SUMO User Conference 2024

Conference paper

https://doi.org/10.52825/scp.v5i.1112

Published: 17 Jul. 2024

Integration Traffic Signal Control From Synchro to SUMO

Yiran Zhang[1], Mingjian Fu[2], and Xuegang (Jeff) Ban[1]

[1] University of Washington, US

[2] INRIX Inc, US

Abstract. This study investigates the feasibility and challenges of transferring traffic signal control schemes from the macroscopic signal timing optimization tool Synchro to the microscopic traffic simulator SUMO, focusing on Downtown Seattle as a case study. The research assesses the process of sharing and importing traffic signal timing plans, a crucial aspect of transportation simulations, between these two platforms. We conduct a detailed analysis of the traffic signal characteristics and data formats unique to each simulator and identify elements suitable for conversion. Subsequently, a four-stage framework is developed for semi-automatic integration of traffic signal control between the two. Our results indicate a successful conversion rate of approximately 85% of signalized intersections from Synchro to SUMO. This research not only illustrates the challenges and solutions in converting signal control across different platforms but also paves the way for future studies aimed at improving the interoperability of various traffic simulation tools.

Keywords: Synchro, SUMO, Traffic Signal Control, Traffic Simulation

1. Introduction

The advancement of computer technology has significantly enhanced the application of traffic simulation in transportation studies. This simulated environment serves as an invaluable tool for, but not limited to transportation researchers, analysts, and policymakers, facilitating the evaluation of traffic management strategies, verification of optimized signal timing plans, assessment of impacts from Connected Autonomous Vehicles, and exploration of a wide range of transportation research questions. Traffic simulation is typically divided into four categories, each distinguished by its specific application objectives, geographical coverage, network characteristics, and scale [1], [2], [3], as shown below. This categorization helps in tailoring simulations to meet various analytical needs and research goals in the field of transportation.

- Macroscopic: Focuses on the traffic at a low level of detail where the traffic stream is represented in an aggregated level, such as speed, flow, and density.
- Microscopic: Details individual vehicle and pedestrian movements, allowing for the simulation of specific behaviors like car-following and lane-changing.
- Mesoscopic: A hybrid approach blending macroscopic and microscopic elements, offering more detailed analysis than macroscopic models but less resource-intensive than microscopic simulations.
- Submicroscopic: Provides intricate details of each vehicle, including internal mechanics such as gear shifting, suitable for in-depth vehicle dynamics analyses.

In response to diverse traffic simulation needs, a range of traffic simulation software has been developed, each catering to different levels of detail and analysis: Synchro and PTV VISUM [4] for macroscopic analysis; PTV VISSIM [5] and SUMO (Simulation of Urban MObility) for microscopic analysis; DTALite (light-weight Dynamic Traffic Assignment, [3]) and MATSim (Multi-Agent Transport Simulation, [6]) for mesoscopic analysis; and Unity [7] for submicroscopic simulations. However, a key challenge across these platforms is efficiently sharing and importing digital traffic network features, especially between software developed at different scales. This issue is particularly critical in large-scale urban transportation networks, where seamless feature transfer can significantly streamline the simulation process.

Efficient simulation conversion is vital for several reasons: In simulations of large urban or regional areas, extracting network features from existing simulations is more efficient than manual coding; In multiscale simulations that study traffic networks and vehicles at varying levels, seamless connectivity and communication across different simulation platforms are essential for effective analysis [8]; For researchers working on multiple simulation projects, the ability to transfer network features between platforms not only conserves valuable time and resources but also ensures consistency and coherence when working within the same study region; Additionally, in cases where data providers offer information derived from different simulation platforms, researchers must adeptly transfer these features to their targeted simulation platforms to maintain data integrity and relevance. The integration of these aspects underscores the importance of developing robust and flexible methods for simulation conversion, enhancing the efficiency, effectiveness, and scope of transportation research and analysis.

A particular challenge of this domain is the conversion of traffic signal timing between different simulation platforms. In contrast to more straightforward geographic parameters like lanes, intersections, and speed limits, simulation platforms often encode traffic controller settings and timing data in distinct ways, making it a critical bottleneck in the process of simulation conversion [9]. The root of this complexity lies in the different methodologies and network encodings each simulation tool employs. This issue is particularly evident in integrating traffic signal control between Synchro and SUMO [8], [9]. Synchro, a macroscopic traffic signal timing tool, often serves as a source of traffic signal data for transportation researchers, necessitating data conversion when integrated into other simulation platforms. Meanwhile, SUMO, a widely used open-source tool in transportation research, lacks direct conversion tools for integrating with Synchro. For instance, Netconvert [10], a SUMO command line application, imports digital road networks from various sources such as Vissim, MATSim, and OpenStreetMap. However, networks generated for SUMO simulations often encounter errors or missing features due to the differing traffic modeling and network design approaches across these platforms [11], [12]. On the other hand, the approach to timing plans differs between Synchro, which uses ring-barrier diagrams for phasing information, and SUMO, which relies on state-based phasing data [9]. This discrepancy, along with the various signal control strategies employed, such as fixed-time, adaptive, and actuated controls, adds complexity to the simulation conversion process. It underscores the need for more standardized and adaptable simulation approaches to facilitate smoother integration between different traffic simulation platforms.

This study addresses the challenges of integrating traffic signal timing across different simulation platforms, focusing on Synchro and SUMO. For this study, we use SUMO 1.7.0 and Synchro 10. We develop a semi-automatic signal timing integration framework and test it in downtown Seattle, which features a variety of signal timing plans and intersection types like five-way intersections and T-intersections. The paper is organized as follows: We first introduce the traffic network features and data formats for signal timing in each simulation. We then compare the two platforms and identify convertible network features. Next, we propose a four-stage approach for traffic signal timing integration. Our results show that approximately 90% of intersections can be successfully converted from Synchro to SUMO. Finally, we summarize our findings and discuss future research directions.

2. Synchro and SUMO Simulation

This study examines the approach of converting simulation data between Synchro and SUMO, each representing different scales of traffic analysis. Leveraging their distinct characteristics, some studies have combined Synchro and SUMO to explore transportation issues from different perspectives. For instance, Synchro is often used to develop various traffic signal timing algorithms, while SUMO is utilized to assess their impacts on vehicles [13], [14]. Additionally, several studies have focused on integrating signal timing control between SUMO and Synchro in smaller-scale areas, such as single corridors with a limited number of intersections [9]. However, the application of both Synchro and SUMO to large road networks remains relatively unexplored. A primary reason for this is the significant challenge posed by expanding the study area across different simulation platforms. Large-scale network simulations introduce increased variability and complexity due to the diversity of signal timing plans, intersection types, and transportation modes. Creating a network for just one simulation demands substantial time, labor, and financial resources, and these costs escalate when constructing two separate simulation platforms. Thus, understanding the network features of these two simulations and exploring the potential for their conversion is of critical importance. This understanding would not only streamline the simulation process but also enable more comprehensive and efficient transportation studies.

To bridge this gap, our study compares the input network features of both Synchro and SUMO, encompassing signal timing control and other geographic features crucial for integration. Synchro's network data, which could be exported to a Comma-Separated Value (CSV) file, focuses on macroscopic traffic signal optimization, encompassing elements like timing, phasing, lanes, traffic volume, and detectors. Conversely, SUMO's network is constructed from multiple eXtensible Markup Language (XML) files, detailing features at a microscopic level, including the road network, vehicle routes, and additional elements like bus stops and advanced signal timing plans. By examining these main network features and their sub-features, we identify which elements are transferable between the two platforms. A comparative table summarizing these features is presented in *Table 1* to facilitate understanding and potential conversion strategies.

***Table 1**. Similar feature comparison between Synchro and SUMO [15], [16], [17]*

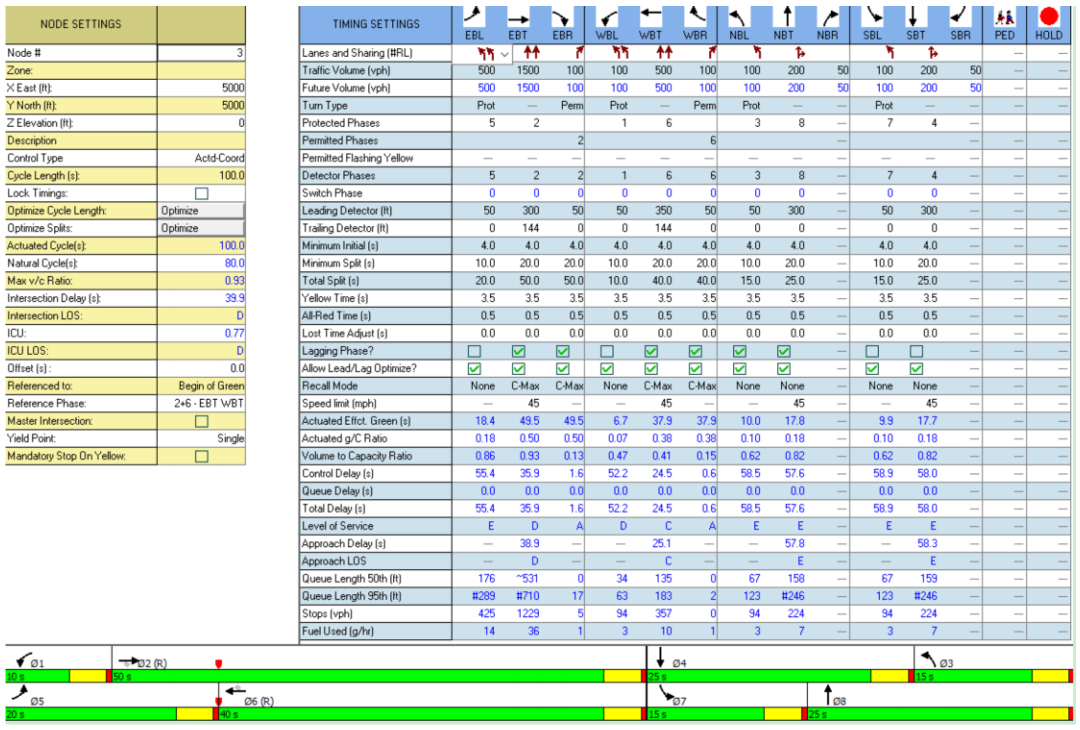
Settings	Synchro Features	Description	SUMO Features	Description
Road Network	Links	Includes road settings such as name, direction (e.g., northbound), distance, grade, and number of lanes.	Edge	Details from the .net.xml file, including road type (permitted vehicle class), priority, and number of lanes.
Road Network	Lanes	Describes the characteristics of individual lanes within a link, including speed limit, width, and storage capacity	Lane & Connections	Lane data from the .net.xml file is a sub-feature of an edge, including road length, coordinates, and allowed vehicle classes. Connection data described in the .net.xml file as the relationship between two lanes (from

				lanes and to lanes), including the direction of the connection (e.g., straight, left), the state (e.g., major link), and related signals.
Intersection	Nodes	Details the location of each intersection, including coordinates (X, Y, Z), and the type of intersection.	Junction	From the .net.xml file, it represents areas where different streams (edges) intersect, covering right-of-way rules, coordinates, and connected lanes.
Traffic Signal	Time-plans & Phases	Timeplan encompasses the signal timing plans, covering control types, cycle length, and offset. Phases specifies the signal timing phases for ring-barrier controller, including barrier, ring, and position (BRP), minimum and maximum green time, yellow time, and red time.	tlLogic	Defined in the .net.xml/.add.xml file, it details the phases of traffic lights, including control types, offset, and phase index.
Simulation	Network	Basic simulation settings that typically applied across the entire network, such as all red time, vehicle length, and scenario date and time.	Configuration	Defined under .sumocfg file, including the basic config for sumo simulation, including the input and output .xml files, simulation time, devices, etc.

After comparing the features of Synchro and SUMO, we identify both the potential and challenges in converting simulation data between these platforms. Firstly, while both simulations share similarities in road network structure and signal timing plans, their definitions of certain features differ significantly, impeding direct feature mapping. For example, Synchro's traffic timing plans use road direction (e.g., northbound) for single intersection control, whereas SUMO implements a clockwise pattern from 0 to 12 o'clock, prioritizing right turns, then straight movements, and finally left turns [18]. Besides, Synchro utilizes ring barrier control for traffic signal design, whereas SUMO typically employs fixed traffic timing plans as the default setting.

Secondly, the task of converting data from a macroscopic level (Synchro) to a microscopic level (SUMO) presents additional difficulties due to missing features. As shown in *Table 1*, although Synchro and SUMO share structural similarities, including lanes, links, and intersections, certain sub-features like road type are absent in Synchro. This discrepancy becomes more problematic in larger traffic networks, where the scale of the simulation amplifies the challenges of unmatched features, leading to increased errors and a labor-intensive process of network verification and correction. Additionally, manual revision of traffic signals for a large network is complex, given the variety of signal control types and signal timing definitions.

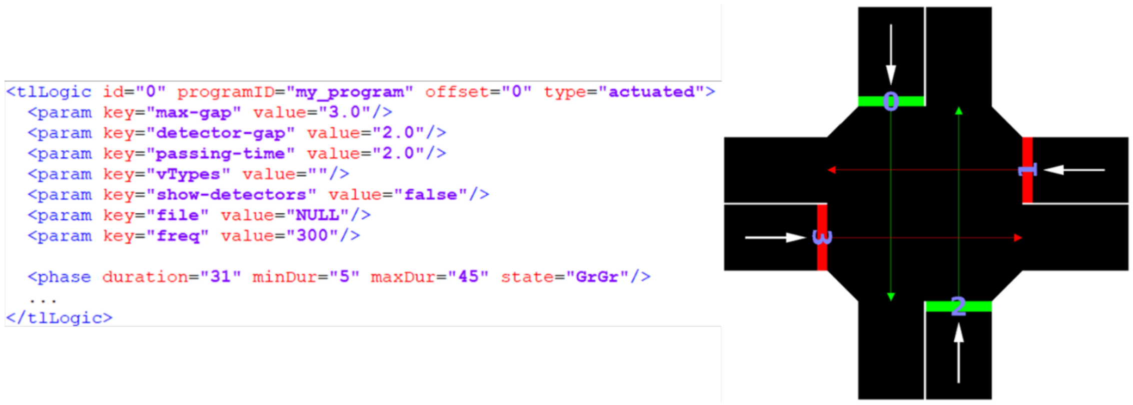
To address these challenges, our study delves into the specific traffic signal settings of both Synchro and SUMO. Synchro’s intuitive design allows for detailed timing settings for each direction, including minimum initial, split, yellow, and all-red times, shown in *Figure 1*. In contrast, SUMO organizes traffic signals in a clockwise sequence, with each phase describing the state of a traffic light signal and its duration, shown in *Figure 2*. This difference in approach necessitates a methodical strategy for efficient signal conversion between the two simulations, ensuring accurate and functional traffic signal integration in large-scale traffic networks.



NODE SETTINGS	
Node #	3
Zone:	
X East (ft):	5000
Y North (ft):	5000
Z Elevation (ft):	0
Description	
Control Type	Actd-Coord
Cycle Length (s):	100.0
Lock Timings:	☐
Optimize Cycle Length:	Optimize
Optimize Splits:	Optimize
Actuated Cycle(s):	100.0
Natural Cycle(s):	80.0
Max v/c Ratio:	0.93
Intersection Delay (s):	39.9
Intersection LOS:	D
ICU:	0.77
ICU LOS:	D
Offset (s) :	0.0
Referenced to:	Begin of Green
Reference Phase:	2+6 - EBT WBT
Master Intersection:	☐
Yield Point:	Single
Mandatory Stop On Yellow:	☐

TIMING SETTINGS	EBL	EBT	EBR	WBL	WBT	WBR	NBL	NBT	NBR	SBL	SBT	SBR	PED	HOLD
Lanes and Sharing (#RL)	↰↰	↑↑	↱	↰↰	↑↑	↱	↰	↱		↰	↱		—	—
Traffic Volume (vph)	500	1500	100	100	500	100	100	200	50	100	200	50	—	—
Future Volume (vph)	500	1500	100	100	500	100	100	200	50	100	200	50	—	—
Turn Type	Prot	—	Perm	Prot	—	Perm	Prot	—	—	Prot	—	—	—	—
Protected Phases	5	2		1	6		3	8	—	7	4	—		
Permitted Phases			2			6			—			—	—	—
Permitted Flashing Yellow	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Detector Phases	5	2	2	1	6	6	3	8	—	7	4	—	—	—
Switch Phase	0	0	0	0	0	0	0	0	—	0	0	—	—	—
Leading Detector (ft)	50	300	50	50	350	50	50	300	—	50	300	—	—	—
Trailing Detector (ft)	0	144	0	0	144	0	0	0	—	0	0	—	—	—
Minimum Initial (s)	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	—	4.0	4.0	—	—	—
Minimum Split (s)	10.0	20.0	20.0	10.0	20.0	20.0	10.0	20.0	—	10.0	20.0	—	—	—
Total Split (s)	20.0	50.0	50.0	10.0	40.0	40.0	15.0	25.0	—	15.0	25.0	—	—	—
Yellow Time (s)	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	—	3.5	3.5	—	—	—
All-Red Time (s)	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	—	0.5	0.5	—	—	—
Lost Time Adjust (s)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	—	0.0	0.0	—	—	—
Lagging Phase?	☐	✓	✓	☐	✓	✓	✓	✓	—	☐	☐	—	—	—
Allow Lead/Lag Optimize?	✓	✓	✓	✓	✓	✓	✓	✓	—	✓	✓	—	—	—
Recall Mode	None	C-Max	C-Max	None	C-Max	C-Max	None	None	—	None	None	—	—	—
Speed limit (mph)	—	45	—	—	45	—	—	45	—	—	45	—	—	—
Actuated Effct. Green (s)	18.4	49.5	49.5	6.7	37.9	37.9	10.0	17.8	—	9.9	17.7	—	—	—
Actuated g/C Ratio	0.18	0.50	0.50	0.07	0.38	0.38	0.10	0.18	—	0.10	0.18	—	—	—
Volume to Capacity Ratio	0.86	0.93	0.13	0.47	0.41	0.15	0.62	0.82	—	0.62	0.82	—	—	—
Control Delay (s)	55.4	35.9	1.6	52.2	24.5	0.6	58.5	57.6	—	58.9	58.0	—	—	—
Queue Delay (s)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	—	0.0	0.0	—	—	—
Total Delay (s)	55.4	35.9	1.6	52.2	24.5	0.6	58.5	57.6	—	58.9	58.0	—	—	—
Level of Service	E	D	A	D	C	A	E	E	—	E	E	—	—	—
Approach Delay (s)	—	38.9	—	—	25.1	—	—	57.8	—	—	58.3	—	—	—
Approach LOS	—	D	—	—	C	—	—	E	—	—	E	—	—	—
Queue Length 50th (ft)	176	~531	0	34	135	0	67	158	—	67	159	—	—	—
Queue Length 95th (ft)	#289	#710	17	63	183	2	123	#246	—	123	#246	—	—	—
Stops (vph)	425	1229	5	94	357	0	94	224	—	94	224	—	—	—
Fuel Used (g/hr)	14	36	1	3	10	1	3	7	—	3	7	—	—	—

***Figure 1**. Signal timing settings in Synchro [15]*



```
<tlLogic id="0" programID="my_program" offset="0" type="actuated">
  <param key="max-gap" value="3.0"/>
  <param key="detector-gap" value="2.0"/>
  <param key="passing-time" value="2.0"/>
  <param key="vTypes" value=""/>
  <param key="show-detectors" value="false"/>
  <param key="file" value="NULL"/>
  <param key="freq" value="300"/>

  <phase duration="31" minDur="5" maxDur="45" state="GrGr"/>
  ...
</tlLogic>
```

***Figure 2.** Traffic signal settings in SUMO [17]*

In this study, we propose a four-stage approach designed to semi-automatically translate select information from one simulation platform to the other. This approach aims to bridge the gap between the two systems, making the conversion process more streamlined and effective, especially where there are discrepancies in signal control methodologies and default configurations.

3. Methodology

3.1. Overview

The flow diagram in *Figure 3* showcases our semi-automatic process of integrating traffic signal control data from Synchro into SUMO is structured into four distinct stages: Data Preparation, Connection-to-Direction Mapping, Signal Phase Mapping, and Data Revision and Output. Central to this approach is a Python 3-based data pipeline, strategically selected for Python's comprehensive libraries adept at parsing and manipulating various data formats. The pipeline processes three key inputs: a Synchro CSV file detailing signal control data, a SUMO network XML file representing the traffic network, and an additional CSV file that aligns intersection IDs between the SUMO and Synchro platforms. In the flow chart, processes derived from Synchro data are marked in green, while those based on SUMO data are indicated in blue. This seamless integration ensures the generation of a new XML file, meticulously formatted to be compatible with SUMO's traffic light system. By bridging these systems, our approach enhances the accuracy and efficiency of traffic simulations in SUMO, reflecting more realistic and dynamic traffic signal behaviors. The portions of the process highlighted in purple require manual interventions, all other steps are automated.

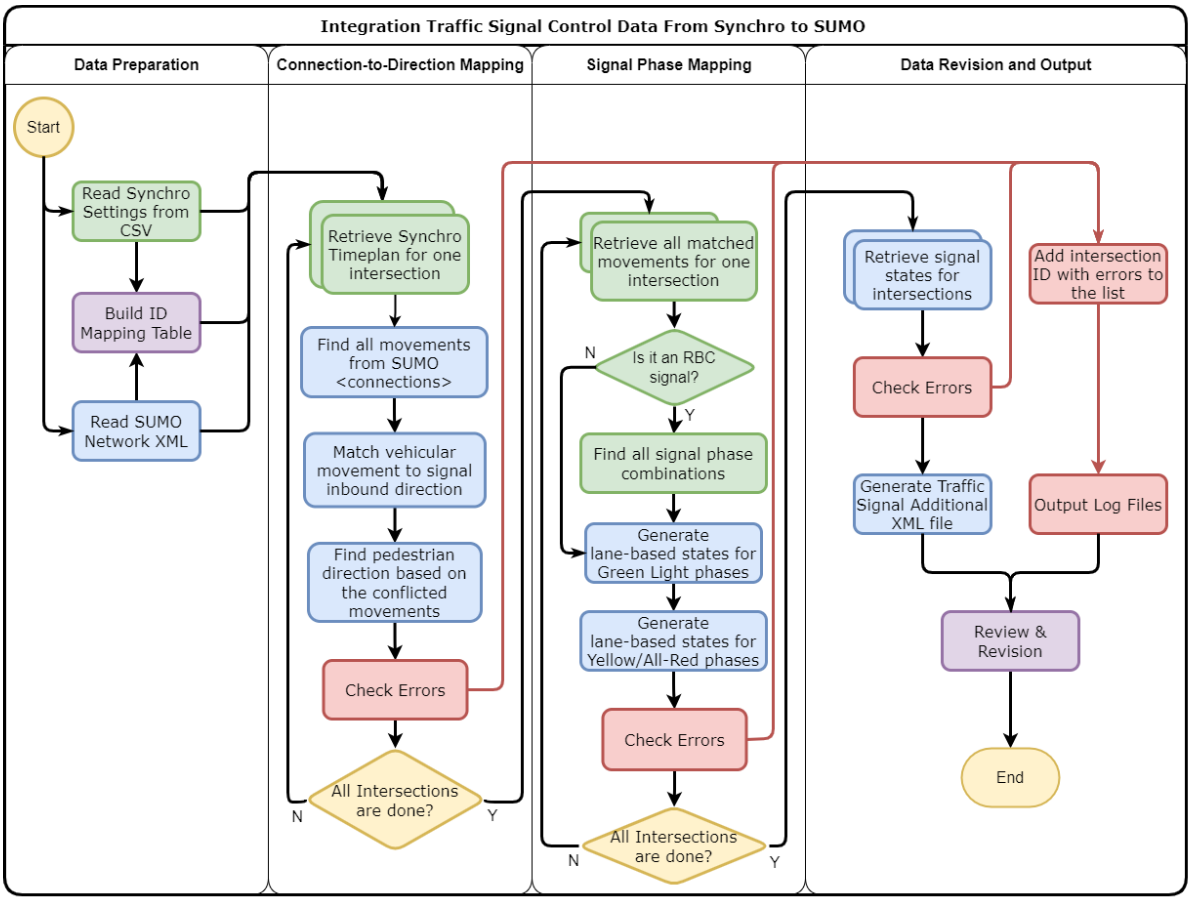

Integration Traffic Signal Control Data From Synchro to SUMO
Data Preparation
Connection-to-Direction Mapping
Signal Phase Mapping
Data Revision and Output
Start
Read Synchro Settings from CSV
Build ID Mapping Table
Read SUMO Network XML
Retrieve Synchro Timeplan for one intersection
Find all movements from SUMO <connections>
Match vehicular movement to signal inbound direction
Find pedestrian direction based on the conflicted movements
Check Errors
All Intersections are done?
N
Y
Retrieve all matched movements for one intersection
Is it an RBC signal?
N
Y
Find all signal phase combinations
Generate lane-based states for Green Light phases
Generate lane-based states for Yellow/All-Red phases
Check Errors
All Intersections are done?
N
Y
Retrieve signal states for intersections
Check Errors
Generate Traffic Signal Additional XML file
Add intersection ID with errors to the list
Output Log Files
Review & Revision
End

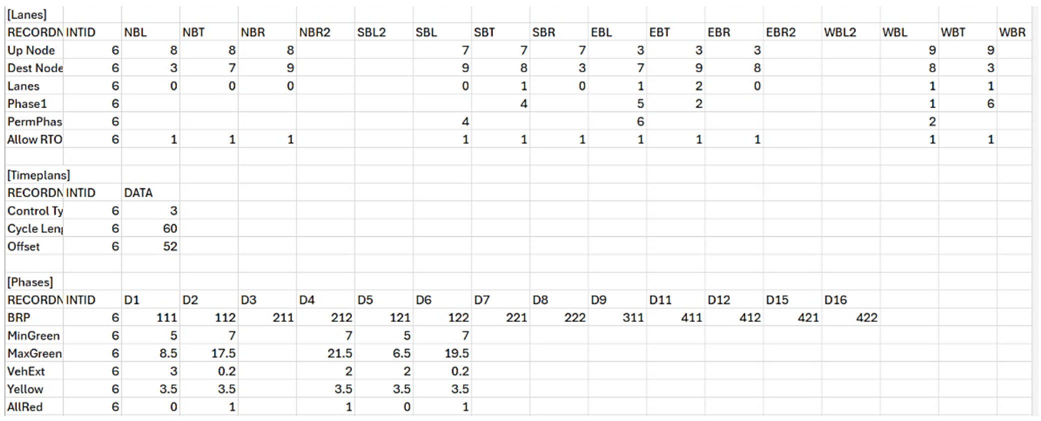
Figure 3. *Overview of the semi-automatic four-stage signal integration process*

3.2. Data Preparation

The Data Preparation stage is pivotal in aligning traffic-signal and intersection movement data from Synchro with the SUMO network. This step encompasses critical tasks such as parsing intersection and signal configurations from Synchro, extracting road network data from SUMO, and creating an intersection ID lookup table to harmonize data across both systems. The exported Synchro CSV file includes diverse model settings like networks, nodes, links, lanes, time plans, and phases. However, only specific settings are pertinent for signal integration. A detailed breakdown of these essential settings from Synchro, as illustrated in *Table 2*, guides the selective extraction of relevant data fields. This step forms the foundation for a streamlined and accurate data integration process, ensuring that the synthesized data reflects the complexity and dynamics of real-world traffic scenarios. *Figure 4* shows a data sample of one intersection in Synchro CSV, after data selective extraction. Each intersection contains multiple lane groups, which are denoted by direction abbreviations (e.g. EBL is an abbreviation of Eastbound Left Turn).

Table 2. *Synchro data key fields required for signal integration*

Section	Field Name	Description
Lanes	Phase{X}	Phase number of protected phase for this lane group. Each group may associate with multiple phases, which will be represented in the order of Phase1, Phase2, etc.
Lanes	PermPhase{X}	Phase number of permitted phase for this lane group. Each group may associate with multiple phases
Timeplans	ControlType	Signal controller types, indicating whether the signal is pretimed or actuated
Timeplans	CycleLength	Length of the cycle
Timeplans	Offset	Offset based on the reference phases
Phases	BRP	Barrier, Ring and Position to support Dual-Ring Barrier Controller
Phases	MinGreen	Minimum green time in seconds
Phases	MaxGreen	Maximum green time in seconds
Phases	VehExt	Vehicle extension time in seconds, for a vehicle passing over a detector.
Phases	Yellow	Yellow time in seconds
Phases	AllRed	All-red time in seconds
Phases	Recall	Recall mode determines whether the phase always shows the maximum/minimum initial time, or it can be skipped



[Lanes]																	
RECORDN	INTID	NBL	NBT	NBR	NBR2	SBL2	SBL	SBT	SBR	EBL	EBT	EBR	EBR2	WBL2	WBL	WBT	WBR
Up Node	6	8	8	8			7	7	7	3	3	3			9	9	
Dest Node	6	3	7	9			9	8	3	7	9	8			8	3	
Lanes	6	0	0	0			0	1	0	1	2	0			1	1	
Phase1	6							4		5	2				1	6	
PermPhas	6						4			6					2		
Allow RTO	6	1	1	1			1	1	1	1	1	1			1	1	

[Timeplans]		
RECORDN	INTID	DATA
Control Ty	6	3
Cycle Len	6	60
Offset	6	52

[Phases]														
RECORDN	INTID	D1	D2	D3	D4	D5	D6	D7	D8	D9	D11	D12	D15	D16
BRP	6	111	112	211	212	121	122	221	222	311	411	412	421	422
MinGreen	6	5	7		7	5	7							
MaxGreen	6	8.5	17.5		21.5	6.5	19.5							
VehExt	6	3	0.2		2	2	0.2							
Yellow	6	3.5	3.5		3.5	3.5	3.5							
AllRed	6	0	1		1	0	1							

Figure 4. *Synchro CSV data sample*

Unlike Synchro, SUMO employs the concept of a ‘connection’, which consists of a pair of from lane and to lane, to depict the movements taking place at an intersection. *Figure 5* illustrates that each entry in the connection data represents a legitimate lane-to-lane movement at an intersection. *Figure 6* visually presents intersection movements between edge/lanes, using the data from *Figure 5.* To accurately map traffic flow and movement patterns in our simulation, we must load the SUMO connection data. The usage of the data and mapping will be further discussed in the Connection-to-Direction Mapping section.

```xml
<connection from="-353690151#9" to="-293753571#1" fromLane="1" toLane="1"
 via=":53073250_5_0" tl="53073250" linkIndex="5" dir="r" state="o"/>
<connection from="-353690151#9" to="-353690151#1" fromLane="1" toLane="2"
 via=":53073250_6_0" tl="53073250" linkIndex="6" dir="s" state="o"/>
<connection from="293753571#0" to="-353690151#1" fromLane="1" toLane="1"
via=":53073250_3_0" tl="53073250" linkIndex="3" dir="r" state="o"/>
<connection from="293753571#0" to="353690151#2" fromLane="1" toLane="2"
via=":53073250_4_0" tl="53073250" linkIndex="4" dir="l" state="o"/>
<connection from="353690151#6" to="353690151#2" fromLane="1" toLane="1"
via=":53073250_0_0" tl="53073250" linkIndex="0" dir="s" state="o"/>
<connection from="353690151#6" to="353690151#2" fromLane="2" toLane="2"
via=":53073250_0_1" tl="53073250" linkIndex="1" dir="s" state="o"/>
<connection from="353690151#6" to="-293753571#1" fromLane="2" toLane="1"
via=":53073250_2_0" tl="53073250" linkIndex="2" dir="l" state="o"/>
<connection from=":53073250_w1" to=":53073250_c0" fromLane="0" toLane="0"
 tl="53073250" linkIndex="7" dir="s" state="M"/>
<connection from=":53073250_w2" to=":53073250_c1" fromLane="0" toLane="0"
 tl="53073250" linkIndex="8" dir="s" state="M"/>
```

Figure 5. *SUMO connection data in XML for intersection with traffic light id (tl=5307250)*

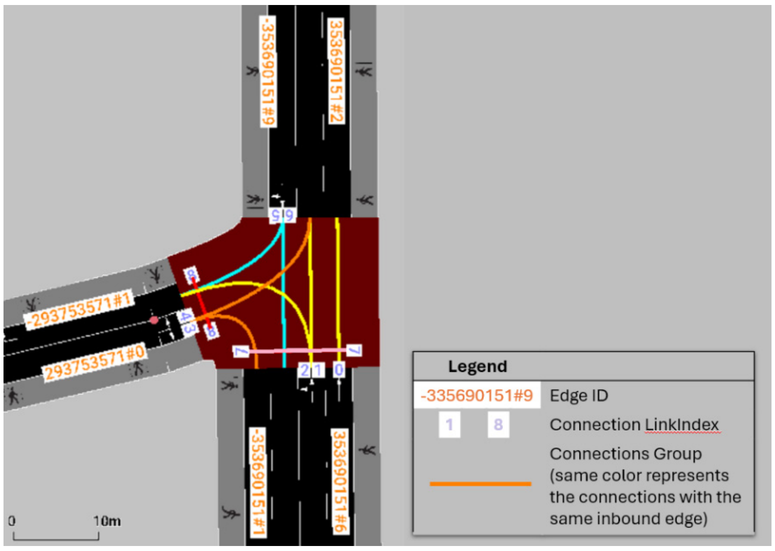


Figure 6. *Intersection layout in SUMO, with traffic light id (tl=5307250)*

In addition to the data from Synchro and SUMO, it's vital to establish a correspondence between signalized intersection node IDs in both networks for effective traffic signal integration. This task is complicated by the differing ID systems used by the two platforms: SUMO employs string-based node IDs, while Synchro uses integers with limited node numbers. Given the possibility of these networks originating from different sources or map versions, with varying levels of detail, nodes might be combined or split across models. To address this, the intersection node mapping table is created using a network alignment algorithm supplemented by manual lookup and revisions. This process is essential for ensuring accurate model alignment, especially in larger networks.

3.3. Connection-to-Direction Mapping

In SUMO, traffic signal phasing data, termed 'tlLogic', is represented as a string where each character denotes a signal state for each lane-to-lane movement. These states correspond to the connection data, with each state's index aligning with the 'linkIndex' in the connection data, as depicted in *Figure 5*. Unlike Synchro, which relies on traffic directions for distinguishing movements, as shown in *Figure 7*, SUMO does not explicitly indicate traffic direction. Due to the varying configurations of real-world intersections, it's crucial to map SUMO's connection data to Synchro's lane group data. This ensures that SUMO's signal states accurately reflect real traffic movements at intersections.

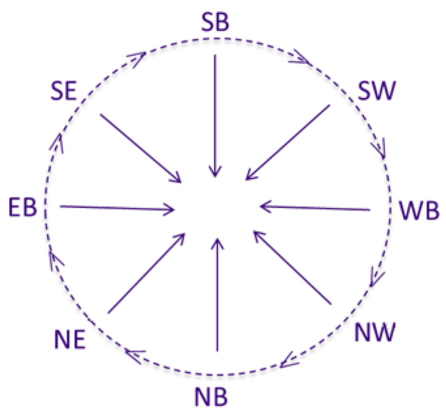


Figure 7. *Direction Order in Synchro*

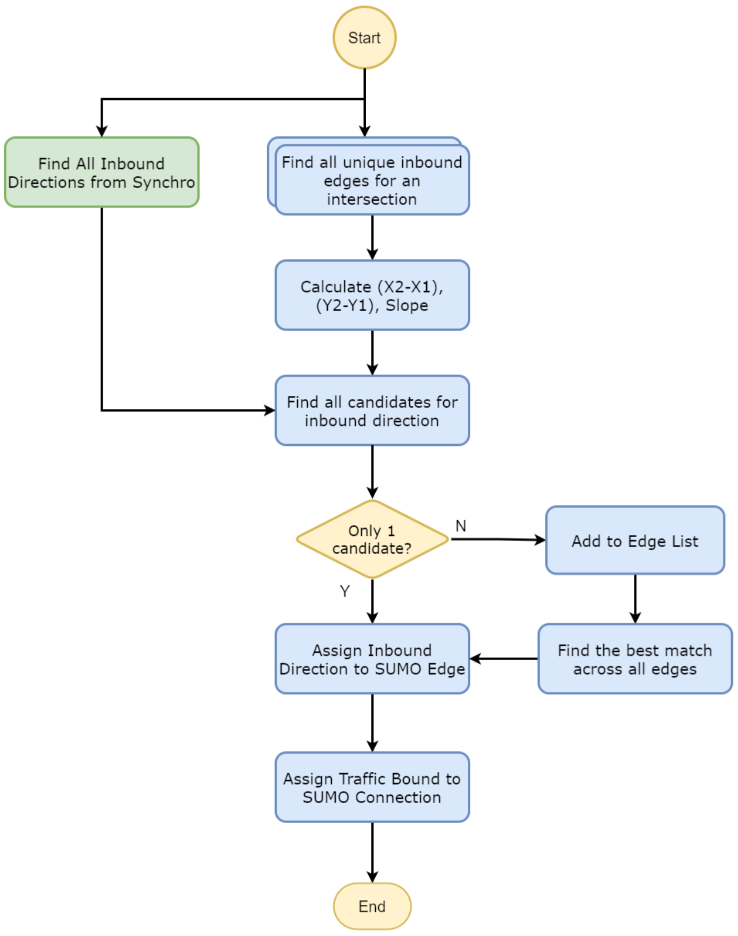

Start
Find All Inbound Directions from Synchro
Find all unique inbound edges for an intersection
Calculate (X2-X1), (Y2-Y1), Slope
Find all candidates for inbound direction
Only 1 candidate?
N
Y
Add to Edge List
Find the best match across all edges
Assign Inbound Direction to SUMO Edge
Assign Traffic Bound to SUMO Connection
End

Figure 8. *SUMO connection to Synchro direction mapping process*

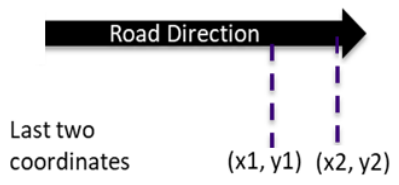


Figure 9. *The last two coordinates on the edge*

Figure 8 outlines the Connection-To-Direction process. At each intersection, we start by grouping the connections by inbound edge ID, assuming the number of inbound directions in Synchro aligns with SUMO's inbound edges, as shown in *Figure 6*. We then select the nearest two points to the intersection on each inbound edge as indicated in *Figure 9*, using them to formulate a linear equation representing the edge's geometry. This simplifies the task of determining the most probable direction by analyzing slope and coordinate differences. In cases where Synchro's directions are multi-faceted (e.g., NB, NE, NW), a single inbound edge might correspond to several direction possibilities. The final direction is determined by comparing the slopes of all edges related to these directions, following the sequence in *Figure 7*.

Next, the assignment of vehicular traffic bound direction in SUMO is determined by both the turning direction under the connection data and the traffic bound direction assigned in the previous step. This assignment follows a specific order of right turn (R), through (T), and left turn (L). In contrast to vehicular bounds, pedestrian directions lack a unique bound direction, as pedestrians can traverse the intersection from east to west (EB to WB) and vice versa. The identification of pedestrian bounds starts with the crossing edges in SUMO corresponding to the first vehicular bound direction, making the identification of corresponding crossing vehicular edges essential for defining the pedestrian bound.

3.4. Signal Phase Mapping

The signal phase mapping step aims to organize signal phase order and generate lane-based signal states. As previously stated, signal phase settings vary in SUMO and Synchro. Synchro uses a Ring-and-Barrier Designer for simulating signal ring-barrier controllers. Once a phase number is assigned to the BRP (barrier, ring, and position) field, Synchro automatically conducts signal phase transitions. SUMO, on the other hand, utilizes a state-based string where each character represents the signal color (green (G, g), yellow (y), red (r)) for each connection, with uppercase indicating protected movements and lowercase for permitted ones. Different models and algorithms have been developed in SUMO to support pretimed, actuated and National Electrical Manufacturers Association (NEMA) type signal controller for managing Dual-Ring-Barrier control signals [19], [20]. Real-world signal scenarios, however, can be more complex, involving multiple rings or specialized pedestrian crossings.

In this study, an alternative approach is applied in SUMO to manage signal phase transitions involves using ‘linkDuration’ and ‘next’ phase settings. The ‘next’ phase setting is used to indicate the next green phase candidates, while the ‘linkDuration’ can limit the maximum green time for a specific movement. This method provides a generic solution for handling more complex signal transition cases and involves three main steps: searching for green phase combinations, generating SUMO lane-based green phases, and interpolating transition phases.

Figure 10 demonstrates our method using a Ring-Barrier-Controlled signal with three rings as an example. The process begins with extracting all valid phases from Synchro. We then apply a Breadth-First Search algorithm [21] to identify all feasible combinations of green light phases. Following this, the Connection-To-Direction mapping is utilized to translate the

Synchro Phase number into SUMO lane-based states. This translation is facilitated by correlating SUMO's 'linkIndex' with the corresponding lane group in Synchro. The final step involves conducting a bitwise comparison between each pair of adjacent green phases. This is complemented by the interpolation of yellow and all-red phases to ensure a seamless transition and accurate signal timing.

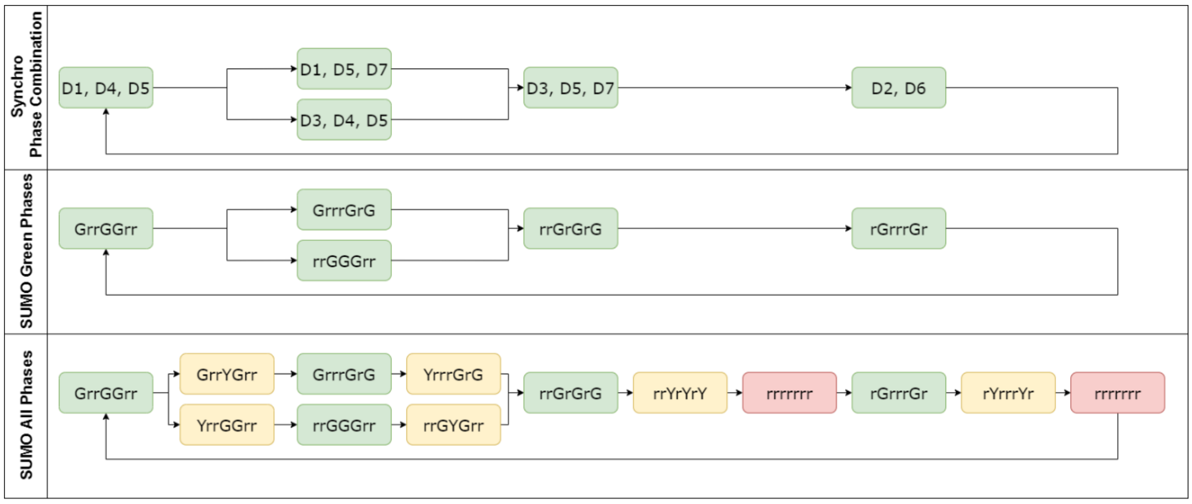


Figure 10. *Signal Phase Mapping Example*

3.5. Data Revision and Output

Upon completing the signal integration between SUMO and Synchro, the signals are exported to an XML file, ready for use as an additional SUMO signal file. In cases where complexities arise, such as combining multiple intersections or mismatches in road geometry, the automated signal phase mapping may not suffice. To counter this, validation steps are integrated at each stage to ensure SUMO's phases match those in Synchro. Discrepancies, which could indicate misalignments, necessitate manual revision at mismatched intersections. Furthermore, logs detailing intersection IDs and issue causes are generated and outputted for further analysis.

4. Case Study

4.1. Study Area

This study selects downtown Seattle as the study area, as depicted in *Figure 11*, from north to Mercer Street, south to South Atlantic St/Edgar Martine Dt St, west to Alaskan Way, and east to 12th Ave. The SUMO network developed for this area comprises three primary components: Network (.net.xml), Route (.rou.xml), and Additional (.add.xml). The network encapsulates the basic structures of the network, detailing edges, lanes, junctions, right-of-way rules, and connections. The route encompasses vehicle paths, pedestrian pathways, and public transit routes. Additionally, it serves as an extended descriptive file, including traffic analysis zones, bus stops, and the traffic signal data integrated with Synchro.

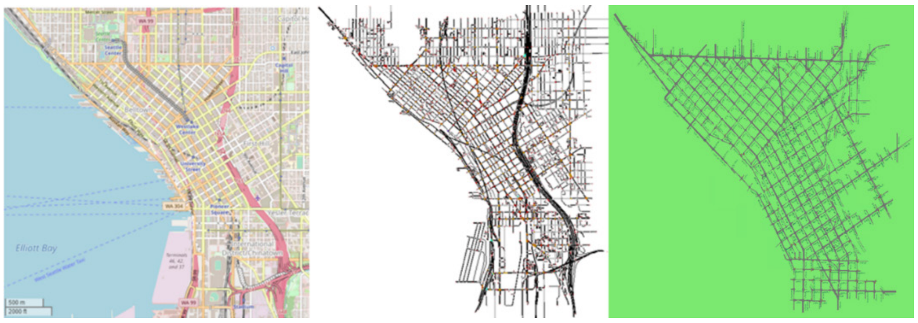


Figure 11. *Range map of downtown Seattle © OpenStreetMap contributors, SUMO network, Synchro network (order from left to right)*

Figure 12 illustrates the complexity of intersections within a large-scale network, highlighting the challenges associated with integrating Synchro and SUMO. The network features a range of intersection types, extending beyond the conventional four-way to include three-way and even five-way and six-way intersections. Additionally, the presence of signal protection at these intersections varies greatly, encompassing factors such as pedestrian phases, protected left or right turns, and different signal timing plans ranging from fixed to adaptive and actuated timing. This diversity significantly complicates the process of implementing integration between Synchro and SUMO, as each intersection type and signalization approach demands careful consideration and unique handling within the simulation models.

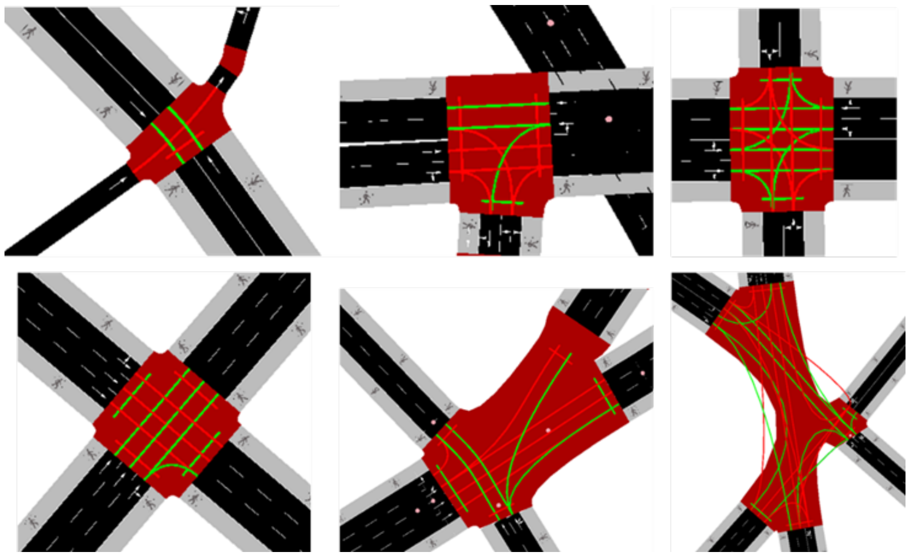


Figure 12. *Diverse intersections display in the SUMO simulation*

4.2. Results

To evaluate the algorithm, we initially applied our semi-automated framework to a smaller arterial (Dearborn Street) network comprising five intersections that were extracted from the Seattle network, before scaling up to the larger downtown Seattle area, featuring 281 intersections. This approach allowed us to refine the algorithm in a controlled environment before tackling the diverse and intricate traffic scenarios in the larger downtown network. Both test scenarios encompassed various controller types, including pretimed and actuated signal controllers.

A specific case study, detailed below, focuses on one cross intersection. Here, we demonstrate the successful integration of signal control in a dual Ring-Barrier-Controller intersection with pedestrian facilities. Referring to the signal phase mapping example in *Figure*

10, *Figure 13* displays the Synchro model of the intersection comprising eight movements, each direction with a signal-protected left turn. *Figure 14* reveals the SUMO output in XML format, achieved through the proposed four-stage algorithm. Meanwhile, *Figure 15* visualizes this intersection based on the Netedit tool. This particular case exemplifies our algorithm's effectiveness and its ability to accurately translate intricate traffic control scenarios into the SUMO framework, thereby enhancing the fidelity of traffic simulations.

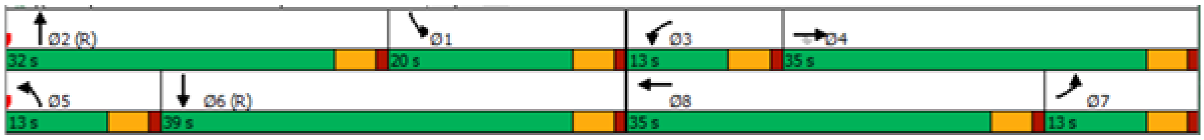
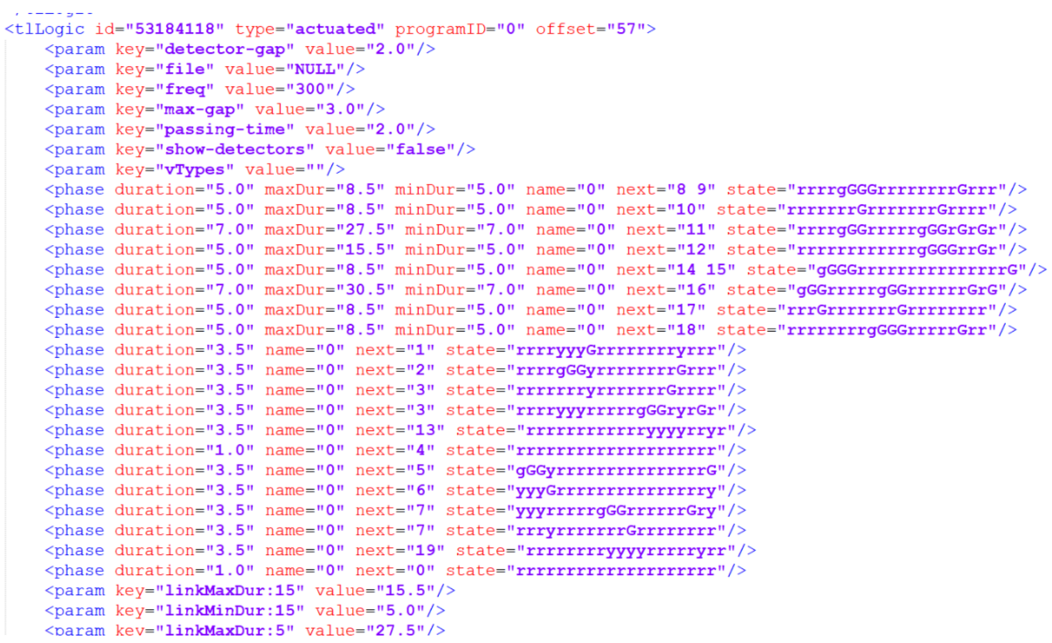
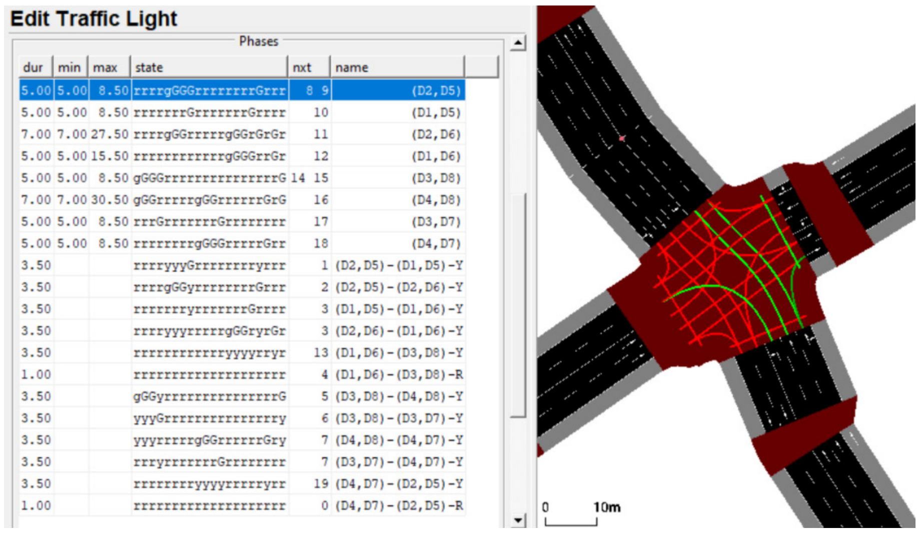


Figure 13. *A Ring-Barrier-Control Signal in Synchro*



```
<tlLogic id="53184118" type="actuated" programID="0" offset="57">
    <param key="detector-gap" value="2.0"/>
    <param key="file" value="NULL"/>
    <param key="freq" value="300"/>
    <param key="max-gap" value="3.0"/>
    <param key="passing-time" value="2.0"/>
    <param key="show-detectors" value="false"/>
    <param key="vTypes" value=""/>
    <phase duration="5.0" maxDur="8.5" minDur="5.0" name="0" next="8 9" state="rrrrgGGGrrrrrrrrGrrr"/>
    <phase duration="5.0" maxDur="8.5" minDur="5.0" name="0" next="10" state="rrrrrrrGrrrrrrrGrrrr"/>
    <phase duration="7.0" maxDur="27.5" minDur="7.0" name="0" next="11" state="rrrrgGGrrrrrgGGrGrGr"/>
    <phase duration="5.0" maxDur="15.5" minDur="5.0" name="0" next="12" state="rrrrrrrrrrrrgGGGrrGr"/>
    <phase duration="5.0" maxDur="8.5" minDur="5.0" name="0" next="14 15" state="gGGGrrrrrrrrrrrrrrrG"/>
    <phase duration="7.0" maxDur="30.5" minDur="7.0" name="0" next="16" state="gGGrrrrrgGGrrrrrrGrG"/>
    <phase duration="5.0" maxDur="8.5" minDur="5.0" name="0" next="17" state="rrrGrrrrrrrGrrrrrrrr"/>
    <phase duration="5.0" maxDur="8.5" minDur="5.0" name="0" next="18" state="rrrrrrrrgGGGrrrrrGrr"/>
    <phase duration="3.5" name="0" next="1" state="rrrryyyGrrrrrrrryrrr"/>
    <phase duration="3.5" name="0" next="2" state="rrrrgGGyrrrrrrrrGrrr"/>
    <phase duration="3.5" name="0" next="3" state="rrrrrrryrrrrrrrGrrrr"/>
    <phase duration="3.5" name="0" next="3" state="rrrryyyrrrrrgGGryrGr"/>
    <phase duration="3.5" name="0" next="13" state="rrrrrrrrrrrryyyyrryr"/>
    <phase duration="1.0" name="0" next="4" state="rrrrrrrrrrrrrrrrrrrr"/>
    <phase duration="3.5" name="0" next="5" state="gGGyrrrrrrrrrrrrrrrG"/>
    <phase duration="3.5" name="0" next="6" state="yyyGrrrrrrrrrrrrrrry"/>
    <phase duration="3.5" name="0" next="7" state="yyyrrrrrgGGrrrrrrGry"/>
    <phase duration="3.5" name="0" next="7" state="rrryrrrrrrrGrrrrrrrr"/>
    <phase duration="3.5" name="0" next="19" state="rrrrrrrryyyyrrrrryrr"/>
    <phase duration="1.0" name="0" next="0" state="rrrrrrrrrrrrrrrrrrrr"/>
    <param key="linkMaxDur:15" value="15.5"/>
    <param key="linkMinDur:15" value="5.0"/>
    <param key="linkMaxDur:5" value="27.5"/>
```

Figure 14. *Signal timing plan in SUMO XML format*



Edit Traffic Light

Phases

dur	min	max	state	nxt	name
5.00	5.00	8.50	rrrrgGGGrrrrrrrrGrrr	8 9	(D2,D5)
5.00	5.00	8.50	rrrrrrrGrrrrrrrGrrrr	10	(D1,D5)
7.00	7.00	27.50	rrrrgGGrrrrrgGGrGrGr	11	(D2,D6)
5.00	5.00	15.50	rrrrrrrrrrrrgGGGrrGr	12	(D1,D6)
5.00	5.00	8.50	gGGGrrrrrrrrrrrrrrrG	14 15	(D3,D8)
7.00	7.00	30.50	gGGrrrrrgGGrrrrrrGrG	16	(D4,D8)
5.00	5.00	8.50	rrrGrrrrrrrGrrrrrrrr	17	(D3,D7)
5.00	5.00	8.50	rrrrrrrrgGGGrrrrrGrr	18	(D4,D7)
3.50			rrrryyyGrrrrrrrryrrr	1	(D2,D5)-(D1,D5)-Y
3.50			rrrrgGGyrrrrrrrrGrrr	2	(D2,D5)-(D2,D6)-Y
3.50			rrrrrrryrrrrrrrGrrrr	3	(D1,D5)-(D1,D6)-Y
3.50			rrrryyyrrrrrgGGryrGr	3	(D2,D6)-(D1,D6)-Y
3.50			rrrrrrrrrrrryyyyrryr	13	(D1,D6)-(D3,D8)-Y
1.00			rrrrrrrrrrrrrrrrrrrr	4	(D1,D6)-(D3,D8)-R
3.50			gGGyrrrrrrrrrrrrrrrG	5	(D3,D8)-(D4,D8)-Y
3.50			yyyGrrrrrrrrrrrrrrry	6	(D3,D8)-(D3,D7)-Y
3.50			yyyrrrrrgGGrrrrrrGry	7	(D4,D8)-(D4,D7)-Y
3.50			rrryrrrrrrrGrrrrrrrr	7	(D3,D7)-(D4,D7)-Y
3.50			rrrrrrrryyyyrrrrryrr	19	(D4,D7)-(D2,D5)-Y
1.00			rrrrrrrrrrrrrrrrrrrr	0	(D4,D7)-(D2,D5)-R

Figure 15. *Signal timing plan display in SUMO Netedit tool [22]*

Table 3 presents the outcomes of signal integration for two scenarios: the Dearborn Street arterial network and the downtown Seattle network. The Dearborn Street arterial network, comprising 5 intersections, achieves a perfect conversion rate of 100%. In contrast, the more extensive downtown Seattle area, with 281 intersections, has a successful conversion rate of 85.1%, with 239 intersections converted. While the success rate in this larger network did not reach 100%, the result is deemed satisfactory considering the network's complexity and varying intersection conditions refer to *Figure 12*.

Table 3. *Signal integration results from Synchro to SUMO*

Scenario	**# of Intersections**	**Successful Converted Intersections**	**Success Rate**
Dearborn Street Arterial	5	5	100%
Downtown Seattle	281	239	85.1%

In addition, we reviewed the intersection that failed to be integrated. The most common issue is misaligned intersection geometry between SUMO and Synchro, which causes a match failure at the Connection-to-Direction stage. Additionally, certain intersections feature exclusive pedestrian phasing, a scenario our framework does not currently address. Moreover, the presence of coordinated signal control at some intersections poses another challenge. Integration of these intersections is conducted manually. Further discussions on potential improvements will be discussed in the next section.

5. Conclusion and Future Work

In this paper, we explored the process of integrating traffic signal data from Synchro to SUMO, identifying both the feasibility and challenges involved. Our approach began with a comparative analysis of traffic network features, data formats, and signal timing schemas between the two platforms. Despite similarities in road network features, significant differences in traffic signal timing plan encodings presented obstacles, especially in large-scale networks.

To overcome these challenges, we introduced a four-stage approach encompassing Data Preparation, Connection-to-Direction Mapping, Signal Phase Mapping, and Data Revision and Output. This methodology focused on extracting relevant signal timing and road features, aligning inbound directions, and generating lane-specific signal states for each timing phase. We then converted this data into SUMO's XML format. Our semi-automated pipeline, developed in Python, was tested on two Synchro models in downtown Seattle. The success rate exceeded 85%, demonstrating the pipeline's ability to handle various intersection layouts and signal controller types. However, the integration is not flawless in large-scale networks, necessitating manual checks for complex intersections.

The proposed four-stage approach can help researchers and modelers integrate the signal timing data from Synchro to SUMO more efficiently. This study delved into the intricacies of signal control mechanisms within both platforms. For future work, this framework could be foundational for extending traffic signal integration from Synchro/SUMO to other platforms. While the results in large-scale networks are promising, they still require manual validations and revisions in complex scenarios, such as the mismatching geometry coding between Synchro and SUMO and intersections with an exclusive pedestrian phase. Further refinement through specific algorithms is needed to enhance signal integration accuracy. Additionally, we envision supporting the integration of advanced signal control features, such as coordinated signal control, to elevate the quality of simulation. Beyond traffic signal control integration, this four-stage approach may hold the potential for integrating a wider range of Synchro features into SUMO, including links, routes, and detectors.

Data availability statement

The SUMO data utilized in this study are publicly accessible via GitHub (link). However, the Synchro data, derived from third-party sources, is subject to access restrictions.

Underlying and related material

GitHub Repo: https://github.com/Yiran6/Synchro2SUMO.

Author contributions

The authors confirm contribution to the paper as follows: study conception and design: Y. Zhang, M. Fu, J. Ban; data collection: Y. Zhang; analysis and interpretation of results: Y. Zhang, M. Fu; draft manuscript preparation: Y. Zhang. M. Fu, J. Ban. All authors reviewed the results and approved the final version of the manuscript.

Competing interests

The authors declare that they have no competing interests.

Funding

National Science Foundation CNS-2034615

Acknowledgement

The research is partially supported by the National Science Foundation (NSF) grant CNS-2034615. Any opinions, findings, conclusions, recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF.

References

[1] N. N. Nor Azlan and M. Md Rohani, “Overview Of Application Of Traffic Simulation Model,” MATEC Web of Conferences, vol. 150, p. 03006, 2018, doi: 10.1051/matecconf/201815003006.
[2] P. A. Lopez et al., “Microscopic Traffic Simulation using SUMO,” in 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Nov. 2018, pp. 2575–2582. doi: 10.1109/ITSC.2018.8569938.
[3] X. Zhou and J. Taylor, “DTALite: A queue-based mesoscopic traffic simulator for fast model evaluation and calibration,” Cogent Engineering, vol. 1, no. 1, p. 961345, Dec. 2014, doi: 10.1080/23311916.2014.961345.
[4] “PTV Visum.” Accessed: Apr. 24, 2024. [Online]. Available: https://www.ptvgroup.com/en-us/products/ptv-visum
[5] “PTV Vissim.” Accessed: Apr. 24, 2019. [Online]. Available: http://vision-traffic.ptvgroup.com/en-us/products/ptv-vissim/
[7] “Unity Real-Time Development Platform | 3D, 2D, VR & AR Engine,” Unity. Accessed: Feb. 10, 2024. [Online]. Available: https://unity.com
[8] J. Ban, O. Angah, Y. Zhang, Q. Guo, Connected Cities for Smart Mobility toward Accessible and Resilient Transportation Center (C2SMART), and University of Washington, “A Multiscale Simulation Platform for Connected and Automated Transportation Systems,” Dec. 2022. Accessed: Feb. 01, 2024. [Online]. Available: https://rosap.ntl.bts.gov/view/dot/67308

[9] S. Coogan and M. Thitsa, “Coordinated Anti-congestion Control Algorithm for Diverging Diamon Interchanges,” FHWA-GA-21-1913, Apr. 2021.

[10] “netconvert - SUMO Documentation.” Accessed: Jul. 27, 2021. [Online]. Available: https://sumo.dlr.de/docs/netconvert.html

[11] “MATsim - SUMO Documentation.” Accessed: Jul. 27, 2021. [Online]. Available: https://sumo.dlr.de/docs/Networks/Import/MATsim.html

[12] “Vissim - SUMO Documentation.” Accessed: Jul. 27, 2021. [Online]. Available: https://sumo.dlr.de/docs/Networks/Import/Vissim.html

[13] K. Udomsilp, T. Arayakarnkul, S. Watarakitpaisarn, P. Komolkiti, J. Rudjanakanoknad, and C. Aswakul, “Traffic Data Analysis on Sathorn Road with Synchro Optimization and Traffic Simulation,” Engineering Journal, vol. 21, no. 6, pp. 57–67, Oct. 2017, doi: 10.4186/ej.2017.21.6.57.

[14] S. K. Singh, P. Komolkiti, and C. Aswakul, “Impact Analysis of Start-Up Lost Time at Major Intersections on Sathorn Road Using a Synchro Optimization and a Microscopic SUMO Traffic Simulation,” IEEE Access, vol. 6, pp. 6327–6340, 2018, doi: 10.1109/ACCESS.2017.2739240.

[15] Trafficware, LLC., “Synchro Studio 10 User Guide.” Oct. 26, 2017. [Online]. Available: https://www.trafficware.com/synchro-studio.html

[16] “SUMO Road Networks - SUMO Documentation.” Accessed: Jan. 23, 2022. [Online]. Available: https://sumo.dlr.de/docs/Networks/SUMO_Road_Networks.html

[17] “Documentation - SUMO Documentation.” Accessed: Jul. 28, 2021. [Online]. Available: https://sumo.dlr.de/docs/index.html

[18] “Traffic Lights - SUMO Documentation.” Accessed: Jan. 23, 2022. [Online]. Available: https://sumo.dlr.de/docs/Simulation/Traffic_Lights.html

[19] M. Schrader, Q. Wang, and J. Bittle, “Extension and Validation of NEMA-Style Dual-Ring Controller in SUMO,” SUMO Conference Proceedings, vol. 3, pp. 1–13, Sep. 2022, doi: 10.52825/scp.v3i.115.

[20] M. Halbach and J. Erdmann, “High fidelity modelling of traffic light control with XML logic representation,” SUMO Conference Proceedings, vol. 3, pp. 45–68, Sep. 2022, doi: 10.52825/scp.v3i.114.

[21] A. Bundy and L. Wallen, “Breadth-First Search,” in Catalogue of Artificial Intelligence Tools, A. Bundy and L. Wallen, Eds., in Symbolic Computation. , Berlin, Heidelberg: Springer, 1984, pp. 13–13. doi: 10.1007/978-3-642-96868-6_25.

[22] “netedit - SUMO Documentation.” Accessed: Feb. 14, 2024. [Online]. Available: https://sumo.dlr.de/docs/Netedit/index.html