Vol. 7, No. 1, February 2025, pp. 103~112, p-ISSN 2685-2381, e-ISSN 2715-2626
http://dx.doi.org/10.28989/avitec.v7i1.2823

Development of Coordinated Control of Vehicle Traffic Flow at Adjacent Intersection

Freddy Kurniawan[1]*, Muzammil Jusoh[2], Bahodir Muminov[3], Hermansyah Alam[4], Denny Dermawan[1], Muhamad Jalu Purnomo[5]
[1]Department of Electrical Engineering, Institut Teknologi Dirgantara Adisutjipto, Indonesia
[2]Faculty of Electronic Engineering Technology, Universiti Malaysia Perlis, Malaysia
[3]Department of Artificial Intelligence, Tashkent State University of Economics, Uzbekistan
[4]Department of Electrical Engineering, Universitas Islam Sumatera Utara, Indonesia
[5]Department of Aeronautics, Institut Teknologi Dirgantara Adisutjipto, Indonesia

Article Info

Article history:
Submitted January 21, 2025
Accepted February 27, 2025
Published February 27, 2025

Keywords:
Coordinated traffic control;
agent;
vehicle platoon;
vehicle wait time.

ABSTRACT

Traffic congestion in areas with two closely situated traffic lights is a complex issue that is often difficult to resolve. To address this issue, a coordination of timing between the two traffic controllers is proposed. This research conducts an experiment with two traffic controllers at two nearby intersections. The vehicle flow at each intersection is managed by the Agent acting as the traffic controller. The agent where more vehicles arrive is designated as the master agent, while the other agent is designated as the slave agent. A coordination algorithm is developed to synchronize the timing of the traffic controller so that the timing at the slave Agent was adjusted according to vehicle platoon arrivals from the master Agent. By this method, the green phase of the slave agent can be synchronized with the master agent, allowing vehicle platoons arriving from the master agent to immediately receive a green phase at the slave agent. This coordinated traffic control can be implemented with a microcontroller-based system, and vehicle movement can be simulated using Matlab's SimEvents. From the experiment conducted for two intersections located 500 meters apart, this scheme can reduce the average vehicle wait time from 40 seconds to just 9.4 seconds.

Corresponding Author:
Freddy Kurniawan,
Department of Electrical Engineering, Institut Teknologi Dirgantara Adisutjipto,
Jl. Janti, Blok R, Lanud Adisutjipto, Yogyakarta, Indonesia.
Email: *freddykurniawan@itda,ac,id

1. INTRODUCTION

Traffic congestion is a major issue in many countries, including Indonesia, leading to various problems such as longer travel times, higher fuel consumption, increased vehicle maintenance costs, and greater carbon emissions [1]. This issue is particularly prominent during peak hours in urban areas. A typical phenomenon of urban congestion is the long queues of vehicles at traffic signals. This is usually caused by the volume of vehicles passing through exceeding the capacity of the intersection [2][3].

One of the main factors contributing to traffic congestion is the improper timing of traffic signals. Many traffic controllers in this country operate on standalone systems with fixed timing settings [4]. During peak hours, traffic jam commonly occurs in areas. When the phenomena occur at closely spaced traffic controllers, long queues form at traffic controllers, sometimes extending to nearby intersections and causing widespread traffic jam [5].

Efforts to alleviate congestion often involve alternative transportation investments such as subways, light rail systems, trams, carpooling, and expanded bicycle infrastructure. These approaches primarily aim to reduce vehicle dependency, particularly in areas prioritizing emission reductions. Another method focuses on increasing road capacity through expansion or optimization. However, expanding road networks is costly and often leads to temporary bottlenecks during construction, with limited long-term impact on reducing congestion [6]. Among the less intrusive and more cost-effective solutions is optimizing existing road infrastructure. This includes dynamic traffic signal adjustments, which have been recognized as one of the most efficient methods for improving traffic flow. In Indonesia, where thousands of traffic signals operate, retiming traffic signals could significantly reduce congestion. However, in practice, signal performance assessments are infrequent, and there is no comprehensive evaluation framework to measure the effectiveness of retiming efforts [7].

One of the efforts made by the Indonesian government is the implementation of the ATCS (Area Traffic Control System). Many traffic controllers in major cities in Indonesia have been equipped with this system. Through ATCS, traffic conditions at intersections can be monitored from the Department of Transportation office via CCTV cameras. Ideally, officers adjust the timing of each traffic controller when it no longer aligns with traffic conditions. However, in practice, the cameras only serve for monitoring purposes and do not contribute to adjusting the timing of the traffic controllers. In reality, the timing for signals at traffic controllers tends to remain fixed, as it is rarely modified by officers. Currently, the ATCS primarily functions as a surveillance facility and a tool for detecting traffic violations (Electronic Law Enforcement) [8][9].

Several developed countries have implemented Intelligent Transportation Systems (ITS) for traffic management. This system, integrated with numerous sensors, can monitor traffic density on all road segments, create timing scenarios for each traffic signal, and send the timing data to each traffic controller [10]. The system also determines the best route for each driver to avoid congestion [11]. Using this method, drivers can get more frequently encounter green lights at closely spaced traffic signals and avoid traffic jams. This system effectively reduces congestion and minimizing vehicle travel time [12]. However, the system has not yet been implemented in Indonesia due to the lack of infrastructure, differences in road segment characteristics, road user behavior, and government regulations [13][14][15].

In this study, we propose one solution to address the issue by coordinating control of vehicle traffic flow, especially at adjacent intersections. This coordination is achieved through the use of coordinated traffic controller system. In this research, the timing of all traffic controllers is coordinated so that the most of vehicles traveling on the main route receive green signals as they arrive at the next traffic controller [16][17], while minimizing delays for other vehicles on the main route [18]. This approach could be more appropriate for implementation in Indonesia and other developing countries, where real-time traffic data is not yet widely accessible. Relying on in-situ sensors near traffic lights may not yield accurate traffic information due to infrastructure constraints and, in certain situations, could even create safety hazards for road users.

A prototype based on this research has been developed in previous studies [19]. The prototype is designed to be developed in traffic controllers in Indonesia, which previously relied on stand-alone and fixed-time traffic controllers. However, a comprehensive analysis of the system's performance has not yet been conducted. Therefore, this study presents the results of an analysis of the performance of coordinated control of vehicle traffic flow at adjacent intersections in addressing traffic congestion issues, particularly at adjacent intersections. The testing was conducted using simulations in SimEvent from MATLAB 2023b Simulink for two adjacent intersections operated with fixed-time traffic control. The benefits of the modelling and simulation are can be used for analyzing traffic flow and predicting congestion at adjacent intersections, as well as possible solutions that can be taken to solve the congestion problems at the wider area.

2. RESEARCH METHODS

2.1 Hardware description

Most traffic lights in Indonesia are designed based on microcontrollers. In light of this fact, a prototype of a coordinated traffic controller based on microcontrollers was developed in previous research [19]. The aim is to enable the implementation of coordination by enhancing the existing system. Figure 1 shows the block diagram of the traffic controller.

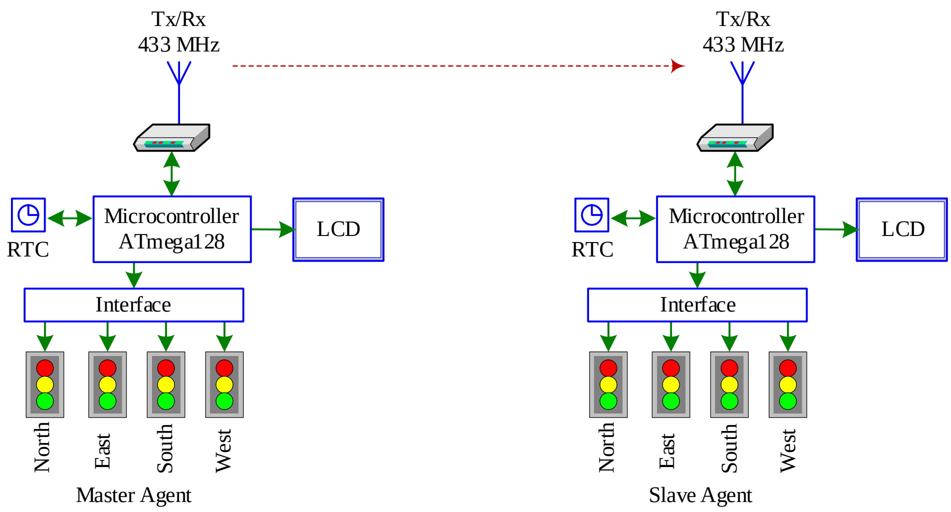

Figure 1. Block diagram of a coordinated traffic controller

The traffic control algorithm was implemented in a program on an ATmega128 microcontroller, forming an Agent. An Agent managed traffic at an intersection using fixed-time schedules. Each Agent has a timing data for red, yellow, and green signals stored in the microcontroller's EEPROM. This data served as the basis for traffic management throughout the day, applicable to both weekdays and weekends. The scheduling consists of 10 time slots per day. An Agent can also coordinate timing with other Agents. This timing coordination adjusts the timing of both Agents according to the predetermined scenario. Coordination commands are transmitted wirelessly using FSK modulation at a frequency of 433 MHz [19].

2.2 Traffic Control and Vehicle Flow Simulation

This objective of this research is to analyze the performance of the developed prototype. The focus is on analyzing the performance of vehicle flow control at two adjacent intersections as shown in a model in Figure 2. Specifically, this study investigated the impact of the proposed system on the average vehicle waiting time of vehicles.

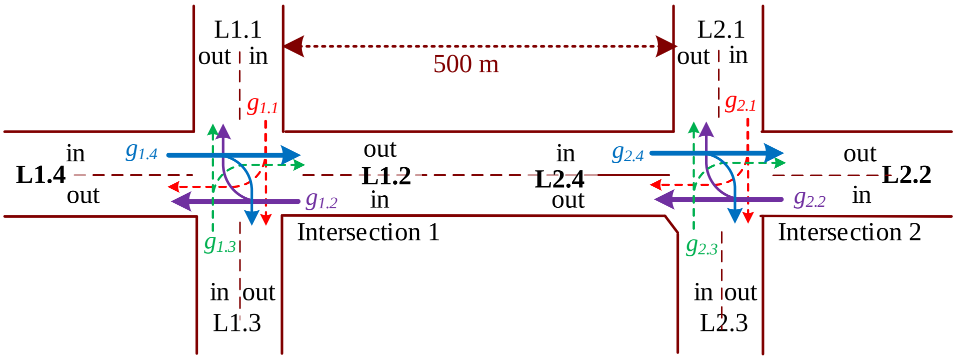

Figure 2. Vehicle flow control at two adjacent intersections

The following assumptions were used in this modeling.

1. The distance between the two intersections is 500 meters.
2. Agent 1 manages vehicle flow at Intersection 1, while Agent 2 manages at Intersection 2.
3. Each intersection has four incoming directions with a clockwise phase sequence, controlled individually by each agent.
4. Each green phase allows for straight-through and right-turn movements, while left turns are allowed continuously. This green phase is denoted by $g_{m,n}$, where m represents the intersection number and n represents the lane number.
5. Vehicle flow from lanes L1.4_in to L2.2_out (depicted by a bold blue arrow) and L2.2_in to L1.4_out (illustrated by a bold purple arrow) represent the main traffic directions, where most vehicles travel.
6. Vehicles are only permitted to use a speed of 40 – 60 km/h.

We employed a computer simulation to comprehensively analyze this phenomenon. The computer simulation was developed using SimEvents in MATLAB-Simulink as shown in Figure 3, offers a more realistic representation of traffic flow, allowing for the analysis of various traffic scenarios to gain a deeper understanding of the factors influencing waiting times at intersections and to identify potential strategies for improving traffic flow.

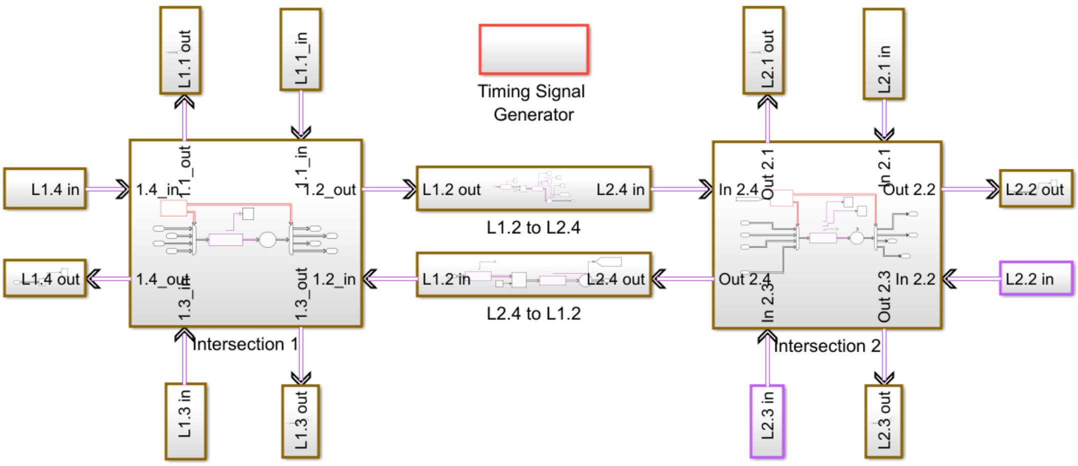

Figure 3. Computer simulation developed using SimEvents in MATLAB-Simulink

In the simulation, each vehicle from each lane is represented by an Entity which is generated by an Entity Generator. To simulate traffic controlling, the flow of each Entity was queued in the Entity Queue. Figure 4 illustrates the model used to queue vehicles in lane L1.1_in. Entity Queue 1.1 simulates the queue of vehicles in lane L1.1_in that are waiting to enter Intersection 1. The model for queuing vehicles from other lanes was created in a similar manner.

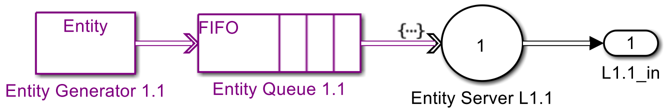

Figure 4. Model to simulate traffic controlling in L1.1_in

Agent 1 acts as the traffic signal controller at Intersection 1. This function is simulated by the Traffic Controller Subsystem 1 as shown in Figure 5. This subsystem regulates the green phase by providing data on the direction receiving the green phase to Entity Input Switch 1. The rotation of the green phase for each lane is simulated by alternating the entity input switch to pass entities sequentially from L1.1_in to L1.4_in. The algorithm in the Traffic Controller 1 subsystem also regulates the simulation of vehicle flow from Entity Input Switch 1. After the vehicle entities pass through Entity Queue 1 and Entity Server Agent 1, each vehicle at Intersection 1 is simulated to exit at the output lane other than the lane where it entered. The routing of all vehicles exiting Intersection 1 is controlled and distributed by Entity Output Switch 1.

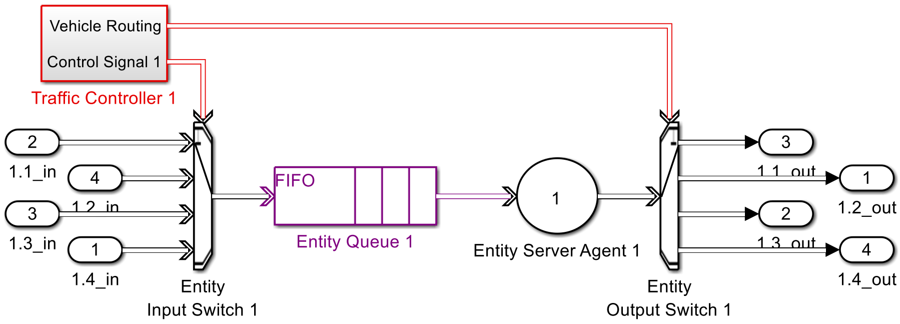

Figure 5. Agent 1 regulated green phase and vehicle movement at intersection 1.

2.3 Coordination Process between Two Agents

The coordination of two Agents is established using a master-slave configuration. The master Agent is designated for an intersection where more vehicles enter the main road. Referring to Figure 2, if there are more vehicles moving from left to right (as indicated by the blue arrow) than those moving from right to left (as indicated by the purple arrow), then Agent 1 acts as the master agent and Agent 2 as the local. Consequently, the traffic signal timing at Agent 2 will be adjusted to align with the timing at Agent 1. Likewise, if the opposite occurs that more vehicles move to the left (in the direction of the purple arrow), then Agent 2 is the master and Agent 1 is the slave and Agent 1's timing follows Agent 2.

Each agent has a designated green phase reference, which is the reference green phase used for coordination purposes. Coordination between agents is implemented to synchronize the reference phase timing of the slave agent with the reference phase timing of the master agent. This ensures that the vehicle platoon traveling from the master agent will arrive at the slave agent during the green phase. We employ the onset of the reference green phase of the master agent denoted as $t_{g:M.n}$ and the onset of the reference green phase of the slave agent symbolized as $t_{g:S.n}$ as the reference time for synchronization. With n being the sequence of the onset of the reference green phase. The following Algorithm 1 is used to synchronize the timing of the slave Agent with the master Agent.

Algorithm 1: Synchronize slave Agent timing with master Agent

1. At $t_{g:M.n}$, the master agent sends a synchronization signal ('sync') to the slave agent.
2. The slave agent timestamps $t_{g:M.n}$.
3. $t_{g:S.n\,(\text{target})} = t_{g:M.n} + \tau_{M\rightarrow S}$
4. $\xi = t_{g:S.n\,(\text{target})} - t_{g:S.n}$; where $t_{g:S.n} = t_{g:S.n\text{-}1} + C$.
5. If $|\xi| < \xi_{th}$, then go to step 6
6. $\Delta g_i = \frac{\xi}{4}$
7. If $\Delta g_i > upper\_limit$ then $\Delta g_i = upper\_limit$
8. If $\Delta g_i < lower\_limit$ then $\Delta g_i = lower\_limit$
9. Return to the main program

In the initial step of the algorithm, at the onset of the reference green phase $t_{g:M.n}$, the master agent sends a synchronization signal (‘sync’) to the slave agent. The slave agent timestamps the reception of the signal. After that, in the third step, the slave agent determines the target of the onset of the reference green phase, denoted as $t_{g:S.n\ (\text{target})}$, by adding the estimated travel time of a vehicle from master agent to slave ($\tau_{M\rightarrow S}$) to $t_{g:M.n}$. Subsequently, the slave agent calculates the synchronization error ξ, which is the time difference between next onset of the reference green phase $t_{g:S.n}$ and $t_{g:S.n\ (\text{target})}$; where $t_{g:S.n}$ can be calculated by adding the cycle length (C) to the onset of the past reference green phase $t_{g:S.n\text{-}1}$. Cycle length is the time required for a complete sequence, composed of the total signal time to serve the all-signal phases that including the green time plus any change interval (yellow and red-clearance). Essentially, if the error is equal to zero, then the timing of agent 2 is already synchronized with the timing of agent 1. Meanwhile, a time-shifting effort of the green phase will be carried out if the error value is not equal to zero. However, to maintain system stability, a threshold is set for tolerance, so that if the absolute value of the error is still below the threshold ξ_{th}, then slave agent's timing is still considered synchronized with master agent. When the timing in slave agent is not synchronized with master, the error value was distributed to the green time of each phase (Δg_i) as in step 6 of Algorithm 1. Consequently, a positive error value would lengthen the green time, while a negative error value would shorten it. The former case extends the current cycle length (C_k), and the latter reduces it.

In this way, the synchronization process can be completed within just one cycle. However, if the synchronization error is significant, it can lead to substantial changes in green time, which may cause discomfort for road users. A significant increase in green time can result in wasted green time, as there may be no more vehicles passing through once the queue has been cleared. Additionally, a significant increase in green time can cause vehicles from other directions, which are currently in the red phase, to wait longer for their green phase. On the other hand, a highly negative synchronization error can lead to a significant reduction in green time. This can also cause discomfort for road users and may prevent all vehicles in the queue from passing during the green time. Therefore, steps 7 and 8 of Algorithm 1 are used to limit the changes in green time. Typically, a considerable increase in green time is still tolerable for users and does not significantly disrupt traffic flow. As a result, the tolerance for green time changes is greater in the positive direction (increasing green time) than in the negative direction (decreasing green time).

Figure 6 illustrates the synchronization process within one cycle, where the change in green time does not exceed the $upper\_limit$ or $lower\_limit$. The master agent periodically sends ‘sync’ to the slave agent at $t_{g:M.n}$ (indicated by a downward blue arrow). The first case in the figure depicts the situation where slave agent timing has been already synchronized with master agent timing. In this case, $t_{g:S.n\ (\text{target})}$ is equal to $t_{g:S.n}$, resulting in $\xi = 0$ and the current cycle time remains unchanged ($C_k = C$).

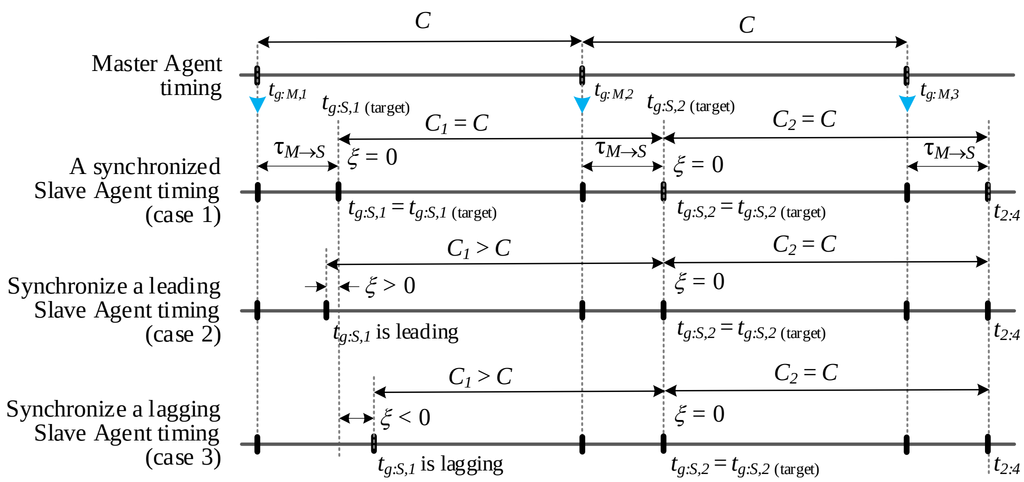

Figure 6. Synchronization process of slave Agent with respect to master Agent

Asynchronous timing conditions can occur, as in case 2, where the next onset of the reference green phase occurs earlier than targeted. The $t_{g:S.1}$ value smaller than $t_{g:S.1\ (\text{target})}$ results in a positive error ξ. To eliminate this error, line 6 of Algorithm 1 distributes the value of ξ to increase the green time (Δg_i) for each phase. As the green time for each phase increases, the current cycle length (C_k) of slave Agent also increases. The next cycle, $C_2 > C$. This cycle time adjustment only applies for one cycle. In the next cycle, since $\xi = 0$, The slave Agent cycle length returns to its initial value.

In the third case, slave Agent timing lags behind master Agent, resulting in a negative error ξ. In this case, line 6 of Algorithm 1 distributes the negative value of ξ to adjust the green time for each direction, so that, the green time for all phases decrease. The reduction in green time decreases slave Agent cycle time. This reduction also only applies for one cycle.

3. RESULTS AND DISCUSSION

The proposed method aims to coordinate the timing of slave agent to synchronize with the master agent. This coordination is intended to ensure that a vehicles platoon departing from the master agent can immediately receive a green light upon arrival at the slave agent, or at least experience a minimal waiting time for the green phase. This method is expected to reduce the travel time of most vehicles traversing both intersections controlled by the two agents. As illustrated in Figure 2, most vehicles travel on the main two-way road. In the first route, vehicles enter intersection 1 via L1.4_in, exit via L1.2_out, and then travel to intersection 2, entering via L2.4_in and exiting via L2.2_out (as indicated by the blue arrows). In the second route, vehicles enter intersection 2 via L2.2_in, exit via L2.4_out, and then go through to intersection 1, entering via L1.2_in and exiting via L1.4_out (as indicated by the purple arrows). Therefore, the development of the vehicle flow coordination control method focuses on the phase adjustment by both agents to minimize the travel time on both main routes.

Agent 1 and Agent 2 are two agents that were previously independent of each other. Both agents have the same green phase duration. Green phases of lane 2 ($g_{1.2}$ and $g_{2.2}$) and 4 ($g_{1.4}$ and $g_{2.4}$) have a duration of 30 seconds, while green phases of lane 1 ($g_{1.1}$ and $g_{2.1}$) and 3 ($g_{1.3}$ and $g_{2.3}$) have a duration of 25 seconds. These durations may change during the synchronization process. Meanwhile, each direction has a yellow phase and red-clearance phase duration of 3 and 2 seconds, respectively. The duration of these phases does not change during the synchronization process.

Lanes L1.4 and L2.2 were identified as having relatively high vehicle arrival rates, while the other lanes exhibited less congested arrival patterns. In the simulation, we modeled lanes L1.4 and L2.2 with a specific arrival pattern, characterized by a mean inter-arrival time of 2.44 seconds, while the remaining lanes were modeled with a mean inter-arrival time of 3.5 seconds.

3.1 Vehicle Movement Analysis

The vehicular movement at intersection 1 is regulated by Agent 1. During the green phase $g_{1.4}$, the majority of vehicles from L1.4_in enter Intersection 1 and exit to L1.2_out. The simulation of vehicle movement can be viewed in the Sequence Viewer window as shown in Figure 7. This figure provides a simulated visualization of vehicle trajectories, illustrating vehicles from lane L1.4_in traversing into intersection 1. At t = 83.7 seconds, a yellow highlighted line shows a vehicle departed from L1.4_in entering intersection 1 and subsequently exiting onto the lane connecting L1.2 and L2.4. Prior to this instance, a vehicle was observed entering and exiting to lane L1.1_out. Several vehicles that entered afterwards can also be observed in the diagram. It is evident that the majority of vehicles entering from L1.4_in exit to L1.2_out towards the lane connecting L1.2 to L2.4.

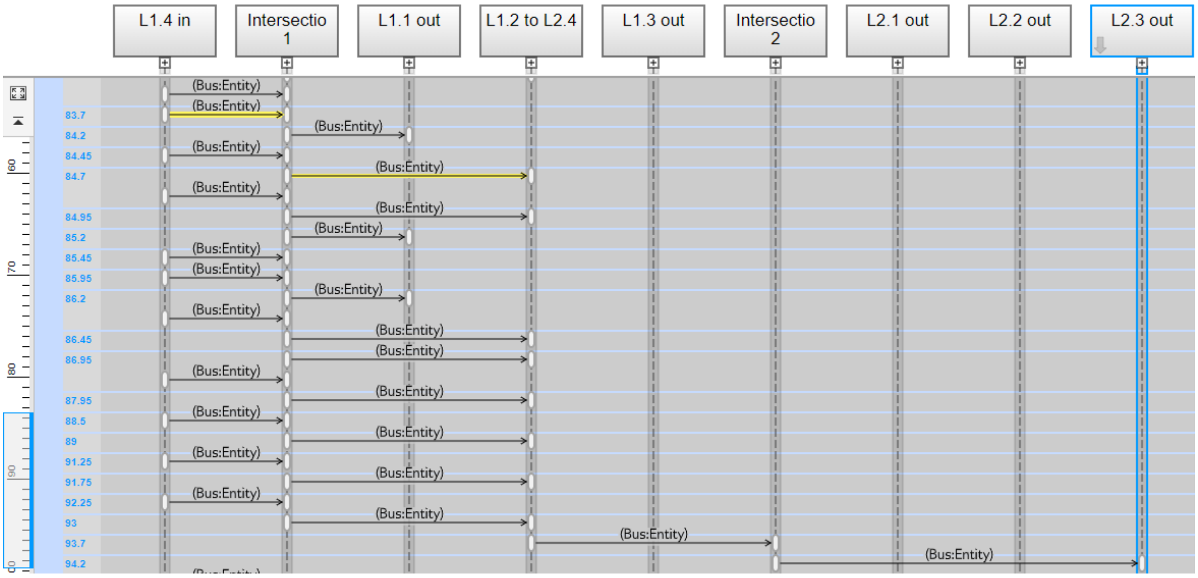

Figure 7. Visualization of vehicle movement from lane L1.4_in entering Intersection 1

Vehicles departing from L1.2_out are sequenced based on the green phase sequence of Intersection 1 and form a vehicle platoon. This platoon is composed of vehicles that departed due to the green phases $g_{1.1}$, $g_{1.3}$, and lastly $g_{1.4}$. However, no vehicles pass during green phase $g_{1.2}$ because no vehicles making a U-turn from L1.2_in to L1.2_out. This phenomenon resulting in a perceived arrival sequence from green phases $g_{1.3}$, $g_{1.4}$, and finally $g_{1.1}$. The entity representing the vehicles exiting through L2.2_out can be seen in the topmost graph in Figure 8.

In this simulation, the green phase of lane 4 of both intersection is designated as the initial active green phase. Thus, $g_{1.4}$ and $g_{2.4}$ will become the first active green phases. The shaded green area in Figure 8 represents the time when the green phase for lane 4 occurs. Since both agents control traffic with the same cycle length, $g_{1.4}$ and $g_{2.4}$ always occur simultaneously.

3.2 Average Vehicle Wait Time

All vehicles departed from L1.2_out traverse the connecting link L1.2 to L2.4 and subsequently arrive as entity arrival patterns at L2.4_in. The average travel time of vehicles on the connecting link is approximately 30 seconds, so that the vehicle platoon arrival is delayed by about 30 seconds. The arrival platoon vehicle is illustrated as entity arrival in the second graph in Figure 8. This figure illustrates the entity graph over a 500-second experiment, depicting five vehicle platoons passing through lanes L1.2 to L2.4.

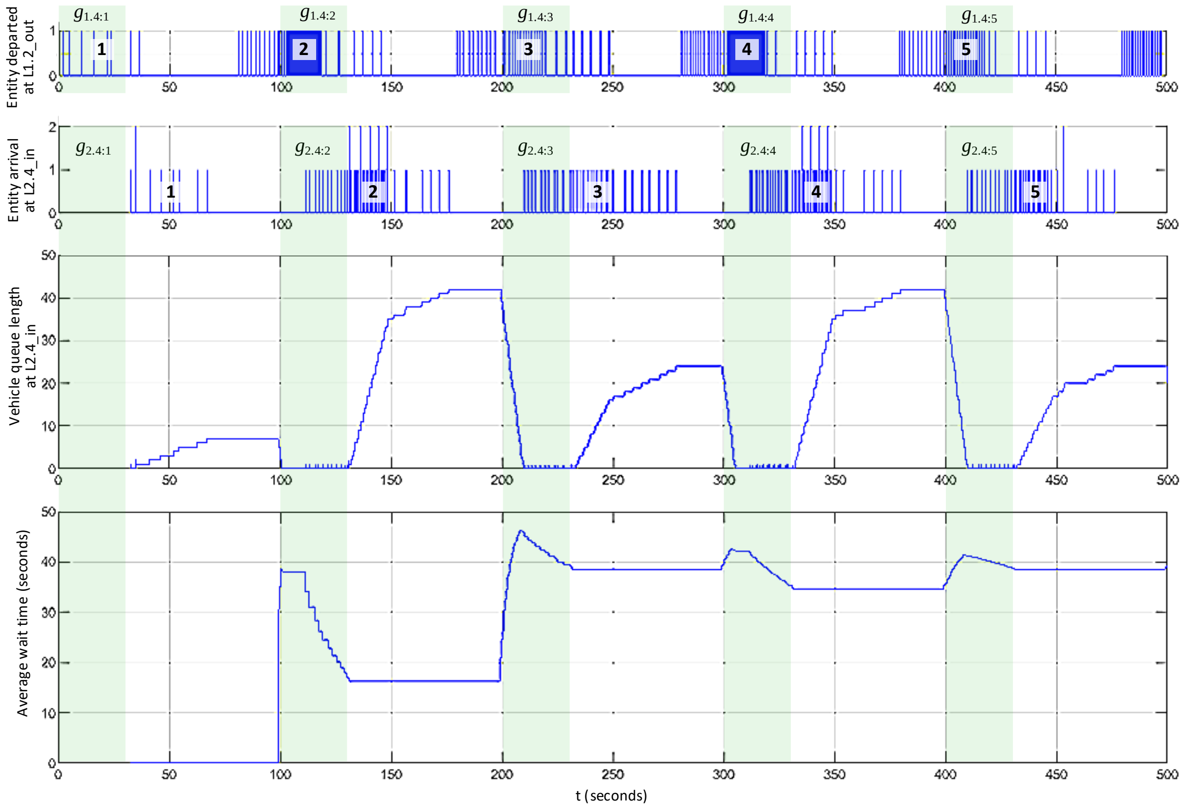

Figure 8. The graph of vehicle departures and arrivals, queue length, and average wait time when both agents operate independently.

The arrival platoon vehicles at lane L2.4_in were queued during the red phase and allowed to move during the green phase $g_{2.4}$. The green phase is depicted by green shading in Figure 8. Other areas are a red phase, a yellow phase, and a red-clearance phase. During the first green phase $g_{2.4:1}$, no vehicles passed through the L2.4_in link as vehicle platoon 1 had not yet arrived at that link. Meanwhile, the second green phase $g_{2.4:2}$ allowed vehicles in vehicle platoon 1 to move. This phase also permitted a portion of vehicles in vehicle platoon 2 to move immediately. These vehicles arrived at L2.4_in and were granted a green phase without delay. They constituted the leading part of vehicle platoon 2 and may be vehicles departed from L1.3_in. Conversely, the middle portion of vehicle platoon 2, departed from L1.4_in, received a red signal after $g_{2.4:2}$. The significant number of these vehicles caused a drastic increase in queue length. These vehicles were allowed to move after $g_{2.4:3}$. The vehicle queue length at L2.4_in during the red phase can be observed in the third graph of Figure 8.

The bottom graph in Figure 8 illustrates the average vehicle waiting time. Due to the fact that the majority of vehicles, departed from L1.4_in, consistently receive red signals, the average vehicle waiting time exhibits a high value. This graph depicts the average vehicle waiting time for all vehicles arriving at L2.4_in from the beginning to the end of the simulation. The average vehicle waiting time increases as the number of vehicles forced to wait at the red light increases, or the longer vehicles have to wait at red phase. The wait time value can vary over time. If the vehicle platoon arrival coincides with the green phase, the wait time can be short. However, conversely, if the vehicle platoon arrival occurs just as the green phase changes to red, the wait time can become significantly longer. In order to achieve a precise average vehicle waiting time, a one-hour simulation was conducted. The statistical analysis revealed that the wait time ranged from 23 to 50 seconds with an average of 40 seconds.

To minimize the average vehicle wait time, the green phase of intersection 2 is adjusted according to the green phase of intersection 1 by Algorithm 1. In this case, the green phase $g_{2.4}$ is adjusted so that most vehicles in each vehicle platoon arrival do not have to wait too long for the green phase. Algorithm 1 has been able to adjust the occurrence of the green phase, so that the green phase $g_{2.4}$ is delayed. This phenomenon can be shown in Figure 9. During the second green phase $g_{2.4:2}$, Agent 2, acting as the slave agent, records the time when the 'sync' signal is received. It then determines the target for the onset of the reference green phase. From these two data points, it is found that there is a synchronization error of 30 seconds. To address this, by executing line 6 of Algorithm 1, the error value is distributed across all green times, resulting in an increase of 7 seconds to the green time for each direction. Consequently, the next green phase will be delayed by 30 seconds.

However, in this experiment, we set the upper limit for green time addition to 4 seconds. As a result, the green time for the 3rd green phase only increases by 4 seconds, and the cycle time increases by 16 seconds. Meanwhile, the remaining error is distributed to the 4th green phase. The green time for this phase increases by 3 seconds from its normal value, thereby extending the cycle time by 12 seconds from its normal value. This phenomenon is illustrated in Figure 9, where the 3rd green phase is delayed by 16 seconds from its intended timing, and the 4th green phase is further delayed by 12 seconds from its intended timing.

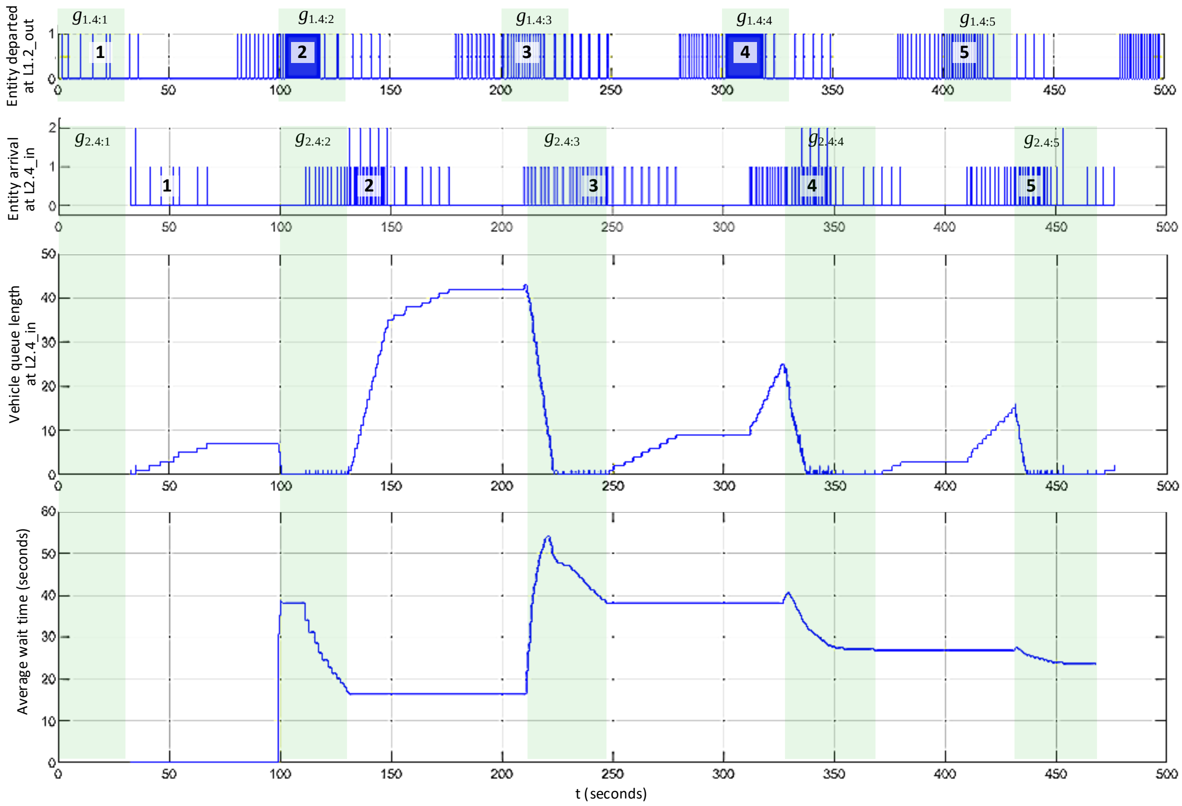

Figure 9. The graph of vehicle departures and arrivals, queue length, and average wait time when Agent 2 is coordinated by Agent 1

The graph of vehicle queue length can be seen in the 3rd graph of Figure 9. It is evident that by delaying the green phase $g_{2.4:4}$ and $g_{2.4:5}$, the vehicle queue waiting for these green phases is significantly reduced. Compared to the graph in Figure 8, where the green phase coincides with the initial vehicle platoon arrival time, in Figure 9, the green phase occurs closer to the peak density of the vehicle platoon arrival. This second scenario allows the majority of vehicles in the platoon to either immediately encounter the green phase or wait only briefly for it.

Thus, by delaying the green phases $g_{2.4:4}$ and $g_{2.4:5}$, the vehicle wait time for these green phases and subsequent phases will be reduced. As a result, the average vehicle wait time will start to decrease from that point onward. The bottommost graph in Figure 9 demonstrates that, during a 500-second experiment, the average vehicle waiting time could be reduced to 23 seconds. By running a simulation for one hour, the average vehicle waiting time for vehicles to wait for the green phase at L2.4_in is only 9.4 seconds.

3.3 Suggestions for Future Research

This study has successfully developed a prototype of a coordinated traffic control system capable of minimizing vehicle waiting times, particularly on main lane sections. However, the timing shift value is determined manually, derived from measurements of vehicle travel time between two intersections. It is important to note that travel time can vary due to fluctuating traffic conditions, weather, or other external factors. The reliance on a fixed timing shift value limits the system's flexibility and its ability to align with real-time traffic dynamics. Therefore, future research should focus on developing an algorithm that dynamically determines the shift value to better adapt to real-time traffic changes.

Additionally, further development of the system's adaptive capabilities is recommended. The coordinated nature of the system imposes constraints on each agent's ability to adjust green times for individual directions based on changes in vehicle arrivals, as all agents must operate under the same cycle time. This adaptability is further restricted by the use of a pre-timed timing mechanism for each agent [20][21]. Consequently, a significant challenge lies in integrating adaptive features into a coordinated pre-timed traffic control system, which could enhance its responsiveness to real-time traffic variations. Addressing this challenge represents a promising direction for future research in intelligent traffic management systems.

4. CONCLUSION

This study demonstrates that the use of a coordinated traffic controller can effectively manage vehicle flow in a synchronized manner. The research involves an experiment with two traffic controllers operating at two adjacent intersections. Each intersection's vehicle flow is regulated by an Agent functioning as the traffic controller. The Agent at the intersection with higher vehicle arrivals is designated as the master Agent, while the other serves as the slave Agent. A coordination algorithm was developed to synchronize the traffic controller timing, ensuring that the slave Agent's timing adjusts according to the arrival of vehicle platoons from the master Agent. This method enables the green phase of the slave Agent to align with the master Agent, allowing vehicle platoons arriving from the master Agent to immediately receive a green phase at the slave Agent. The coordinated traffic control system can be implemented using a microcontroller-based system, with vehicle movement simulated using Matlab's SimEvents. By shifting the reference timing of the slave Agent's green phase, the waiting time for most vehicles in a platoon can be significantly reduced. Simulation results for two intersections 500 meters apart show that this method reduces the average wait time from 40 seconds to 9.4 seconds.

REFERENCES

[1] Z. Lv and W. Shang, “Impacts of intelligent transportation systems on energy conservation and emission reduction of transport systems: A comprehensive review,” *Green Technologies and Sustainability*, vol. 1, no. 1, p. 100002, 2023. https://doi.org/10.1016/j.grets.2022.100002

[2] M. Eom and B. I. Kim, “The traffic signal control problem for intersections: a review,” Dec. 01, 2020, *Springer Science and Business Media Deutschland GmbH*. https://doi.org/10.1186/s12544-020-00440-8

[3] R. T. Wakkumbura, B. Hettige, and A. Edirisuriya, “Real-time traffic controlling system using multi-agent technology,” *Journal Europeen des Systemes Automatises*, vol. 54, no. 4, pp. 633–640, 2021. https://doi.org/10.18280/jesa.540413

[4] A. Susanty, B. Purwanggono, and V. A. Putri, “The barriers to the implementation of intelligent transportation system at Semarang City,” in *Procedia Computer Science*, Elsevier B.V., 2021, pp. 312–319. https://doi.org/10.1016/j.procs.2021.07.068

[5] R. Yue, G. Yang, Y. Zheng, J. Wu, D. Lin, and Z. Tian, “Feasibility of using a single traffic signal controller to accommodate adjacent intersections,” *International Journal of Transportation Science and Technology*, vol. 9, no. 1, pp. 52–63, 2020. https://doi.org/10.1016/j.ijtst.2019.09.002

[6] D. Metz, “Economic benefits of road widening: Discrepancy between outturn and forecast,” *Transp Res Part A Policy Pract*, vol. 147, pp. 312–319, 2021. https://doi.org/10.1016/j.tra.2021.03.023

[7] X. Wang, D. A. Rodríguez, and A. Mahendra, “Support for market-based and command-and-control congestion relief policies in Latin American cities: Effects of mobility, environmental health, and city-level factors,” *Transp Res Part A Policy Pract*, vol. 146, pp. 91–108, 2021. https://doi.org/10.1016/j.tra.2020.12.004

[8] A. Triani and B. Budiman Rusli, “Evaluasi Program ATCS (Area Traffic Control System) di Kota Bandung,” *Jurnal Administrasi Negara*, vol. 13, no. 1, pp. 50–61, 2021. https://doi.org/10.24198/jane.v13i1.28709

[9] P. Devia, “Implementasi Fasilitas Intelligent Transportation System (ITS) Di Kota Palangka Raya,” *Media Ilmiah Teknik Sipil*, vol. 8, no. 2, pp. 72–78, 2020. https://doi.org/10.33084/mits.v8i2.1404

[10] M. C. E. L. Chen, “Cooperative Intersection Management: A Survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, pp. 570–586, 2016. https://doi.org/10.1109/TITS.2015.2471812

[11] S. C. Rai, S. P. Nayak, B. Acharya, V. C. Gerogiannis, A. Kanavos, and T. Panagiotakopoulos, “ITSS: An Intelligent Traffic Signaling System Based on anIoTInfrastructure,” *Electronics* (*Basel*), vol. 12, p. 1177, 2023.

[12] X. Zhou, R. Ke, H. Yang, and C. Liu, “When Intelligent Transportation Systems Sensing Meets Edge Computing: Vision and Challenges,” *Applied Sciences*, vol. 11, no. 20, p. 9680, 2021. https://doi.org/10.3390/app11209680

[13] P. S. D. Balaji, “Multi-Agent System in Urban Traffic Signal Control,” *IEEE Comput Intell Mag*, vol. 5, pp. 43–51, 2010. https://doi.org/10.1109/MCI.2010.938363

[14] A. Basuhail, M. Khemakhem, F. E. Eassa, and J. M. Qurashi, “N-Versions-Based Resilient Traffic Control Systems,” *Electronics* (*Basel*), vol. 11, p. 2414, 2022. https://doi.org/10.3390/electronics11152414

[15] F. Gunes, S. Bayrakli, and A. H. Zaim, “Smart cities and data analytics for intelligent transportation systems: An analytical model for scheduling phases and traffic lights at signalized intersections,” *Applied Sciences (Switzerland)*, vol. 11, no. 15, 2021. https://doi.org/10.3390/app11156816

[16] S. Chen, C. Shang, and F. Zhu, “A Flexible Traffic Signal Coordinated Control Approach and System on Complicated Transportation Control Infrastructure,” *Sensors*, vol. 23, p. 5796, 2023. https://doi.org/10.3390/s23135796

[17] S. M. A. B. Al Islam and A. Hajbabaie, “Distributed coordinated signal timing optimization in connected transportation networks,” *Distributed coordinated signal timing optimization in connected transportation networks*, vol. 80, pp. 272–285, 2017. http://dx.doi.org/10.1016/j.trc.2017.04.017

[18] Z. Haoa, R. Boelb, and Z. Li, “Model based urban traffic control, part II: Coordinated model predictive controllers,” *Transportation Research Part C*, vol. 97, pp. 23–44, 2018. https://doi.org/10.1016/j.trc.2018.09.025

[19] F. Kurniawan, D. Dermawan, O. Dinaryanto, and M. Irawati, “Pre-Timed and Coordinated Traffic Controller Systems Based on AVR Microcontroller,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 12, no. 4, p. 787, 2014. http://doi.org/10.12928/telkomnika.v12i4.497

[20] D. Sun, A. Jamshidnejad, and B. De Schutter, “Adaptive Parameterized Control for Coordinated Traffic Management Using Reinforcement Learning,” in *IFAC-PapersOnLine*, Elsevier B.V., Jul. 2023, pp. 5463–5468. http://doi.org/10.1016/j.ifacol.2023.10.198

[21] F. Kurniawan, H. Sajati, O. Dinaryanto, “Adaptive Traffic Controller Based On Pre-Timed System,” *Telkomnika*, vol. 14, no. 1, pp. 56–63, 2016. http://doi.org/10.12928/telkomnika.v14i1.2798