийные ситуации [4], заранее оценить последствия решения [5] и сравнить варианты на одной модели [2]. Для обучения с подкреплением (reinforcement learning, RL), при котором алгоритм-агент учится методом проб и ошибок по сигналу «вознаграждения», они необходимы: реальный город даёт слишком мало примеров [6].

Симуляторов много, они относятся к разным классам [4], и выбрать подходящий непросто [7]; часто его выбирают по доступности, а не по технической пригодности [8]. Сравнительные работы охватывают два [5], три [8] или пять [7] симуляторов без трёхмерных (3D) симуляторов с датчиками, а обзор [4] написан с позиции беспилотного вождения и не рассматривает совместное моделирование. В рассмотренных нами работах не найдено сравнения, которое ставило бы рядом симуляторы транспортного потока, агентные и 3D-симуляторы с датчиками и давало рекомендации для задач ИТС, включая управление светофором по изображениям камер.

Цель статьи — систематизировать возможности симуляторов дорожного движения разных классов и сформулировать рекомендации по их выбору для задач ИТС. Задачи статьи: 1) классифицировать симуляторы по уровню детализации и назначению (подраздел 3.1); 2) описать семь распространённых симуляторов по единой схеме (подраздел 3.2); 3) рассмотреть совместное моделирование: зачем и как связывают симуляторы, какие связки применяют и каковы их ограничения (подраздел 3.3); 4) сравнить симуляторы по девяти критериям и предложить рекомендации по выбору для типовых задач ИТС (подраздел 3.4); 5) на демонстрационном эксперименте в SUMO проверить пригодность симулятора для обучения RL-агента и зависимость результатов управления светофором от нагрузки (подраздел 3.5).

Материалы и методы

Статья — обзорная; для проверки двух её выводов дополнительно проведён демонстрационный вычислительный эксперимент (подраздел 3.5). Источники 2016–2025 гг. (более ранние — только основополагающие) искали в Google Scholar, arXiv, КиберЛенинке и eLibrary (примеры запросов: «SUMO VISSIM Aimsun comparison», «SUMO CARLA co-simulation», «имитационное моделирование дорожного движения SUMO», «traffic simulation Kazakhstan»); отбирали рецензируемые публикации и препринты с полным текстом, которые описывают, классифицируют или сравнивают симуляторы либо показывают их применение в задачах ИТС. Семь рассмотренных симуляторов (SUMO, CityFlow, MATSim, CARLA, MetaDrive, VISSIM, Aimsun) широко применяются в исследованиях [4; 8], представляют все три семейства из подраздела 3.1 и включают открытые и коммерческие программы; коммерческие описаны по открытым данным (см. подраздел 3.6). Для обзора отобрано 24 источника, в том числе работы из Казахстана [1; 2] и России [3; 9]; ещё пять — официальное сообщение о системе «Сергек» и программы и методы, использованные в эксперименте.

Мы применяем качественное сравнение [7] по девяти критериям в четырёх группах: модель (уровень детализации, масштаб сети), возможности для ИТС (светофоры и RL, камеры и датчики, калибровка по реальным данным), интеграция (интерфейс программирования — API — и совместное моделирование) и доступность (лицензия, порог входа). Количественные данные взяты из опубликованных сравнений [5; 6; 8; 10]; шкала оценок приведена в примечании к таблице 1.

Демонстрационный эксперимент поставлен в SUMO на перекрёстке из примеров библиотеки sumo-rl: четыре подхода по две полосы длиной около 140 м. Спрос синтетический, двух уровней: средний (1920 машин/ч) и высокий, близкий к насыщению (2880 машин/ч); машины появляются в случайные моменты. Сравнивались три способа управления светофором: фиксированный план, рассчитанный по формуле Вебстера отдельно

для каждого уровня спроса; встроенное адаптивное регулирование SUMO, продлевающее зелёный по детекторам; RL-агент DQN, который раз в 5 с выбирает фазу по загрузке и очередям полос. Каждый прогон длится 1 ч модельного времени, для каждого сочетания выполнено 5 повторов с разными случайными числами, одинаковыми для всех способов. Показатели: среднее время ожидания на машину с учётом ожидания въезда на подход, средняя очередь, число машин, проехавших за час, и среднее время поездки.

При подготовке статьи использовался ИИ-ассистент Claude (Anthropic) — для поиска и систематизации литературы, подготовки черновика текста и кода эксперимента. Все сведения проверены автором по первоисточникам, эксперимент запущен автором; ответственность за содержание статьи несёт автор.

Результаты и обсуждение

3.1 Классификация симуляторов

По уровню детализации. Макроскопические модели описывают поток усреднёнными величинами — плотностью, средней скоростью, интенсивностью [9]; микроскопические воспроизводят движение каждого автомобиля с помощью моделей следования за лидером, где ускорение машины зависит от её скорости, скорости впереди идущей машины и дистанции до неё [9]. Промежуточное положение занимают кинетические [9] и мезоскопические [11] модели, а в очередных моделях машина просто ждёт в очереди на участке дороги [11]. Чем подробнее модель, тем больше вычислений [9], поэтому для крупных сетей поведение машин упрощают [4; 11]. В агентном подходе моделируется житель с планом дня, который меняет время выезда, маршрут, вид транспорта и пункт назначения; так воспроизводится спрос на поездки [11].

По назначению. Авторы обзора [4] выделяют пять классов симуляторов: транспортного потока, сенсорных данных, политики вождения, динамики транспортного средства (ТС) и комплексные; агентных симуляторов спроса среди них нет. Для задач ИТС мы предлагаем выделять три семейства: (1) симуляторы транспортного потока микро- и мезоуровня; (2) агентные симуляторы спроса; (3) 3D-симуляторы вождения с моделированием датчиков, которые строят трёхмерную сцену и выдают по ней «показания» виртуальных камер и LiDAR (лазерных дальномеров) [4; 12; 13]. Семейство (3) объединяет два класса [4] по общему свойству — виртуальным датчикам.

Место рассматриваемых симуляторов. MATSim относится к семейству (2); его очередную модель движения разработчики называют микроскопической [11], авторы [14] — мезоскопической, а по нашей оценке она промежуточная: машины учитываются по отдельности, но без следования за лидером [11]. Семейство (1) — открытые SUMO [15] и CityFlow [6] и коммерческие VISSIM [16] и Aimsun [5]; их 3D-визуализация [5; 15] — показ модели, а не моделирование датчиков. Семейство (3) — CARLA (по [4] — комплексный симулятор) и MetaDrive (по [4] — симулятор политики вождения).

3.2 Обзор симуляторов

Ниже симуляторы описаны по единой схеме: назначение, отличительные черты, ограничения и примеры применения.

SUMO (Simulation of Urban MObility) — открытый микроскопический симулятор Немецкого аэрокосмического центра [5] с лицензией EPL 2.0 [15]. Кроме модели следования за лидером Краусса [5; 8] в нём есть, например, IDM (Intelligent Driver Model) и модель Видемана [17], а через API TraCI им управляют из Python [15]. Ограничения: нужны навыки программирования, визуализация только двумерная [5], изображений камер SUMO не создаёт [18]. Пример: компания NoTraffic калибрует в SUMO модели перекрёстков по видео- и радарным данным; новая логика светофора снизила задержку на 50 % в модели и на 43 % на реальном перекрёстке [17].

CityFlow — открытый микроскопический симулятор, созданный в Шанхайском университете Цзяо Тун как среда RL для управления светофорами [6]. По данным авторов, он более чем в 20 раз быстрее SUMO при расхождении средней длительности поездки не более 1,06 %, но сравнивался только с SUMO и не калиброван по реальным данным [6],

поэтому эти цифры требуют независимой проверки. Датчиков CityFlow не моделирует [19] и неактивно поддерживается [4].

MATSim — открытый агентный симулятор спроса на Java из ETH Zurich: на каждой итерации часть агентов (часто 10 %) меняет время выезда, маршрут, вид транспорта или пункт назначения, пока оценка планов не стабилизируется, а очередная модель движения позволяет считать сценарии с 10^7 и более «частиц» [11]. Пример: в [14] адаптивное управление в модуле светофоров MATSim снизило задержки в регионе Котбуса на 35 % по сравнению с фиксированным планом.

CARLA — открытый 3D-симулятор беспилотного вождения на игровом движке Unreal Engine 4 с клиентом на Python, созданный Intel Labs, Toyota Research Institute и Центром компьютерного зрения (Барселона) [12]. Его сила — датчики: RGB-камеры, карты глубины и семантической сегментации [12], а также LiDAR и радар [4]. Ограничения: фоновые машины следуют простым правилам, поэтому массового реалистичного трафика CARLA не воспроизводит [20], а реалистичная графика требует много вычислений [12].

MetaDrive — открытый лёгкий 3D-симулятор исследователей из CUHK, ETH Zurich и UCLA для изучения обобщающей способности RL, то есть умения агента действовать в новых условиях [13]. Сцены собираются из типовых блоков или импортируются из наборов Waymo и Argoverse; есть LiDAR, камеры RGB и глубины, фоновые машины движутся по модели IDM [13]. Ограничения: графика проще, чем в CARLA, нет пешеходов и велосипедистов [13], сцены короткие, вокруг одной машины [19].

VISSIM — коммерческий микроскопический поведенческий симулятор компании PTV (с 1994 г.), описывающий решения водителя: следование за лидером (психофизическая модель Видемана [8]), смену полосы, проезд конфликтных зон [16]. Есть API, внешние контроллеры светофоров и внешняя модель водителя [16]. Ограничения: закрытый код [4; 10] и сложные модели, требующие тщательной калибровки [10]. Пример: модель перекрёстка в Санкт-Петербурге показала, что одной оптимизацией светофоров при текущей нагрузке ситуацию не улучшить [3].

Aimsun — коммерческий симулятор испанской компании TSS [5] с эвристическим динамическим распределением потоков, то есть расчётом маршрутов с учётом меняющихся во времени задержек [21]. Он совмещает уровни детализации, не требует программирования и выдаёт готовые результаты [5]. Ограничения: закрытый код и дорогая лицензия — по данным [5], 20–25 тыс. евро. Пример из Казахстана: в модели центра Сатпаева, построенной в Aimsun по данным 80 точек наблюдения, лучший сценарий сократил время поездки на 9,8 % [2].

3.3 Совместное моделирование

Зачем связывать симуляторы. SUMO моделирует поток во всей сети, но не создаёт изображений камер [18]; CARLA, наоборот, даёт 3D-сцену и датчики, но не массовый реалистичный трафик [20; 22]; симуляторы связи ns-3 и OMNeT++ моделируют передачу сообщений при упрощённом движении машин [23]. Поэтому задачи на стыке областей решают с помощью совместного моделирования (co-simulation): несколько симуляторов работают одновременно и на каждом шаге времени обмениваются состоянием модели [18; 23].

Как устроено совместное моделирование. Симуляторы продвигаются общими фиксированными шагами, например по 0,05 с (CARLA для этого переводят в синхронный режим); ход времени задаёт ведущий участник, например SUMO, принимающий через интерфейс TraCI [15] нескольких клиентов; на каждом шаге передаются положения ТС и состояния светофоров, а сеть одного симулятора конвертируют в формат другого [18; 19; 20; 23; 24].

Типичные связки. По назначению связки делятся на три группы. Для испытаний беспилотных и кооперативных ТС SUMO и CARLA связаны в OpenCDA [20] и CARLA-SUMO-Gym [22]; в [22] полнота обнаружения машин ночью составила 0,23 против 0,41 утром. Для моделирования связи транспортных средств между собой и с инфраструктурой

(V2X) и кооперативного восприятия к ним добавляют симулятор связи: ns-3 в ms-van3t-CARLA [24], OMNeT++ в OpenCAMS [23]. Для управления светофорами по камерам SUMO связывают с CARLA [18] или с MetaDrive в TrafficDojo [19].

Ограничения и проблемы. Первое — вычислительная нагрузка: в [18] 100 с модельного времени с распознаванием на 43 камерах рассчитывались за 145 с, поэтому распознавание выполняли раз в 5 с. Второе — согласование: возможны рассогласования во времени [23], а конвертация карт из OpenStreetMap требует ручных исправлений [19]. Третье — разрыв «симуляция — реальность»: изображения CARLA заметно отличаются от реальных [4], что признают и авторы [19]. Поэтому опыты с виртуальными камерами (в [18] RL-агенты с вознаграждением по средней скорости сохранили выигрыш при ошибках детектора YOLO) пока показывают устойчивость алгоритма к ошибкам распознавания, но не его точность на реальном перекрёстке.

3.4 Сравнительный анализ

Сравнение по критериям. Результаты сравнения сведены в таблицы 1 и 2.

Таблица 1 – Сравнение симуляторов: модель и возможности для ИТС

Симулятор	Масштаб сети	Светофоры и RL	Камеры и датчики	Калибровка по реальным данным
SUMO	+ до ~100 тыс. агентов [8]	+ частая среда RL [6; 18]	− только детекторы [18]	+ Cadyts, калибраторы [15]; видео и радар [17]
CityFlow	+ сетка 30×30, часть Манхэттена* [6]	+ создан для RL [6]	− [19]	− только в планах [6]
MATSim	+ 10^7 «частиц» [11]; лучшая масштабируемость [8]	± модуль светофоров (расширение) [14]; RL н/д	н/д	± объёмы, 96 % [8]
CARLA	− малые карты [12]; слабый фоновый трафик [20]	± RL вождения [12]; светофоры — через SUMO [18]	+ RGB, глубина, LiDAR, радар [4; 12]	н/д
MetaDrive	− сцена вокруг одного ТС [19]	± RL вождения [13]; светофоры — через SUMO [19]	+ RGB, глубина, LiDAR [13]	± импорт сцен Waymo, Argoverse [13]
VISSIM	± замедление > 12 тыс. агентов [8]	± внешние контроллеры [16]; RL по виду сверху (по [18])	± вид сверху (по [18]); 3D-визуализация [15]	+ методы калибровки [16]; объёмы, 95 % [8]
Aimsun	+ гибрид «область + детальные зоны» [5]	± планы по поворотам [10]; RL н/д	н/д; 2D и 3D-визуализация [5]	+ проверка по детекторам (GEH) [5]; 80 точек [2]

Примечание. «+» — возможность есть и подтверждена источником (сильная сторона); «±» — есть с ограничениями или источники расходятся; «−» — нет или слабая сторона; «н/д» — в использованных источниках данных нет. Уровень детализации — см. подраздел 3.1. * — данные разработчиков симулятора. «по [n]» — источник [n] приводит сведение со ссылкой на другую работу. Источники описывают разные версии программ (см. подраздел 3.6).

Таблица 2 – Сравнение симуляторов: доступность и интеграция

Симулятор	Лицензия и стоимость	Порог входа	API и языки	Совместное моделирование
SUMO	+ открытый, EPL 2.0 [15]	± Python/C++, XML; 2D [5]	+ TraCI, Python [15]	+ ns-3, OMNeT++ [15]; CARLA [18; 20]; MetaDrive [19]
CityFlow	+ открытый код [4; 6]; лицензия н/д	± прост для RL; не поддерживается [4]	+ Python (pybind11) [6]	н/д
MATSim	+ открытый, бесплатно [7; 11]	± Java [11]; сильная документация [7]	± Java, заменяемые модули [11]	± внешние симуляторы флота (по [18])

Симулятор	Лицензия и стоимость	Порог входа	API и языки	Совместное моделирование
CARLA	+ открытый, бесплатные 3D-ресурсы [12]	− 20 ГБ диска, 8 ГБ видеопамяти [4]	+ Python, клиент–сервер [12]	+ с SUMO [18; 20]
MetaDrive	+ открытый, Apache 2.0 [13]	+ ~100 МБ, установка одной командой* [13]	+ Python, OpenAI Gym [13]	+ с SUMO [19]
VISSIM	− коммерческий, закрытый код [4; 10]	± удобный интерфейс [15], сложные модели [10]	± COM: C, C++, Python, VB.Net [10]	± внешние модели через API [16]
Aimsun	− коммерческий, 20–25 тыс. € [5]	± без программирования [5], но платный	± Python-скрипты, API, SDK [10]	± обмен с VISSIM, Visum и др. [5]

Примечание. Обозначения — как в таблице 1. Оценки в столбце «Порог входа» — экспертная оценка автора по указанным фактам.

Из таблиц 1 и 2 следуют четыре вывода.

1) Ни один симулятор не получил «+» по всем критериям; к такому же итогу пришли авторы двух других сравнений [7; 8]. Значит, вопрос «какой симулятор лучше» имеет смысл только для конкретной задачи.

2) Из таблицы 1 видно, что столбцы «Камеры и датчики» и «Масштаб сети» почти зеркальны: у симуляторов с «+» по масштабу (SUMO, CityFlow, MATSim, Aimsun) по датчикам стоит «−» или «н/д», у симуляторов с «+» по датчикам (CARLA, MetaDrive) масштаб ограничен небольшой сценой, VISSIM занимает промежуточное положение. Этот разрыв делает совместное моделирование (подраздел 3.3) необходимым для задач вида «камера → управление».

3) Открытые и коммерческие симуляторы сильны в разном. Открытые бесплатны и у большинства есть API на Python (таблица 2), поэтому эксперимент может повторить любой читатель; коммерческие дают удобный интерфейс [5; 15], средства калибровки [16] и готовые к анализу результаты [5]. Это согласуется с оценкой авторов [7]: промышленные программы проще в работе, но хуже расширяются. Поэтому для научной статьи важнее воспроизводимость, а для проекта заказчика — готовые инструменты.

4) Масштаб и детальность конкурируют: MATSim заменяет следование за лидером очередной моделью [11] и показал лучшую масштабируемость, VISSIM подробно описывает поведение водителя [16], но замедляется на крупных сценариях, SUMO находится посередине [8], а реалистичная графика CARLA требует больших вычислений [12]. Это частный случай закономерности из подраздела 3.1: чем подробнее модель, тем меньше сеть, которую можно рассчитать за разумное время. CityFlow — частичное исключение: скорость он получает за счёт оптимизации вычислений, но это данные только его авторов [6].

Что показывают сравнительные исследования. *Согласованность.* В [10] одна сеть с 11 регулируемыми перекрёстками построена в VISSIM, Aimsun и SUMO по одинаковым данным: при базовом и сниженном спросе результаты согласовались, а при спросе выше на 25 % VISSIM показал большее время поездки и больший разброс; Aimsun и SUMO в [5] разошлись на 1–13 %. Расхождение при высокой нагрузке особенно важно для управления светофорами: именно в насыщенном режиме от управления ждут наибольшего эффекта. *Производительность.* В малом сценарии [8] SUMO рассчитывал модель примерно в 6,5 раза быстрее VISSIM; VISSIM замедлялся при числе агентов более 12 тыс., SUMO давал сбои выше 100 тыс., а лучшую масштабируемость показал MATSim. Авторы CityFlow сообщают об ускорении более чем в 20 раз относительно SUMO [6]; расхождение с [8] мы объясняем разными версиями (2019 и 2025 гг.) и метриками. Нас настораживает, что в [6] нет результатов обучения RL-агентов, ради которых создан CityFlow.

Точность. В [8] эталоном служило время поездки по участку длиной 3 км по данным Google Maps (8–13 мин): VISSIM дал 8,2 мин, SUMO — 13,1 мин, MATSim — 17,7 мин. В аннотации [8] ближе к реальности назван VISSIM, в обсуждении — SUMO, поэтому вывод о

превосходстве одного из них мы считаем необоснованным: одного участка и одного эталона (Google Maps, один день [8]) для такого вывода мало. Общий вывод источников другой: результаты сильно зависят от калибровки по реальным данным [5; 10; 17]; так, из трёх моделей следования, проверенных более чем на 20 перекрёстках, требование к точности задержки выполнила только одна [17].

Рекомендации по выбору симулятора сведены в таблицу 3; это наш вывод из таблиц 1, 2 и подраздела 3.2.

Таблица 3 – Рекомендуемые симуляторы для типовых задач ИТС

Задача	Рекомендуемый симулятор	Обоснование
1. Адаптивное управление светофорами по данным детекторов	SUMO; для одного перекрёстка — также VISSIM	внешнее управление через TraCI, калибраторы по детекторам [15]; калибровка по видео и радару и перенос результата на реальный перекрёсток [17]; внешние контроллеры VISSIM [16]
2. RL-управление светофорами крупной сети	CityFlow — отладка; SUMO — итоговые эксперименты	CityFlow создан для RL и быстрее SUMO* [6], но не поддерживается [4] и не калиброван [6]; SUMO устойчив до ~100 тыс. агентов [8]
3. Управление светофором по изображениям камер	SUMO + MetaDrive — обучение; SUMO + CARLA — проверка устойчивости к ошибкам распознавания	симуляторы потока не дают изображений [18; 19]; MetaDrive быстрее* [19]; CARLA реалистичнее, погода и освещение [18]; обучение в SUMO, проверка с камерами [18]
4. Испытания беспилотных ТС, моделирование датчиков	CARLA (+ SUMO для фонового трафика); MetaDrive — обобщающая способность RL-политики	полный набор датчиков [4; 12]; слабый фоновый трафик CARLA [20]; процедурная генерация сцен [13]
5. Городской спрос, маршруты, сценарии на сутки	MATSim	агентный спрос, 10^7 «частиц» [11]; моделирование суток [14]; лучшая масштабируемость [8]
6. Детальный анализ перекрёстка, безопасность, проекты для заказчика	VISSIM, Aimsun	детальная модель поведения, рекомендация авторов [8]; готовые показатели, уровень обслуживания [5]; опыт в Казахстане и России [2; 3]
7. Учебные и исследовательские проекты с ограниченным бюджетом	SUMO; MetaDrive	бесплатны [13; 15]; низкие требования к компьютеру [4]; открытые симуляторы снижают барьер входа [4]

Примечание. * — данные разработчиков симулятора. Все рекомендации — вывод автора на основе таблиц 1, 2 и подраздела 3.2.

Для управления светофором по камерам нужна связка SUMO с 3D-симулятором, но авторы [19] считают CARLA слишком медленной для обучения по изображениям, а авторы [18] — графику MetaDrive недостаточно реалистичной для проверки детектора. По нашему мнению, это два этапа одной работы: обучение, где важна скорость, — в SUMO + MetaDrive, проверка устойчивости к ошибкам распознавания при разной погоде и освещении — в SUMO + CARLA; в [18] агентов так и обучали в SUMO на точных данных, а проверяли уже с камерами.

Для дальнейшего нашего исследования ошибок компьютерного зрения при управлении светофором в идеале подходит связка SUMO + CARLA + YOLO [18] — ради реализма. Но требования CARLA к компьютеру [4] выше доступных нам, поэтому выбрана связка SUMO + MetaDrive [19], в которой модель можно обучать непрерывно.

VISSIM и Aimsun оправданы в инженерных проектах: VISSIM рекомендуют для анализа регулируемых перекрёстков и безопасности [8], а Aimsun выдаёт готовые показатели, например уровень обслуживания [5]. Для RL они менее удобны: готовых RL-сред для них в наших источниках нет, а пример обучения по изображениям VISSIM ограничен одним перекрёстком [18].

Главный недостаток симуляторов. Данные в модель вводит сам исследователь, поэтому нужны многократные прогоны и калибровка [5; 10; 17], но и тогда результат остаётся модельным. По нашему мнению, главное, чего не хватает, — человеческого фактора: люди реагируют по-разному. Расхождение у NoTraffic (−50 % в модели, −43 % на перекрёстке [17]) невелико, но, на наш взгляд, его можно сократить, вводя не суммарные, а детальные данные о поведении участников. Поэтому приоритетом мы считаем реализм на этапе проверки, а скорость — на этапе обучения.

Применимость в Казахстане. Казахстанских работ, где светофор обучался бы в симуляторе по изображениям камер, в рассмотренных источниках не найдено. Препятствиями мы считаем финансирование, саму разработку и климат: снег и сильные морозы, на которые оборудование может быть не рассчитано; CARLA, например, не моделирует снег [4]. Нужна и подготовленная инфраструктура, возможно — изменения ПДД. Но основа есть: система «Сергек» в пилотном режиме распознаёт по камерам непристёгнутый ремень и непропуск пешехода [25]. Поэтому мы считаем направление перспективным, хотя и требующим времени.

3.5 Демонстрационный эксперимент

Цель эксперимента — проверить на собственной модели два вывода обзора: пригоден ли SUMO как среда обучения RL-агентов (таблица 3, задача 2) и зависят ли результаты управления от нагрузки (подраздел 3.4). Модель построена в SUMO 1.28 [15] через библиотеку sumo-rl [26]. Фиксированный план рассчитан по формуле Вебстера [27]: цикл 62 с при среднем спросе и 120 с при высоком. RL-агент — глубокая Q-сеть (DQN) [28] из библиотеки Stable-Baselines3 [29]; его вознаграждение — уменьшение суммарного времени ожидания на подходах. Агент обучался 100 тыс. шагов (около 139 ч модельного времени) при случайной смене уровня спроса; обучение заняло около 26 мин на процессоре в Google Colab. Результаты приведены в таблице 4 и на рисунке 1.

Таблица 4 – Результаты демонстрационного эксперимента (среднее ± стандартное отклонение по 5 повторам)

Спрос	Способ управления	Время ожидания, с/маш.	Очередь, маш.	Пропущено, маш./ч	Время поездки, с
Средний (1920 маш./ч)	Фиксированный план	17,7 ± 0,8	9,4 ± 0,7	1885 ± 45	46,8 ± 1,1
	Адаптивное регулирование SUMO	14,3 ± 1,2	7,6 ± 0,8	1888 ± 47	43,2 ± 1,3
	RL-агент DQN	18,7 ± 1,7	9,8 ± 1,1	1883 ± 48	53,5 ± 2,8
Высокий (2880 маш./ч)	Фиксированный план	43,4 ± 2,4	32,6 ± 2,0	2824 ± 59	74,4 ± 2,7
	Адаптивное регулирование SUMO	52,6 ± 15,8	32,8 ± 3,5	2819 ± 55	83,3 ± 16,4
	RL-агент DQN	212,8 ± 36,6	35,8 ± 1,9	2531 ± 25	255,0 ± 32,4

Примечание. Составлено автором по результатам моделирования в SUMO. Время ожидания включает ожидание въезда на подход; очередь — среднее число стоящих машин на восьми входящих полосах; время поездки — по машинам, завершившим поездку за 1 ч.

При среднем спросе все три способа пропустили практически весь поток — около 1885 машин/ч. Наименьшее время ожидания дало адаптивное регулирование: 14,3 с на машину против 17,7 с у фиксированного плана. RL-агент показал 18,7 с; его отличие от фиксированного плана не выходит за разброс повторов. При высоком спросе различия резко выросли. Фиксированный план и адаптивное регулирование дали 43,4 и 52,6 с, причём у адаптивного регулирования разброс между повторами велик. У RL-агента ожидание составило 212,8 с — в 4–5 раз больше. За час через перекрёсток проехала 2531 машина против 2824 при фиксированном плане, а около 270 машин так и не смогли въехать на подходы.

© Central Asian Scientific Journal, 2026

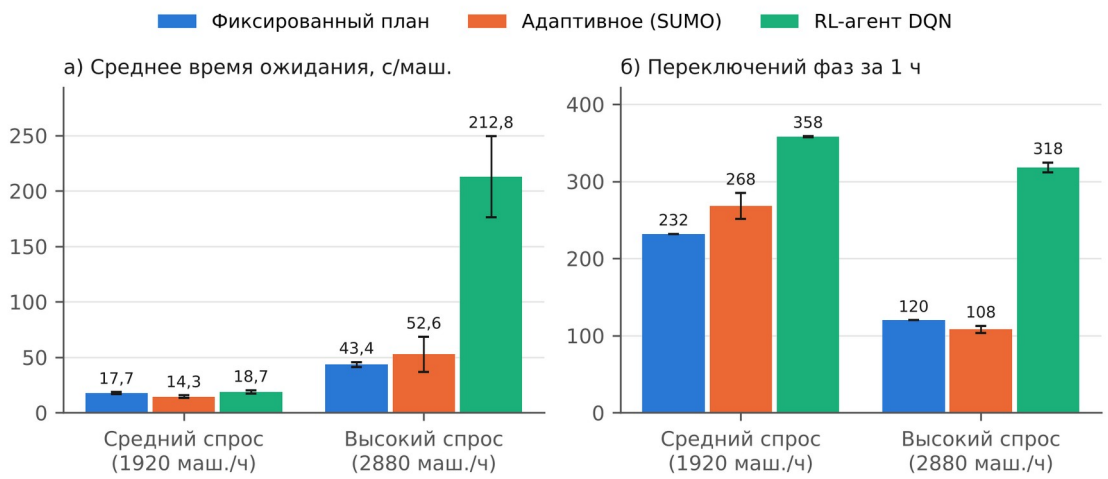

Рисунок 1 - Среднее время ожидания (а) и число переключений фаз (б) при разных способах управления светофором (составлено автором)

Неожиданным для нас оказался результат RL-агента. На наш взгляд, главная причина — слишком частые переключения фаз (рисунок 1, б). При переходе к высокому спросу фиксированный план и адаптивное регулирование переключались примерно вдвое реже (232 → 120 и 268 → 108 раз в час), то есть удлиняли зелёный. Агент почти не изменил поведения (358 → 318). Каждое переключение — это 2 с жёлтого и время на трогание; возможно, время теряется и пока машины, уже въехавшие на перекрёсток, завершают манёвр (это не проверялось). При среднем спросе запас пропускной способности покрывает эти потери, а вблизи насыщения число проехавших машин падает, и очередь выходит за пределы подходов. Итак, SUMO технически пригоден как среда обучения: агент обучается примерно за 26 мин на обычном процессоре, но хорошее управление этим не гарантируется — при нашей единственной настройке агент не превзошёл простые способы. Порядок способов меняется с нагрузкой: при среднем спросе лучшим было адаптивное регулирование, вблизи насыщения оно потеряло преимущество перед фиксированным планом, а агент проиграл обоим в 4–5 раз. Это согласуется с выводом обзора о том, что результаты нужно проверять при нескольких уровнях нагрузки.

Эксперимент демонстрационный: один изолированный перекрёсток, синтетический спрос без калибровки по реальным данным, идеальные детекторы, одна настройка агента с ограниченным обучением и 5 повторов. Кроме того, фиксированный план рассчитан по формуле Вебстера под заранее известный спрос каждого уровня, а RL-агент был один на оба уровня, что даёт плану преимущество. Поэтому эксперимент показывает свойства постановки, а не преимущество какого-либо способа управления в реальном городе. Потенциал RL-агента, на наш взгляд, этим не исчерпан: его можно обучать дольше, штрафовать за каждое переключение или обучать отдельных агентов для разных уровней нагрузки, а затем проверить на реальном перекрёстке Астаны.

3.6 Ограничения обзора

Во-первых, VISSIM и Aimsun описаны без изучения полных текстов [16] и [21]. Во-вторых, часть характеристик — данные разработчиков [6; 13; 19], а порог входа — экспертная оценка автора. В-третьих, источники сравнивали разные версии программ на разных сценариях (статус поддержки по [4] — на конец 2023 г.), поэтому таблицы 1 и 2 собраны из разных работ. В-четвёртых, собственный эксперимент (подраздел 3.5) демонстрационный и не проверяет точность симуляторов по реальным данным.

Заключение

Для задач ИТС предложено выделять три семейства симуляторов: симуляторы транспортного потока, агентные симуляторы спроса и 3D-симуляторы вождения с моделированием датчиков. Универсального симулятора нет: по данным рассмотренных источников, симуляторы для крупных сетей не моделируют камеры, а 3D-симуляторы

ограничены небольшой сценой; результаты разных программ расходятся при высокой нагрузке и зависят от калибровки по реальным данным. Разрыв между масштабом сети и датчиками восполняет совместное моделирование — ценой роста вычислительной нагрузки и сложной настройки; опыты с виртуальными камерами пока показывают устойчивость алгоритма к ошибкам распознавания, но не его точность на реальном перекрёстке.

Демонстрационный эксперимент согласуется с выводами обзора: SUMO технически пригоден как среда обучения — RL-агент для одного перекрёстка обучается примерно за 26 мин на обычном процессоре, но при нашей настройке он не превзошёл простые способы управления; порядок способов меняется с нагрузкой, а вблизи насыщения агент проиграл обоим в 4–5 раз по времени ожидания. Значит, способы управления нужно сравнивать при нескольких уровнях нагрузки.

Выбор симулятора определяется задачей: для RL-управления светофорами подходят SUMO и CityFlow (последний — для отладки), для городского спроса — MATSim, для детального инженерного анализа — VISSIM и Aimsun, а для задач «камера → управление» — связка SUMO с MetaDrive для обучения и с CARLA для проверки.

Предложенные рекомендации могут использоваться при выборе инструментов для исследований и проектов ИТС, в том числе в Казахстане. Перспективное направление — сравнить симуляторы на одном калиброванном сценарии с виртуальными камерами для задачи управления светофором. По нашему мнению, дальше важно изучать совместное моделирование: модель, обученная в связке симуляторов, проще будет реализовать на практике.

Конфликт интересов: автор заявляет об отсутствии конфликта интересов.

Финансирование: исследование не имело внешнего финансирования.

Список литературы:

1. Tulegulov A.D., Akishev K.M., Zhamangarin D.S. et al. The use of digital technologies for intelligent traffic management // Вестник КазУТБ. – 2023. – Т. 2. – С. 146–155. – DOI: 10.58805/kazutb.v.2.23-485.

2. Saiynov A.M., Tashenov S.K., Kartashov M.V. et al. Detailed transport analysis and modelling focused on the key network of streets in Satpayev city // Вестник ЕНУ им. Л.Н. Гумилева. Серия: Технические науки и технологии. – 2025. – № 3 (152). – С. 257–267. – DOI: 10.32523/2616-7263-2025-152-3-257-267.

3. Селиверстов С.А., Селиверстов Я.А., Шаталова Н.В. и др. Разработка модели и оценка сложного нагруженного перекрестка улично-дорожной сети с использованием современных программных комплексов // SCM-2020. – СПб.: СПбГЭТУ «ЛЭТИ», 2020. – С. 166–170.

4. Li Y., Yuan W., Zhang S. et al. Choose Your Simulator Wisely: A Review on Open-source Simulators for Autonomous Driving // IEEE Transactions on Intelligent Vehicles. – 2024. – Vol. 9, No. 5. – P. 4861–4876. – DOI: 10.1109/TIV.2024.3374044.

5. Baza-Solares N., Velasquez-Martínez R., Torres-Bohórquez C. et al. Traffic Simulation with Open-Source and Commercial Traffic Microsimulators: A Case Study // Communications – Scientific Letters of the University of Zilina. – 2022. – Vol. 24, No. 2. – P. E49–E62. – DOI: 10.26552/com.C.2022.2.E49-E62.

6. Zhang H., Feng S., Liu C. et al. CityFlow: A Multi-Agent Reinforcement Learning Environment for Large Scale City Traffic Scenario // The World Wide Web Conference (WWW '19). – New York: ACM, 2019. – P. 3620–3624. – DOI: 10.1145/3308558.3314139.

7. Diallo A.O., Lozenguez G., Doniec A. et al. Comparative Evaluation of Road Traffic Simulators based on Modeler's Specifications: An Application to Intermodal Mobility Behaviors // Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART 2021). – SciTePress, 2021. – Vol. 1. – P. 265–272. – DOI: 10.5220/0010238302650272.

8. Anyin P., Wittenberg D., Pannek J. Quantitative Methodology for Comparing Microscopic Traffic Simulators // Future Transportation. – 2025. – Vol. 5, No. 4. – Art. 201. – DOI: 10.3390/futuretransp5040201.

9. Швецов В.И. Математическое моделирование транспортных потоков // Автоматика и телемеханика. – 2003. – № 11. – С. 3–46.

10. Saroj A., Xu G., Shao Y. et al. A Systematic Comparison for Consistent Scenario Development Using Microscopic Simulation Software // Proceedings of the 2024 Winter Simulation Conference (WSC 2024). – IEEE, 2024. – P. 194–205.

11. The Multi-Agent Transport Simulation MATSim / ed. by A. Horni, K. Nagel, K.W. Axhausen. – London: Ubiquity Press, 2016. – 618 p. – DOI: 10.5334/baw.

12. Dosovitskiy A., Ros G., Codevilla F. et al. CARLA: An Open Urban Driving Simulator // Proceedings of the 1st Annual Conference on Robot Learning (CoRL 2017). – PMLR, 2017. – Vol. 78. – P. 1–16.

13. Li Q., Peng Z., Feng L. et al. MetaDrive: Composing Diverse Driving Scenarios for Generalizable Reinforcement Learning // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2023. – Vol. 45, No. 3. – P. 3461–3475. – DOI: 10.1109/TPAMI.2022.3190471.

14. Thunig T., Kühnel N., Nagel K. Adaptive traffic signal control for real-world scenarios in agent-based transport simulations // Transportation Research Procedia. – 2019. – Vol. 37. – P. 481–488. – DOI: 10.1016/j.trpro.2018.12.215.

15. Alvarez Lopez P., Behrisch M., Bieker-Walz L. et al. Microscopic Traffic Simulation using SUMO // 2018 21st International Conference on Intelligent Transportation Systems (ITSC). – Maui: IEEE, 2018. – P. 2575–2582. – DOI: 10.1109/ITSC.2018.8569938.

16. Fellendorf M., Vortisch P. Microscopic Traffic Flow Simulator VISSIM // Fundamentals of Traffic Simulation / ed. by J. Barceló. – New York: Springer, 2010. – P. 63–93. – (International Series in Operations Research & Management Science; vol. 145). – DOI: 10.1007/978-1-4419-6142-6_2.

17. Dobrilko O., Bublil A. Leveraging SUMO for Real-World Traffic Optimization: A Comprehensive Approach // SUMO Conference Proceedings. – 2024. – Vol. 5. – P. 179–194. – DOI: 10.52825/scp.v5i.1120.

18. Azfar T., Huang K., Tracy A. et al. Traffic Co-Simulation Framework Empowered by Infrastructure Camera Sensing and Reinforcement Learning. – arXiv:2412.03925v2. – 2025.

19. He P., Li Q., Yuan X. et al. A Simple Framework Towards Vision-Based Traffic Signal Control with Microscopic Simulation. – arXiv:2403.06884v2. – 2025.

20. Xu R., Guo Y., Han X. et al. OpenCDA: An Open Cooperative Driving Automation Framework Integrated with Co-Simulation // 2021 IEEE International Intelligent Transportation Systems Conference (ITSC). – IEEE, 2021. – P. 1155–1162.

21. Barceló J., Casas J. Dynamic Network Simulation with AIMSUN // Simulation Approaches in Transportation Analysis / ed. by R. Kitamura, M. Kuwahara. – Boston: Springer, 2005. – P. 57–98. – (Operations Research/Computer Science Interfaces Series; vol. 31). – DOI: 10.1007/0-387-24109-4_3.

22. Li P., Kusari A., LeBlanc D.J. A Novel Traffic Simulation Framework for Testing Autonomous Vehicles Using SUMO and CARLA // SAE Technical Paper 2022-01-0110 (WCX SAE World Congress Experience). – 2022.

23. Ahmad M.U., Abrar A., Dasgupta S. et al. OpenCAMS: An Open-Source Connected and Automated Mobility Co-Simulation Platform for Advancing Next-Generation Intelligent Transportation Systems Research. – arXiv:2507.09186. – 2025. – DOI: 10.48550/arXiv.2507.09186.

24. Risma Carletti C.M., Casetti C., Härri J. et al. MS-VAN3T-CARLA: An Open-Source Co-Simulation Framework for Cooperative Perception Evaluation // 19th Wireless On-Demand Network Systems and Services Conference (WONS 2024). – Chamonix: IEEE, 2024. – P. 93–96. – DOI: 10.23919/WONS60642.2024.10449533.

25. Sergek будет фиксировать водителей, не пропускающих пешеходов, в Алматы [Электронный ресурс] // Министерство внутренних дел Республики Казахстан. – 2023. – 19 дек. – URL: https://www.gov.kz/memleket/entities/qriim/press/news/details/673554?lang=ru (дата обращения: 23.11.2026).

26. Alegre L.N. SUMO-RL [Электронный ресурс]. – GitHub, 2019. – URL: https://github.com/LucasAlegre/sumo-rl (дата обращения: 22.11.2026).

27. Webster F.V. Traffic Signal Settings. – London: HMSO, 1958. – (Road Research Technical Paper; No. 39).

28. Mnih V., Kavukcuoglu K., Silver D. et al. Human-level control through deep reinforcement learning // Nature. – 2015. – Vol. 518, No. 7540. – P. 529–533. – DOI: 10.1038/nature14236.

29. Raffin A., Hill A., Gleave A. et al. Stable-Baselines3: Reliable Reinforcement Learning Implementations // Journal of Machine Learning Research. – 2021. – Vol. 22, No. 268. – P. 1–8.