Computing, Telecommunication and Control, 2024, Vol. 17, No. 4, Pp. 56–63.
Информатика, телекоммуникации и управление. 2024. Том 17, № 4. С. 56–63.

Research article
DOI: https://doi.org/10.18721/JCSTCS.17405
UDC 004.021

DEVELOPMENT OF AN ALGORITHM FOR SYNCHRONIZATION OF TRAFFIC LIGHT CONTROLLER PROGRAMS

A.M. Sazanov ✉

Peter the Great St. Petersburg Polytechnic University,
St. Petersburg, Russian Federation

✉ arseny.sazanov@gmail.com

Abstract. This article develops an algorithm for synchronizing traffic light controller programs on one route in accordance with the general plan from the core. The relevance of the problem under consideration is described within the framework of improving the quality of functioning of adaptive and intelligent traffic control systems. An analysis of domestic and foreign modern solutions is carried out that ensure synchronization of traffic light modes in both local and distributed control systems of traffic light objects. Based on the necessary initial data and entered conditions that provide the required performance indicators of traffic control systems, an algorithm for synchronizing traffic controller programs is developed. For clarification, a general diagram of the synchronization algorithm is given. Conclusions are made on the results of testing the algorithm in real conditions and the prospects for further development of this study are described, including integration with existing control systems and the possibility of adaptation to various urban infrastructures.

Keywords: traffic light controller, synchronization algorithm, traffic light control, control, traffic management

Citation: Sazanov A.M. Development of an algorithm for synchronization of traffic light controller programs. Computing, Telecommunications and Control, 2024, Vol. 17, No. 4, Pp. 56–63. DOI: 10.18721/JCSTCS.17405

Научная статья
DOI: https://doi.org/10.18721/JCSTCS.17405
УДК 004.021

РАЗРАБОТКА АЛГОРИТМА СИНХРОНИЗАЦИИ ПРОГРАММ ДОРОЖНОГО КОНТРОЛЛЕРА

А.М. Сазанов ✉

Санкт-Петербургский политехнический университет Петра Великого,
Санкт-Петербург, Российская Федерация

✉ arseny.sazanov@gmail.com

Аннотация. В данной статье осуществляется разработка алгоритма синхронизации программ дорожного контроллера на одном маршруте в соответствии с общим планом от ядра. Описывается актуальность рассматриваемой задачи в границах повышения качества функционирования адаптивных и интеллектуальных систем управления дорожным движением. Проводится анализ отечественных и зарубежных современных решений, обеспечивающих синхронизацию светофорных режимов как в локальных, так и в распределенных системах управления светофорными объектами. На основе необходимых исходных данных и введенных условий, обеспечивающих требуемые показатели эффективности систем управления дорожным движением, реализуется разработка алгоритма синхронизации программ дорожного контроллера. Для пояснения приводится общая схема алгоритма синхронизации. Делаются выводы о результатах тестирования алгоритма в реальных условиях и описываются перспективы дальнейшего развития данного исследования, включая интеграцию с существующими системами управления и возможность адаптации под различные городские инфраструктуры.

Ключевые слова: дорожный контроллер, алгоритм синхронизации, управление светофором, управление, контроль дорожного движения

Для цитирования: Sazanov A.M. Development of an algorithm for synchronization of traffic light controller programs // Computing, Telecommunications and Control. 2024. Т. 17, № 4. С. 56–63. DOI: 10.18721/JCSTCS.17405

Introduction

In the modern world of cities with increasing mobility and movement of the population, traffic flow is a vital part of the logistical, economic, and social life of the city [1, 2]. Therefore, the development of traffic management systems is important and relevant. By improving its functionality [3], a wide range of complex transport problems is solved today, such as: traffic jams and reduced speed caused by increased traffic density; accidents and road traffic incidents resulting from traffic rule violations or decreased driver attentiveness; optimization of priority passage modes for accelerated and continuous movement of special transport and others. The key role in such traffic management systems is assigned to traffic light objects, ensuring safety, efficiency, and smooth traffic flow on urban road networks.

The concepts of modern urban development within the framework of innovative technological approaches [4, 5] also indicate the improvement of traffic light objects' performance. The latter, according to [5], affects aspects of various types of management in both local and network modes. When implementing transitional modes, activating priority passage modes, changing coordination plans and other management policies, the relevance and critical importance of ensuring program synchronization with smooth transition to the appropriate phase without introducing additional service modes (e.g., a special “all red” phase after which synchronization occurs and the traffic light begins to operate in the planned mode) increase.

Thus, the development of a synchronization algorithm for traffic light objects is a serious scientific task, the solution of which can improve the efficiency of the urban transport.

Domain Analysis

To date, domestic and foreign scientists have investigated a significant number of approaches, methods and algorithms, regarding the issues of synchronization of modes and programs of traffic light objects.

In [6], a comprehensive analysis of modern approaches to optimizing traffic control systems depending on the structural and functional features of intersections was performed.

In [7], a two-level approach to local and global synchronization based on traffic flow was proposed. Local synchronization ensures autonomous adjustment of traffic light operating times based on current traffic density within the traffic lane boundaries, while global synchronization determines the green signal time of the traffic light based on the density of peer connections at various levels, as well as a set of related parameters.

In [8], modern approaches to managing traffic light objects in real time at road junctions in smart cities were analyzed. Features of traffic light signal synchronization algorithms on various busy routes to ensure uninterrupted traffic at intersections were also disclosed.

In [9], the influence of vehicle density on optimizing traffic light retiming policies on Jalan Bukit Gambir was investigated.

In [10], an IoT-based method for synchronizing traffic light signals was proposed to increase non-stop runs between intersections by adapting traffic light phases, reducing average travel time compared to other fixed-time traffic management strategies and unsynchronized control.

In [5], procedures for centralized and decentralized management of traffic light objects were developed and analyzed based on the formulation of the Hamilton-Jacobi kinematic wave model. The proposed control strategies were tested on the real road network of London.

In [11, 12], modern architectures of intelligent traffic management systems were analyzed, allowing the implementation of modern synchronization algorithms and ensuring compatibility when expanding the preferred functionality through additional services.

Problem Statement

The development of a synchronization algorithm for traffic controller (TC) programs on a single route, in accordance with the master plan from the core to the end point through precise time control, is an important step towards optimizing transport infrastructure, ensuring efficient traffic functioning on urban transport networks. In this work, we focus on the task of developing algorithms for synchronizing current and new traffic light plans, which are changed by dispatchers (or a remote computer) of the Traffic Control Center. Each program update command at a specific time plays a key role in balancing safety, efficiency and convenience for all road users.

The object of the research is the traffic light control system. The subject of the research is the methods and algorithms for synchronizing traffic light plans.

Main Part

Initial Data:

- Traffic light objects (TCs) managed by an intermediate device (ID);
- Minimum number of phases in the program: 2;
- Maximum number of phases in the program: not specified (infinite);
- TC can be:
    - o Under the control of the ID-coordinated mode, in this case, the core knows exactly the current TC program and its current phase;
    - o In local mode:
        - If at least one TC cycle has passed, its local program and current TC phase are known;
        - If not a single cycle has passed, we only know the number of the current phase.

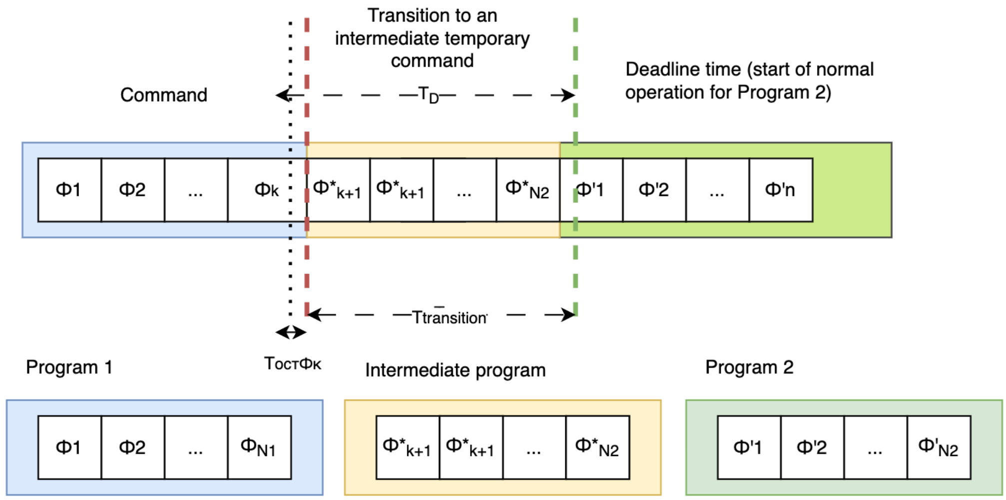

Fig. 1. Scheme of program synchronization on the traffic light controller

Conditions:

1. **System time** on the ID (at least within one route) is synchronized with an accuracy of **±1 second**.
2. The **current operating phase of the TC** is known.
3. $\mathbf{T_D}$ is the time after which (**deadline**) the new program should be running on the TC, starting from the first phase, and it is greater ($T_D > T_{\text{remains } \Phi}$) than the remaining operating time of the current phase of the old TC program;

a. If we apply a hard switch, then:

$$T_D > \min[T_{\min\Phi}; T_{\min\Phi} - T_{\text{completed part } \Phi}],$$

where $T_{\text{completed part } \Phi} = T_\Phi - T_{\text{remains } \Phi}$.

General scheme of the program synchronization on the TC is shown in Fig. 1.

Known:

- **Program No.1** consists of $\mathbf{N_1}$ phases: $\mathbf{\Phi_1}$, $\mathbf{\Phi_1}$, $\mathbf{\Phi_k}$, ..., $\mathbf{\Phi_{N1}}$;
- **Program No.2** consists of $\mathbf{N_2}$ phases: $\mathbf{\Phi'_1}$, $\mathbf{\Phi'_2}$, $\mathbf{\Phi'_n}$, $\mathbf{\Phi'_{n+1}}$, ..., $\mathbf{\Phi'_{N2}}$;

where

- $\mathbf{\Phi_1}$ is phase 1 of Program No. 1 (initial phase);
- $\mathbf{\Phi_2}$ is phase 2 of Program No. 1;
- $\mathbf{\Phi_k}$ is the k-th phase of Program No. 1 (current phase of the program at the moment the message about the program change arrives);
- $\mathbf{\Phi'_n}$, $\mathbf{\Phi'_{n+1}}$ are the transitional phases based on the phases of the new Program No. 2, considering the phase time correction matrix;
- $\mathbf{T_{transition}}$ is the transition process time (the time during which special transitional phases based on the new program phases operate on the TC);
- **n** is the program number following k → k+1 (provided: k+1 if $k+1 \le N_2$, otherwise k = 1);
- $\mathbf{\Phi_1}$ is the phase with corrected phase 1, according to the phase time correction matrix;
- $\mathbf{\Phi'_1}$ is the phase 1 of the new Program No. 2 (the initial phase of the new program; we assume that it is from this phase that the program should start after T_D occurs);
- $\mathbf{T_{\Phi i}}$ is the duration of phase No. i;

- $\mathbf{T_{min\Phi}}$ is the minimum duration of the phase, usually individually set for each phase; if not set, we assume the value ≥ T_{int} (duration of the intermediate phase);
- $\mathbf{T_{transition} = T_{\Phi i} + T_{corr\Phi i}}$ is the duration of each specific transitional synchronization phase until time T_D; this formula is the essence of the synchronization algorithm (which will be explained below).

Description of the Algorithm

Assumptions:

- All calculations are performed on the ID.
- The algorithm is the same for all IDs controlling TCs on the route; calculations are performed on the TC to reduce the load on the core, and because, when synchronizing times for program synchronization, one ID does not need to synchronize (knowing the phase times of another ID) with another ID.

General scheme of the program synchronization algorithm for TC on a single route according to the master plan from the core via program update command at a specific time ($T_D = T_{Deadline}$) is shown in Fig. 2.

ALGORITHM START

Upon receiving a command to switch the phase at the time when this switch should occur (i.e., upon receiving a new program and T_D):

1. Perform a check of the received T_D to ensure it is not less than:

a. The minimum time required to perform a hard program switch:

$$T_D > \min[T_{int}; T_{int} - T_{elapsed\Phi}]$$

(If this condition is not met, the program ends with an error) END

b. The minimum time required to perform a soft program switch:

$$T_D > T_{remaining\_phase}$$

(If this condition is not met and an additional option for a hard switch is not set – return an error; in case of a set option, from the next phase instead of the next phase of the old program, immediately activate Φ_1 of the new program).

2. If the ID does not have information about which program the TC is operating on (e.g., the TC was operating on a local program, the ID has recently turned on and has not accumulated statistical information on the TC's operation), then the ID must perform a hard switch to Φ'_1 of the new program at the moment TD occurs (ensuring traffic safety by observing T_{min_Φ}). END

3. If the ID has information about which program the TC is operating on, the ID calculates the remaining time of the current phase $T_{remaining\_phase}$. After the end of the current phase (Φ_k) of Program No. 1, it is further assumed that the time of the next phase (Φ_n*) is taken from the new program. Then $T_{remaining\_phase}$ is compared with TD. If $T_{remaining\_phase} > T_D$, the ID must return an error. END

4. After determining $T_{remaining\_phase}$, the ID calculates the remaining time for the transition process:

$$T_{transition} = T_D - T_{remaining\_phase}$$

5. Then, it calculates the number of sequential transitional phases that can complete within $T_{transition}$ and their total time $T_{sum\_transition\_phases}$

$$\text{(Important! } T_{sum\_transition\_phases} \le T_{transition}\text{)}$$

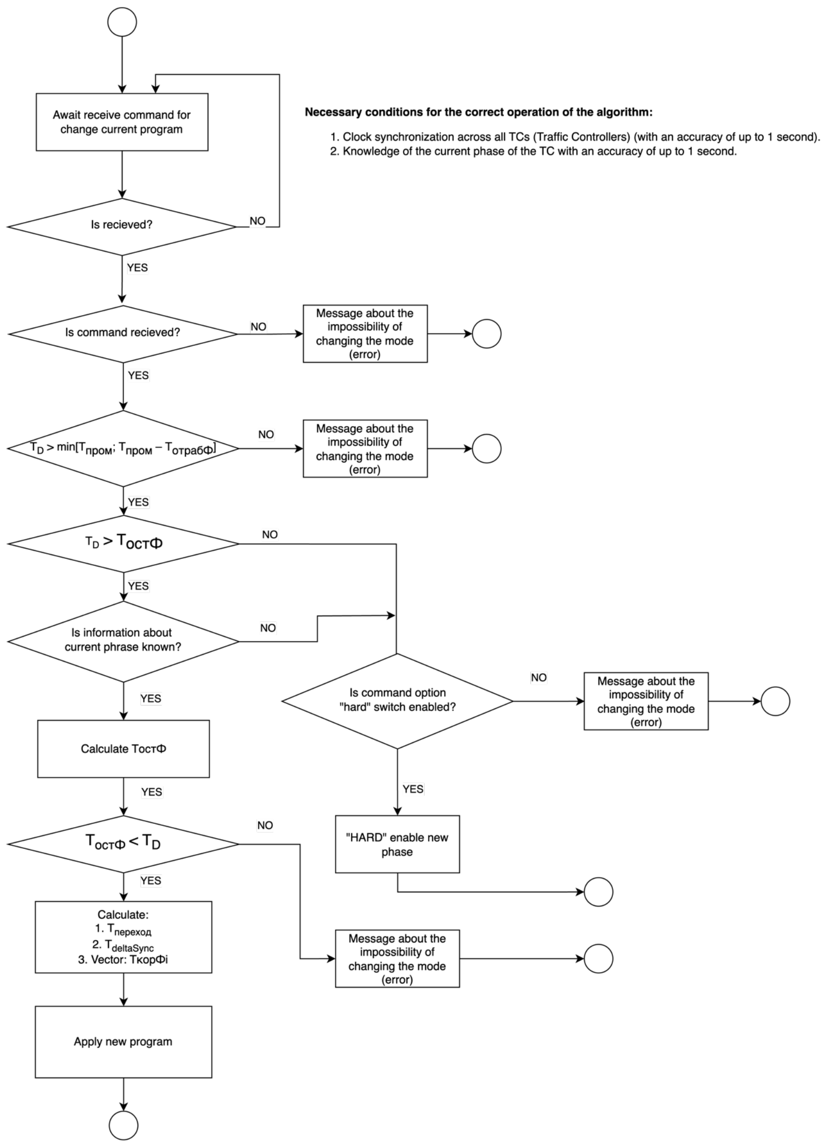

Fig. 2. Scheme of program synchronization algorithm
for a traffic light object (road controller) on a single route

$$T_{sum\_transition\_phases} = SUM(T_{Фn}, T_{Фn+1}, \ldots)$$

$$N_{transition\_phases} = COUNT(T_{Фn}, T_{Фn+1}, \ldots)$$

6. Next, it calculates the remaining time (which is insufficient to process the next phase of the new program cycle):

$$T_{delta\_sync} = T_{transition} - T_{sum\_transition\_phases}$$

7. Then, it calculates the average number of corrected seconds per transition cycle:

$$T_{avg\_corr} = DIV(T_{delta\_sync} / N_{transition\_phases}) \text{ (integer division)}$$

8. It calculates the final correction vector for all phases:

$T_{corr\_Фi} = T_{avg\_corr}$, where i is for all transitional phases from 2 to N_2; for i = 1, a special formula is applied:

$T_{corr\_Ф1} = T_{avg\_corr} + MOD(T_{delta\_sync} / N_{transition\_phases})$ (accounts for the undistributed remainder from integer division)

END OF ALGORITHM

Thus, after receiving the command to change the program with synchronization, the intermediate device, having calculated the correction vector of phases according to the algorithm described above, after the end of the current phase and until the deadline (T_D), begins to operate according to the new program, with the phase durations of this program changing according to the formula:

$T_{transition\_Фi} = T_{Фi} + T_{corr\_Фi}$ (this is why the phase is called transitional).

The developed algorithm ensures smooth phase synchronization and transition to the new program after its receipt, i.e., bringing the TC from the current state to the required one in the time necessary to activate the first phase of the new program, i.e., within T_D.

Considering time synchronization, if such an algorithm is successfully executed on all TCs, then all TCs are synchronized by the time T_D.

The algorithm was implemented in Go 1.18. The program text of the developed algorithm is posted on GitHub[1].

Testing and practical approbation of the algorithm were carried out in the traffic light control systems of the Moscow city transport system in 2023.

Conclusions

In this work, an algorithm for synchronizing TC programs on a single route in accordance with the master plan from the core was developed.

The developed algorithm will allow for effective implementation of synchronization modes both within a single intersection (traffic light object) and in coordination mode (on urban networks and city highways), when organizing priority passage, activating the green wave mode, or switching to a preferred traffic light plan. The algorithm's operation was successfully tested in the traffic light control systems of the Moscow Traffic Control Center. The accuracy of program synchronization for TC according to the master plan from the core was less than 1 second.

The proposed algorithm effectively aligns with the software of existing traffic light control systems and can be integrated into next-generation intelligent control systems.

REFERENCES

1. **Seliverstov S.A., Seliverstov Ya.A., Lukomskaya O.Yu., Sazanov A.M., Shkodyrev V.P.** Analysis of the functional features of intelligent traffic management systems. Transport: science, equipment, management (Scientific Information Collection), 2023, Vol. 12, Pp. 25–31. DOI: 10.36535/0236-1914-2023-12-3

2. **Seliverstov Ya.A., Gergel G.Yu., Seliverstov S.A., Nikitin K.V.** Improvement of intellectual transport systems by mobile technologies and procedures of analysis of social activity of urban population. St. Petersburg State Polytechnical University Journal. Computer Science. Telecommunications and Control Systems, 2018, Vol. 11, No. 1, Pp. 47–64. DOI: 10.18721/JCSTCS.11105

3. **Seliverstov S.A., Seliverstov Ya.A., Shatalova N.V., Borodina O.V., Lukomskaya O.Yu., Vydrina E.O.** Transformation of digital technologies in the process of transition to cognitivity in transport. Transport: science, equipment, management (Scientific Information Collection), 2021, Vol. 8, Pp. 3–9. DOI: 10.36535/0236-1914-2021-08-1

4. **Seliverstov S., Seliverstov Ya., Gavkalyk B., Fahmi S.** Development of transport infrastructure organization model for modern cities with growing effectiveness. Transportation Research Procedia, 2020, Vol. 50, Pp. 614–625. DOI: 10.1016/j.trpro.2020.10.073

5. **Chow A.H.F., Sha R., Li S.** Centralised and decentralised signal timing optimisation approaches for network traffic control. Transportation Research Procedia, 2019, Vol. 38, Pp. 222–241. DOI: 10.1016/j.trpro.2019.05.013

6. **Seliverstov S.A., Sazanov A.M., Lukomskaya O.Y., Nikitin K.V., Shatalova N.V., Benderskaya E.N.** Analysis of modern approaches to optimizing traffic control systems. 2021 XXIV International Conference on Soft Computing and Measurements (SCM), 2021, Pp. 106–108. DOI: 10.1109/SCM52931.2021.9507147

7. **Sankara Nayaki K., Nithin Krishnan N.U., Joby V., Sreelakshmi R.** Traffic light synchronization. Advances in Computing and Information Technology. Advances in Intelligent Systems and Computing, 2013, Vol. 177, Pp. 97–106. DOI: 10.1007/978-3-642-31552-7_11

8. **Tomar I., Sreedevi I., Pandey N.** State-of-Art review of traffic light synchronization for intelligent vehicles: Current status, challenges, and emerging trends. Electronics, 2022, Vol. 11, No. 3, Article no. 465. DOI: 10.3390/electronics11030465

9. **Ahmad Rafidi M.A., Abdul Hamid A.H.** Synchronization of traffic light systems for maximum efficiency along Jalan Bukit Gambier, Penang, Malaysia. SHS Web of Conferences. Sustanable Built Environment Symposium (SBES 2014), 2014, Vol. 11, Article no. 01006. DOI: 10.1051/shsconf/20141101016

10. **Aleko D.R., Djahel S.** An IoT enabled traffic light controllers synchronization method for road traffic congestion mitigation. 2019 IEEE International Smart Cities Conference (ISC2), 2019, Pp. 709–715. DOI: 10.1109/ISC246665.2019.9071667

11. **Lee W.-H., Chiu C.-Y.** Design and implementation of a smart traffic signal control system for smart city applications. Sensors, 2020, Vol. 20, No. 2, Article no. 508. DOI: 10.3390/s20020508

12. **Seliverstov S.A., Sazanov A.M., Seliverstov Y.A., Benderskaya E.N., Nikitin K.V.** Development of the intelligent traffic management system architecture. 2021 XXIV International Conference on Soft Computing and Measurements (SCM), 2021, Pp. 200–203. DOI: 10.1109/SCM52931.2021.9507125

INFORMATION ABOUT AUTHOR / СВЕДЕНИЯ ОБ АВТОРЕ

Sazanov Arseniy M.
Сазанов Арсений Михайлович
E-mail: arseny.sazanov@gmail.com

Submitted: 11.07.2024; Approved: 10.10.2024; Accepted: 28.10.2024.
Поступила: 11.07.2024; Одобрена: 10.10.2024; Принята: 28.10.2024.