



Design and implementation of a density-based intelligent traffic-light control system

Dr. Sonu Rana¹, Ashish², Ranjan Kumar², Shahnwaj Alam²

¹ Assistant Professor, Global Institute of Technology and Management (GITM), Gurugram, Haryana, India

² Global Institute of Technology and Management (GITM), Gurugram, Haryana, India

DOI: <https://doi.org/10.66856/ijraet.2025.11.1.12031>

Abstract

Urban traffic congestion is a major transportation problem caused by increasing vehicle ownership, fixed-time traffic signals, irregular traffic demand, and inefficient allocation of green time. Conventional traffic lights operate according to predetermined timing plans and may continue displaying a green signal even when no vehicles are present. This results in unnecessary vehicle idling, increased fuel consumption, longer travel time, and reduced road-user satisfaction. This paper presents the design and working of a density-based intelligent traffic-light control system that dynamically adjusts signal timing according to the number of vehicles waiting on each approach. The proposed system uses vehicle-detection sensors, a microcontroller, signal-control hardware, and programming or simulation tools to monitor traffic conditions and determine the appropriate signal sequence. The controller assigns longer green periods to approaches with higher traffic density while maintaining minimum green time, yellow transition time, pedestrian safety, and maximum waiting limits. The system is evaluated under low, medium, and high traffic-load conditions. The proposed approach demonstrates the application of embedded control and intelligent transportation concepts to real-time traffic management. It can reduce unnecessary waiting, improve traffic flow, minimize idling, and support safer and more sustainable urban transportation.

Keywords: Intelligent transportation system, density-based traffic control, traffic-light automation, embedded systems, microcontroller, vehicle detection, urban congestion, real-time control

Introduction

Traffic congestion is one of the most common problems in urban areas. Rapid growth in population, vehicle ownership, commercial activity, and road usage has placed significant pressure on existing transportation infrastructure. Intersections are particularly vulnerable to congestion because vehicles from multiple directions compete for limited road capacity.

Most conventional traffic-light systems use fixed timing schedules. Each traffic direction receives a predetermined green interval regardless of the actual number of waiting vehicles. This method is simple and reliable, but it is inefficient when traffic conditions vary throughout the day. For example, a signal may provide equal green time to two roads even when one road contains many vehicles and the other road is nearly empty.

A density-based traffic-light control system provides a more flexible alternative. It uses sensors to estimate the number of vehicles or the occupied length of each road approach. A microcontroller then processes the sensor information and changes the signal timing according to current traffic conditions.

The project presented in this paper focuses on the design and working of an embedded density-based traffic-light control system. The system is intended for a four-way intersection and can be implemented using infrared sensors, ultrasonic sensors, cameras, magnetic sensors, or other vehicle-detection technologies. A prototype can be constructed using a microcontroller and LED traffic signals, while the same control logic can be evaluated using traffic simulation and programming tools.

Background and Related Work

Traditional traffic-light controllers generally use fixed-time operation. Under this method, every lane receives a predefined green period. Fixed-time control is easy to install and maintain, but it cannot react to changes in traffic volume. It may lead to long queues during peak periods and unnecessary delays during low-demand periods.

Vehicle-actuated traffic signals improve on fixed-time systems by detecting vehicle presence. A signal can extend the green period when vehicles are still waiting or terminate the green period when no vehicles are detected. However, basic vehicle-actuated systems may only identify whether a vehicle is present and may not estimate the complete traffic density of an approach.

Modern intelligent transportation systems use cameras, radar, inductive loops, wireless sensors, and connected vehicle data. Camera-based systems can provide detailed traffic information, including vehicle count, queue length, vehicle class, and lane occupancy. However, they may require higher processing power and can be affected by weather, lighting, and image occlusion.

Microcontroller-based traffic control systems provide a low-cost platform for implementing adaptive signal control. Sensors collect traffic information, while the microcontroller performs decision-making and generates signal commands. Such systems are suitable for educational prototypes, small intersections, and experimental traffic-management projects.

The system proposed in this paper focuses on density-based control using a modular embedded architecture. It supports sensor-based vehicle detection, dynamic signal timing, emergency priority, pedestrian safety, and performance evaluation under different traffic loads.

Proposed System Architecture

a. System Overview

The proposed system consists of the following major units:

1. Vehicle-detection sensors.
2. Signal-processing and input interface.
3. Microcontroller-based control unit.
4. Traffic-light output stage.
5. Pedestrian and emergency-priority interface.
6. Display and communication unit.
7. Simulation and data-logging software.

Each road approach is equipped with one or more sensors. The sensors provide information about the presence or density of vehicles. The microcontroller receives this information and determines which direction should receive the next green phase.

The controller does not simply select the approach with the highest number of vehicles. It also considers waiting time, previous signal state, safety intervals, and fairness between approaches. This prevents a low-traffic road from being blocked indefinitely.

b. Four-Way Intersection Model

The prototype considers a four-way intersection with four approaches:

- North approach.
- South approach.
- East approach.
- West approach.

Opposing approaches can be grouped into one signal phase. For example, northbound and southbound traffic may receive a common green phase, followed by eastbound and westbound traffic.

The controller must prevent conflicting traffic movements. Before changing the active green direction, the system provides a yellow transition period followed by an all-red safety interval. This ensures that vehicles already crossing the intersection have time to clear the conflict area.

c. Sensor Options

The system can use different sensor configurations depending on the project scale.

1. Infrared Sensors

Infrared sensors can detect vehicles passing through a fixed point. They are inexpensive and suitable for small-scale prototypes. Multiple sensors can be placed at different positions along an approach to estimate queue length.

2. Ultrasonic Sensors

Ultrasonic sensors can detect the distance between the sensor and the nearest vehicle. They are useful for identifying whether a lane is occupied and for estimating queue length in a model intersection.

3. Magnetic or Inductive Sensors

Magnetic sensors detect changes caused by the presence of metal vehicles. Inductive loops are widely used in real traffic systems, although they are more difficult to install in a laboratory prototype.

4. Camera-Based Detection

A camera and computer-vision algorithm can detect and count vehicles. This method can provide more detailed

information but requires greater processing capability and careful lighting control.

For the proposed educational prototype, infrared or ultrasonic sensors are suitable because they reduce system complexity while clearly demonstrating density-based control.

Hardware Design

a. Microcontroller Unit

The microcontroller is the central component of the system. It performs the following functions:

- Reads signals from vehicle-detection sensors.
- Estimates the traffic condition of each approach.
- Selects the next traffic phase.
- Controls red, yellow, and green signal outputs.
- Maintains safety transition periods.
- Monitors sensor faults.
- Records traffic and controller data.
- Communicates with a display or computer when required.

Possible platforms include Arduino-compatible boards, ESP32, STM32 development boards, or similar embedded controllers. The selected device should provide sufficient input pins, output pins, processing capability, and communication interfaces.

b. Traffic-Light Output Stage

Each direction uses red, yellow, and green LEDs to represent the traffic signal. In a full-scale implementation, the microcontroller output would control relay modules, solid-state drivers, or industrial traffic-signal controllers. The output stage must protect the microcontroller from excessive current. Driver transistors or dedicated driver circuits should be used when the signal lamps require higher power than the microcontroller can provide directly.

c. Vehicle-Detection Interface

The sensor interface converts the sensor output into a form that can be read by the microcontroller. Digital sensors provide simple presence signals, while analog sensors may require signal conditioning and analog-to-digital conversion. The controller should include debouncing or filtering because vehicle-detection signals may fluctuate when a vehicle is moving slowly or when sensor readings are affected by environmental noise.

d. Display and Communication

An LCD, OLED, or computer interface can display:

- Current signal phase.
- Traffic condition of each approach.
- Remaining green time.
- Number of detected vehicles.
- Queue condition.
- Emergency or sensor-fault status.

A serial, Bluetooth, Wi-Fi, or Ethernet connection can be added for remote monitoring and data logging. In an advanced version, the system can send traffic information to a central management server.

e. Power Supply

The prototype can be powered using a regulated low-voltage adapter or rechargeable battery. Separate regulation may be

required for the microcontroller, sensors, and signal lamps. Proper grounding and electrical protection are important to avoid unstable sensor readings.

Density Estimation Method

a. Traffic-Density Zones

Each approach can be divided into several detection zones. For example:

- **Low-density zone:** Few or no vehicles detected.
- **Medium-density zone:** A moderate queue is present.
- **High-density zone:** A long queue or continuous vehicle presence is detected.

The density level can be estimated from the number of active sensors. If the sensors are placed at increasing distances from the intersection, the furthest active sensor can indicate the approximate queue length.

b. Vehicle Counting

A pair of sensors can be used to count vehicles entering and leaving an approach. The first sensor detects vehicle entry, while the second sensor detects vehicle movement or exit. The controller updates the estimated queue according to the sensor sequence.

Counting accuracy depends on vehicle spacing, vehicle speed, sensor alignment, and the ability to distinguish consecutive vehicles. For a prototype, the system can use simplified detection assumptions. For real deployment, multiple sensors or camera-based verification may be necessary.

c. Occupancy-Based Estimation

Instead of counting individual vehicles, the controller can estimate how much of a lane is occupied. Continuous vehicle presence over multiple detection zones indicates a higher traffic load.

Occupancy-based estimation is simpler than exact vehicle counting and is often suitable for adaptive signal control. It also reduces the effect of occasional missed vehicle detections.

d. Waiting-Time Consideration

Traffic density alone is not sufficient for fair signal control. A lightly occupied approach may still have been waiting for a long time. Therefore, the controller records the time since each approach last received a green phase.

If an approach reaches a maximum waiting threshold, it is given service even when another approach has a higher density. This prevents starvation and provides a fairer traffic-control strategy.

Control Algorithm

a. General Working Principle

The controller operates in the following sequence:

1. Initialize the microcontroller, sensors, traffic lights, and display.
2. Read the traffic condition of each approach.
3. Verify that sensor values are valid.
4. Identify the highest-priority traffic approach.
5. Check the maximum waiting condition.
6. Select the next signal phase.
7. Provide green time according to traffic density.
8. Provide yellow and all-red safety intervals.
9. Update waiting times and traffic records.

10. Repeat the process continuously.

b. Green-Time Assignment

The green period is selected according to the estimated traffic density.

- A low-density approach receives a short green period.
- A medium-density approach receives a moderate green period.
- A high-density approach receives a longer green period.
- An approach with excessive waiting time receives priority even if its density is low.

The controller also imposes a minimum and maximum green period. The minimum period ensures that vehicles have enough time to start moving. The maximum period prevents one direction from occupying the intersection for too long.

c. Signal-Transition Safety

Every change between conflicting traffic directions follows a safe sequence:

1. Current green signal changes to yellow.
2. All signals briefly change to red.
3. The next direction changes to green.

This sequence reduces the possibility of collisions caused by overlapping green signals or vehicles crossing during the transition.

d. Emergency Vehicle Priority

An optional emergency-priority input can be added to the system. When an emergency vehicle is detected through a wireless signal, RFID tag, acoustic detector, or manual control input, the controller can provide a green path after completing the required safety transition.

Emergency priority must be carefully managed so that it does not create unsafe signal conflicts or indefinitely block other road users.

e. Pedestrian Safety

A pedestrian request button or pedestrian sensor can be integrated into the design. When a request is received, the controller schedules a pedestrian-crossing phase at a safe point in the signal sequence.

The pedestrian phase should not be activated immediately if doing so would create a conflict with vehicles already moving through the intersection. The controller should complete the current safe transition before granting pedestrian access.

f. Fault Handling

If a sensor fails or provides inconsistent data, the controller should not generate unpredictable signal changes. The system can enter a fallback mode with predefined timing while notifying the operator of the fault.

The fallback mode ensures that the intersection continues to operate even when density information is temporarily unavailable.

Software Design

a. Embedded Program Structure

The embedded program is divided into several modules:

- Hardware initialization.
- Sensor-reading routine.

- Traffic-density estimation.
- Waiting-time management.
- Phase-selection routine.
- Green-time management.
- Safety-transition routine.
- Pedestrian and emergency handling.
- Fault monitoring.
- Display and communication.
- Data logging.

This modular structure simplifies testing and allows individual functions to be modified without rewriting the complete program.

b. Simulation and Programming Tools

The system may be designed and evaluated using the following tools:

- Arduino IDE or Platform IO for embedded programming.
- MATLAB or Simulink for controller modelling.
- Proteus for microcontroller and circuit simulation.
- Python for traffic-data generation and performance analysis.
- SUMO for traffic-flow simulation.
- Serial-monitoring tools for real-time debugging.
- Spreadsheet or database software for experimental data analysis.

A hardware prototype demonstrates physical sensor and signal operation, while simulation tools allow testing under a wider range of traffic conditions.

c. Data Logging

The controller can record:

- Time of each signal change.
- Number of vehicles detected.
- Estimated queue length.
- Waiting time for each approach.
- Signal phase duration.
- Sensor faults.
- Emergency requests.
- Average and maximum delay.

Data logging helps evaluate controller performance and identify weaknesses in the algorithm.

Results and Discussion

a. Expected Performance under Low Traffic

Under low traffic conditions, a fixed-time controller may continue assigning long green phases even when no vehicles are waiting. The proposed controller can shorten unnecessary phases and transfer service to approaches with detected vehicles.

This reduces idle signal time and allows vehicles to pass through the intersection with less delay. The system also avoids keeping a high-priority green signal active when its approach has become empty.

b. Performance under Balanced Traffic

When traffic is balanced, the controller should produce signal behaviour similar to a coordinated fixed-time system. However, it can still adjust phase duration according to temporary changes in traffic density.

The use of maximum waiting limits ensures that all approaches receive regular service. This prevents small differences in sensor readings from creating unfair signal allocation.

c. Performance under Unequal Traffic Loads

When one approach experiences heavy traffic, the density-based controller increases its green duration within the allowed limits. This helps reduce queue growth and prevents vehicles from remaining stationary for unnecessary periods. The controller does not allow the high-density approach to occupy the entire cycle. Other approaches are still served according to their waiting conditions. This balance is important for practical urban traffic management.

d. Performance under Sudden Changes

When traffic changes rapidly, the controller updates its density estimate during each sensing cycle. The response time depends on the sensor-read interval, filtering method, and decision logic.

A system with overly slow filtering may respond too late to a growing queue. Conversely, a system with insufficient filtering may react to temporary sensor noise. The design therefore requires appropriate sensor-processing settings.

e. Congestion and Environmental Benefits

Reducing waiting time can reduce the duration for which vehicle engines remain active without movement. In conventional internal-combustion vehicles, this may reduce fuel wastage and exhaust emissions. In electric vehicles, smoother traffic flow can reduce repeated acceleration and braking, helping improve battery efficiency.

The environmental benefit depends on traffic volume, vehicle type, road gradient, weather, and driver behaviour. Nevertheless, adaptive signal control provides a practical method for reducing avoidable idling at intersections.

f. Road-Safety Performance

The system improves safety by enforcing non-conflicting signal phases, yellow transitions, and all-red clearance intervals. Emergency and pedestrian functions can also be integrated into the same safety framework.

The controller should never prioritize traffic efficiency over basic safety. If traffic information is uncertain, the system should use a conservative fallback mode rather than producing unpredictable signal behaviour.

Conclusion

This paper presented the design and working of a density-based intelligent traffic-light control system. The proposed system uses vehicle-detection sensors to estimate traffic conditions and a microcontroller to dynamically select signal phases. Green time is adjusted according to traffic density while maintaining minimum service time, maximum waiting limits, yellow transitions, all-red safety intervals, and optional pedestrian or emergency priority.

The system provides a practical embedded solution for improving traffic flow at urban intersections. Compared with fixed-time control, it can reduce unnecessary green periods, shorten vehicle waiting time, reduce queue formation, and minimize avoidable idling. These improvements can support lower fuel consumption and reduced emissions.

The project also provides a suitable platform for evaluating controller performance under different traffic loads. It develops practical skills in sensor interfacing, embedded programming, real-time decision-making, traffic simulation, data logging, and safety-oriented control design.

Future work may include camera-based vehicle classification, wireless communication between intersections, emergency-vehicle detection, pedestrian and bicycle recognition, cloud-based traffic prediction, vehicle-to-infrastructure communication, and integration with real-time urban traffic-management platforms.

References

1. Roess RP, Prassas ES, McShane WR. Traffic Engineering, 4th ed. Upper Saddle River, NJ, USA: Pearson, 2011.
2. Hall FL. Traffic Flow Theory and Characteristics. Toronto, ON, Canada: University of Toronto Press, 1996.
3. Papageorgiou M, Diakaki C, Dinopoulou V, Kotsialos A, Wang Y. Review of road traffic control strategies. *Proceedings of the IEEE*, 2003;91(12):2043–2067.
4. Diakaki C, Papageorgiou M, Aboudolas K. A multivariable regulator approach to traffic-responsive network-wide signal control. *Control Engineering Practice*, 2002;10(2):183–195.
5. Poole RW. Rethinking Traffic Congestion. Washington, DC, USA: Reason Foundation, 2015.
6. Treiber M, Kesting A. Traffic Flow Dynamics: Data, Models and Simulation. Berlin, Germany: Springer, 2013.
7. Institute of Electrical and Electronics Engineers. IEEE Standard for Intelligent Transportation Systems—Communications Access for Land Mobiles, IEEE standards, Piscataway, NJ, USA.
8. National Highway Traffic Safety Administration. Traffic Safety Facts, Washington, DC, USA.
9. U.S. Department of Transportation, Federal Highway Administration. Traffic Signal Timing Manual, Washington, DC, USA.