SCALABLE PARALLEL SIMULATION OF URBAN TRAFFIC WITH DIGITAL TWINS FOR ADAPTIVE SPEED CONTROL

Adrian V. OLTEANU[1*], Maximilian NICOLAE[2], Stefan MOCANU[3]

Urban traffic management requires rapid evaluation of control strategies, yet traditional sequential simulations are too slow for real-time decision support. This paper presents a scalable Digital Twin framework that leverages parallel microscopic simulations in SUMO to assess alternative speed settings in congested areas. The system compares throughput, delay, and simulation-to–real-time ratios, enabling optimal speed recommendations within strict time constraints. Results show that anticipatory simulations can run faster than real-time on commodity hardware, demonstrating the potential of parallel digital twins to provide timely, data-driven support for adaptive speed control in complex urban mobility systems.

Keywords: Digital Twin; Parallel Simulation; Adaptive Speed Control; Intelligent Driver Model (IDM); SUMO.

1. Introduction

The increasing complexity of urban mobility systems poses significant challenges for traffic management, especially in congested areas where small variations in vehicle speed can trigger large-scale effects on throughput and delay. Traditional simulation-based approaches have provided valuable insights into urban traffic dynamics, yet they are often limited by their sequential execution and inability to evaluate multiple scenarios within realistic time constraints [1,2]. As traffic management strategies become more data-driven and adaptive, there is a growing need for scalable methodologies capable of exploring alternative control options in parallel and delivering actionable results rapidly.

The Digital Twin (DT) concept offers a promising foundation for such capabilities. By maintaining a continuously updated virtual representation of the traffic system, a DT can integrate real-time data, simulate alternative strategies, and

[1*] PhD Student, Dep. Automation and Industrial Informatics, Faculty of Automation and Computers, National University of Science and Technology Politehnica Bucharest, Romania, corresponding author, e-mail: vasile.olteanu@upb.ro

[2] Assoc. prof., Dep. Automation and Industrial Informatics, Faculty of Automation and Computers, National University of Science and Technology Politehnica Bucharest, Romania, e-mail: max.nicolae@upb.ro

[3] Assoc. prof., Dep. Automation and Industrial Informatics, Faculty of Automation and Computers, National University of Science and Technology Politehnica Bucharest, Romania, e-mail: stefan.mocanu@upb.ro

assess their impact on mobility performance. While most DT research in transportation has focused on macroscopic control or centralized optimization [3,4], less attention has been given to the computational aspects of executing large numbers of simulations concurrently in order to anticipate and compare different control decisions.

This paper addresses this gap by proposing a scalable parallel simulation framework for adaptive speed control in congested urban traffic. The approach leverages parallel execution of microscopic simulations—using the Simulation of Urban Mobility (SUMO) platform—to evaluate alternative speed settings for platoons of vehicles in a given region of congestion. By systematically comparing throughput, average delay, and simulation-to-real-time ratios across scenarios, the framework enables the identification of optimal speed recommendations within strict time budgets.

The contribution of this work is twofold: (1) the design of an experimental setup for parallel, headless simulation of traffic scenarios with varying speed configurations; and (2) the evaluation of scalability and performance trade-offs when running multiple instances concurrently. The results demonstrate that anticipatory simulations can be executed faster than real-time under realistic hardware constraints, thus enabling Digital Twins to provide timely support for adaptive speed recommendations.

The remainder of this paper is organized as follows. Section 2 reviews related work on Digital Twins, parallel traffic simulation, and adaptive speed control. Section 3 presents the proposed methodology and simulation setup. Section 4 describes the experimental infrastructure and scenarios. Section 5 reports results on throughput, delay, and scalability. Finally, Section 6 concludes the paper and outlines future research directions.

2. Related work

Traffic management in congested urban environments has been widely studied through the lens of Intelligent Transportation Systems (ITS). Recent literature identifies six recurring principles in contemporary ITS implementations: adaptive real-time control, multi-objective optimization, prediction-based strategies, decentralized and distributed architectures, simulation-driven validation, and multimodal integration [5-13]. These principles highlight the need for dynamic traffic management systems capable of balancing multiple objectives such as throughput, emissions, and travel delays.

Prediction-based approaches have become increasingly prominent, with methods such as long short-term memory networks (LSTM), hybrid optimization, and graph neural networks (GNN) being applied to forecast short-term traffic fluctuations [8]. Decentralized, agent-based control has also been investigated, where vehicles, intersections, and even pedestrians are modeled as independent agents cooperating through graph optimization [9,10]. A key enabler of these frameworks is microscopic simulation, supported by platforms such as SUMO, AIMSUN, or VISSIM, which allow the testing of algorithms under controlled yet realistic traffic scenarios [11,12].

Digital Twins (DTs) have emerged as a complementary concep for traffic management, enabling the synchronization of real-world data with virtual simulations [14-16]. In urban mobility, DTs provide a bidirectional link that allows both monitoring and optimization. Applications include adaptive signal control, congestion reduction, emission evaluation, and incident detection. Projects such as NREL in the United States have demonstrated significant delay reductions by integrating DTs with high-performance computing [16], while European initiatives such as DT-GM highlight the feasibility of dynamic synchronization of microscopic models with live traffic streams [17,18].

Despite these advances, several limitations remain. Current DT implementations often face the trade-off between scalability and model granularity [19], data gaps due to heterogeneous sensor coverage [20], and constraints related to privacy and governance [21]. Furthermore, while DTs have been leveraged for visualization and predictive analytics, their integration into closed-loop decision support with fast, parallel execution remains limited [22]. This motivates the approach presented in this paper, which focuses on scalable, parallel simulations as a means to identify optimal speed configurations for adaptive control under congestion.

3. Methodology

3.1. Framework Overview

The proposed methodology integrates the Digital Twin concept with scalable traffic simulation, enabling parallel execution of multiple speed scenarios in congested urban areas. The Digital Twin is employed as a computational framework that continuously synchronizes with the simulated traffic state and anticipates alternative control options (Fig. 1). The approach is designed to address two challenges:

1. *Latency of decision-making* – mitigated by simulating multiple speed options on various time horizons in advance for a selected region of congestion (ROC).
2. *Scalability of computation* – achieved through parallel execution of independent traffic simulations, allowing results to be delivered within strict time budgets.

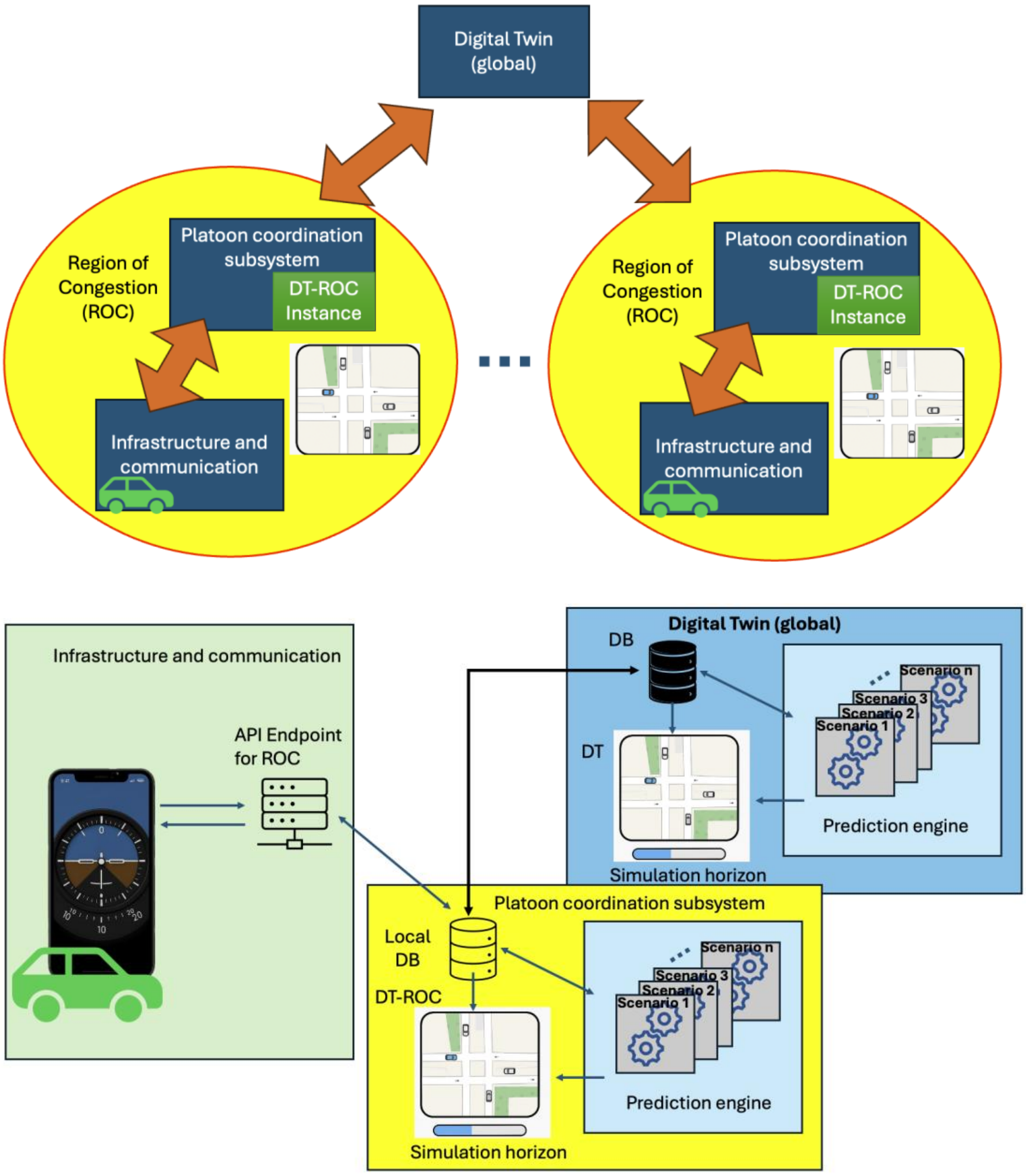

Fig. 1. Architecture of the Digital Twin framework showing bidirectional data flow: vehicle telemetry and environmental data sent to the DT, and speed recommendations transmitted back to drivers via mobile interface.

3.2. Simulation Platform and Tools

Microscopic simulation was selected due to its ability to capture vehicle-level dynamics, particularly the effects of speed variations on platooning and stopand- go waves. The SUMO (Simulation of Urban Mobility) platform was used as the core simulation engine, with the following settings:

- Networks generated from OpenStreetMap (OSM) data via *netconvert*.
- Traffic demand defined through flows using the Intelligent Driver Model (IDM).
- Maximum vehicle speed as the main experimental variable, ranging from 15 km/h to 60 km/h with an increment of 5 km/h.
- Independent SUMO processes executed headless for efficiency.
- Simulation orchestration managed via Python scripts, with automatic collection of trip info and summary outputs.

3.3 Vehicle Dynamics: Intelligent Driver Model (IDM)

The IDM is a well-established car-following model widely used in microscopic traffic simulations [23][24]. It determines a vehicle's acceleration as a function of its own speed, the relative speed, and the headway to the vehicle in front. The acceleration is given by:

$$\frac{\mathbf{dv}}{\mathbf{dt}} = \mathbf{a}\left[\mathbf{1} - \left(\frac{\mathbf{v}}{\mathbf{v_0}}\right)^{\boldsymbol{\delta}} - \left(\frac{\mathbf{s}^*(\mathbf{v}, \boldsymbol{\Delta}\mathbf{v})}{\mathbf{s}}\right)^{\mathbf{2}}\right]$$

where:

- v = current speed of the vehicle,
- v_0= desired speed,
- a = maximum acceleration,
- δ = acceleration exponent (commonly 4),
- s = actual gap to the leading vehicle,
- Δv = relative speed to the leading vehicle ($\mathrm{v} - \mathrm{v}_{\mathrm{leader}}$),
- $s^*(v, \Delta v)$ = desired dynamic gap, defined as:

$$s^*(v, \Delta v) = s_0 + v \cdot T + \frac{v \cdot \Delta v}{2\sqrt{ab}}$$

with:

- s_0 = minimum distance (jam distance),
- T = safe time headway (τ),
- b = comfortable deceleration.

This formulation allows the IDM to realistically capture both free-flow acceleration towards the desired speed and braking behavior under congestion or sudden changes in traffic flow.

3.4 Parallel Execution Strategy

The core novelty of the methodology lies in its parallel execution of simulations and the way this is integrated. Instead of testing speed configurations sequentially, the Digital Twin launches multiple SUMO instances simultaneously, each configured with a different maximum speed parameter.

- *Scenario replication*: identical network and demand across simulations.
- *Speed variation*: each instance explores a distinct maximum speed value.
- *Parallel processing*: simulations executed concurrently on separate CPU cores.
- *Time measurement*: both simulated time and wall-clock execution time recorded for performance analysis.

3.5 Performance Metrics

Two categories of metrics are evaluated:

- *Traffic performance metrics*
  - *Throughput* (vehicles completed per simulation horizon).
  - *Average travel time* (mean trip duration).
  - *Average waiting time* (time spent in queues).

2. *Computational performance metrics*
  - *Wall-clock time* (real execution duration).
  - *Simulation-to-real ratio* (ratio of simulated time vs. real execution time, values >1 indicate faster-than-real-time performance).

The optimal speed configuration is selected by minimizing average travel delay while maximizing throughput.

3.6 Workflow Summary

The complete workflow can be summarized in six steps:

1. Define the Region of Congestion (ROC) and extract its network topology.
2. Configure baseline traffic flows and vehicle behavior using IDM.
3. Launch N parallel SUMO simulations, each with a distinct recommended speed.
4. Collect simulation outputs (*tripinfo*, *summary*, *edgedata*).
5. Compute traffic and computational performance metrics.
6. Select the optimal speed configuration and report results.

4. Experimental Setup

4.1 Region of Congestion (ROC) and Network Derivation

We target a single urban ROC extracted from OpenStreetMap (OSM) (Fig. 2) and converted to SUMO using *netconvert* (Fig. 3). Two network variants were prepared to compare preprocessing effects: a basic conversion and a cleaned conversion (geometry simplification and junction merging).

- Conversion (basic):

netconvert --osm-files eval_apaca/apaca.osm -o \ eval_apaca/apaca_basic.net.xml

- Conversion (cleaned):

*netconvert --osm-files eval_apaca/apaca.osm -o \ eval_apaca/apaca.net.xml *

- *geometry.remove --junctions.join *

remove-edges.by-vclass rail_urban,rail

The cleaned network is used in all reported results unless stated otherwise.

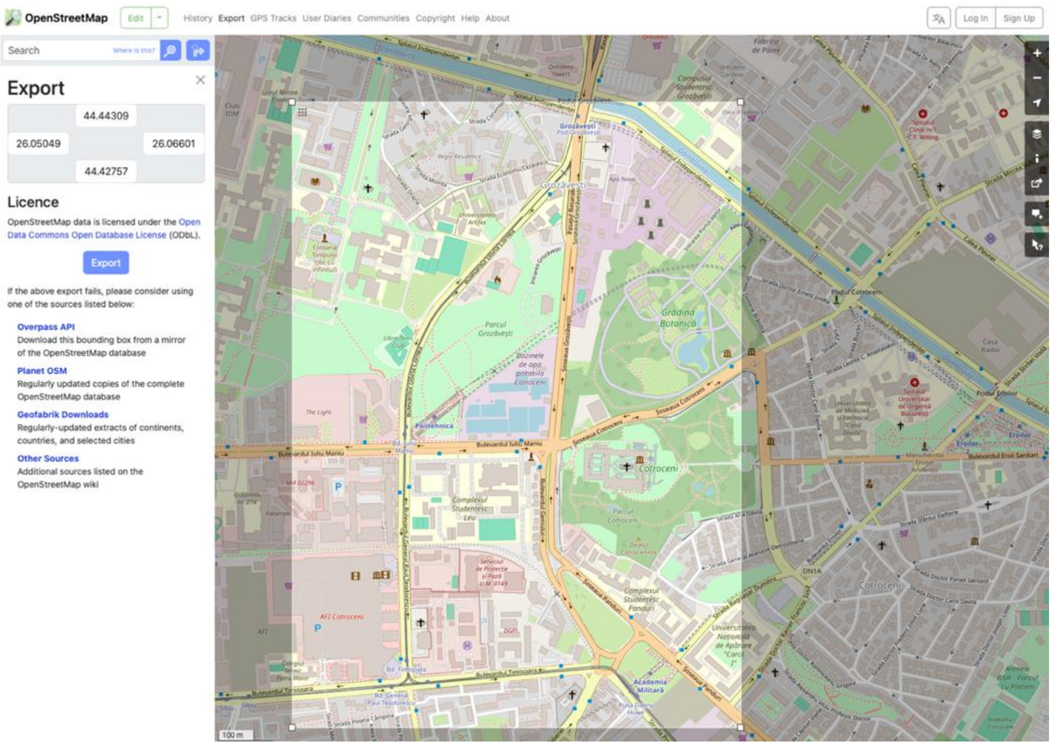

Fig. 2. OSM extraction: rectangular bounding box around the chosen intersection

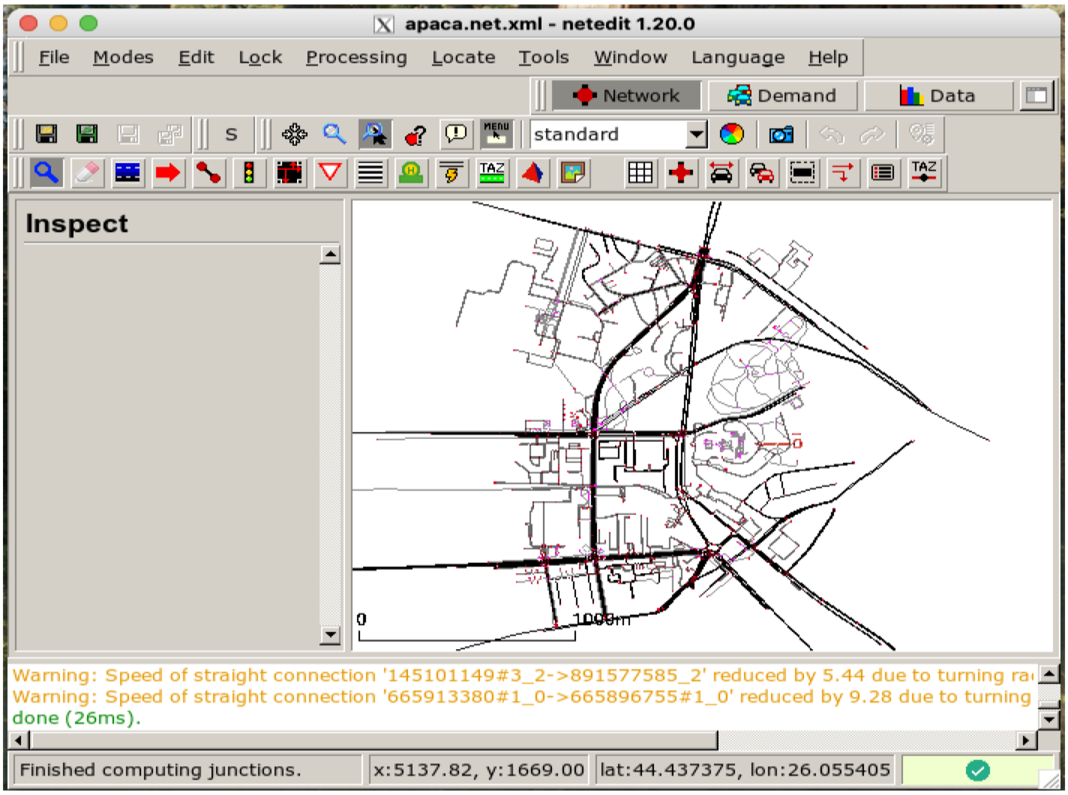

Fig. 3. Simulated network as result of netconvert from OSM map extract

4.2 Traffic Demand and Driver Model (IDM)

Traffic demand is defined through flows over selected *from–to* edges that traverse the ROC. The Intelligent Driver Model (IDM) governs car-following dynamics (parameters consistent with [25–29]:

- Base *vType* (IDM):

```
<vType id="idmCar" vClass="passenger"
        accel="2.6" decel="4.5" sigma="0.5"
        length="5.0" maxSpeed="55"
        carFollowModel="IDM"
        tau="1.2" minGap="2.5" />
```

- Experimental variable: v_0 = *maxSpeed*.
- Optional platoon stress-test (not in main results): τ = 0.3–1.2 s (τ = T - see Section 3.3).
- *minGap* is important for cooperative following but our scenario involves human drivers, so it was kept larger than usual for autonomous vehicles traveling in platoons.

The reported scenarios in this article address the *dynamic flows* as continuous injection using <flow> with *vehsPerHour* ∈ [600, 2000] distributed over 300–3600 s, matched across all speed scenarios. Our interest was to create a demand that resembles the daily traffic jams we are witnessing on our way to work (Fig. 4).

4.3 Parallel Execution Environment

We execute N parallel, headless SUMO instances, each testing a distinct maximum speed. Parallelism matches physical cores to avoid contention. The computing infrastructure consisted of a MacBook Pro with Apple M2 Max processor and 32 GB RAM. All runs were headless (*sumo* executable, not *sumo-gui*) with logging minimized to reduce I/O overhead. Simulations were executed in headless mode to allow multiple parallel instances without GUI overhead, ensuring deterministic performance measurement across identical scenarios. Each simulation scenario assumes a uniform target speed for all vehicles, representing the effect of speed recommendations rather than natural inter-driver variability. The intersection was modeled without signal control to isolate the influence of speed on throughput

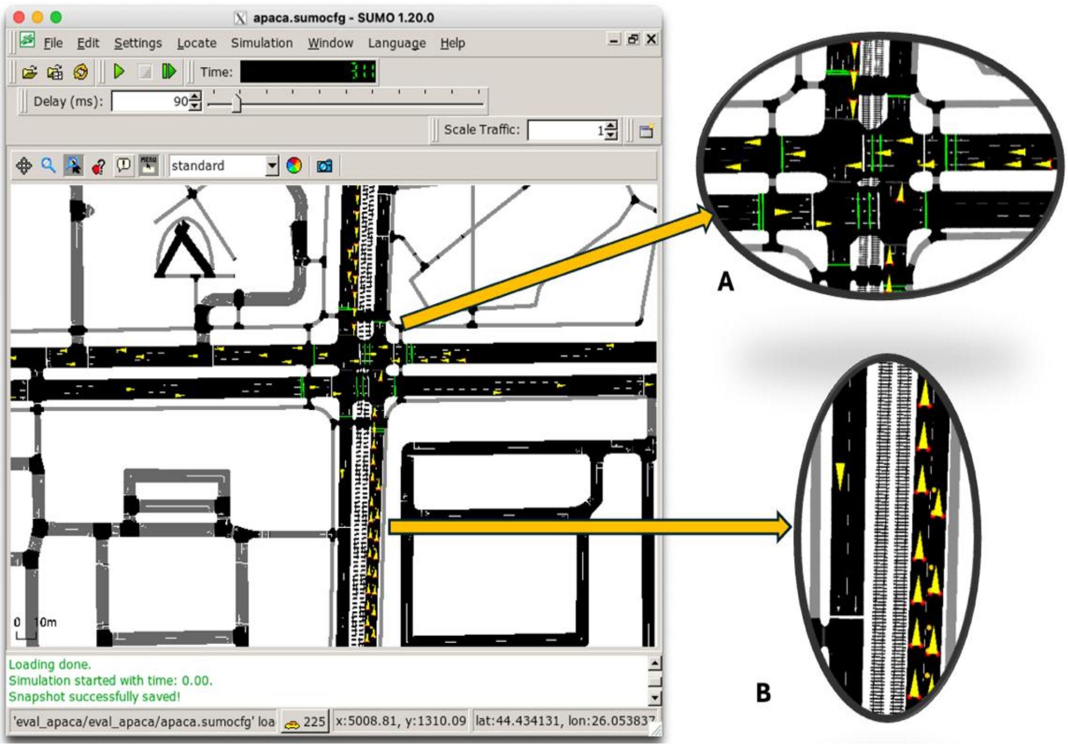

Fig. 4. Visual simulation (*sumo-gui*) for a junction in the ROC illustrating shared access of the junction without semaphores (A) and the accumulation of queues (B)

4.4 Instrumentation and Outputs

Each SUMO run emits:

- *Trip metrics*: --tripinfo-output tripinfo_<tag>.xml → per-vehicle duration, waitingTime.
- *Global summary*: --summary-output summary_<tag>.xml → per-step running, ended.
- *Optional edges*: --edgedata-output edgedata_<tag>.xml (used for queue diagnostics, not central to reported plots).

A Python orchestration script (batch runner) clones the base routes, sets maxSpeed per run, launches processes in parallel, and timestamps wall-clock execution:

- *Per-run timers*: perf_counter() around each sumo call.
- *Batch timer*: global wall-clock around the thread pool.
- *Collected KPIs*: ended (throughput), average duration, average waitingTime, simulated horizon (end - begin), wall-clock, and sim/real ratio.

Outputs are written per speed tag (e.g., runs_speed_sweep/40kmh/…) and aggregated into a CSV (speed_sweep_summary.csv) for farther analyze.

All experiments were conducted under realistic hardware constraints, referring to typical mid-range computing hardware (Intel Core i5 CPU, 8GB RAM), to reflect performance achievable in practical edge-based DT deployments

4.5 Scenario Factor: Speed Sweep

To evaluate how speed limits affect congestion in the ROC, we sweep 10 speed configurations:

- *Speed set (km/h)*: 15, 20, 25, 30, 35, 40, 45, 50, 55, 60.
- *Mapping to SUMO*: maxSpeed per <vType> (m/s).
- *Simulation horizon*: begin = 0 s, end = 600 s (unless otherwise stated).
- *Constancy across runs*: identical network, flows, and IDM parameters; only maxSpeed differs.

This design isolates the effect of speed on *throughput* and *delays*, while enabling parallel evaluation within a fixed time budget.

4.6 Evaluation Metrics

We report two categories of metrics, consistent with practice in microscopic traffic studies [11, 30]:

- Traffic performance
  - Throughput: number of vehicles with completed trips (ended).
  - Average travel time: mean duration over completed trips.
  - Average waiting time: mean waitingTime over completed trips (proxy for queueing).
- Computational performance
  - Wall-clock time: per-run elapsed time.

- Simulation-to-real ratio: (simulated horizon) / (wall-clock); values > 1 indicate faster-than-real-time.

Speed recommendations are selected by minimizing average travel time (delay) while keeping throughput high. Where trade-offs exist, we report Pareto points and discuss sensitivity.

5. Results and Discussion

5.1 Throughput Patterns Across Speed Sweeps

Tables 1–5 summarize the throughput, average travel time, waiting time, and computational performance for different target speeds under varying simulation horizons and headway values (τ).

Table 1

Speed sweep summary τ =1.2, Simulation horizon - 3600s
(Total batch simulation time - 20.484 s)

Target speed [Km/h]	Throughput [no of vehicles]	Average travel time [s]	Average waiting time [s]	Instance simulation time [s]	Simulation-to-real ratio
15	237	295.544	16.211	19.241	187.104
20	1862	431.954	184.708	12.954	277.906
25	295	209.851	26.925	20.477	175.806
30	186	151.957	7.543	20.108	179.036
35	1892	368.808	188.098	11.608	310.139
40	1895	365.448	191.833	11.623	309.722
45	1880	374.03	205.941	11.662	308.705
50	426	118.434	21.049	19.11	188.379
55	1906	352.493	188.93	11.617	309.893
60	1895	355.551	191.436	11.534	312.126

Table 2

Speed sweep summary τ =1.2, Simulation horizon - 600s
(Total batch simulation time - 2.846 s)

Target speed [Km/h]	Throughput [no of vehicles]	Average travel time [s]	Average waiting time [s]	Instance simulation time [s]	Simulation-to-real ratio
15	203	285.764	17.877	2.731	219.737
20	225	248.827	30.973	2.842	211.137
25	276	208.446	27.797	2.73	219.796
30	186	151.957	7.543	2.735	219.388
35	342	139.538	17.518	2.441	245.846
40	352	122.29	14.878	2.387	251.388
45	311	134.666	30.437	2.496	240.38
50	337	113.582	17.039	2.407	249.256

55	360	108.833	16.278	2.338	256.591
60	366	108.276	18.175	2.331	257.353

Table 3

Speed sweep summary τ =0.5, Simulation horizon - 600s
(Total batch simulation time - 2.804 s)

Target speed [Km/h]	Throughput [no of vehicles]	Average travel time [s]	Average waiting time [s]	Instance simulation time [s]	Simulation-to-real ratio
15	223	283.565	26.906	2.724	220.291
20	257	223.416	23.374	2.466	243.3
25	224	176.991	11.92	2.586	232.015
30	328	156.387	19.942	2.352	255.154
35	225	128.484	10.289	2.442	245.729
40	331	153.091	41.26	2.49	240.929
45	357	114.269	19.51	2.296	261.324
50	43	89.721	2.488	2.793	214.798
55	329	97.219	13.635	2.251	266.51
60	346	116.338	31.156	2.271	264.253

Table 4

Speed sweep summary τ =0.8, Simulation horizon - 600s
(Total batch simulation time – 2.715 s)

Target speed [Km/h]	Throughput [no of vehicles]	Average travel time [s]	Average waiting time [s]	Instance simulation time [s]	Simulation-to-real ratio
15	212	281.307	23.118	2.713	221.14
20	282	214.089	16.766	2.554	234.957
25	307	211.029	44.407	2.527	237.462
30	330	148.764	13.327	2.393	250.75
35	335	143.809	23.006	2.341	256.266
40	353	118.034	13.788	2.216	270.716
45	355	115.073	18.924	2.335	257.002
50	364	122.118	25.632	2.366	253.599
55	327	116.052	24.511	2.284	262.684
60	330	109.979	22.809	2.259	265.561

Table 5

Speed sweep summary τ =0.8, Simulation horizon - 3600s
(Total batch simulation time - 19.*266* s)

Target speed [Km/h]	Throughput [no of vehicles]	Average travel time [s]	Average waiting time [s]	Instance simulation time [s]	Simulation-to-real ratio
15	263	298.7	21.517	19.263	186.884
20	2141	354.44	136.421	12.267	293.47
25	2202	333.839	146.747	11.673	308.413
30	2214	317.868	153.91	11.468	313.915
35	582	155.084	32.828	16.612	216.714
40	2236	291.394	152.78	11.102	324.26
45	2237	299.827	170.502	10.871	331.153
50	2224	299.742	170.933	11.098	324.383
55	429	131.114	32.872	19.132	188.167
60	2263	283.961	156.688	11.076	325.017

A consistent observation is that throughput increases with target speed up to a point, then stabilizes or fluctuates. For example, in Table 1 (τ=1.2, 3600s), throughput stabilizes between 1892–1906 vehicles in the 35–55 km/h range, with minor variations. In contrast, extreme values (15 km/h or 25–30 km/h) result in significantly lower throughput, revealing the nonlinear effects of speed control on congestion.

When the horizon is reduced to 600s (Table 2), throughput levels are naturally lower, yet the same pattern emerges: speeds of 35–60 km/h provide stable capacity, while lower speeds degrade performance. The 600 s horizon was selected not only as a representative short-term control window, but also as a computationally equivalent substitute for more complex real-world scenarios. The extended simulated duration compensates for the relative simplicity of the modeled intersection, ensuring that computational effort per simulation remains realistic. This choice also demonstrates that the observed acceleration of the algorithm (simulation-to-real-time ratio > 200) would persist—or improve—for shorter real-world time frames.

At reduced headways (τ=0.5, Table 3), throughput increases beyond 300 vehicles even at moderate speeds (30–40 km/h), reflecting the platooning effect where vehicles can safely follow more closely. Similarly, with τ=0.8 (Tables 4–5), throughput stabilizes around 330–360 vehicles for 35–60 km/h under short horizons, and above 2200 vehicles for long horizons (3600s). These results confirm that optimal throughput is achieved at intermediate target speeds (35–50 km/h) and lower headways (τ < 1.0), supporting the hypothesis that predictive speed recommendations can enhance capacity in congested regions. This behavior is consistent with previous observations of fundamental diagrams of traffic flow, where throughput saturates once capacity is reached [11].

5.2 Travel Time and Waiting Time Trade-offs

While throughput reflects system capacity, average travel time and waiting time provide insight into user-level experience.

- In Table 1 (τ=1.2, 3600s), average travel times exceed 350s for 35–55 km/h, but are accompanied by waiting times above 180s, indicating persistent queues.
- In short horizon runs (Table 2), waiting times drop significantly (~15–30s) at 35–55 km/h, showing that congestion can be reduced over smaller temporal windows.
- With τ=0.5 (Table 3), waiting times are minimized (~10–20s) at speeds of 25–35 km/h, but extreme fluctuations occur at 40–60 km/h (up to 41s), highlighting instability when vehicles follow too closely under higher speed regimes.
- With τ=0.8 (Table 4–5), waiting times remain in a stable band (13–25s for short horizons, ~150–170s for long horizons), confirming robustness of intermediate speeds under more realistic driver reactions.

There exists a U-shaped curve between speed and delay: both very low and very high speeds induce longer waiting times, while intermediate values minimize delays.

This highlights the importance of speed recommendations not as static limits, but as adaptive values that consider congestion levels. By running simulations in parallel, the Digital Twin can dynamically select the configuration that minimizes delay under current conditions.

5.3 Computational Performance

A crucial contribution of this work is demonstrating that such evaluations can be performed faster than real-time.

- For the 3600s horizon (Tables 1 and 5), each instance required ~11–20 seconds wall-clock, yielding simulation-to-real ratios above 180 and up to 331.
- For the 600s horizon (Tables 2–4), runtimes per instance were ~2.3–2.7s, yielding ratios consistently above 210 and peaking at 266.

These results show that even on modest hardware, parallel execution of 10 simulations can explore multiple scenarios within a fraction of real-time. The entire batch of 10 runs completed in ~20s (for 3600s horizons) and <3s (for 600s horizons), confirming near-linear scalability.

The computational feasibility of this framework allows the Digital Twin to act as a real-time anticipator, testing candidate strategies before decisions need to be applied in the physical system.

5.4 Synthesis of Findings

The results across Tables 1–5 suggest that optimal operation does not hinge on a single target speed, but rather on a robust neighborhood of speeds, typically in the 35–50 km/h range. Within this interval, throughput remains consistently high and average delays are minimized, even under varying headways and demand levels. This indicates that the system can tolerate moderate deviations without significant performance degradation, which is particularly important in real-world conditions where exact speed compliance is unlikely.

Another key observation is that similar throughput values can be achieved at lower speeds compared to higher speeds. For instance, in Tables 2 and 4 (600s horizon), speeds of 35–40 km/h yield throughput levels comparable to 55–60 km/h, yet with reduced volatility in waiting times. This finding is critical: it suggests that lower, safer speeds may deliver nearly the same capacity as higher speeds, while enhancing traffic safety and predictability. In practice, such robust lower-speed regimes could allow more effective sharing of intersection resources, operating in a manner similar to roundabouts where continuous but moderated flows prevent gridlock and minimize severe conflicts.

Finally, the computational performance of the framework demonstrates that complex predictive simulations over long horizons (up to 3600 s) can be executed within a few seconds of wall-clock time. This confirms the feasibility of using the Digital Twin not only for monitoring, but also for anticipatory updates of recommended speeds in real-time. By running parallel simulations, the system can identify robust speed ranges that balance throughput, safety, and computational efficiency, and update recommendations adaptively as conditions evolve.

While the experiments were conducted on a particular ROC, the methodology generalizes to larger networks and more complex flows. Limitations include the reliance on simulation fidelity (IDM assumptions) and the absence of real-world sensor noise or behavioral variability. These will be addressed in future extensions.

6. Conclusions

This paper introduced a scalable parallel simulation framework for adaptive speed control in congested urban traffic, leveraging the Digital Twin concept, defined as a continuously synchronized digital model of a physical system, as the computational backbone of the framework. By running multiple SUMO instances in parallel with varying maximum speeds, the system can identify the configuration that optimizes throughput and minimizes delay within operational time constraints.

The main contributions can be summarized as:

- Integration of Digital Twin principles with microscopic traffic simulation for anticipatory decision support.

- A parallel execution strategy enabling near real-time evaluation of multiple speed scenarios.
- Validation of the framework through a speed sweep experiment, highlighting the balance point as optimal for congestion mitigation.

Future work will focus on:

- Extending the framework to larger and more heterogeneous networks, including multi-intersection corridors.
- Incorporating stochastic driver behavior and sensor uncertainty for more realistic validation.
- Exploring integration with real-world traffic data streams, including mobile sensing and roadside units.
- Benchmarking the framework on more powerful and distributed infrastructures, such as cloud or edge clusters, to assess scalability beyond a single machine.

Ultimately, the results demonstrate that parallel Digital Twin simulations can act as a practical enabler for adaptive speed recommendations, bridging the gap between computational feasibility and real-time traffic management needs.

R E F E R E N C E S

[1] Ge, C.; Qin, S. Digital Twin Intelligent Transportation System (DT-ITS)—A Systematic Review. IET Intelligent Transport Systems 2024, 18, 2325–2358, doi:10.1049/ITR2.12539.

[2] Dasgupta, S.; Rahman, M.; Professor, A.; Lidbe, A.D.; Research Engineer, A.; Lu, W.; Jones, S.; Cudworth Professor, J.R. A Transportation Digital-Twin Approach for Adaptive Traffic Control Systems. 2021.

[3] Wu, K.; Li, P.; Cheng, Y.; Parker, S.T.; Ran, B.; Noyce, D.A.; Ye, X. A Digital Twin Framework for Physical-Virtual Integration in V2X-Enabled Connected Vehicle Corridors. IEEE Transactions on Intelligent Transportation Systems 2024, doi:10.1109/TITS.2025.3564351.

[4] Wang, Z.; Han, K.; Tiwari, P. Digital Twin-Assisted Cooperative Driving at Non-Signalized Intersections. IEEE Transactions on Intelligent Vehicles 2021, 7, 198–209, doi:10.1109/TIV.2021.3100465.

[5] Meess, H.; Gerner, J.; Hein, D.; Schmidtner, S.; Elger, G.; Bogenberger, K. First Steps towards Real-World Traffic Signal Control Optimisation by Reinforcement Learning. Journal of Simulation 2024, 18, 957–972, doi:10.1080/17477778.2024.2364715;WEBSITE:WEBSITE:TFOPB;PAGEGROUP:STRING:PUBLICATION.

[6] Mirbakhsh, S.; Azizi, M. Adaptive Traffic Signal's Safety and Efficiency Improvement by Multi-Objective Deep Reinforcement Learning Approach. International Journal of Innovative Research in Multidisciplinary Education 2024, 03, doi:10.58806/ijirme.2024.v3i7n10.

[7] Ulvi, H.; Yerlikaya, M.A.; Yildiz, K. An Innovative Mathematical Model for Integrated Traffic Flow Optimization in Organized Industrial Zones. Sci Rep 2025, 15, 1–31, doi:10.1038/S41598-025-14761-Y;SUBJMETA=1041,1046,531,639,705;KWRD=APPLIED+MATHEMATICS,SCIENTIFIC+DATA,STATISTICS.

[8] Hammoumi, L.; Farah, S.; Benayad, M.; Maanan, M.; Rhinane, H. Leveraging Machine Learning to Predict Traffic Jams: Case Study of Casablanca, Morocco. Journal of Urban Management 2025, 14, 813–826, doi:10.1016/J.JUM.2025.02.004.
[9] Flormann, M.; Meyer, F.; Henze, R. Applying Real World Traffic Data on Machine Learning Methods for Graph Based Cooperation Strategies of Automated Vehicles. SAE Technical Papers 2025, doi:10.4271/2025-01-0286.
[10] Talluri, K. K.; Stang, C.; Weidl, G. Green Wave as an Integral Part for the Optimization of Traffic Efficiency and Safety: A Survey. arXiv preprint arXiv:2507.22511, July 2025. DOI:10.48550/arXiv.2507.22511
[11] Dobrilko, O.; Bublil, A. Leveraging SUMO for Real-World Traffic Optimization: A Comprehensive Approach. SUMO User Conference Proceedings, Vol. 5, pp. 179-194, July 2024. DOI: 10.52825/scp.v5i.1120
[12] Singh, A.; Lin, C.-Y.; Huang, C.-I.; Lin, F.-P. Quantum Annealing Approach for the Optimal Real-Time Traffic Control Using QUBO. Proceedings - 22nd IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD 2021-Fall 2024, 74–81, doi:10.1109/SNPD51163.2021.9704985.
[13] Poudel, B.; Wang, X.; Li, W.; Zhu, L.; Heaslip, K. Joint Pedestrian and Vehicle Traffic Optimization in Urban Environments using Reinforcement Learning. arXiv preprint arXiv:2504.05018 (2025). DOI: 10.48550/arXiv.2504.05018
[14] Wang, Y.; Wang, H.; Wang, W.; Song, S.; Fu, X. Architecture, Application, and Prospect of Digital Twin for Highway Infrastructure. Journal of Traffic and Transportation Engineering (English Edition) 2024, 11, 835–852, doi:10.1016/J.JTTE.2024.03.003.
[15] Almeaibed, S.; Al-Rubaye, S.; Tsourdos, A.; Avdelidis, N.P. Digital Twin Analysis to Promote Safety and Security in Autonomous Vehicles. IEEE Communications Standards Magazine 2021, 5, 40–46, doi:10.1109/MCOMSTD.011.2100004.
[16] Digital-Twin Project Green-Lights Traffic Congestion Improvements | NREL Available online: https://www.nrel.gov/news/detail/program/2023/digital-twin-project-green-lights-traffic-congestion-improvements (accessed on 10 September 2025).
[17] In NYC, Digital Twin Project Tackles Traffic | ASCE Available online: https://www.asce.org/publications-and-news/civil-engineering-source/civil-engineering-magazine/article/2022/09/in-nyc-digital-twin-project-tackles-traffic (accessed on 10 September 2025).
[18] Digital Twin Available online: http://128.122.136.17/DigitalTwin/ (accessed on 10 September 2025).
[19] Kušić, K.; Schumann, R.; Ivanjko, E. A Digital Twin in Transportation: Real-Time Synergy of Traffic Data Streams and Simulation for Virtualizing Motorway Dynamics. Advanced Engineering Informatics 2023, 55, 101858, doi:10.1016/J.AEI.2022.101858.
[20] Yan, Y.; Ni, L.; Sun, L.; Wang, Y.; Zhou, J. Digital Twin Enabling Technologies for Advancing Road Engineering and Lifecycle Applications. Engineering 2025, 44, 184–206, doi:10.1016/J.ENG.2024.12.017.
[21] Nag, D.; Brandel-Tanis, F.; Pramestri, Z.A.; Pitera, K.; Frøyen, Y.K. Exploring Digital Twins for Transport Planning: A Review. European Transport Research Review 2025, 17, doi:10.1186/S12544-025-00713-0.
[22] Cantisani, G.; Serrone, G. Del; Afshari, A.; Lee, J.; Besenski, D. A Digital Twin Platform for Real-Time Intersection Traffic Monitoring, Performance Evaluation, and Calibration. Infrastructures 2025, Vol. 10, Page 204 2025, 10, 204, doi:10.3390/INFRASTRUCTURES10080204.
[23] Treiber, M.; Kesting, A. Traffic Flow Dynamics: Data, Models and Simulation; Springer Berlin Heidelberg, 2013; ISBN 9783642324604

[24] Treiber, M.; Hennecke, A.; Helbing, D. Congested Traffic States in Empirical Observations and Microscopic Simulations. Phys Rev E Stat Phys Plasmas Fluids Relat Interdiscip Topics 2000, 62, 1805–1824, doi:10.1103/PhysRevE.62.1805.
[25] Meess, H.; Gerner, J.; Hein, D.; Schmidtner, S.; Elger, G.; Bogenberger, K. First Steps towards Real-World Traffic Signal Control Optimisation by Reinforcement Learning. Journal of Simulation 2024, 18, 957–972, doi:10.1080/17477778.2024.2364715;WEBSITE:WEBSITE:TFOPB;PAGEGROUP:STRING:PUBLICATION.
[26] Mirbakhsh, S.; Azizi, M. Adaptive Traffic Signal's Safety and Efficiency Improvement by Multi-Objective Deep Reinforcement Learning Approach. International Journal of Innovative Research in Multidisciplinary Education 2024, 03, doi:10.58806/ijirme.2024.v3i7n10.
[27] Ulvi, H.; Yerlikaya, M.A.; Yildiz, K. An Innovative Mathematical Model for Integrated Traffic Flow Optimization in Organized Industrial Zones. Sci Rep 2025, 15, 1–31, doi:10.1038/S41598-025-14761-Y;SUBJMETA=1041,1046,531,639,705;KWRD=APPLIED+MATHEMATICS,SCIENTIFIC+DATA,STATISTICS.
[28] Hammoumi, L.; Farah, S.; Benayad, M.; Maanan, M.; Rhinane, H. Leveraging Machine Learning to Predict Traffic Jams: Case Study of Casablanca, Morocco. Journal of Urban Management 2025, 14, 813–826, doi:10.1016/J.JUM.2025.02.004.
[29] Flormann, M.; Meyer, F.; Henze, R. Applying Real World Traffic Data on Machine Learning Methods for Graph Based Cooperation Strategies of Automated Vehicles. SAE Technical Papers 2025, doi:10.4271/2025-01-0286.
[30] Singh, A.; Lin, C.-Y.; Huang, C.-I.; Lin, F.-P. Quantum Annealing Approach for the Optimal Real-Time Traffic Control Using QUBO. Proceedings - 22nd IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD 2021-Fall 2024, 74–81, doi:10.1109/SNPD51163.2021.9704985.