Bringing AI to the PLCs: Realization of Artificial Intelligence Based Traffic Light Control

Tamás Wágner[1], István Varga[2] and Tamás Tettamanti[3]

***Abstract*— A plethora of Artificial Intelligence (AI) based methods have been published for adaptive traffic control. However, the demonstration of real-world applications is often overlooked. An effective and widely accepted solution for traffic signalization is the use of Programmable Logic Controllers (PLCs), especially in safety-critical infrastructures. This paper presents a reinforcement learning (RL)-based traffic control system deployed directly on an industrial PLC, representing a fundamentally novel integration of AI in a deterministic control environment. In contrast to prior approaches that focus solely on optimization, our system ensures compliance with strict traffic safety rules by restricting the agent's action space using a safety constraint matrix. Despite limited computational resources, the system achieved measurable improvements in CO_2 emissions while maintaining safe and stable runtime behavior. The results confirm that AI-based control is not only feasible but reliable on PLCs when supported by appropriate architectural and safety mechanisms, opening pathways for further industrial AI deployments.**

I. INTRODUCTION

Traffic signal heads, which display the different signal stages, serve as the visual actuators of traffic control—i.e., the timing of all displayed signals (green, yellow, red-yellow, red, and flashing) is managed by the traffic light controller hardware. Traffic lights are a fundamental component of modern urban transportation systems, enabling safe and efficient traffic flow control [1] at intersections or pedestrian crossings. Today, the complexity of urban environments demands more effective traffic signal control, making traffic-responsive systems increasingly important. The first electric traffic light controllers emerged in the 1960s. Since then, the technology has spread and evolved rapidly, driven by the ongoing expansion of road transport and technological advancement. From the 1980s onward, microcontroller-based traffic signal controllers—and later, embedded CPU-based systems—have become widespread and dominant.

Over the past decades, traffic controller development has mainly focused on enhancing control algorithms to achieve optimal, traffic-responsive signaling. At the same time, signal heads have also undergone technological evolution: traditional high-voltage bulbs have been replaced by modern systems using low-voltage, energy-efficient LED technology, typically operating on 24-volt power supplies.

In Europe, most urban intersections are equipped with products from Swarco [2] and Yunex [3]. These systems generally operate with fixed-time control, meaning they follow predetermined phase plans that cycle repeatedly. While the controller unit can select from multiple periodic timing plans, these systems cannot directly respond to real-time traffic conditions.

As an alternative to fixed-time systems, Adaptive Traffic Control Systems (ATCS) have emerged, operating on fundamentally different principles. These systems continuously adjust phase timings based on current traffic data, allowing them to dynamically adapt to changing conditions. The first such systems appeared in Australia during the 1970s and have since been implemented in hundreds of cities worldwide [4].

Although these adaptive systems were groundbreaking, they are not always sufficient to meet the complex demands of today's smart cities. A core component of the smart city concept is real-time communication and information sharing between various transportation actors and devices—enabled by technologies such as Vehicle-to-Everything (V2X). In these environments, where devices from multiple manufacturers operate together, it is crucial for traffic control systems to be open and flexible, enabling easy customization and integration [5], [6].

However, most currently available traffic control systems function as closed, vendor-dependent platforms. The control logic is often a "black box" inaccessible to users, making it impossible to modify [7]. These systems install predefined rule sets in static form, with limited or no adaptability. This closed nature poses a significant barrier to the adoption of advanced, artificial intelligence (AI)-based control solutions.

This is particularly true for algorithms based on Reinforcement Learning (RL), which enable adaptive and predictive control but require substantial computational capacity and a flexible architecture [8]. Traditional closed-system

*The research was supported by the European Union within the framework of the National Laboratory for Autonomous Systems. (RRF-2.3.1-21-2022-00002); Project no. 2024-2.1.1-EKÖP-00003 has been implemented with the support provided by the Ministry of Culture and Innovation of Hungary from the National Research, Development and Innovation Fund, financed under the EKÖP-24-3 funding scheme.

[1]Tamás Wágner is with the Department of Control for Transportation and Vehicle Systems, Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics, Műegyetem rkp. 3, Budapest, H-1111, Hungary wagner.tamas@edu.bme.hu

[2]István Varga is with the Department of Control for Transportation and Vehicle Systems, Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics, Műegyetem rkp. 3, Budapest, H-1111, Hungary and with the Systems and Control Laboratory, Institute for Computer Science and Control (SZTAKI), Hungarian Research Network (HUN-REN), Kende u. 13-17., Budapest, H-1111, Hungary

[3]Tamás Tettamanti is with the Department of Control for Transportation and Vehicle Systems, Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics, Műegyetem rkp. 3, Budapest, H-1111, Hungary and with the Systems and Control Laboratory, Institute for Computer Science and Control (SZTAKI), Hungarian Research Network (HUN-REN), Kende u. 13-17., Budapest, H-1111, Hungary

traffic controllers are unsuitable for directly running such algorithms—neither their performance nor their accessibility meets the required standards, especially in safety-critical environments where reliability and deterministic operation are essential.

These limitations drive the need for new, more open and intelligent traffic control systems. The goal is to create a flexible architecture that overcomes current technological constraints and allows for the seamless integration of cutting-edge technologies—especially artificial intelligence—into traffic signal control.

At the heart of our proposed system are Programmable Logic Controllers (PLCs), which have proven their reliability across various industrial domains, particularly in safety-critical control applications. One key advantage of PLCs is that they can be programmed using safety-critical languages under the IEC 61131-3 standard [9], making them suitable for a wide range of control tasks.

Since PLCs are primarily used in industrial environments, they come equipped with features that can be particularly advantageous for managing traffic intersections. They support multiple communication interfaces (e.g., serial ports, Ethernet, optical, Controller Area Network (CAN) bus), enabling seamless integration with other smart city devices, such as V2X units.

Modern PLCs are no longer limited to traditional runtime systems—they increasingly run full-fledged embedded Linux operating systems, allowing access to advanced technologies such as Docker containerization. Our solution enables the safe and reliable integration of AI-based control logic—including RL algorithms—within a PLC-based environment. Our system architecture ensures that intelligent control can function not only in simulations or lab conditions, but also in real-time, live traffic scenarios, complying with the stringent requirements of safety-critical systems [7].

Our results so far indicate that the developed system is not only applicable to traffic control but also provides a relevant alternative for any environment requiring PLC-based control—such as production lines or industrial automation systems.

The primary contribution of this study is the presentation of a traffic control system based on artificial intelligence—specifically, reinforcement learning—running on an industrial PLC platform. This approach is novel, as AI-based traffic control has not previously been widely implemented or technologically supported in such environments. The innovation lies in integrating AI-driven logic into a deterministic, safety-critical context where no general solutions were previously available. Our system enables real-time application of AI-based decision-making without compromising the reliability and industrial compatibility that characterize PLCs.

II. STATE OF THE ART AND RESEARCH GAPS

In contrast to previous studies, we have established a complete AI-based traffic signal control system using an industrial standard controller — a programmable logic controller — which is robustly capable of efficient, adaptive traffic management. The primary contributions to the state of the art can be summarized as follows:

- A novel AI-based traffic control system deployed directly on industrial hardware (PLC).
- A RL-based controller integrated with real-world traffic safety constraints (e.g., clearance time).
- A hybrid architecture combining adaptive AI decision-making with deterministic, safety-certified control logic.
- A sustainable and resource-efficient implementation of AI-based control systems, optimized for PLC deployment without sacrificing responsiveness or reliability.

A. *Traffic Management AIs*

AI-based approaches to traffic control—particularly those utilizing RL—have received significant attention in recent years. The literature includes numerous models [10], [11] that aim to optimize traffic signal control at intersections using various RL agents, primarily within simulation environments such as the Simulation of Urban MObility (SUMO) [12] or VISSIM [13].

While these models may prove effective from a traffic engineering perspective, they frequently neglect essential real-world safety regulations. For example, some omit consideration of the minimum clearance time between phase transitions or the prohibition of conflicting green signals. Several published models allow agents to select phase plans that do not guarantee conflict-free operation [11].

Moreover, most implementations are confined to simulation environments and do not demonstrate feasibility for deployment in real-time, safety-critical settings or on actual controller hardware. Their decision-making cycles are not embedded within deterministic systems, and their action spaces often disregard regulatory constraints. Safety considerations are typically relegated to post-validation, rather than being integrated into the control logic itself.

This work addresses these critical limitations by ensuring that the proposed RL system selects only phase time combinations that conform to a predefined safety matrix. The safety logic runs deterministically on an industrial PLC under real-time control. This approach surpasses the limitations of purely simulated systems, enabling the practical application of AI in real-world road infrastructure under safety-critical conditions.

B. *AI on PLCs*

Integrating AI into industrial automation has become one of the most significant technological developments in recent years. Applying such technologies to PLCs presents unique challenges, as these systems are traditionally designed to operate under deterministic, real-time, and safety-critical constraints [14]. In contrast, neural network-based decision-making is computationally intensive, inherently non-deterministic, and typically executed on cloud or edge devices. Bridging this gap is feasible only on a limited number of industrial platforms, and often subject to manufacturer-specific constraints.

Several industrial vendors have attempted to address this challenge. Siemens, for instance, provides the S7-1500 TM NPU module, which incorporates an Intel Movidius-based coprocessor for AI inference. However, in this configuration, the AI execution occurs on an external computational unit rather than the PLC's central processing unit, requiring tight system integration and increasing overall complexity [15].

Beckhoff's TwinCAT 3 Machine Learning (TF3810) offers direct execution of Open Neural Network Exchange (ONNX)-format [16] neural networks within the PLC's control cycle. This capability is particularly notable, as it supports deterministic neural network execution directly on the PLC. Nonetheless, it is available only on high-performance industrial PC (IPC)-type Beckhoff controllers, operates under closed licensing terms, and functions within a tightly vendor-dependent ecosystem [17].

The Bosch Rexroth ctrlX CORE is a modern, Linux-based edge-PLC hybrid that supports Docker or Snap containerized deployments, including neural network inference. However, this platform is primarily optimized for edge processing and does not guarantee real-time, deterministic execution. For safety-critical applications, an additional ctrlX SAFETY module is required. This separation of AI and PLC logic introduces integration challenges and increases system cost [18].

Other manufacturers—such as Schneider Electric, Rockwell Automation, Omron, and Mitsubishi—primarily offer AI solutions compatible with cloud or edge computing, generally involving external computational units, such as industrial PCs (IPC). In these configurations, the PLC functions mainly as a data collector or executor, with AI logic executed outside the PLC environment.

In our research, we selected the WAGO PFC200 PLC [7] for implementing AI-based inference. This choice was based on a set of technical and application-oriented criteria. First, the WAGO PFC200 series natively supports a manufacturer-certified Docker engine, enabling the deployment of containerized applications directly on the PLC. This capability offers significant flexibility for development, updates, and integration. Second, platform-independent AI execution is supported, allowing the trained neural network to be executed in real time using any inference engine, irrespective of programming language or model format. This enables an open toolchain and eliminates reliance on proprietary vendor ecosystems.

Third, while the PLC lacks a dedicated AI coprocessor, its performance was sufficient to execute a resource-efficient, optimized neural network. Fourth, the PLC's 24-volt digital outputs support direct control of energy-efficient LED traffic signals, obviating the need for additional interface hardware, power supply for the traffic lights and it is enhancing the system's applicability to traffic management. Finally, alongside the Docker engine, the PLC simultaneously runs deterministic control logic programmed via CODESYS [19], ensuring that AI-proposed phase plans are executed only if they conform to predefined safety constraints. If any rule is violated, the system automatically transitions into a fail-safe mode.

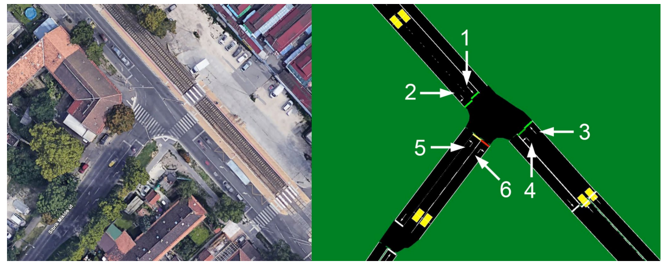

Fig. 1. Comparison between the SUMO model and the real satellite image, with signal head numbering (Source: Google Earth)

The use of CODESYS offers particular advantages, as it is one of the most widely adopted development environments in the PLC domain. It adheres to the IEC 61131-3 standard and supports structured programming languages commonly used in industrial automation (e.g., ladder diagram, structured text, function block diagram). Owing to its comprehensive library support, built-in safety features, and compatibility with industrial standards, CODESYS enables rapid, secure development and facilitates portability across different manufacturers' hardware platforms.

III. REALIZATION OF THE SYSTEM

The primary objective of the research was to ensure that the simulation environment used during the training process accurately reflected real-world traffic conditions, both structurally and in terms of data fidelity. To this end, a specific intersection in Budapest, Hungary—located at the junction of Maglódi Road and Sibrik Miklós Road in District X (4728'38"N; 1909'56"E) — was selected. A digital replica of the intersection was developed using the SUMO platform, based on satellite imagery and traffic measurement data. A visual comparison between the generated SUMO model and the actual satellite image is presented in Figure 1.

The model comprises six distinct traffic signals controlling the intersection. However, to simplify control during training, only two key parameters—the green times for Signal 1 (G_1) and Signal 4 (G_4)—were directly optimized. The remaining phase durations were derived using mathematical expressions, thereby significantly reducing the dimensionality of the action space that the RL agent was required to manage. The yellow and red-yellow signal timings were fixed at 3 and 2 seconds, respectively, for all phase transitions, while the red signal duration was determined as the remaining time in the cycle based on the calculated green times. The equations applied for the other green times from G_1 and G_4 were as follows:

$$G_2 = 40 \text{ sec} - G_1 - G_4 \tag{1}$$

$$G_3 = 6 \text{ sec} + G_1 + G_4 \tag{2}$$

$$G_5 = 39 \text{ sec} - G_1 - G_4 \tag{3}$$

$$G_6 = G_4 \quad (4)$$

During training, a Double Deep Q-Network (DDQN)-based [20] agent was employed, relying exclusively on input features that can realistically be obtained at actual intersections via sensor technologies such as inductive loop detectors or camera systems [4], [21]. In the simulation model, each incoming lane was equipped with a loop detector positioned approximately 40–50 meters upstream of the intersection. These sensors collected data over 295-second intervals, capturing the number of passing vehicles as well as the relative occupancy time—defined as the proportion of time the sensor was activated by a vehicle.

The neural network made control decisions at five-minute intervals, selecting a new phase plan from among 36 pre-defined green time combinations. Details of the network's architecture and its parameters are provided in Table II.

Training was conducted within a custom-built RL environment based on the Gymnasium framework, which interfaced with the SUMO simulation through the Traffic Control Interface (TRACI) API [22]. Each learning episode consisted of 3,900 simulation steps, corresponding to approximately one hour of real-world traffic. The reward function was formulated to encourage both traffic efficiency—by minimizing average travel time—and environmental sustainability—by reducing vehicle emissions. Reducing emissions is a critical goal for modern urban transport systems to mitigate environmental impacts, making it a key metric for optimizing traffic flow [23]. Upon completion of the training process, the trained neural network was exported in ONNX format [16], allowing for execution in resource-constrained environments such as industrial PLCs. The same simulation environment can also be utilized for validation purposes.

A. Deployment on PLC Hardware

For system deployment, we utilized industrial PLCs manufactured by WAGO, as these devices offer significantly greater computational capacity than most alternatives and support Docker-based container execution. Specifically, we employed the WAGO 750-8211/040-000 model, which features a 1 GHz Cortex A8 processor, 512 MB of RAM, and 4 GB of internal storage—expandable up to 32 GB via SD card. The ability to run containerized applications was a critical requirement for integrating the neural network inference module.

As previously mentioned, an additional advantage of this platform is that the PLC's 24-volt digital outputs can directly control modern, energy-efficient LED traffic signals. This enables the system to replace traditional traffic control units at the hardware level, substantially enhancing its industrial applicability.

The timing logic for traffic signal control was implemented within a CODESYS project. Given that traffic control is a safety-critical application, all signal timings and phase transitions are executed on this deterministic and reliable platform. The system was designed to be fully parameterizable, allowing phase timing parameters to be modified during runtime without requiring a system restart. To further enhance safety, mechanisms were incorporated to detect violations of the minimum clearance time between consecutive green phases. If such a violation is identified, the system automatically transitions into a fail-safe mode. In the case of repeated violations, a warning is issued, and the system can revert to manual control.

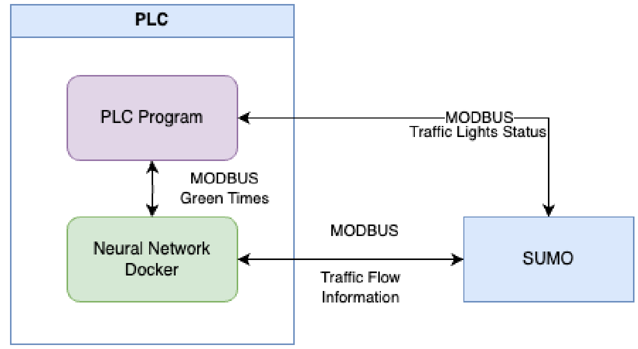

Fig. 2. System communication architecture on the PLC

The RL agent operates inside a Docker container hosted on the PLC and communicates with the CODESYS environment via the internal network. This container maintains a persistent connection with the SUMO simulation platform, which provides updated traffic data in five-minute intervals. At the 295th second of each cycle, the container retrieves sensor data via the MODBUS protocol, processes it, and computes new green time parameters.

Within the container, a custom-developed application written in Rust is responsible for executing the neural network inference. Rust was selected due to its high performance and low memory footprint—key considerations in the resource-constrained PLC environment. The inference process is thus both efficient and fast, while preserving operational safety. The computed green time values are subsequently transmitted from the Rust application to the CODESYS control logic via MODBUS, along with a feedback signal to confirm successful data exchange.

The complete system communication architecture is illustrated in Figure 2.

B. Results

The performance of the trained RL-based controller was compared to a baseline system employing a fixed, real-world phase plan across various traffic load levels. Simulations were conducted using the SUMO platform, with traffic scaling factors ranging from 0.5 to 1.5. In each case, a 3,900-second simulation episode was executed, and total carbon dioxide (CO_2) emissions were used as the primary evaluation metric.

For assessing the AI-based controller, we employed the neural network model previously trained on the Weights & Biases (W&B) platform [24]. Among several training experiments, the selected model demonstrated one of the strongest overall performances across multiple simulation scenarios.

The model was integrated into the simulation environment using the ONNX Runtime framework. These evaluations were conducted outside the PLC environment. Decision-making was based on the output with the highest activation value from the neural network, which received inputs consisting of realistically measurable observations—namely, loop detector occupancy and vehicle count.

The key training parameters for the selected RL model are presented in Table I.

TABLE I
SELECTED RL AGENT TRAINING CONFIGURATION

Parameter	**Value**
Neural network layers	[48, 64] (fully connected)
Batch size	256
Learning rate	0.0355
Discount factor (γ)	0.0328
Exploration start (ϵ_{start})	0.844
Exploration decay	2000 steps
Number of episodes	650
Episode simulation time	3900 seconds
Traffic input (veh/h) (in traffic light order)	210, 210, 510, 195, 210, 195

Figure 3 displays the CO_2 emissions under varying traffic load levels for both the AI-based and the fixed-timing baseline systems. The results indicate that the RL-based controller consistently achieved lower emissions across all load scenarios, with especially notable performance under heavy traffic conditions (1.25 and 1.5 scaling). This suggests that the trained controller adapts more effectively to dynamic traffic patterns, reducing unnecessary stops and delays, whereas the baseline system demonstrates instability under increased demand.

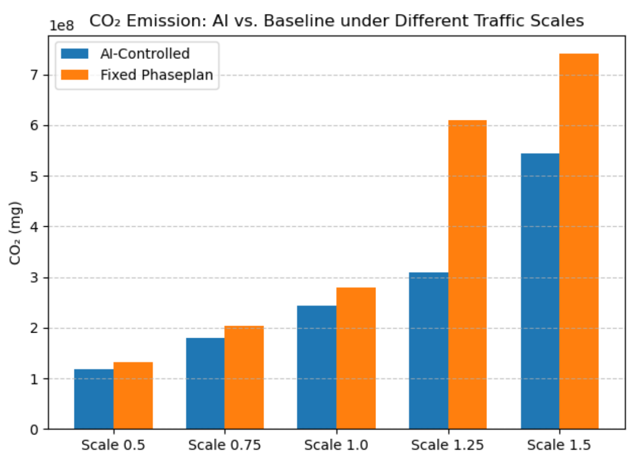

Fig. 3. Comparison of the total CO_2 emissions for AI-based vs. fixed-timing control at varying traffic levels

These results confirm that the trained RL controller can optimize traffic signal operation in a way that yields measurable environmental benefits—even without explicitly modeling fuel consumption.

To evaluate the system's runtime performance in an embedded environment, we monitored CPU utilization on the PLC during active operation of the RL inference module. Measurements focused on the initial 15 minutes (900 seconds) of execution, as longer time series would have resulted in overly dense and less interpretable graphs. Overall PLC CPU usage was sampled at 100 ms intervals, while usage data specific to the CODESYS runtime was collected at two-second intervals.

Figure 4 illustrates the total CPU usage, which includes the Docker container running the inference module, the CODESYS runtime environment, and background operating system processes. Additionally, Figure 5 presents CPU time specifically allocated to the CODESYS runtime. In this system architecture, CODESYS was assigned a dedicated CPU core to ensure deterministic and safe execution of control tasks.

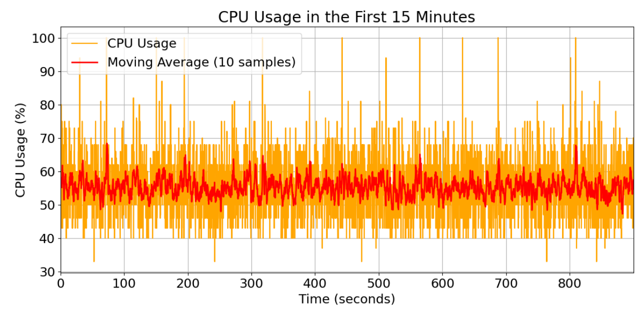

Fig. 4. Total CPU usage during the first 15 minutes (AI inference + system processes, 100 ms sampling)

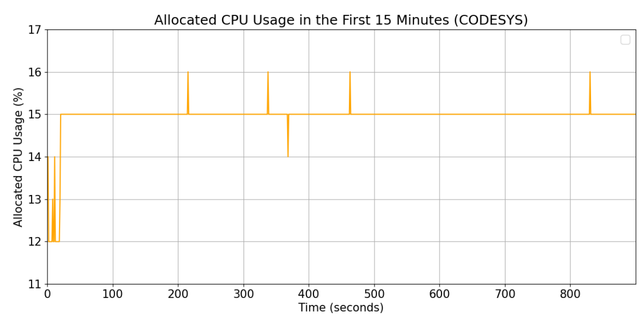

Fig. 5. CPU time usage allocated to CODESYS runtime (2 s sampling)

A summary of the statistical characteristics of total CPU utilization is provided in Table II. On average, the AI inference process accounted for a CPU load of 55.3%, while maintaining stable system behavior. According to the moving average trend, load typically ranged between 50% and 60%, with occasional peaks slightly above 70%. These values demonstrate that the decision-making process performed by the neural network aligns well with the computational capacity of the PLC, without inducing system overload.

In contrast, the CPU quota allocated to the CODESYS runtime demonstrated significantly greater stability. The processor utilization of the PLC control program consistently remained around 15%, with only minor fluctuations. These

TABLE II
CPU USAGE STATISTICS (FIRST 15 MINUTES)

Metric	Value (%)
Mean	55.32
Median	56.00
Standard deviation	8.56
Minimum	33.00
Maximum	100.00
25th percentile	50.00
75th percentile	62.00

results indicate that deterministic control and AI-based inference operate in a well-isolated manner, and that the stability of the CODESYS runtime remains assured—even in the presence of containerized AI integration.

IV. CONCLUSIONS

This article presents a traffic control system based on AI that is fully capable of operating on a PLC. The proposed approach combines RL with deterministic, safety-critical control logic, effectively addressing the challenge of integrating AI into real-time, reliable infrastructure.

The system was shown to be functional not only in simulation environments but also on actual PLC hardware. The compact neural network developed during training was deployed via a Rust-based inference engine, executed as a Docker container on the controller. The AI-driven control strategy achieved notable reductions in emissions while consistently adhering to strict safety requirements.

The contribution of this work lies not only in improving traffic management efficiency but also in demonstrating that AI can operate cooperatively with deterministic control systems within a real, embedded industrial environment in an environmentally sustainable manner. Future research will focus on scaling the system to manage multiple intersections, introducing federated learning and multi-agent coordination, and enabling deeper integration with V2X communication technologies.

REFERENCES

[1] M. Papageorgiou, "Overview of road traffic control strategies," *IFAC Proceedings Volumes*, vol. 37, no. 19, pp. 29 – 40, 2004, 4th IFAC Workshop DECOM-TT 2004: Automatic Systems for Building the Infrastructure in Developing Countries, Bansko, Bulgaria, October 3-5, 2004. [Online]. Available: https://doi.org/10.1016/S1474-6670(17)30657-2

[2] SWARCO, "SMART GREEN – Adaptive Traffic Management Brochure," Swarco, Tech. Rep., 2020. [Online]. Available: https://www.swarco.com/sites/default/files/public/downloads/2020-08/SMART%20GREEN%20BROCHURE%202020_EN%20VERSION%20WEBEDITION_2.pdf

[3] YUNEX TRAFFIC, "Yutraffic FUSION, Adaptive control for today's road users - real life deployments, real life learnings: from February to May 2023," Yunex, Tech. Rep., 2023. [Online]. Available: https://www.yunextraffic.com/portfolio/urban/mobility-management/adaptive-traffic-control/

[4] L. Studer and M. Ketabdari, "Analysis of adaptive traffic control systems design of a decision support system for better choices," *Journal of Civil & Environmental Engineering*, vol. 05, no. 06, 2015. [Online]. Available: http://dx.doi.org/10.4172/2165-784x.1000195

[5] H. Hejazi and L. Bokor, "V2X-equipped smart intersections - survey of surveys, use cases, and deployments," *Journal of Communications*, p. 722–738, Nov. 2023. [Online]. Available: http://dx.doi.org/10.12720/jcm.18.11.722-738

[6] T. Wágner, T. Ormándi, T. Tettamanti, and I. Varga, "SPaT/MAP V2X communication between traffic light and vehicles and a realization with digital twin," *Computers and Electrical Engineering*, vol. 106, p. 108560, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790622007753

[7] T. Wágner, T. Tettamanti, B. Varga, and I. Varga, *PLC-Based Traffic Light Control for Flexible Testing of Automated Mobility*. Springer Nature Singapore, 2024, p. 177–185. [Online]. Available: http://dx.doi.org/10.1007/978-981-97-4581-4_13

[8] M. Li, X. Pan, C. Liu, and Z. Li, "Federated deep reinforcement learning-based urban traffic signal optimal control," *Scientific Reports*, vol. 15, no. 1, Apr. 2025. [Online]. Available: http://dx.doi.org/10.1038/s41598-025-91966-1

[9] IEC, "Programmable controllers – Part 3: Programming languages," International Electrotechnical Commission, Geneva, CH, Standard, Feb 2013.

[10] S. S. S. M. Qadri, M. A. Gökçe, and E. Öner, "State-of-art review of traffic signal control methods: challenges and opportunities," *European Transport Research Review*, vol. 12, no. 1, Oct. 2020. [Online]. Available: http://dx.doi.org/10.1186/s12544-020-00439-1

[11] H. Wei, G. Zheng, V. Gayah, and Z. Li, "Recent advances in reinforcement learning for traffic signal control: A survey of models and evaluation," *ACM SIGKDD Explorations Newsletter*, vol. 22, no. 2, p. 12–18, Jan. 2021. [Online]. Available: http://dx.doi.org/10.1145/3447556.3447565

[12] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018. [Online]. Available: https://elib.dlr.de/124092/

[13] M. Fellendorf and P. Vortisch, *Microscopic Traffic Flow Simulator VISSIM*. Springer New York, 2010, p. 63–93. [Online]. Available: http://dx.doi.org/10.1007/978-1-4419-6142-6_2

[14] N. Dhameliya, "Revolutionizing plc systems with ai: A new era of industrial automation," *American Digits: Journal of Computing and Digital Technologies*, vol. 1, no. 1, pp. 33–48, 2023, accessed: 2025-04-07. [Online]. Available: https://hal.science/hal-04648969

[15] "SIMATIC S7-1500 TM NPU Manual," Siemens AG, Nuremberg, Germany, Manual, Sep 2022, accessed: 2025-04-07. [Online]. Available: https://cache.industry.siemens.com/dl/files/877/109765877/att_979771/v3/S71500_tm_npu_manual_en-US_en-US.pdf

[16] J. Bai, F. Lu, K. Zhang *et al.*, "ONNX: Open neural network exchange," https://github.com/onnx/onnx, 2019.

[17] "TwinCAT 3 AI Toolkit TF3800," Beckhoff Automation GmbH & Co. KG, Verl, Germany, Product Page, 2023, accessed: 2025-04-07. [Online]. Available: https://www.beckhoff.com/en-en/products/automation/twincat/tfxxxx-twincat-3-functions/tf3xxx-measurement/tf3800.html

[18] "ctrlX SAFETY Compact Safety Controller Devices SAFEX-C.12 / SAFEX-C.15 Programming Manual," Bosch Rexroth AG, Lohr am Main, Germany, Programming Manual, Aug 2023, edition 03, Document number: DOK-XSAFE*-SAFEX-C.1XC-PR03-EN-P.

[19] 3S-Smart Software Solutions GmbH, "Codesys." [Online]. Available: https://www.codesys.com

[20] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with Double Q-learning," 2015. [Online]. Available: https://arxiv.org/abs/1509.06461

[21] T. Kovács and L. Kovács, "Economical camera-based measuring scheme for roundabout traffic," *Periodica Polytechnica Transportation Engineering*, vol. 53, no. 2, p. 235–241, Feb. 2025. [Online]. Available: http://dx.doi.org/10.3311/pptr.37617

[22] A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer, and J.-P. Hubaux, "TraCI: an interface for coupling road traffic and network simulators," in *Proceedings of the 11th communications and networking simulation symposium*, ser. SCS SSM'08. ACM, Apr. 2008. [Online]. Available: http://dx.doi.org/10.1145/1400713.1400740

[23] Z. Xiao, Z. Xiao, D. Wang, and X. Li, "An intelligent traffic light control approach for reducing vehicles CO2 emissions in VANET," in *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*. IEEE, Aug. 2015, p. 2070–2075. [Online]. Available: http://dx.doi.org/10.1109/fskd.2015.7382270

[24] L. Biewald, "Experiment tracking with weights and biases," 2020, software available from wandb.com. [Online]. Available: https://www.wandb.com/