SUMO User Conference 2026
TIBconferencesession
TIB-OP_will_set_DOI

ATSUM: An `Attlas`-SUMO framework for Traffic Signal Optimization and Controller-in-the-Loop Simulation

Rodolfo V. Valentim[1], Tiago Pinheiro[1], Vincen Santaella[1], Artur Souza[1],
João P. M. Clevelares[1], Daniel E. Garcia A.[1], Fernando Martinelli[1], Víctor M. G. Martínez[1]

[1]Atman Systems

*Correspondence: Rodolfo V. Valentim, rodolfo.valentim@atmansystems.com

Abstract. Microscopic traffic simulation is widely used to evaluate traffic signal control strategies. However, integrating operational traffic management platforms into simulation environments typically requires manual translation of network topologies, signal settings, and timing plans, which is time-consuming and error-prone. This process limits the rate of iteration and makes it more difficult to evaluate alternative control strategies. For this reason, this work presents `ATSUM`, a framework that connects an industrial traffic management and control platform, such as `Attlas`, with the microscopic simulator SUMO (Simulation of Urban Mobility) to support offline planning and reproducible simulation-based evaluation. The framework provides a deterministic pipeline that converts platform-level network and control descriptions into simulation-ready assets, ensuring consistency between planning data and simulation inputs. In addition, detector measurements and network semantics are combined to estimate traffic demand and derive coordination inputs, and the resulting optimization outputs are automatically translated into executable signal programs. The approach was demonstrated through a case study of a representative urban corridor. The results highlight the framework's ability to capture nuanced operational trade-offs, such as prioritizing travel time reductions versus minimizing emissions, enabling a systematic comparison between baseline and optimized plans under a controlled, reproducible setup. Overall, this work introduces an end-to-end framework that integrates traffic management data with a microscopic simulator, providing a unified framework for evaluating traffic signal control.

Keywords: SUMO, Traffic Signal Control, Traffic Simulation, Optimization, Controller-in-the-loop.

1. Introduction

Microscopic traffic simulation is a core tool for evaluating and designing traffic signal control strategies, supporting both offline planning and the validation of adaptive control algorithms before field deployment [1]. In this context, SUMO (Simulation of Urban Mobility) is widely adopted due to its open architecture, high extensibility, and detailed support for traffic signal logic and detector modeling [2]. Moreover, its interfaces enable closed-loop runtime coupling with external controllers, while explicit seed control supports deterministic, reproducible benchmarking across control experiments.

However, in practical traffic engineering workflows, integrating signal planning and control management systems with SUMO typically requires manual translation of network topology, signal configurations, and timing plans [3]. This process involves converting controller definitions, movement groups, and detector layouts into SUMO-specific representations, thereby introducing an additional layer of complexity between planning tools and simulation environments. As reported in prior work, this translation is time-consuming, error-prone, and it slows iterative optimization, especially when real-time or coordinated control algorithms are involved [4]. Additionally, inconsistent mappings between operational data and simulation artifacts can introduce discrepancies in phase structure, detector placement, or offset coordination, thereby reducing the reliability of simulation results as a proxy for field behavior [5].

Consequently, the lack of an integrated workflow between signal planning platforms and microscopic simulators limits the systematic evaluation of alternative timing plans and hinders reproducible experimentation across scenarios and networks [3]. Several approaches in the literature attempt to bridge this gap by coupling optimization models with simulation environments or by interfacing traffic controllers through middleware layers [3], [5]. While these solutions demonstrate the feasibility of simulation-based validation, they often rely on ad hoc network conversions, manual signal mapping, or scenario-specific scripts, which limit scalability, portability, and reproducibility. Commercial traffic engineering tools provide tighter integration between planning and simulation, but they are typically closed, inflexible with respect to custom optimization models, and difficult to embed in reproducible research workflows.

Although SUMO provides low-level control interfaces through TraCI, it does not natively offer an end-to-end workflow that connects planning data, optimization outputs, and operational signal abstractions in a consistent manner [6]. As a result, validating both fixed-time plans and coordinated strategies within a reproducible pipeline remains challenging, particularly when multiple iterations and comparative evaluations are required.

In this context, this work introduces `ATSUM` (Attlas–SUMO), a framework that integrates Atman Systems' commercial adaptive traffic control software `Attlas`[1] with the microscopic traffic simulator SUMO. `ATSUM` supports deterministic topology translation, automated generation of simulation assets, demand estimation, and coordination optimization. It further formalizes the simulation lifecycle by persisting all inputs, outputs, and metadata in versioned run directories, enabling repeatable experiments and traceable comparisons between alternative signal plans [7].

Overall, the main contributions of this work are:

- A deterministic translation pipeline that converts platform-level network definitions into SUMO networks, detectors, routes, and traffic light logic.
- A demand estimation layer that combines detector data with network semantics to compute movement-level flows.

---

[1] https://attlas.atmansystems.com/

- Integration of coordination optimizers (MAXBAND [8], MultiBand [9], and PM-BAND [10]) with automatic generation of executable SUMO programs.
- A reproducible simulation runner with versioned outputs and a unified metrics layer.

The remainder of this paper is organized as follows: Section 2 provides background on the `Attlas` platform and the bandwidth-based optimization models used in this work. Section 3 describes the `ATSUM` system architecture, including topology conversion, demand estimation, and signal optimization. Section 4 demonstrates the framework through a coordinated-corridor case study, while Section 5 introduces the graphical user interface. Finally, Section 6 discusses results and limitations, and Section 7 concludes the paper.

2. Background

2.1 `Attlas`

`Attlas` is a commercial adaptive traffic control software platform developed by Atman Systems, which maintains an internal, control-oriented representation of the road network and signalized intersections. This industrial traffic management platform is summarized in Figure 1. Figure 1 summarizes the main entities of this abstraction.

The network is composed of a set of `Node` and `Link` objects and, optionally, a collection of predefined `Path` definitions. Each `Node` represents a signalized or unsignalized intersection and is associated with geographic coordinates, as well as one or more `MovementGroup` entities. A node may also be associated with a `Controller` object that defines the operational signal control plan. Each `Link` connects an origin node to a destination node and contains an ordered set of `Lane` objects. Lanes carry geometric and operational attributes (e.g., width and maximum speed) and may include one or more `Detector` entities that provide traffic measurements. Turning movements are explicitly represented by `Movement` objects, each defined by an origin lane and a destination lane, and grouped into `MovementGroup` abstractions that correspond to controller signal groupings.

Signal control logic is encoded in the `Controller` entity through a hierarchical `Plan` structure composed of `Stage` and `Interval` elements. Each interval specifies a duration and a vector of group activation states (`groupStatus`), which together define the temporal evolution of signal groups during operation. Besides, optional `Path` objects represent predefined traffic routes as sequences of `PathLink` elements, each referencing a pair of origin and destination nodes. These paths provide a network-level abstraction for representing recurrent travel patterns and can be used as inputs for route-based demand generation and metric collection, depending on client requirements.

This data model is semantically aligned with operational traffic control concepts, such as movement groups, signal groups, and controller stages, rather than with geometric or lane-level simulation primitives. While the `Attlas` topology can be exported as a structured JSON description for external processing, its underlying abstraction is designed to support production control logic and coordination strategies.

In contrast, microscopic simulators such as SUMO require a geometric network representation based on nodes, edges, lanes, and explicit lane-to-lane connections,

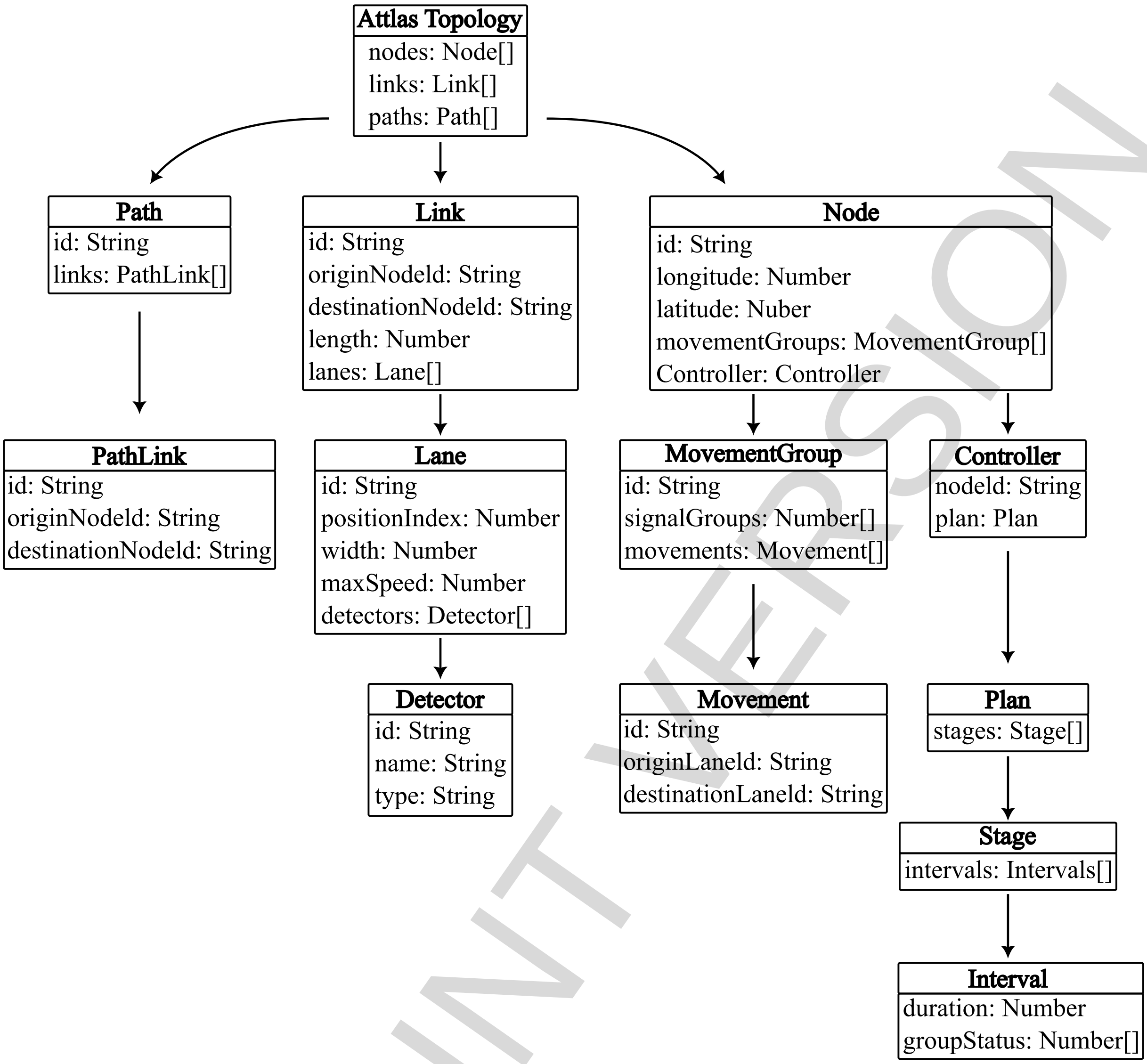


Figure 1. *`Attlas` topology and control abstraction showing nodes, links, lanes, movements, detectors, and controller plans.*

as well as separate definitions for traffic-light logic and detectors. Consequently, integrating `Attlas` with SUMO requires a structured translation process that preserves the platform's movement- and group-level semantics while transforming the `Attlas` topology into SUMO's connection-based simulation model.

2.2 SUMO

SUMO is organized as a modular traffic-simulation ecosystem, with a high-performance microscopic simulation core complemented by specialized tools for network construction, demand modeling, route generation, visualization, and post-processing [2]. At its core, the simulator executes time-discrete, space-continuous vehicle dynamics over large urban networks, while auxiliary components (e.g., for network import/editing and routing workflows) transform heterogeneous mobility data into executable scenarios. This architecture is further extended through APIs, most notably TraCI, which enable external applications to monitor and control vehicles, traffic lights, and other simulation entities online. As a result, SUMO is well-suited for closed-loop traffic management experiments, algorithm prototyping, and reproducible evaluation pipelines that couple planning data with simulation outputs [6].

During execution, SUMO can be integrated with external low-level controllers via TraCI's TCP client–server interface, and performance-critical integrations can use libsumo with equivalent function signatures. This separation between core simulation, preprocessing tools, and online control interfaces makes SUMO suitable for large-scale and reproducible experimentation pipelines in traffic management [2].

Offline signal planning methods compute cycle lengths, green splits, and offsets to improve traffic progression along coordinated corridors [11], [12]. Bandwidth-based coordination models, such as MAXBAND and its extensions, are widely used to maximize progression bandwidth under signal timing and speed constraints [8]–[10]. These optimization models are typically evaluated using microscopic simulation to assess their operational performance and feasibility under realistic traffic conditions. In addition, they incorporate turn phasing constraints, clearance intervals, and speed bounds to ensure implementability in field deployments [9], [10]. However, it should be noted that in these models, demand is typically assumed to be exogenous and fixed. In reality, traffic demand can adapt to signal timing changes via route choice or departure time shifts, adding a layer of complexity to the solution space validation.

However, validating the output of signal optimization models through microscopic simulation poses additional challenges, particularly in large, complex networks. As network size increases, the number of intersections, coordination variables, and feasible timing configurations grows rapidly, leading to a combinatorial explosion of candidate signal plans [8], [9]. As a result, exhaustively testing these alternatives becomes computationally expensive and operationally cumbersome, since each configuration requires regenerating simulation assets, executing over extended horizons, and post-processing performance metrics. Additionally, large topologies amplify the effect of small inconsistencies between optimization outputs and simulation inputs. Minor discrepancies in phase structure or offset interpretation can propagate across coordinated corridors, making it difficult to attribute observed performance differences to the optimization model itself rather than to translation artifacts [10]. Together, these factors hinder systematic exploration of the solution space and limit the practical use of simulation as a validation tool for large-scale coordination problems.

3. Proposed System

`ATSUM` implements a modular pipeline that connects platform-level traffic signal planning data to microscopic simulation. Each processing stage produces structured artifacts consumed by subsequent stages, forming a deterministic workflow from network definition to performance evaluation. As illustrated in Figure 2, a user-defined scenario is converted from the `Attlas` network abstraction into a SUMO-compatible representation.

During this step, lane connectivity is analyzed to infer geometric movement types (left, through, and right), and additional detectors are inserted when needed to ensure coverage of incoming approaches. Based on the converted network, traffic demand is then constructed using one of two mechanisms. When detector data are available, measured flows are aggregated and transformed into route-level demand inputs. Conversely, when detector data are missing or incomplete, synthetic demand is generated using topology-based source-sink identification and prescribed temporal profiles.

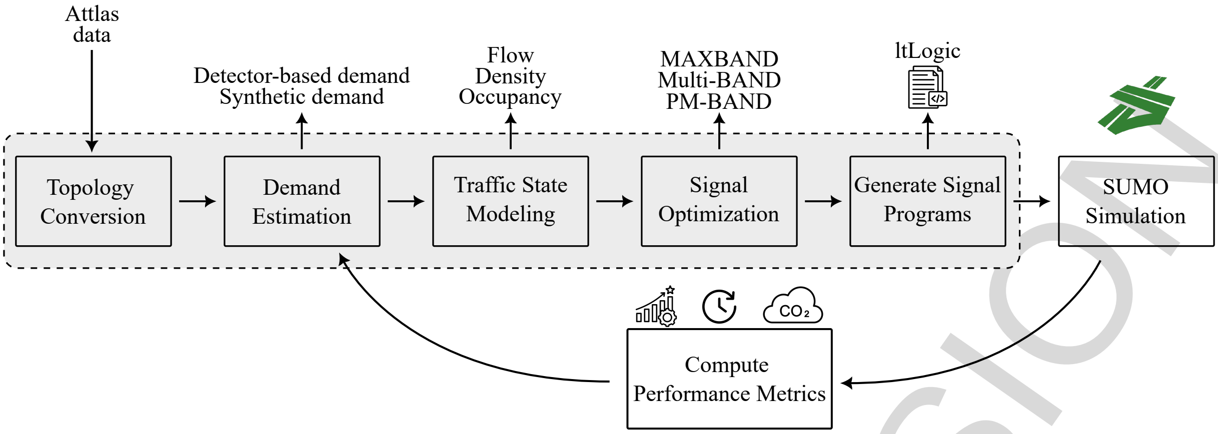


Figure 2. *`ATSUM` system overview.*

Simulation with inferred or synthesized demand produces a traffic-state representation that aggregates lane-level and movement-level quantities. These data are then used to derive movement demands and green splits through a critical movement formulation for standard eight movement intersections. Signal coordination is subsequently computed using bandwidth-based optimization models (MAXBAND, MultiBand, and PM-BAND). The optimization operates over the `Attlas` movement and signal-group abstractions and yields timing plans defined by movement states, offsets, and cycle lengths. Finally, the resulting plans are translated into executable SUMO traffic light programs by mapping movement-level decisions to lane-to-lane connections.

Finally, the generated signal programs are executed in SUMO, and performance metrics (such as efficiency, quality of service, and sustainability) are extracted from the simulation outputs. These metrics enable comparative evaluation of baseline and optimized plans and support iterative refinement of demand and control parameters. All stages are implemented within a unified software package and exposed through both programmatic and graphical interfaces, enabling batch execution and automated experimentation while maintaining consistent data flow across topology conversion, demand estimation, optimization, and simulation.

3.1 Topology Conversion and SUMO Integration

The topology conversion module implements the interface between the `Attlas` control abstraction (presented in Section 2.1) and the geometric, connection-based network model required by SUMO. Its function is to transform the platform-level representation of nodes, links, lanes, movements, and controller plans into a complete set of SUMO-compatible simulation assets while preserving identifiers, movement semantics, and signal-group structure.

Starting from an exported `Attlas` topology, the adapter generates intermediate network descriptions in terms of SUMO nodes, edges, and lane-level connections. `Node` entities associated with a `Controller` are mapped to traffic-light-controlled intersections, while each `Link` is converted into a directed edge with an explicit ordering of `Lane` objects. During this step, a lane-correspondence map is constructed to maintain a bidirectional association between `Attlas` lane identifiers and SUMO lane indices. This mapping is reused across all subsequent pipeline stages, including traffic-light logic synthesis, detector placement, and route generation.

Turning movements defined by `Movement` objects are translated into SUMO connections by resolving their origin and destination `Lane` references. For each movement, the adapter records both the `Attlas` identifiers and the corresponding SUMO edge-lane indices, producing an explicit movement-to-connection mapping. This artifact allows later workflow components to operate on movement-level quantities while interfacing with SUMO's connection-indexed signal logic. Static traffic light logic is derived from `Controller` plans by converting `Stage` and `Interval` definitions into SUMO `tlLogic` programs. Group-level signal states specified by `MovementGroup` entities are expanded into per-connection phase strings, thereby preserving the temporal structure and group semantics of the operational control plans in the simulation model. This establishes a deterministic correspondence between `Attlas` signal groups and SUMO traffic light states.

To support traffic-state estimation and performance analysis, the adapter also generates auxiliary simulation assets. Existing `Detector` entities defined in the `Attlas` topology are translated into SUMO induction-loop sensors, while additional lane-area detectors are automatically inserted on approaches to signalized nodes to ensure full coverage of incoming movements. Timed event loggers are also instantiated to capture traffic light state transitions during simulation. Together, these assets provide the sensing infrastructure required for demand estimation and traffic-state reconstruction. When `Path` definitions are present, the adapter can also produce route and flow specifications by reconstructing ordered edge sequences from the corresponding sets of `PathLink` elements. This enables simulation of predefined traffic patterns and supports integration of observed or planned routes into the microscopic model.

All generated artifacts, including intermediate network files, traffic light programs, detector definitions, and optional routes, are compiled into a final SUMO network using the standard `netconvert` tool. The adapter maintains a structured map of all output file paths, serving as a single source of truth for subsequent simulation, optimization, and analysis stages. By enforcing a deterministic and explicit translation from the `Attlas` control abstraction to SUMO's geometric, connection-based representation, this component eliminates manual network reconstruction and reduces semantic drift between operational and simulated configurations. However, it is worth noting that other sources of discrepancy, such as demand modeling accuracy, routing behavior, and the calibration of vehicle dynamics, can still contribute to deviations between simulated and real-world behavior. It therefore provides the foundation for consistent demand estimation, signal optimization, and reproducible simulation-based evaluation.

3.2 Demand Estimation and Traffic State Modeling

`ATSUM` supports two complementary mechanisms for constructing traffic demand inputs, depending on sensor-data availability: (1) demand inference from existing detectors and (2) synthetic demand generation based on network topology and prescribed temporal profiles. In the synthetic case, demand profiles are defined in terms of time-varying arrival rates, and the resulting traffic load is evaluated using an estimated vehicle-accumulation curve. This curve approximates the number of vehicles in the network over time by combining generated departure times with an assumed average travel time, providing a coarse but informative representation of temporal congestion patterns before full simulation execution.

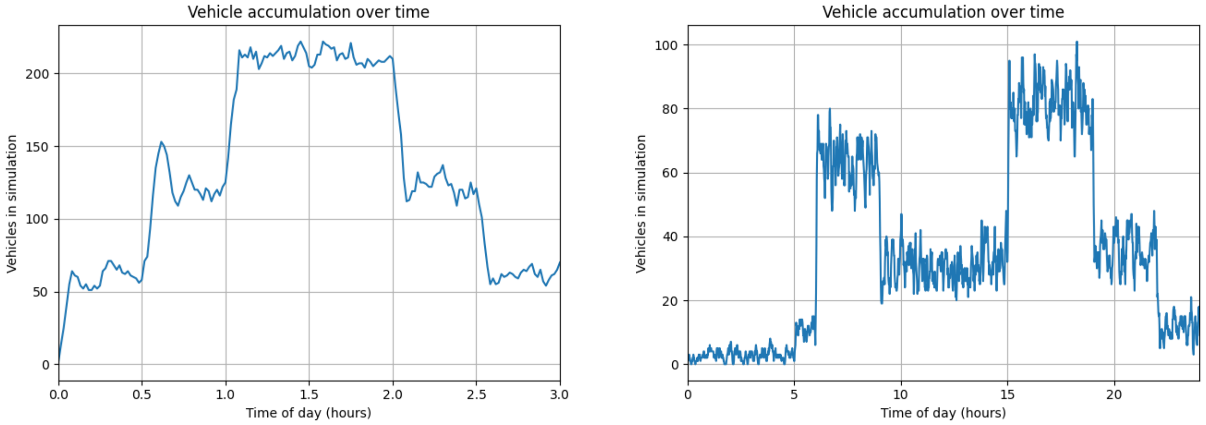
3.2.1 Detector-Based Demand Estimation

When detector measurements are available, `ATSUM` derives traffic demand from observed flows by invoking SUMO's demand generation tools on detector time series. Specifically, detector outputs are aggregated and used as inputs to `dfrouter` to reconstruct route flows consistent with measured link-level traffic volumes. This approach enables the simulation to be driven by empirically observed traffic patterns rather than by assumed distributions.

3.2.2 Synthetic Demand Generation

When detector data are unavailable or incomplete, `ATSUM` can alternatively synthesize traffic demand based on the network topology and prescribed temporal profiles. Source and sink edges are identified using topological criteria: edges whose upstream or downstream nodes are not controlled by traffic signals and lie at the boundary of the modeled subnetwork are classified as entry and exit points, respectively. Vehicles are generated only at source edges and are assigned destinations among reachable sink edges, ensuring that trips traverse the modeled network rather than originating or terminating internally.

Temporal variation in demand is modeled through parametric flow profiles that emulate commonly reported traffic patterns, such as multi-hour rush periods and full diurnal cycles. These profiles modulate a base arrival rate to produce time-dependent vehicle injection rates, yielding demand patterns that reproduce characteristic morning and evening peaks. Generated trips are subsequently routed using standard shortest-path assignment to obtain executable SUMO route definitions. Alternatively, existing SUMO tools such as `routeSampler.py` can be integrated to support demand generation from available data points. Figures 3a and 3b show examples of the resulting vehicle accumulation over time obtained from rush-hour and diurnal synthetic demand profiles, respectively. In these synthetic scenarios, the vehicle count is an input option that allows users to create demand based on possible routes and defined vehicle volumes for demonstration purposes.



(a) *Rush-hour profile (3 hours)*

(b) *Diurnal profile (24 hours)*

Figure 3. *Example vehicle accumulation generated by the synthetic demand model. Both profiles illustrate the temporal evolution of the number of vehicles present in the network, reproducing characteristic peak and off-peak traffic patterns commonly reported in the literature [13], [14].*

3.2.3 Traffic State Modeling and Movement Demand

`ATSUM` derives movement-level demand by combining standardized detector measurements with the network topology of turning movements. First, raw outputs from heterogeneous detector types are normalized into a common representation in terms of flow, occupancy, and density.

Flow is computed from vehicle counts over the aggregation interval Δt as:

$$q = \frac{N}{\Delta t} \cdot 3600, \tag{1}$$

where N denotes the number of detected vehicles. Likewise, density is obtained either directly from lane-area detectors or inferred from occupancy using an effective vehicle length L_{eff}:

$$k = \frac{\text{occ}}{L_{\text{eff}}} \cdot 1000. \tag{2}$$

Classical signal-control formulations rely on estimates of average arrival flow and saturation flow for each movement. However, saturation flow is difficult to observe directly from detector data. Therefore, `ATSUM` approximates it using capacity estimates derived from the observed flow-density relationship. When sufficient data are available, a Greenshields fundamental diagram is fitted:

$$q(k) = u_f, k\left(1 - \frac{k}{k_j}\right), \tag{3}$$

where u_f is the free-flow speed and k_j is the jam density. From this fit, capacity is computed as:

$$C = \frac{u_f k_j}{4}. \tag{4}$$

If calibration is not feasible or yields non-physical values, default parameter values are used instead. Next, standardized detector measurements and capacity estimates are mapped to their associated lanes and aggregated at the incoming edge of each signalized intersection. Accordingly, total approach flow and capacity are obtained by summing over contributing lanes while avoiding double counting.

Because optimization is formulated at the movement level, approach demand is then distributed across turning movements using lane-to-lane connectivity. Let Q_a and C_a denote the total arrival flow and capacity of incoming approach a. For each feasible movement $m \in \mathcal{M}_a$, demand is assigned as:

$$Q_m = \alpha_m Q_a, \qquad C_m = \alpha_m C_a, \tag{5}$$

where α_m denotes the turning proportion of movement m. When turning proportions are not directly observed, default split ratios are applied and normalized over $\mathcal{M}_a$.

Finally, Figure 4 illustrates an example intersection annotated with estimated movement flows and capacities obtained through this procedure. In this context, the estimated capacity derived from the fundamental diagram is used as a proxy for the saturation flow rate within the signal timing formulas, assuming the observed peak throughput represents the operational limit of the movement. As a result, each signalized node is represented by a set of movement-level demand values (Q_m, C_m), which constitutes the traffic-state representation used by subsequent signal optimization and green-split computation.

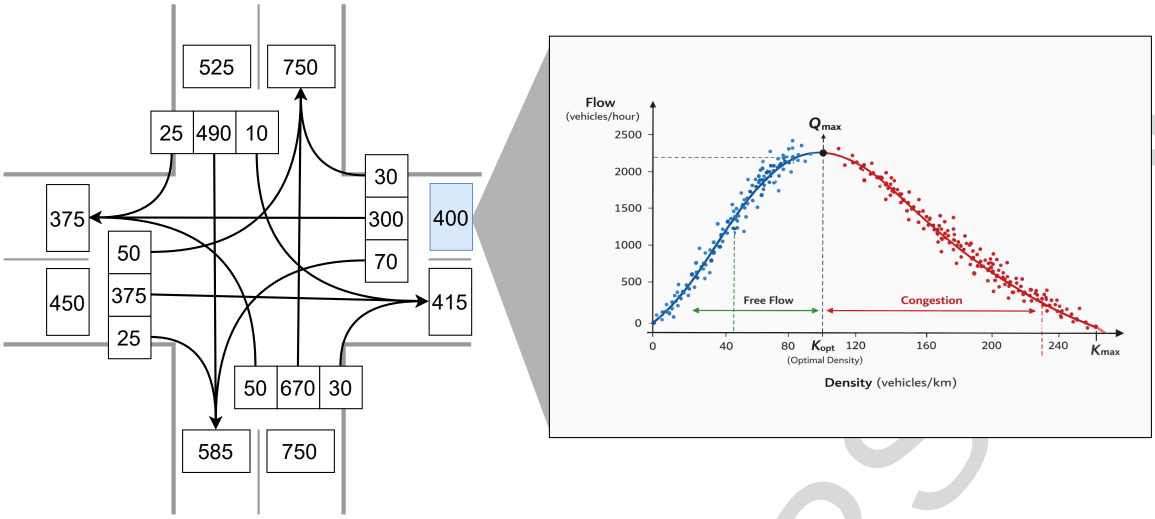


Figure 4. *Example intersection annotated with estimated movement-level arrival flows and capacities. Lane-level detector measurements are aggregated by approach and distributed across turning movements to obtain the quantities used for signal optimization.*

3.3 Signal Optimization and Program Generation

`ATSUM` integrates bandwidth-based coordination models to compute optimized signal timing plans from the traffic-state information derived in the previous stage. Several formulations are supported, including MAXBAND [8], MultiBand [9], and PM-BAND [10]. In this work, the emphasis is not on the optimization formulations themselves, but on the mechanism that translates their abstract outputs into executable traffic light programs for microscopic simulation.

The optimization layer produces timing plans expressed in terms of movements and signal groups, together with offsets and cycle lengths defined over the `Attlas` network abstraction. These results are represented as intersection-level plans composed of phase intervals, with each movement assigned a logical state (e.g., green, protected left, or red). However, SUMO requires traffic light programs to be defined over lane-to-lane connections using fixed-length phase strings.

To bridge this representational gap, `ATSUM` implements a translation layer that maps movement-level optimization outputs to SUMO-compatible `tlLogic` definitions. This process relies on the topology mapping established during network conversion, which associates each movement with a set of lane-to-lane connections. Even with this mapping, a central challenge in this translation is establishing correspondence between abstract optimization movements and concrete SUMO connections. Optimization models index movements by artery name, travel direction, and turn type, whereas SUMO represents signal control at the lane-to-lane connection level without an explicit notion of arterial structure. Consequently, movements cannot be mapped directly to simulation elements.

`ATSUM` resolves this mismatch by constructing an intermediate classification of incoming edges at each intersection in terms of artery and direction. If an edge explicitly belongs to a defined artery, the association is direct. Otherwise, the mapping is inferred by examining movements at the intersection and propagating arterial labels from downstream edges. As a result, arterial optimization variables are consistently grounded in the physical network, even when explicit corridor definitions are incomplete. Then, for each intersection, the optimized plan is decomposed into a sequence of phases with constant signal states. Within each phase, movement states are matched to their corresponding connections and expanded into connection-level signal assignments. Finally, movements within the same signal group are consistently mapped to a common connection state, thereby preserving the group-level constraints imposed by the optimizer in the simulation.

Phase boundaries are derived from the critical time points of the optimized plan. Whenever successive phases differ in their assigned states, yellow and all-red transition intervals are automatically inserted. In addition, the optimizer's offsets and cycle lengths are retained and applied directly to the generated `tlLogic` programs, ensuring temporal consistency between the analytical solution and its simulated execution. The overall translation procedure is summarized in Algorithm 1.

Algorithm 1 Translation of optimized timing plan into SUMO `tlLogic`

```
Require: Optimized timing plan $\mathcal{P}$ (movements, phases, offsets), connection map $\mathcal{C}$
Ensure: SUMO traffic light program $\mathcal{T}$
for all intersections $n$ in $\mathcal{P}$ do
    Extract movement-level phase sequence and cycle length
    Determine critical time boundaries from phase endpoints
    for all time intervals $[t_k, t_{k+1})$ do
        Initialize empty connection state vector
        for all movements $m$ active in interval do
            Retrieve associated connections $\mathcal{C}(m)$
            for all connections $c \in \mathcal{C}(m)$ do
                Assign signal state of $m$ to $c$
            end for
        end for
        Append phase with duration $(t_{k+1} - t_k)$ to $\mathcal{T}$
    end for
    Insert yellow and all-red transitions where state changes occur
    Apply offset and cycle length to $\mathcal{T}$
end for
return $\mathcal{T}$
```

By explicitly separating optimization from execution and introducing a deterministic translation layer between them, `ATSUM` enables direct evaluation of signal coordination models in microscopic simulation while preserving their structural assumptions and timing semantics. This design allows different optimization methods to be interchanged without modifying the simulation logic and ensures that simulation results faithfully reflect the optimized control strategy. Finally, although NEMA-style controller mappings for SUMO have been proposed, `ATSUM` follows an end-to-end translation approach that directly maps optimization movements to simulation connections. This choice provides finer control over the optimization-simulation interface and preserves the semantics of the underlying coordination models.

3.4 Simulation Execution and Metrics

`ATSUM` encapsulates SUMO execution within a dedicated simulation runner that manages configuration, execution, and result persistence in a unified manner. Each simulation run is executed in an isolated, version-controlled directory that contains all inputs, outputs, and metadata required to reproduce the experiment.

To ensure deterministic replay, all simulation inputs (i.e., network, routes, detector definitions, and signal programs) are explicitly resolved and recorded in a structured input map. Auxiliary files that declare output paths, such as detector and traffic-light state loggers, are rewritten at run time, and their outputs are redirected to run-specific locations. This mechanism prevents file collisions across experiments and enables parallel or batch execution of multiple scenarios. Additionally, simulation results are serialized along with the exact command-line invocation and the resolved input map, allowing each run to be reloaded and re-executed with the same configuration. `ATSUM` also supports execution in containerized environments with Docker, which isolates SUMO dependencies, simplifies deployment across heterogeneous platforms, and preserves the same run directory structure.

Building on this execution setup, performance metrics are computed from standard SUMO outputs. `ATSUM` then aggregates them into a unified metrics layer covering three categories:

- **Efficiency:** average travel time, delay per vehicle, and network throughput (vehicles/hour).
- **Quality of Service:** average number of stops and waiting time.
- **Sustainability:** fuel consumption and CO_2 emissions derived from SUMO's HBEFA-based model. Note that the mapping from emission classes to instantaneous emissions in SUMO is an approximation and may introduce modeling uncertainties.

By indexing all selected metrics by run identifier and linking them to the corresponding input map, `ATSUM` ensures comparisons between baseline and optimized scenarios are fully traceable and reproducible. Furthermore, by integrating simulation execution, versioned data management, and metric extraction into a single workflow, the framework provides a controlled experimental environment for the systematic evaluation of traffic signal control strategies.

4. Case Study

This section presents a case study of `ATSUM` in a representative urban signal coordination problem. The goal is to validate the proposed end-to-end workflow, from topology translation and demand construction to signal optimization and simulation-based evaluation, using a realistic but controlled network.

4.1 Study Area and Network Description

The study area corresponds to a signalized urban corridor in Rio de Janeiro, Brazil. The selected topology reflects a common arterial-collector configuration found in dense urban environments, where a dominant arterial is intersected by several lower-capacity cross streets. The modeled network comprises six signalized intersections arranged along a slightly curved main arterial, with multiple orthogonal cross streets. All intersections follow a four-leg configuration and are closely spaced, creating a coordination-sensitive environment in which offsets and progression are critical.

Figure 5 shows the microscopic network geometry constructed in the Attlas' traffic model described in Section 2.1. The traffic model is used in the experiments after conversion to a SUMO-compatible representation. Nodes in red correspond to signalized intersections, while edges represent individual roadway segments with explicit lane geometry. This topology was selected because it exhibits realistic geometric irregularities (non-orthogonal links and curvature) while remaining sufficiently compact to allow detailed inspection of signal coordination effects.

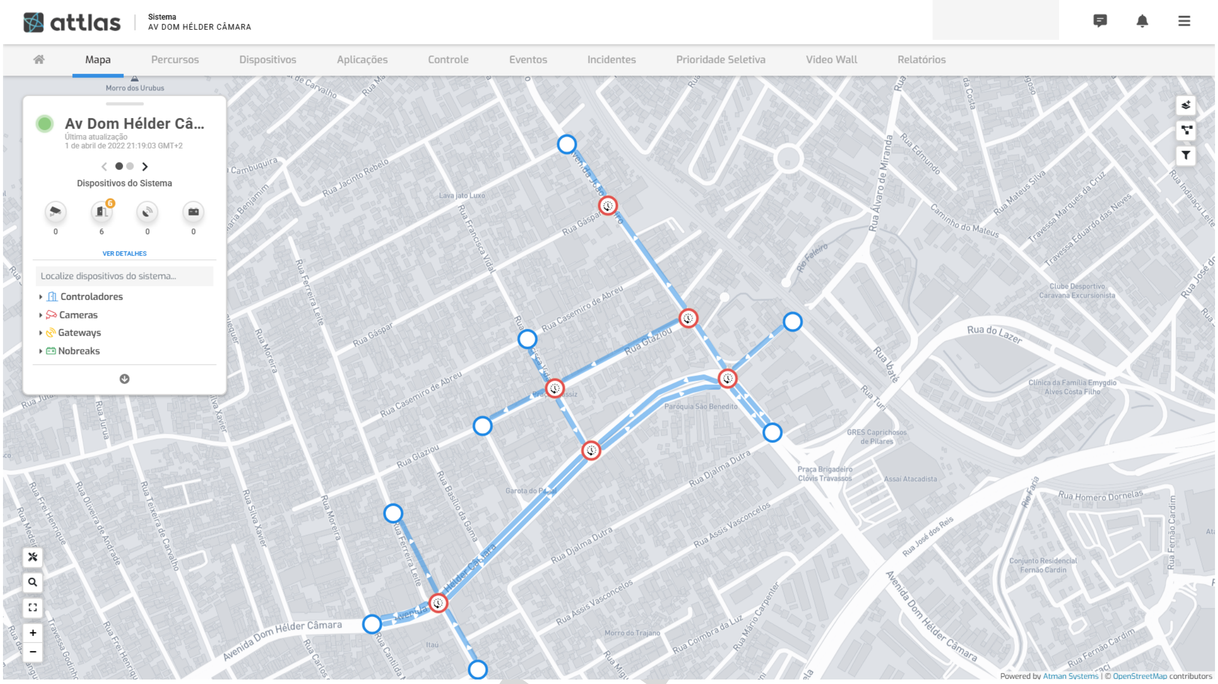


Figure 5. *Network topology used in the case study. The network represents a six-intersection urban corridor in Rio de Janeiro with a dominant arterial and multiple signalized cross streets.*

4.2 Signal Control and Coordination Setup

All six intersections are equipped with fixed-time signal control and participate in the coordination scheme. Each intersection is modeled with a standard eight-movement structure, including straight, left-turn, and right-turn movements. Protected left-turn phases are enabled where allowed by the geometry.

A common cycle length is enforced across the corridor to enable offset-based coordination. In the baseline scenario, the original fixed-time plans derived from the planning platform are used, preserving phase order, clearance times, and safety constraints. In contrast, optimized scenarios apply bandwidth-based coordination models, as described in Section 3.3. The optimization is performed along the arterial direction, and its outputs, such as cycle length, offsets, and movement-level green intervals, are automatically translated into executable SUMO `tlLogic` programs.

4.3 Traffic Demand

Traffic demand for the case study is generated using the synthetic-demand mechanism described in Section 3.2.2. This reflects a realistic planning situation in which comprehensive detector coverage is not available for all approaches. Specifically, demand follows a rush-hour profile, with higher arrival rates along the main arterial than on cross streets. Vehicle entry points are restricted to network-boundary edges, and all trips traverse the modeled corridor internally without originating or terminating. Finally, the same demand realization is used for both baseline and optimized scenarios to ensure that observed performance differences are attributable solely to signal-control strategies (see Figure 6).

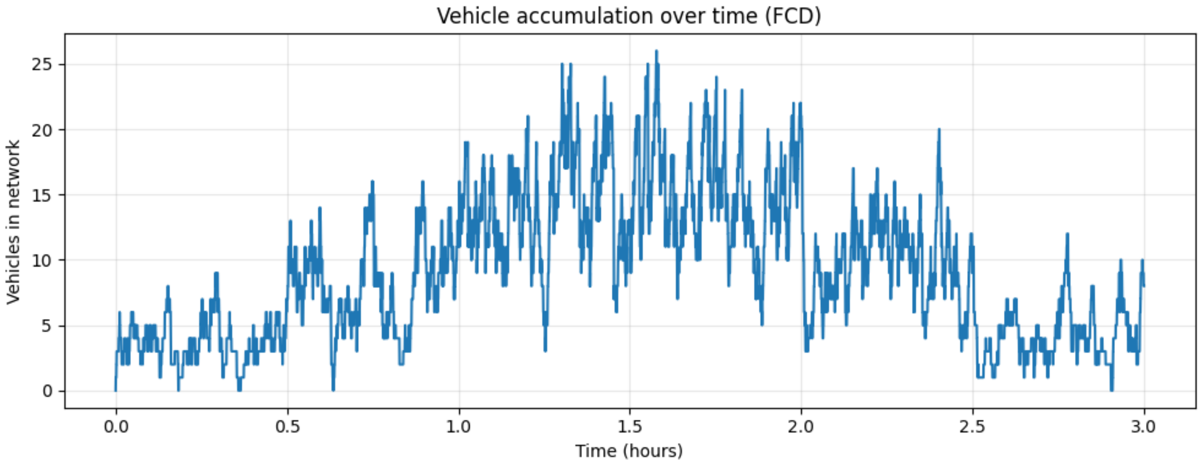


Figure 6. *Visualization of the synthetic rush-hour traffic demand used in the case study.*

4.4 Experimental Setup and Evaluation

Each scenario was simulated over a three-hour horizon. Three signal coordination configurations were evaluated:

- The current fixed-time coordination plan.
- An optimized coordination plan obtained using the MAXBAND formulation.
- An optimized coordination plan obtained using the MultiBand formulation.

All simulations were executed using the versioned simulation runner described in Section 3.4. For each run, all inputs, outputs, and metadata were persisted, ensuring deterministic replay and traceable comparisons. Given that this study serves as a demonstration of the integration framework rather than a performance benchmark, no warm-up period was applied to the synthetic demand realization. Performance was evaluated using standard microscopic traffic indicators, including average trip duration, average queue length, average waiting time, space mean speed, and total CO_2 emissions. Table 1 summarizes the performance metrics obtained for the three coordination scenarios.

Table 1. *Performance comparison between baseline and optimized coordination scenarios under identical demand conditions. Values are reported as mean ± standard deviation over all simulated vehicles where applicable.*

Run	Avg. Trip Duration (s)	Avg. Queue Length (m)	Avg. Waiting Time (s)	Space Mean Speed (km/h)	CO_2 Emission Total (g)
Current Configuration	65.16 ± 23.0	6.11 ± 2.5	20.10 ± 17.6	33.54	106293.60
Maxband Optimized	55.64 ± 21.7	5.84 ± 2.6	10.44 ± 11.6	39.33	100479.87
Multiband Optimized	64.85 ± 28.0	6.11 ± 2.6	18.72 ± 17.9	33.71	108759.54

As shown in Table 1, the MAXBAND-optimized configuration produced the strongest overall improvement among the evaluated scenarios. Relative to the current configuration, MAXBAND reduced the average trip duration from 65.16 s to 55.64 s, corresponding to an improvement of approximately 14.6%. Average waiting time decreased from 20.10 s to 10.44 s, representing a reduction of approximately 48.1%, while the average queue length decreased slightly from 6.11 m to 5.84 m. The space mean speed also increased from 33.54 km/h to 39.33 km/h, indicating smoother progression along the corridor. In addition to these operational improvements, total CO_2 emissions decreased from 106293.60 g to 100479.87 g, corresponding to a reduction of approximately 5.5%. This result also illustrates an important consideration when comparing emissions across signal timing scenarios: emission rates expressed per simulation step or per unit time may not directly reflect total environmental impact when scenarios produce different trip durations. A timing plan that reduces travel time may involve fewer simulated time steps and different cruising-speed profiles, so aggregate emissions over the completed trips provide a more consistent basis for cross-scenario comparison.

The MultiBand-optimized configuration produced more limited operational benefits. Its average trip duration, 64.85 s, was only slightly lower than that of the current configuration, while the average queue length remained unchanged at 6.11 m. Average waiting time decreased from 20.10 s to 18.72 s, indicating a modest improvement, and the space mean speed increased marginally from 33.54 km/h to 33.71 km/h. However, total CO_2 emissions increased to 108759.54 g, exceeding both the current configuration and the MAXBAND-optimized case. These results show that mobility improvements do not necessarily translate into lower aggregate emissions, particularly when changes in speed profiles, stopping behavior, and trip duration interact over the full simulation horizon.

These results indicate that, for the evaluated corridor and demand realization, the MAXBAND formulation provided the most effective timing plan in terms of both operational efficiency and environmental performance. By prioritizing arterial progression, MAXBAND reduced travel time and waiting time while also lowering the total mass of CO_2 emitted. In contrast, the MultiBand solution produced only marginal improvements in travel time and waiting time and did not reduce total emissions. This distinction highlights the importance of evaluating signal coordination strategies using both mobility and environmental indicators, as improvements in one dimension do not necessarily imply improvements in all others.

Overall, this case study demonstrates how `ATSUM` enables systematic, reproducible evaluation of coordinated signal timing strategies in a realistic urban topology. Reproducibility is achieved by enforcing a deterministic pipeline in which network topology, demand generation parameters, optimization outputs, and simulation configurations are fully specified and versioned for each run. All scenarios are executed using identical demand realizations, fixed optimization inputs, and explicitly recorded simulation parameters, ensuring that observed performance differences are attributable solely to changes in signal control logic. This design maintains a clear, traceable link between planning abstractions, optimization results, and microscopic simulation behavior, enabling experiments to be repeated and compared consistently across alternative timing plans.

5. Graphical User Interface

In addition to its programmatic interfaces, `ATSUM` provides a graphical user interface (GUI) that exposes the full planning-to-simulation workflow in an interactive and reproducible manner. The GUI is designed to support exploratory analysis and rapid prototyping, while relying on the same deterministic pipeline and versioned artifacts described in the previous sections. Figure 7 summarizes the main GUI components, which are organized as a sequence of tabs corresponding to the major stages of the workflow.

The *Project Setup* interface allows users to load an Attlas topology, configure the working directory, and trigger automated generation of all SUMO assets. The *Route Generation* tab supports synthetic demand specification using parametric temporal profiles, accompanied by an estimated vehicle accumulation plot that provides a diagnostic view of traffic loading prior to simulation.

Simulation runs are configured and executed through the *Simulation* tab, with optional containerized execution and asynchronous run tracking. The *Metrics* interface aggregates standard SUMO outputs into comparable performance indicators, enabling traceable comparison between baseline and optimized scenarios.

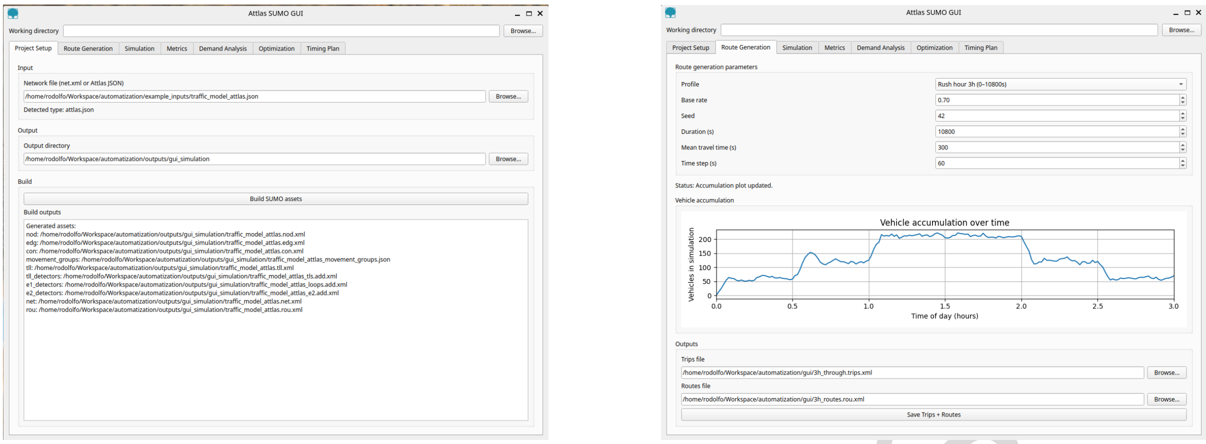
Finally, the *Optimization* and *Timing Plan* interfaces allow users to compute bandwidth-based coordination plans, visualize time-space diagrams, and export executable traffic light programs, directly reflecting the optimization-to-simulation translation described in Section 3.3.

6. Discussion

Beyond functional integration, a central motivation of `ATSUM` is reducing the manual engineering effort required to validate signal-control strategies in SUMO from the Attlas platform. To assess this aspect, we conducted a controlled comparison between a traditional manual integration workflow and the automated `ATSUM` pipeline using the six-intersection corridor described in Section 4.

In the manual workflow, engineers were required to construct SUMO-compatible artifacts starting from an exported planning-level topology. This process included explicit generation of nodes and edges, definition of lane-to-lane connections, manual specification of traffic light programs (tlLogic), detector placement, route configuration, and simulation setup. In contrast, the ATSUM workflow required only scenario-level configuration, with all simulation artifacts generated deterministically by the pipeline.

The total preparation time from topology availability to the first successful simulation execution was measured. Manual integration required qualitatively more effort, often spanning several hours due to variable tasks such as loading data from datasheets and manual movement verification. The ATSUM pipeline eliminated these steps by preserving a deterministic correspondence between movement abstractions and SUMO connections, thereby producing valid simulation artifacts on the first execution attempt. While the exact time depends heavily on network size and the complexity of in-field data, the automated approach structurally reduces this overhead.



Working directory

Network file (net.xml or Attlas JSON): /home/rodolfo/Workspace/automatization/example_inputs/traffic_model_attlas.json

Detected type: attlas.json

Output directory: /home/rodolfo/Workspace/automatization/outputs/gui_simulation

```
Generated assets:
nod: /home/rodolfo/Workspace/automatization/outputs/gui_simulation/traffic_model_attlas.nod.xml
edg: /home/rodolfo/Workspace/automatization/outputs/gui_simulation/traffic_model_attlas.edg.xml
con: /home/rodolfo/Workspace/automatization/outputs/gui_simulation/traffic_model_attlas.con.xml
movement_groups: /home/rodolfo/Workspace/automatization/outputs/gui_simulation/traffic_model_attlas_movement_groups.json
tll: /home/rodolfo/Workspace/automatization/outputs/gui_simulation/traffic_model_attlas.tll.xml
tll_detectors: /home/rodolfo/Workspace/automatization/outputs/gui_simulation/traffic_model_attlas_tls.add.xml
e1_detectors: /home/rodolfo/Workspace/automatization/outputs/gui_simulation/traffic_model_attlas_loops.add.xml
e2_detectors: /home/rodolfo/Workspace/automatization/outputs/gui_simulation/traffic_model_attlas_e2.add.xml
net: /home/rodolfo/Workspace/automatization/outputs/gui_simulation/traffic_model_attlas.net.xml
rou: /home/rodolfo/Workspace/automatization/outputs/gui_simulation/traffic_model_attlas.rou.xml
```

Route generation parameters

Profile: Rush hour 3h (0–10800s)
Base rate: 0.70
Seed: 42
Duration (s): 10800
Mean travel time (s): 300
Time step (s): 60

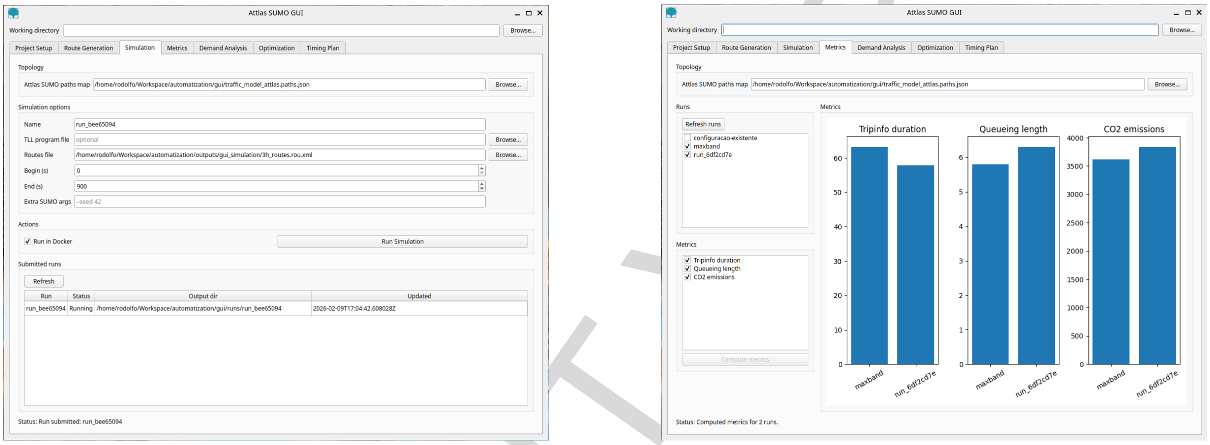
Status: Accumulation plot updated.

Vehicle accumulation

Trips file: /home/rodolfo/Workspace/automatization/gui/3h_through.trips.xml
Routes file: /home/rodolfo/Workspace/automatization/gui/3h_routes.rou.xml

(a) *Project setup and topology conversion.*

(b) *Synthetic demand and route generation.*



Working directory

Attlas SUMO paths map: /home/rodolfo/Workspace/automatization/gui/traffic_model_attlas.paths.json

Name: run_bee65094
TLL program file: optional
Routes file: /home/rodolfo/Workspace/automatization/outputs/gui_simulation/3h_routes.rou.xml
Begin (s): 0
End (s): 900
Extra SUMO args: --seed 42

☑ Run in Docker

Run	Status	Output dir	Updated
run_bee65094	Running	/home/rodolfo/Workspace/automatization/gui/runs/run_bee65094	2026-02-09T17:04:42.608028Z

Status: Run submitted: run_bee65094

Attlas SUMO paths map: /home/rodolfo/Workspace/automatization/gui/traffic_model_attlas.paths.json

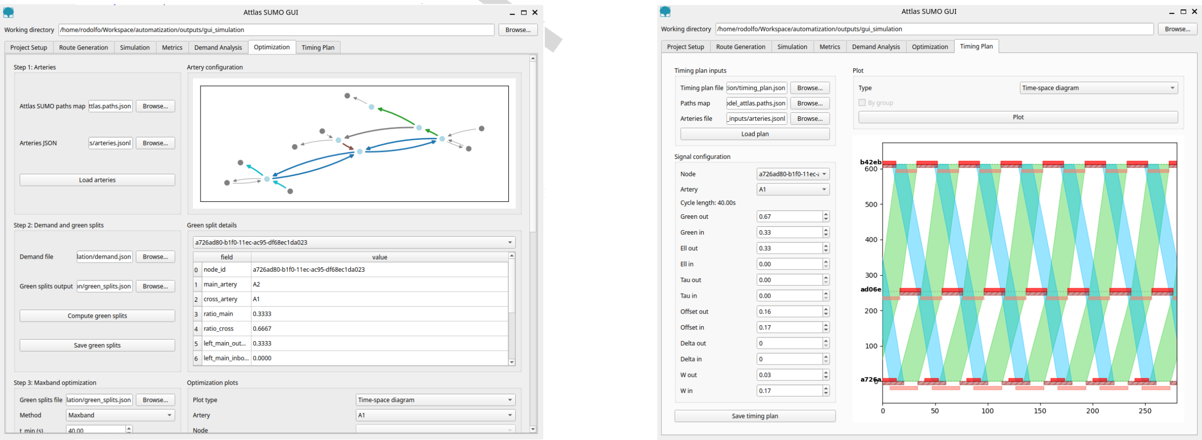
Runs
☐ configuracao-existente
☑ maxband
☑ run_6df2cd7e

Metrics
☑ Tripinfo duration
☑ Queueing length
☑ CO2 emissions

Status: Computed metrics for 2 runs.

(c) *Simulation execution and run tracking.*

(d) *Metrics computation and comparison.*



Working directory: /home/rodolfo/Workspace/automatization/outputs/gui_simulation

Attlas SUMO paths map: ttlas.paths.json
Arteries JSON: s/arteries.jsonl
Demand file: lation/demand.json
Green splits output: n/green_splits.json

a726ad80-b1f0-11ec-ac95-df68ec1da023

	field	value
0	node_id	a726ad80-b1f0-11ec-ac95-df68ec1da023
1	main_artery	A2
2	cross_artery	A1
3	ratio_main	0.3333
4	ratio_cross	0.6667
5	left_main_out...	0.3333
6	left_main_inbo...	0.0000

Green splits file: lation/green_splits.json
Method: Maxband
t_min (s): 40.00
Plot type: Time-space diagram
Artery: A1
Node

Working directory: /home/rodolfo/Workspace/automatization/outputs/gui_simulation

Timing plan file: ion/timing_plan.json
Paths map: odel_attlas.paths.json
Arteries file: _inputs/arteries.jsonl

Node: a726ad80-b1f0-11ec-a
Artery: A1
Cycle length: 40.00s
Green out: 0.67
Green in: 0.33
Ell out: 0.33
Ell in: 0.00
Tau out: 0.00
Tau in: 0.00
Offset out: 0.16
Offset in: 0.17
Delta out: 0
Delta in: 0
W out: 0.03
W in: 0.17

Type: Time-space diagram

(e) *Signal optimization and timing plan.*

(f) *Signal optimization and timing plan.*

Figure 7. *Overview of the `ATSUM` graphical user interface. The GUI exposes the complete workflow, from topology conversion and demand generation to simulation execution, performance evaluation, and signal optimization, while preserving deterministic inputs and reproducible run artifacts.*

From a scalability perspective, manual effort increased proportionally with the number of signalized intersections, as each additional node required explicit phase and connection specification. In contrast, ATSUM's preparation time was largely independent of network size beyond network-conversion overhead, since movement-to-connection mappings and signal-group translations are performed programmatically.

These findings suggest that ATSUM does not merely automate file generation but structurally reduces semantic translation overhead between planning-level abstractions and microscopic simulation models. This reduction directly supports reproducible experimentation, rapid iteration of signal strategies, and systematic comparison of alternative coordination schemes. However, several limitations remain.

First, the accuracy of demand estimation is constrained by the spatial coverage and quality of available detectors. When measurements are sparse or incomplete, the system relies on default turning ratios and synthetic demand generation, which may introduce uncertainty into the derived traffic state and optimization inputs.

Second, the coordination models currently employed are based on mixed-integer linear programming formulations. Although these models provide well-structured solutions, their computational complexity increases rapidly with corridor length and network size, potentially limiting scalability for large urban networks or real-time applications.

Finally, the fidelity of simulation results depends on the appropriate calibration of driver behavior parameters and saturation flows. While `ATSUM` ensures consistency between planning data and simulation inputs, it does not yet automate the calibration process and therefore relies on externally defined parameter settings.

7. Recommendations and Conclusion

The development of `ATSUM` highlighted a fundamental semantic gap between operational traffic control and microscopic simulation. Based on our observations, we recommend that future SUMO releases consider native movement abstractions to group lane-to-lane connections into logical phases, and standardized schemas for signal-timing exchange. Such features would significantly reduce the translation overhead described in Section 3.1 and lower the barrier for validating industrial control logic.

In conclusion, this work presented `ATSUM`, an end-to-end framework that connects traffic management data with the SUMO microscopic simulator to enable automated network translation, demand estimation, signal optimization, and reproducible simulation-based evaluation. By unifying these steps within a single workflow, `ATSUM` establishes a consistent interface between planning-level abstractions and simulation-level execution, thereby reducing semantic drift and shortening iteration cycles in traffic signal engineering.

The proposed framework enabled systematic comparison of baseline and optimized signal plans and provided a foundation for controller-in-the-loop validation of adaptive strategies in a safe simulated environment. Future work will focus on extending the optimization layer to support multimodal coordination, incorporating automated calibration procedures, and improving scalability through heuristic or decomposition-based methods. We also plan to release curated reference scenarios and documentation to facilitate the adoption and replication of the workflow across other networks and deployment contexts.

Regarding availability, `ATSUM` is currently developed as a proprietary internal tool at Atman Systems to support commercial traffic engineering projects. However, the authors are evaluating releasing a simplified open-source version of the topology translation and simulation-runner modules to foster collaboration within the SUMO community.

References

[1] D. Rowan, H. He, F. Hui, A. Yasir, and Q. Mohammed, “A systematic review of machine learning-based microscopic traffic flow models and simulations,” *Communications in Transportation Research*, vol. 5, p. 100 164, 2025.

[2] P. A. Lopez, M. Behrisch, L. Bieker-Walz, *et al.*, “Microscopic traffic simulation using sumo,” in *The 21st IEEE International Conference on Intelligent Transportation Systems*, IEEE, 2018.

[3] M. Bottazzi, L. C. Touko Tcheumadjeu, J. Trumpold, J. Erdmann, and R. Oertel, “Lisum: Design and development of a middleware to couple virtual lisa+ tls controller and sumo simulation,” *SUMO 2017–Towards Simulation for Autonomous Mobility*, vol. 31, pp. 179–192, 2017.

[4] K. Malena, C. Link, S. Gausemeier, and A. Trächtler, “Implementation and results of a multi-stage model predictive traffic light control system,” in *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2024, pp. 331–337.

[5] M. Barthauer and B. Friedrich, “Connecting microscopic traffic simulation and lisa+ external signal control,” *Transportation Research Procedia*, vol. 27, pp. 420–427, 2017.

[6] D. Krajzewicz, J. Erdmann, M. Behrisch, L. Bieker, *et al.*, “Recent development and applications of sumo-simulation of urban mobility,” *International journal on advances in systems and measurements*, vol. 5, no. 3&4, pp. 128–138, 2012.

[7] K. Riehl, A. Kouvelas, and M. A. Makridis, “Revisiting reproducibility in transportation simulation studies,” *European Transport Research Review*, vol. 17, no. 1, p. 22, 2025.

[8] J. Little, M. Kelson, and N. Gartner, “Maxband: A program for setting signals on arteries and triangular networks. transportation research record 795: Trb, national research council. washington,” 1981.

[9] N. H. Gartner, J. D. C. Little, and H. Gabbay, “Multiband—a variable-bandwidth arterial progression scheme,” *Transportation Research Part B*, 1991.

[10] W. Ma, L. Zou, K. An, N. H. Gartner, and M. Wang, “A partition-enabled multi-mode band approach to arterial traffic signal optimization,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 313–322, 2018.

[11] F. V. Webster, *Traffic signal settings*. Road Research Laboratory, 1958.

[12] R. P. Roess, E. S. Prassas, and W. R. McShane, *Traffic Engineering*. Pearson, 2010.

[13] J. Hansson, F. Pettersson-Löfstedt, H. Svensson, and A. Wretstrand, “Patronage effects of off-peak service improvements in regional public transport,” *European Transport Research Review*, vol. 14, no. 1, p. 19, 2022.

[14] L. L. Vitanova, E. Shirinyan, T. Trendafilova, and D. Petrova-Antonova, “Understanding sofia’s travel dynamics: Study of private traffic patterns and urban mobility,” *Transportation Research Interdisciplinary Perspectives*, vol. 30, p. 101 352, 2025.