SUMO User Conference 2022
https://doi.org/10.52825/scp.v3i.115

Published: 29 Sept. 2022

Extension and Validation of NEMA-Style Dual-Ring Controller in SUMO

Max Schrader[1], Qichao Wang[2], and Joshua Bittle[1][https://orcid.org/0000-0003-4524-3316]

[1] University of Alabama, Tuscaloosa, Alabama, USA
mcschrader@crimson.ua.edu, jbittle@eng.ua.edu
[2] National Renewable Energy Laboratory,
Golden, Colorado, U.S.A.
qichao.wang@nrel.gov

Abstract

Until recently, SUMO users could not model the behavior of a ring-and-barrier traffic signal via existing signal types, which left North American SUMO users without the direct ability to capture traffic dynamics of their local networks. This work presents the meth-ods, implementation overview, and validation of a 'dual-ring' NEMA style traffic controller which has recently been added to the main SUMO code base. A brief explanation of the 'dual-ring' implementation is also provided as context for those new to this type of traffic controller. The foundation for this work was presented at the SUMO User Conference 2021 by researchers at the US Department of Energy's National Renewable Energy Laboratory, but was not integrated into SUMO code base at the time. Following the initial inclusion of the controller to SUMO, the authors began validation of the SUMO controller against an Econolite software-in-the-loop (SIL) traffic signal controller configured with actual setup parameters from controllers in a real-world three-intersection corridor in Tuscaloosa, Al-abama, USA. This paper documents the process of adding new features to the controller code as well as validating their implementation through simulation-based and automated grey-box testing is presented in this paper. Key features such as fully-actuated operation, various timing offset plans, proper next-phase fit algorithms and more, have been added and validated against this SIL system. Though not an exhaustive demonstration of fea-tures, this work is intended make more users aware of this extension of SUMO capabilities.

1 Introduction

Prior to Wang, Li and Jones's presentation at the 2021 SUMO User's Conference, SUMO could not capture the dynamic behavior of a standard traffic signal in North America [7, 4]. Users that wanted to model a network with 'dual-ring' traffic signals had to either sacrifice speed by building a custom SIL simulation or sacrifice accuracy by approximating the dual-ring controller in limited capacity through the existing SUMO traffic signal controller types. The remaining alternative for prospective SUMO users was to forgo it altogether in favor PTV Vissim, which has a built-in dual-ring controller module, as well as an add-on package for software-in-loop simulation with an Econolite traffic signal controller [2]. It is clear that a native implementation of the control logic within SUMO's core code base would be desirable for many current and potential future users.

Enabled by the open-source model of SUMO, the integration of the dual-ring controller into SUMO's main branch allowed for the extension of its capabilities. In both literature and practice, there are several terms used synonymously for the ring-and-barrier signal controller. Two such terms that will be used throughout this paper are "dual-ring controller" or "NEMA-type controller", which is a reference to the National Electrical Manufacturing Association (NEMA) Standards to which the controllers adhere. When testing the controller against an

Econolite software-in-the-loop (SIL) controller, it became clear that the code from [7] would have to be extended to capture the broad range of behavior possible for NEMA-based traffic signal controllers. This paper aims to describe the process of extending the ring-and-barrier controller presented in [7] to model additional operation modes. First a brief explanation of the dual-ring controller is presented. This is followed by a description of the test setup / environment which was aided by a SIL traffic signal controller. Finally, the results of validation are presented along with a summary of current features of the NEMA type controller within SUMO code base along with known features that may be added in the future.

2 Background

2.1 NEMA Dual-Ring Controller

Traffic signals in the United States adhere to the NEMA Standards, which enforce a concept of rings and barriers on the traffic signal switching logic. For succinctness, only a standard, four-way intersection is discussed below; however, the same logic can be applied to any intersection configuration.

Under NEMA standards, a phase is used to represent a certain movement at the intersection. A phase is named by a number which is usually between 1 and 8. Conventionally, the even numbers represent the through movements and the odd numbers represent the left-turn movements. The right-turn movements usually share the same phase numbers as the associated through movements. Figure 1 shows standard phase numbering for a four-way intersection. As foundation of the control logic, there are typically two barriers, which represent the separation between serving 'side' or 'main/major' streets [6]. The main side of the barrier is denoted as the side that serves the most traffic volume.

Using the intersection in Figure 1 as reference, the dual-ring phase diagram can be drawn as Figure 2. The top row in the figure comprises one ring, phases {1, 2, 3, 4}), and the bottom the other, phases {5, 6, 7, 8}. The horizontal axis in the dual-ring diagram represents cycle-time and the barriers are denoted by the double vertical grey lines. They must not be crossed unless both rings move across the barrier at the same time. On either side of a barrier, the top ring may be served with any combination of the bottom ring. For example, phases {[1, 5], [1, 6], [2, 5], [2, 6]} are all potential combinations on the mainline side of the barrier. In the same manner, phases {[3, 7], [3, 8], [4, 7], [4, 8]} are all valid for the side street barrier. Intuitively, it is clear that control should not serve both main and side streets simultaneously for safety reasons

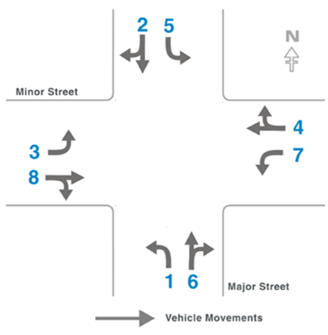

Figure 1: Typical phase numbering for a four-way intersection. Adapted from [5].

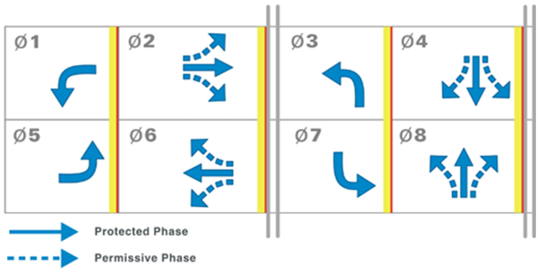

Figure 2: Typical ring-and-barrier diagram for the intersection in Figure 1. Adapted from [5].

and thus the barrier crossings must be synchronized between the two rings. In determining transitions, vehicle detectors are used in combination with a minimum and maximum time that a given phase should be served, though maximum time can be extended in certain situations.

All the phase combinations presented are valid, but the NEMA controller does constrain the transitions between states, and it depends on various manufacturer-specific settings as well as operation mode. Two of the most common operation modes are 'coordinated' and 'free' operation.

The goal of coordination is to synchronize multiple intersections, which will ideally minimize vehicle stops on the mainline roads. Coordination happens by enforcing a cycle length on the NEMA controller. The coordinated phases must be served at a regular interval equal to the cycle length, though the specifics of when and how to return to this coordinated state can vary. Figure 3 displays a ring-and-barrier diagram for the intersection in Figure 1, with added coordination annotations. It is important to note that Figure 3 is drawn with all phases at their maximum duration. In coordinated mode, each phase has a maximum and a minimum duration. Whether it lasts for the maximum, minimum or somewhere in between depends on the vehicle extension timer, which will be explained below. The cycle length is equal to the sum of each phase's maximum duration plus its transition time (yellow and red time) per ring, however it is often the case that the side street phases are not served for their maximum time. If this happens, the additional cycle time is returned to the coordinated phases and they are actually served longer than their 'maximum' duration. In this example, phases 2 & 6 are the coordinated phases (i.e. main road through movements).

The differences between three common NEMA controller conventions are displayed in the bubble callouts [5]. The ring-and-barrier diagram in Figure 3 has a leading left turn on the mainline street, meaning that phase 1 is served in conjunction with phase 6 (one of the coordinated phases) before phase 2 turns green. This is a more complex example than Figure 2, but it is helpful in illustrating the different offset types. For example, TS1 style-offsets designate the offset reference point (0 cycle time) as the time when **both** coordinated phases must be green, so the offset reference point in Figure 3 is not until phase 2 turns green as well. A TS2-syle offset designates the start of the coordinated cycle as the point when the **first** phase should be green. In the case of Figure 3 below, the first coordinated phase is 6. A Type-170 style offset sets 0 cycle time as the beginning of yellow on the earliest coordinated phase to end.

Having the offset reference point at the beginning of yellow makes the coordination easy to identify in the field. In the case of TS1 and TS2, the offset is referenced to the start of green, but only when all phases have been served their maximum allotted time. In the case when all phases haven't been served their maximum duration, the controller will return to green on the coordinated phases before the offset point. In TS2-style controllers, the coordinated phases can

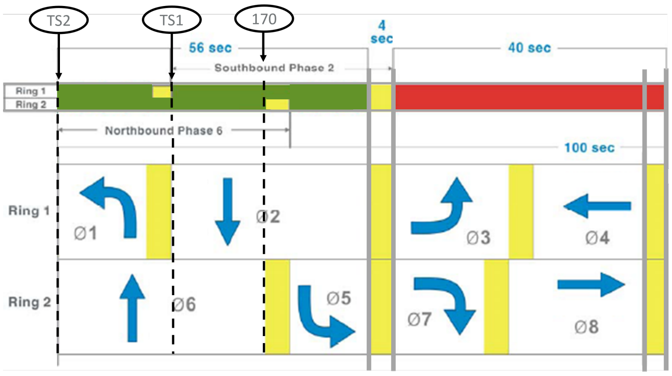

Figure 3: Dual-ring diagram for intersection in Figure 1 displaying the 3 different types of coordination with a 100 second cycle length. Adapted from [5].

also 'rest in green' if there are no vehicles detected on the side streets, which further obfuscates the coordination. While this discussion is not comprehensive, it provides some context to the motivation for including options for each style of offset.

As alluded to above, the non-coordinated phases may vary in duration and occurrence depending on controller settings. If phase skipping is possible, the controller can go directly from [2, 5] to [4, 8] if there is not a vehicle detected on either the 3 or the 7 phases' actuating detector. If phase skipping is disabled, the controller must progress from [2, 5], to [3, 7] for at least the minimum green time and then finally [4, 8]. The duration of non-coordinated phases can vary between the minimum and maximum green time, depending each phase's vehicle extension timers. Vehicle extension timers are also referred to as passage gap or passage timers. They serve to extend a phase past its minimum green time. When a phase is active and a vehicle crosses it's actuating detector, the duration of the phase is extended by the extension timer amount, as long as the addition of the extension timer to current phase duration will be greater than the minimum phase green time and less than the maximum time.

When the NEMA traffic signal controller is used at a stand-alone intersection or where traffic is sparse, traffic engineers will often use the controller in 'fully-actuated' or 'free' operation. It varies only slightly from coordinated operation, with the main difference being there is no cycle length. There is also more variation allowed in phase transition, as well as the phases which are typically coordinated ([2, 6] in case of Figure 1) being actuated. When there is infrequent traffic on the side-streets, a traffic signal in free operation will "rest-in-green" on designated phases (typically the mainline straight). In free operation and assuming that [2, 6] has been served for at least its minimum time, a transition from [2, 6] to [2, 5] or [1,5] is always valid, which is not the case in coordinated operation. During coordinated operation, a transition from [2, 6] to [1, 5] will have to wait until the possibility of serving [3, 7] or [4, 8] is exhausted. Put another way, [1, 5] cannot be served in coordinated mode unless the latest possible start time of the prior phases in the sequence has past.

The target of the initial development by Wang, Li and Jones was a coordinated, Type-170 Dual-Ring traffic signal [7]. As this controller was applied to other simulation networks, it became apparent that certain dual-ring settings and operation modes were missing. The term 'ring and barrier' traffic light describes only the core of each traffic light controller, and does not necessarily capture the additional functionality that each controller manufacture bundles with the core logic.

3 Methods

3.1 SUMO Integration

At a high level, the NEMA logic was incorporated into SUMO as a subclass of the SUMO `MSSimpleTrafficLightLogic` class, and is called `NEMAController`. The code is located at `src/microsim/traffic_lights/NEMAController.cpp` relative to the SUMO repository. The `NEMAController` logic fundamentally operates as a state machine, with the numbered phases being the state space. There are sets of transition conditions, depending on the mode of operation. Further details on the code layout are omitted for brevity.

A SUMO user indicates that a traffic light should utilize the NEMA logic by providing `type="NEMA"` in the traffic light configuration file. Indicating the traffic light is of type NEMA gives the user access to the NEMA controller settings. A typical configuration is displayed in the code block below. Information about detectors, cycle length (if coordinated), ring mapping, barrier phases, specific minimum/maximum and transition timing for each phase, and more. More details of features of the NEMA controller implementation in SUMO are provided in Section 5. There is also further explanation of the configuration parameters on the NEMA page of SUMO's website.

```
<tlLogic id="2881" offset="0" programID="NEMA" type="NEMA" offset="10">
    <param key="detector-length" value="20"/>
    <param key="detector-length-leftTurnLane" value="10"/>
    <param key="total-cycle-length" value="130"/>
    <param key="ring1" value="3,4,1,2"/>
    <param key="ring2" value="7,8,5,6"/>
    <param key="barrierPhases" value="4,8"/>
    <param key="coordinate-mode" value="true"/>
    <param key="barrier2Phases" value="2,6"/>
    <param key="minRecall" value="2,6"/>
    <param key="maxRecall" value=""/>
    <param key="whetherOutputState" value="true"/>
    <param key="fixForceOff" value="false"/>
    <phase duration="99" minDur="5"  maxDur="25" vehext="2" yellow="3" red="2" name="3" state="rrrrrrrrGrrr"/>
    <phase duration="99" minDur="5"  maxDur="25" vehext="2" yellow="3" red="2" name="7" state="rrGrrrrrrrrr"/>
    <phase duration="99" minDur="5" maxDur="30" vehext="2" yellow="3" red="2" name="4" state="GGrrrrrrrrrr"/>
    <phase duration="99" minDur="5" maxDur="30" vehext="2" yellow="3" red="2" name="8" state="rrrrrrGGrrrr"/>
    <phase duration="99" minDur="5"  maxDur="20" vehext="2" yellow="3" red="2" name="1" state="rrrrrGrrrrrr"/>
    <phase duration="99" minDur="5"  maxDur="20" vehext="2" yellow="3" red="2" name="5" state="rrrrrrrrrrrG"/>
    <phase duration="99" minDur="5"  maxDur="35" vehext="2" yellow="3" red="2" name="2" state="rrrrrrrrrGGr"/>
    <phase duration="99" minDur="5"  maxDur="35" vehext="2" yellow="3" red="2" name="6" state="rrrGGrrrrrrr"/>
</tlLogic>
```

3.2 SIL Setup

Both the development and validation of the SUMO NEMA dual-ring controller were aided by Econolite's EOS virtual controller. The virtual controller emulates a Econolite Cobalt or ATC controller running the Econolite EOS signal control software. Using the virtual controller running on local PC, configurations used by real intersections could be loaded into the virtual controller and importantly - confidently used as a ground truth.

To compare the behavior of the SUMO controller vs. Econolite, a software-in-the-loop (SIL) simulation framework was developed that coupled the Econolite EOS to SUMO. Similar to [1], the SIL framework is a Python program that maps detector calls in SUMO to the Econolite EOS and the traffic light state in the Econolite EOS to SUMO. Figure 4 depicts the SIL framework in more detail.

Communication between the Econolite EOS and the python script uses RFC 6455, also known as a websocket. The Econolite EOS broadcasts its traffic light state at a regular interval

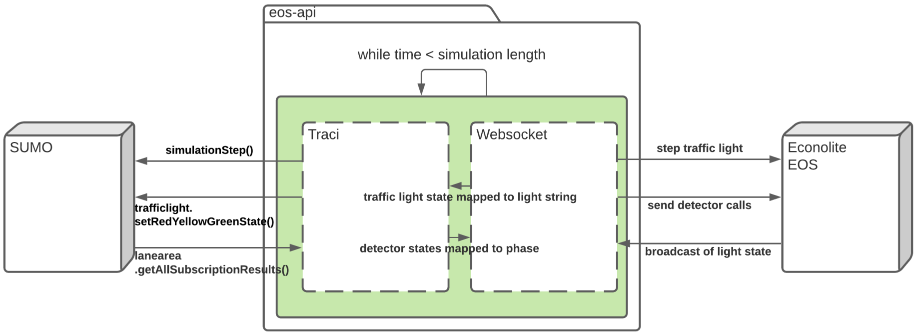

Figure 4: Schematic of the SIL framework.

and the message queue must be consumed quickly to ensure that the most up-to-date information is used. The Econolite EOS has both a pause and a 'step' feature, which allows the middleman python script to keep SUMO and the Econolite in sync.

The framework is scaleable to multi-intersection networks by running multiple instances of the Econolite EOS software. Care must be taken as the Econolite EOS defaults to using the host computers date-time and must be configured to match the real-world time that the simulation begins. The Econolite EOS's settings can be configured via the Python script, which sends websocket messages that match user inputs on the Econolite EOS GUI. Both date and time are important, as the EOS can have different day plans (similar to the 'Wochenschaltautomatik' is SUMO's traffic light logic). Further information about the Econolite implementation specifics can be shared upon request.

Developing the SIL framework and using it as a ground-truth against the SUMO NEMA controller was essential to enable building the additional features referenced in this paper. Significant time was spent investigating the various Econolite EOS settings and their corresponding responses to simulation traffic. Without coupling the Econolite EOS to the simulation, it would have been difficult to capture the true behavior of many of the features implemented and perhaps several other features such as cross-phase switching or locking detectors would have gone unnoticed. Again, an overview of currently features and remaining features to be implemented are provided in Section 5.

3.3 Test Description

Validation tests for the SUMO implementation of the NEMA controller were split into two different categories: realistic, simulation based tests and automated fuzz testing. The simulation-based tests show that the state machine transitions adhered to NEMA switching logic, or more specifically the Econolite EOS switching logic. Fuzz testing was used to send a barrage of random detector call combinations at the NEMA controller, with the intention of breaking the logic if bugs were present.

3.4 Simulation-Based Tests

For the simulation-based tests, a calibrated SUMO network representing a three intersection corridor of Tuscaloosa, Alabama was utilized. This was advantageous as the authors had access to the physical controllers in the network, and so were able to download and copy the configurations to the virtual controllers introduced in Section 3.2. The simulation encapsulated traffic from 7AM to 9AM on a representative work day where real-world detector logs allow for generating simulated traffic volumes at appropriate volume per hour at all network edges.

Figure 5 shows the SUMO model of the target network overlaid on geo-located satellite images. Each of the three intersections in the network have a different layout. The left-most is a three-way intersection, whereas the other two are four-way intersections. The two four-way intersections were specifically selected to test various scenarios, as each receives different volumes on its non-coordinated (side-street) phases from shopping centers and residential areas.

Ultimately, the goal of the simulation-based tests was to match the NEMA controller's phase and duration to that of the SIL controller exactly. The initial efforts sparked the inclusion of many new features to the Dual-Ring controller in SUMO and the testing was highly iterative. As the controller in SUMO matured, comparison on a multi-intersection network scale became possible. The results are presented in Section 4.

In the early stages the development and validation cycle, a one intersection cutout of the three intersection network was used to drive development. Figure 5 shows this one-intersection cutout as the intersection inside of the white-dashed box. The traffic signals impact traffic flow and thus comparing the SIL behavior to the SUMO-native traffic lights in a multi-intersection network is difficult unless the traffic signals operate in a very similar manner. The development of the NEMA controller was accompanied by SUMO test cases which are available in the SUMO repository, with the relative path being `/tests/sumo/basic/tls/NEMA`. Each feature added to the NEMA controller in SUMO has a corresponding test case that subsequent changes can be compared against, preventing regression. SUMO's documentation provides information on how to run each test.

3.5 Fuzz Testing

The NEMA traffic light agent interacts with the larger SUMO simulation in two fundamental ways: detector states and simulation time. Because the simulation time is intrinsically tied to the progress of the simulation, the behavior of the traffic light at any particular time is easy to analyze. On the other hand, when the NEMA traffic light has some level of actuation, different combinations and durations of detector calls are what trigger state transitions. Adding addi-

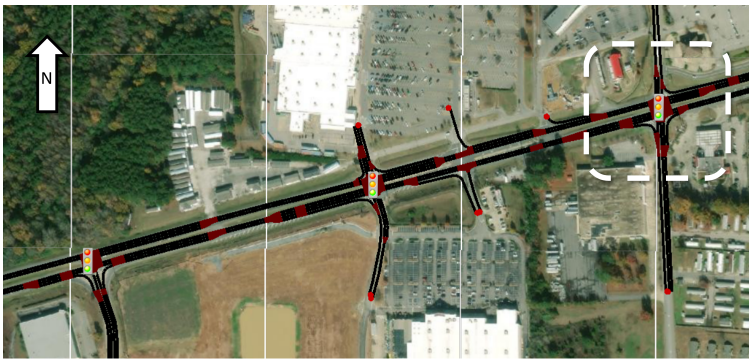

Figure 5: SUMO model of the simulated network including three intersections. Initial development was completed with the outlined sub-network.

tional features to the NEMA controller during the iterative design process scaled the complexity of the state transition logic. Though the simulation-based NEMA tests described in Section 3.4 give an idea of how the controllers perform under common traffic situations, they cover only a subset of potential traffic situations. To combat the lack of test coverage, a modified method of functional grey-box testing was employed.

There are numerous methods of grey-box testing utilized during software development. One such method is all-pairs testing, but considering the combinatorial detector state space quickly makes a comprehensive detector sweep unmanageable. In the same way, analyzing only 'critical' detector situations was considered, but designing such a test would likely been subject to the same logical issues that the code could contain. As such, an approach similar to a comprehensive detector sweep was employed, except the detectors on/off times and combinations were chosen at random. In software testing, this approach is sometimes referred to as targeted fuzz-testing, where automated tests generate random inputs to find software bugs and vulnerabilities [3, 8].

The automated detector tests were implemented using TraCI and a newly built API to override detector calls in SUMO. Prior to running the simulation, a set of randomly generated detector 'on' (corresponding to 1 vehicle on the detector) and 'off' (0 vehicles on the detector) times for each detector in the network were generated by iteratively sampling from a uniform distribution. Equations 1 and 2 below show how a series of detector calls was generated. Starting with the detector off ($\text{Off}[0] = 0$), then generating first on time ($\text{On}[0]$), then the second off time ($\text{Off}[1]$), and so on.

$$\text{On}[i] = \sum_{k=0}^{i} U(0, N] + \text{Off}[k], \text{ for } i = 0, 1, \ldots \text{Off}[i] \geq T \tag{1}$$

$$\text{Off}[i] = \sum_{k=1}^{i} U(0, N] + \text{On}[k-1], \text{ for } i = 1, 2, \ldots \text{Off}[i] \geq T \tag{2}$$

$U(0, N]$ represents a sample of a uniform distribution between 0 and N (the cycle length). Off and On represent vectors of simulation times where the detector should turn off and on respectively. The summation continues until the calculated detector off time is greater than the specified simulation time.

Assertions were added to the NEMALogic code to forcefully highlight bugs in the logic. In fully-actuated tests, there were two basic assertions:

- Active phases must be on the same side of the barrier, i.e. in Figure 2, phases 2 & 7 should never be served together.
- Each phase must last at least as long as it's minimum time.

In coordinated mode, an additional assertion was added which ensured that:

- The coordinated phases must be green at the start of their coordinated period.

This pass-fail logic was then applied to the tests in the SUMO repository, which include various intersection layouts as well as combinations of configuration s ettings. T he fuzz testing was also applied to a single intersection cut-out of the network presented in Section 3.4. It should be noted that proper testing would also sweep all combinations of user configurations. While the authors have such tests planned, the results are not included in this paper.

4 Validation

This section includes validation results of the SUMO NEMA Controller. Results are presented broken down into the two primary operation modes (coordinated and free) and for both simulation-based and the grey-box fuzz testing.

4.1 Coordinated Operation

The NEMA controller was developed and extended with all operation modes in mind, but the main target was coordinated mode. Coordination is favored by traffic engineers where there are several intersections that are close together and is the operation mode utilized by the target simulation network in the field. Coordination is enabled in the SUMO NEMA controller by passing `<param key="coordinate-mode" value="true"/>` in the traffic light configuration file.

4.1.1 Simulation-Based Testing

In the real network presented in Section 3.4, the three intersections operate almost exclusively in coordinated mode, with a 20 second offset between each controller. Figure 6 presents the result of controller development: identical response to traffic as a time history of the active green/yellow phases and coresponding detector calls for both controllers. The SUMO controller's behavior is shown above the phase on the y-axis, and the Econolite SIL controller is below. The dark vertical lines show the configured controller's cycle reference-point, which is a TS2 style offset. The offset type can be set via `<param key="cabinetType" value="TS2"/>` parameter in the traffic light configuration file.

In Figure 6, there are several side street phase progressions that occur. At 3250 seconds into the simulation, the controller progresses from [2, 6] to [3, 8] and back to [2, 6], which indicates light traffic on the side street. At 3700 seconds, the controller goes from [2, 6] to [3, 7], then to [3, 8], then [2, 5] and finally [2, 6]. This progression shows the "green transfer" functionality, as phase 3 stays green going from [3, 7] to [3, 8]. There are no detector calls on either phases 3 or 4 during this transition, so the controller behavior is to leave the existing phase (3) green. Phase 1 is never served explicitly in this simulation period, but by analyzing the detector calls in Figure 6 it is clear that there were no detector calls on phase 1 during transition periods.

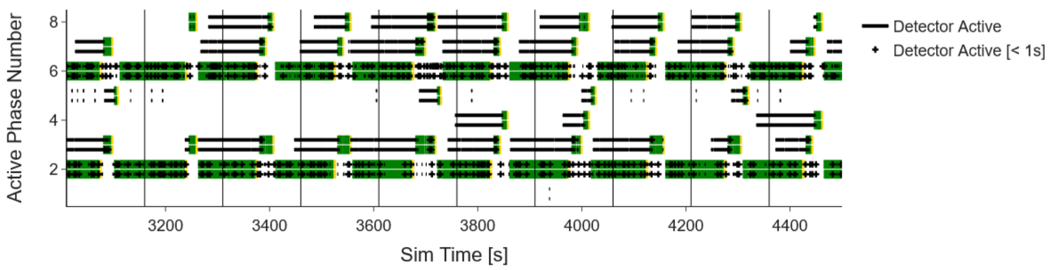

Figure 6: Visual comparison of SUMO NEMA Logic vs. Econolite in the presence of the same traffic demand. SUMO behavior is displayed slightly above the phase number and Econolite EOS below. Detector calls are shown as black crosses when their duration is less than one second and as a horizontal black line when longer than 1 second.

While it was unreasonable to present a plot comparing all phases of all intersections, the test will be added to SUMO as a test case, meaning that the results can be reproduced and analyzed by all SUMO users.

In addition to looking at all 8 phases of a single intersection, it was important to verify that all three NEMA controllers in the SUMO simulation worked together in coordinated mode. The effects of coordination are frequently viewed through the use of space-time diagrams, which show the effect of multiple traffic signals on traffic flow. Figure 7 presents the space-time diagrams of two simulations: one with SUMO NEMA controllers and one with SIL traffic signal controllers. In both Figure 7b & Figure 7a the eastbound (EB) vehicles are shown as the solid black lines and westbound (WB) as dashed. Only phase 6 of each of the three intersections is plotted, with the color of the horizontal line corresponding to the three intersections' light state. The light states are plotted at the distance each intersection is from the EB network edge.

The benefits of coordination on traffic flow are clear, with traffic progressing with constant velocity through the network during periods of all green. Comparing the two sub-figures reveals little to no difference, which gives the authors confidence that TS2-style offsets and coordination is working as expected in the SUMO NEMA controller. In fact, the two simulations are indistinguishable in the period analyzed.

4.1.2 Fuzz Testing

Fuzz testing the controller in coordinated mode surfaced several bugs in the logic that have been addressed. As an example, the algorithm which computes whether a phase will 'fit' inside of the cycle time was incomplete. This becomes important with phase-skipping functionality.

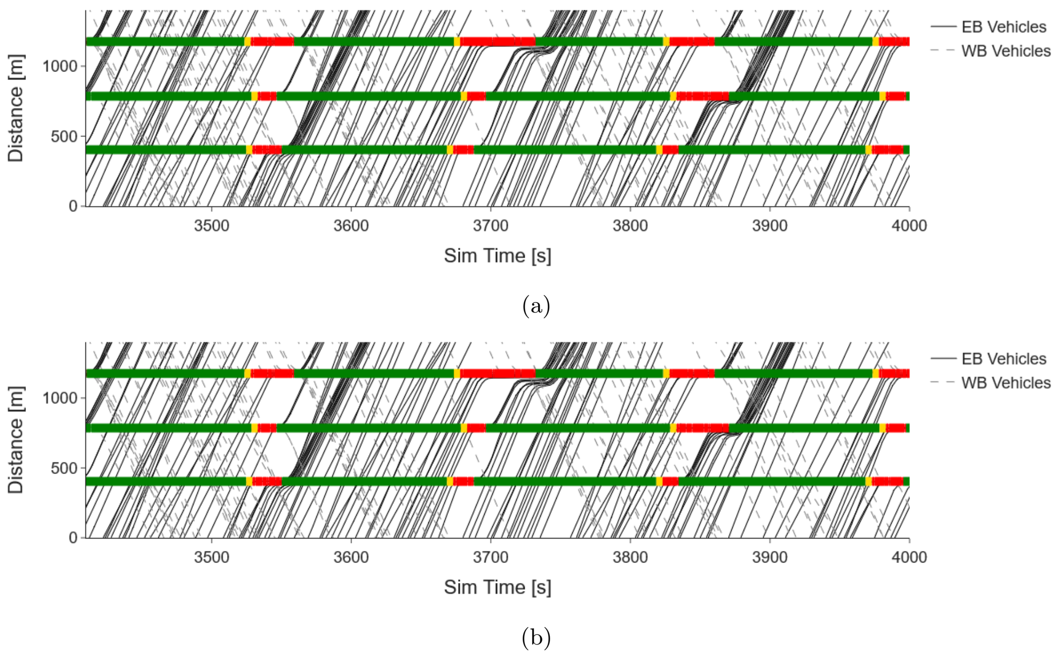

Figure 7: Space-time diagram of SUMO traffic lights (a) and the Econolite traffic lights in SIL (b) described in Section 3.4. The space-time diagrams are virtually identical.

In addition to the phase fit algorithm, certain combinations of detector calls during a yellow to red transition were found to cause the controller to 'reverse' away from a barrier, meaning that in Figure 2, phase 8 was transitioning to 7, which should not happen in coordinated mode.

4.2 Free Operation

In free mode, the controller has more freedom than in coordinated mode, thus the SIL controller and SUMO NEMA controller diverge more frequently, especially when the divergence of one is propagated through the three intersections of the network under study.

4.2.1 Simulation-Based Testing

An example of differences between the SIL controller and SUMO during simulation testing of free operation can be seen in Figure 8. As with Figure 6, the SUMO NEMA controller phases have been plotted above the phase number and Econolite SIL below. In the first half of plotted simulation time (3500 - 3800s), some of their behavior looks quite different such as during the period encircled with the red-dashed overlay. Inspection of the detector calls inside of the overlay reveal that vehicles cross the phase 6 detector in the SIL simulation around 3740 seconds into simulation, which extends the [2, 6] phases. Those vehicles do not cross the detector in the SUMO-native simulation, which is likely due to a difference in the upstream behavior of a different traffic signal.

Because of a limitation in the SIL controller implementation, there is a one simulation step delay on detector calls. This lag between SUMO detectors and what the SIL controller sees leads to differences in the vehicle extension timer and then ultimately the phase length. Knowing the limitations of the SIL setups and the degrees of freedom that a free dual-ring controller has, the authors are confident that SUMO is capturing the behavior of the SIL controller correctly.

4.2.2 Fuzz Testing

As in Section 4.1.2, fuzz testing the controller in free mode also surfaced bugs. For example, combinations of detector calls that occurred during a transition from [2, 5] to [1, 6] could ultimately lead to the barrier being crossed by one ring and not the other. The bug was since fixed by enforcing stricter logic on barrier cross transitions.

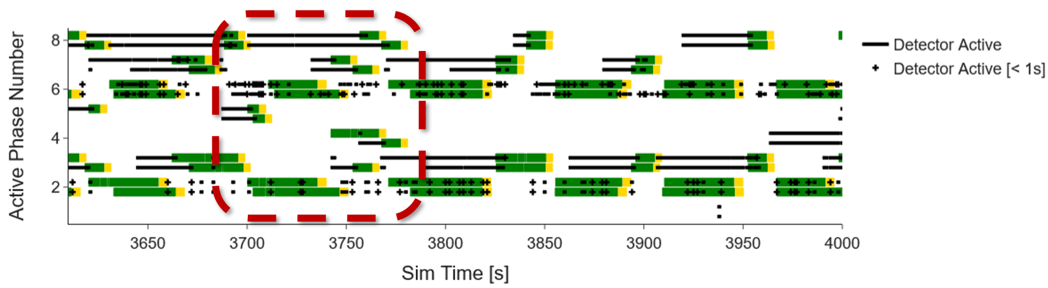

Figure 8: Visual comparison of SUMO NEMA Logic vs. Econolite for a select period of the simulation. The phase are not identical, but the behavior in response to detector calls is.

4.3 Simulation Speed

Preserving SUMO's standalone simulation speed was one of the main goals of integrating the NEMA controller into SUMO. As discussed, there were SIL alternatives, but SIL simulations are slower and more resource intensive. Table 1 presents a comparison of simulation real-time factor for the two options, which is the equivalent to $\frac{\text{simulated time}}{\text{computation time}}$. The two columns (1 Intersection and 3 Intersections) represent the simulations discussed in Section 3.4. Each simulation was ran with a 0.1 second step length for and lasted 6900 seconds. The route file and random seed was the same for each 1 Intersection and 3 Intersection simulation.

Table 1: Comparison of simulation real time factor for the SIL and SUMO-native NEMA methods.

	Network Size	
Controller Method	**1 Intersection**	**3 Intersections**
SIL With EOS	31.5	21.6
SUMO NEMA	205.2	180.0

While it's not a comprehensive simulation speed test, the brief comparison of the Econolite SIL simulation presented in Section 3.2 against the built-in NEMA controller makes the speed penalty of the SIL implementation clear. With three intersections, the standalone SUMO simulation has a real-time factor of 180.0, which is roughly 8.5x faster than the same SIL simulation. The one intersection simulation is 6.5x faster in standalone mode. The ratio between SUMO-standalone real-time factor and the SIL real-time factor will continue to increase as intersections are added to the simulation network.

5 Conclusions and Future Work

As this paper has shown, the authors have attempted generality in their implementation of the NEMA controller. Care was taken to test against multiple configurations and intersection layouts. At the same time, the only virtual traffic signal controller available to the authors was the Econolite EOS and thus there is potential that the SUMO integration is 'overfit' to the Econolite EOS traffic signal controller software.

Table 2 presents some of the features implemented, as well as features that may be useful for other users but have not been implemented yet.

The authors are hopeful that the SUMO community will see the newly-integrated controller as a big step forward for North American users and will be willing to contribute to the code-base or reach out to the authors when they see a missing feature. In addition to the features not included in Table 2, one of largest outstanding tasks at the time of writing is to incorporate the NEMA controller configuration into netedit.

6 Acknowledgments

Incorporating the NEMA controller would have been impossible without the support of the entire SUMO team. Specifically, the authors would like to thank Jakob Erdmann for his guidance integrating the NEMA Controller into SUMO's source code as well as being a sounding board for new features and parameters.

This work was supported by The University of Alabama's Center for Advanced Vehicle Technology (CAVT). This material is also based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the the Vehicle Technologies Office (VTO) and through Energy Efficient Mobility Systems (EEMS) program award number DE-EE0009210. The views expressed herein do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

Table 2: Coverage of NEMA-type controller settings

Feature	**Included**	**Notes**
Green Rest	✓	Econolite Implementation
Green Transfer	✓	Econolite Implementation
Fully-actuated	✓	
Latching Detectors	✓	Basic Implementation
Cross-phase Switching	✓	Econolite-style Implementation
Detector Delay		
Detector Lock-In Time		
Phase Recall	✓	Min/Max Recall. Detector Recall Missing
Fix/Float Force Off	✓	Bool on/off, not per phase
Dual Entry		
Red Revert		
Type-170 Offset	✓	
TS1 Offset		
TS2 Offset	✓	

References

[1] Mirko Barthauer and Bernhard Friedrich. External signal control: Integrating LISA+ into SUMO. In *SUMO User Conference*, pages 143–153, 2017.

[2] Martin Fellendorf. VISSIM: A microscopic simulation tool to evaluate actuated signal control including bus priority. In *64th Institute of Transportation Engineers Annual Meeting*, volume 32, pages 1–9. Springer, 1994.

[3] Patrice Godefroid. Fuzzing: Hack, art, and science. *Communications of the ACM*, 63(2):70–76, 2020.

[4] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018.

[5] Tom Urbanik, Alison Tanaka, Bailey Lozner, Eric Lindstrom, Kevin Lee, Shaun Quayle, Scott Beaird, Shing Tsoi, Paul Ryus, Doug Gettman, Srinivasa Sunkari, Kevin Balke, and Darcy Bullock. *Signal Timing Manual - Second Edition*. The National Academies Press, Washington, DC, 2015.

[6] Qichao Wang and Montasir Abbas. Optimal Urban Traffic Model Predictive Control for NEMA Standards. *Transportation Research Record*, 2673(7):413–424, 2019.

[7] Qichao Wang, Tianxin Li, and Wesley Jones. Augmenting SUMO with Ring-and-Barrier Structured Traffic Signal Controller Module [Conference Presentation]. In *SUMO User Conference*, 2021.

[8] Andreas Zeller, Rahul Gopinath, Marcel Böhme, Gordon Fraser, and Christian Holler. The Fuzzing Book, 2019.