INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Adaptive Traffic Signal Timer for A Signal in Chennai Metropolitan City Using Python and OpenCV

P Joyson Silva[1]*, R Vignesh[2], Binu Sukumar[3], Nithish Kumar[4]

1 Assistant Professor, Department of Civil Engineering, R.M.K. Engineering College, RSM Nagar, Kavaraipettai, 601 206, Tamil Nadu, India
2 GET, Cognizant, Chennai, Tamil Nadu, India
3 Professor and Head, Department of Civil Engineering, R.M.K. Engineering College, RSM Nagar, Kavaraipettai, 601 206, Tamil Nadu, India
4 Student, Department of Civil Engineering, R.M.K. Engineering College, RSM Nagar, Kavaraipettai, 601 206, Tamil Nadu, India

OPEN ACCESS

Received: 03-11-2022
Accepted: 04-03-2023
Published: 03-04-2023

Citation: Silva PJ, Vignesh R, Sukumar B, Kumar N (2023) Adaptive Traffic Signal Timer for A Signal in Chennai Metropolitan City Using Python and OpenCV. Indian Journal of Science and Technology 16(13): 1007-1013. https://doi.org/10.17485/IJST/v16i13.2131

* **Corresponding author**.

civiljoyson@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Background: The current traffic control system in India's metropolises is ineffective because of the randomization of traffic density patterns throughout the day. For a predetermined period of time, the traffic signal timers switch traffic's direction. Vehicles must therefore wait for a long time even when there is little traffic. **Objectives:** To continuously adjust the traffic signal timer based on the varying real-time traffic density and to significantly lessen traffic congestion. **Methods:** An adaptive traffic light timer is offered by this research using image processing algorithms. The real-time image data will be analyzed by the Raspberry Pi and OpenCV, which will determine when to regulate traffic flow. **Findings:** The adaptive timer minimizes vehicle waiting time and increases the number of turns granted to cars in each lane to cross the signal, which results in an improvement in traffic flow. **Novelty:** The use of image processing algorithms, the Raspberry Pi, and OpenCV makes the proposed system highly efficient and effective in adapting to the real-time traffic conditions. The suggested methodology will be more affordable than the currently used methodologies.

Keywords: Adaptive Traffic Signal Timer; Intelligent Traffic Control

1 Introduction

Traffic congestion is a result of metropolitan areas' rapid growth in many developing countries. Major cities around the world are experiencing severe challenges due to the exponential growth of vehicles. Two important elements in reducing traffic congestion are improving road infrastructure or limiting the number of vehicles on the road. However, infrastructure space is scarce in today's globalized world, making it impossible to control the increasing number of vehicles.

It is therefore necessary to have an advanced and effective way for controlling the growing traffic congestion problem. As per reports in Chennai there are 408+ traffic

signals at various locations[1]. All these signals have conventional fixed time loop for regulating the traffic flow. Even now in Tamil Nadu, during peak hours traffic police take control over the signal and control them manually.

To govern the flow of traffic, the traditional traffic system goes through stages. The orientations of structured motions determine the phases. The intersection in the diagram below depicts the sequence in which each phase operates. All phases are controlled by a pre-programmed traffic algorithm that distributes the time across all active lanes. For a predetermined cycle, all of the lights will turn on and off. Traditional fixed-time traffic signals are inefficient in their operation[2]. The cycle time, length, and interval sequence are all fixed and specified by a pre-programmed programme. When a red light appears, all vehicles must come to a complete stop, regardless of the scenario. Adaptive traffic signals adjust the timing of their green light cycles to match current traffic conditions on the ground[3]. They constantly collect data about approaching vehicles and create a new timing sequence to match them. The duration of traffic signal's red-green phases and green waves is automatically changed every cycle by examining the real time traffic conditions at intersections along the corridors. It's a very efficient way to move traffic through a busy corridor.

The system proposed by A. A. Zaid et al.[4] makes use of two fuzzy controllers with 3 inputs and one output for primary and secondary driveways. A simulation was done using VISSIM and MATLAB and for low traffic density, it improved traffic conditions by a factor of several hundred times.

Wang C et al.[5], a zebra-crossing detection method was proposed using an inverse perspective mapping picture or a bird's-eye view image. The method comprises two steps. In the first step, a morphological filter was used, followed by horizontal projection, to quickly identify possible zebra-crossing regions while utilizing the size and structure information of zebra-crossings. In the second step, a self-similarity-based identification approach was used to select potential areas.

The study focuses on reducing traffic congestion in intersection zones using a Raspberry-pi based traffic controlling and monitoring system. OpenCV software and machine learning algorithms are used to detect vehicular count and alter the timer into an adaptive one based on real-time traffic density. Results show a 75% vehicular flow even during high congestion, leading to improved traffic efficiency and reduced congestion.

2 Methodology

In this project we have used OpenCV, Yolo, python and Raspberry Pi to develop the adaptive traffic signal timer and the step-by-step procedure carried out in this project is given below (Figure 1)

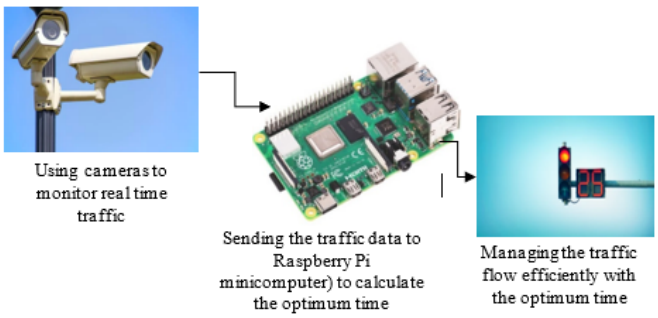

Fig 1. Schematic Diagram

- Developed an algorithm that can identify vehicles in an image.
- The algorithm will give the count of total number of vehicles in the given image as the output.
- By using that output, the optimum time required for the passage of vehicles will be calculated.
- In real time, cameras installed at the junction will be used or a camera has to be installed in order to take the real time image of the current traffic conditions prevailing at a junction.
- The captured image is sent to Raspberry Pi.
- The Raspberry Pi will have the pre-installed algorithm. That algorithm will calculate the optimum time required to regulate the traffic condition prevailing in the ground.
- Calculated optimum time is given as input to traffic light timer.
- Now traffic flow will be controlled by using the optimum time automatically without the intervention of traffic police.

2.1 Data Collection and Simulation

2.1.1 Traffic Survey

A traffic survey has been conducted on a junction near Anna Nagar tower in Chennai Metropolitan city (Figure 2). The survey was carried out on a weekday at peak hour time. The aim of this survey is to find the total number of vehicles crossing the junction and to feed the data to the simulation software in order to compare the efficiency of fixed time signal and an adaptive traffic signal. The required traffic data was collected by using two video cameras in which a single camera can cover 2 lanes, hence all the 4 lanes were covered using 2 cameras. Later the recorded video was used to count the total number of vehicles crossing each lane.

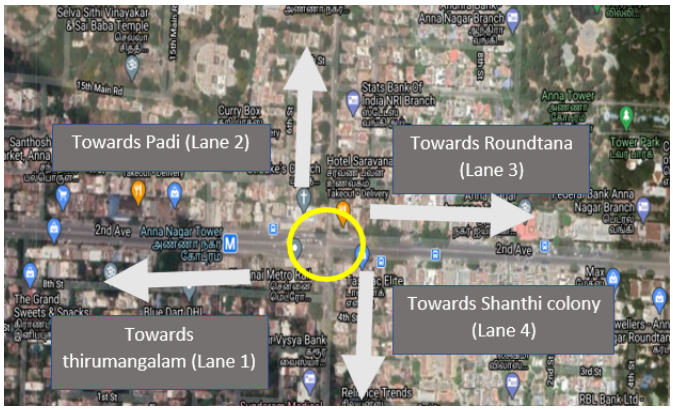

Fig 2. Location selected for the present study

From the traffic survey conducted the following details were collected about that junction.

- Max time(T) given for the green signal in the conventional fixed time loop.
- Max no. of vehicles(N) that can go in the given max time.
- Min time(M) that should be given for green signal (It is found based on the space constrain of the junction)

Formula to find optimum time:

If X is the number of vehicles standing in the lane, then

Optimum Time (OT) = (X*T)/N seconds

Optimum time is chosen such that, it should not be greater than the max green time and less than the minimum time.

(i.e) T >= OT >= M

For the surveyed junction

- The max time given for the green signal is 30 secs
- Max no. of vehicles that can pass in 30 secs is 96
- Minimum time for the specified junction is 5 secs

The Table 1 shows the total number of vehicles passing through the above specified junction in a time span of 1 hour.

Table 1. Number of Vehicle crossing the junction

Lane1	Lane 2	Lane 3	Lane 4	Total
1482	876	3432	960	6690

2.2 Simulation software

A simulation software tool was used to compare the performance of conventional fixed timer and an adaptive traffic light timer. It was originally developed by Mihir Gandhi [6] from Mumbai. This software was developed using python and pygame module and in this software tool we can control the total number of vehicles generated per second, density of vehicles on each lane and speed of vehicles. Hence it was easy to reproduce a real word traffic environment in this simulation software. By making minor change in the code used in this software it is possible to manage the traffic flow by both conventional fixed timer and an adaptive traffic light timer. Therefore, it is an ideal tool to compare the efficiency of conventional fixed timer and an adaptive timer.

2.2.1 Feeding data into the software

The simulation software is developed using python and has 525 lines of code in it. It can be used to simulate and control traffic by both conventional method and adaptive method. That change in functionality can be achieved in traffic simulation just by changing few values in the source code. Now we will see what values has to be changed in the simulation code.

In Figure 3, line number 30 has a variable named "defaultMinimum". The value of that variable is now set to 5 seconds. It indicates the minimum time given to green signal in a junction. So, now if the code runs, the simulation software behaves like an adaptive traffic signal. If the value of "defaultMinimum" is changed to 30 seconds the simulation behaves like a conventional fixed time loop. In other words, the "defaultMinimum" and "defaultMaximum" must have same value in order to make the simulation run like a conventional fixed time loop.

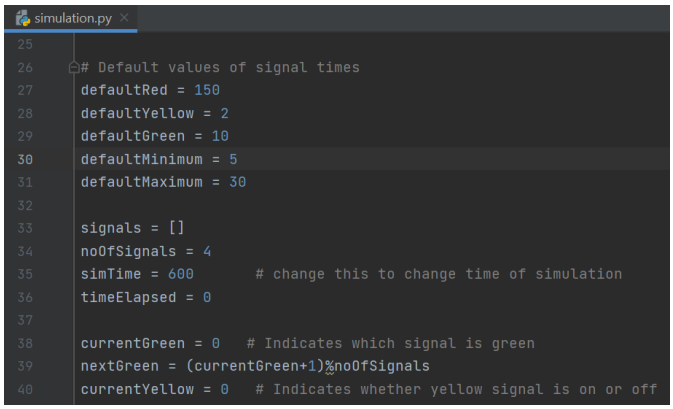

Fig 3. Adaptive timer code

Now it can be noted that both the variables are having the same value. Hence it behaves like conventional timer.

There is a variable "simTime" in line number 35 in Figure 4. It indicates simulation time of the software. The value 600 indicates 600 seconds (i.e 10 minutes). By defaults the variable "simTime" will have 300 as the default value. We can change its value according to our needs.

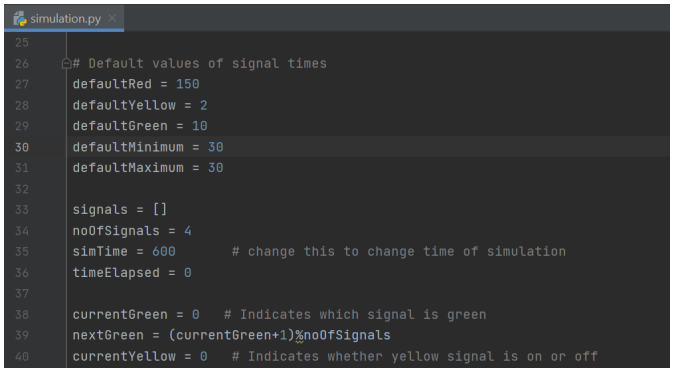

Fig 4. Conventional timer code

The line number 61 in Figure 5 has the value of speed of vehicles. We can run the simulation and adjust the speed of each vehicle according to our need. By using the default values used in the code, on an average 80 vehicles can be passed in a time span of 25 seconds.

The above shown lines of code in Figure 6 is very important because this is what decide how many vehicles will be generated and in which lane they will be generated. The variable in line number 403 basically indicates the weightage on which lane the vehicle should be generated. We can see that it has 4 different values. If all the 4 values are equally divided then vehicles will be generated equally in all lanes. If we want to have more vehicles in one lane and less vehicles on other lane then accordingly the values have to be changed in line number 403. If we take a look at Table 1, we can see that there is a frequent vehicle movement in lane number 1 and 3 (i.e towards thirumangalam and roundtana respectively) whereas lane number 2 and 4 has less vehicle

```
# Red signal time at which cars will be detected at a signal
detectionTime = 5

speeds = {'car':2.25, 'bus':1.8, 'truck':1.8, 'rickshaw':2, 'bike':2.5}  # average speeds of vehicles

# Coordinates of start
x = {'right':[0,0,0], 'down':[755,727,697], 'left':[1400,1400,1400], 'up':[602,627,657]}
y = {'right':[348,370,398], 'down':[0,0,0], 'left':[498,466,436], 'up':[800,800,800]}
```

Fig 5. Speed of vehicles code

movement in comparison with the other 2 lanes. We have to calculate the percentage of vehicles passing on each lane with respect to total no.of vehicles passed in an hour. According to our surveyed data 22%, 13%, 51% and 14% of vehicles pass through the lanes 1,2,3 & 4 respectively. This is how the values has to be entered in line no 403.

```
        elif(temp>2):
            will_turn = 0
    temp = random.randint(0,99)
    direction_number = 0
    a = [13,27,49,100]
    if(temp<a[0]):
        direction_number = 1
    elif(temp<a[1]):
        direction_number = 3
    elif(temp<a[2]):
        direction_number = 0
    elif(temp<a[3]):
        direction_number = 2
    Vehicle(lane_number, vehicleTypes[vehicle_type], direction_number, directionNumbers[direction_number], will_turn)
    totalnoofvehicles += 1
    print("total vehicles = ", totalnoofvehicles)
    time.sleep(0.51)
```

Fig 6. Vehicle density code

- First the least 13% has to be entered as 13.
- Next add the second least percentage to the first one and enter the value (i.e 13%+14% =27%). Enter 27 as the second value.
- Now add the third least value to the above and enter it (27%+22%= 49)
- Finally enter 100 as the last value.

There is a variable named “directionnumber” in line no 402. It indicates in which direction vehicles will be generated.

- Direction number 0 generates vehicles from west to east.
- Direction number 1 generates vehicles from north to south.
- Direction number 2 generates vehicles from east to west.
- Direction number 3 generates vehicles from south to north.

According to our needs the value of “directionnumber” in line numbers 405,407,409 & 411 has to be changed. The variable “time.sleep(0.51)” in line no 415 controls how many vehicles will be generated per second. The number 0.51 indicates 0.51 second. By changing this value, we can control the overall traffic generated in a junction. If we change that value to 1, 1 vehicle will be generated every second. If we change that value to 0.35, 1 vehicle will be generated for every 0.31 second.

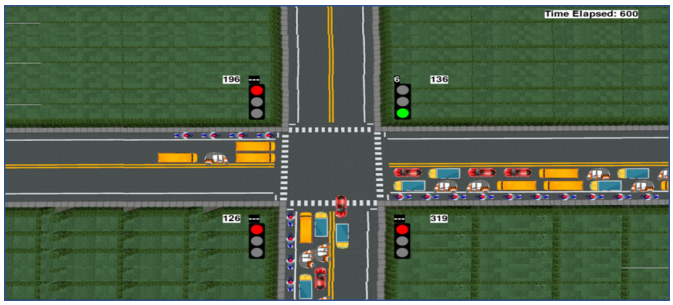

Fig 7. Simulation No 1 : Traffic controlled by Fixed time signal

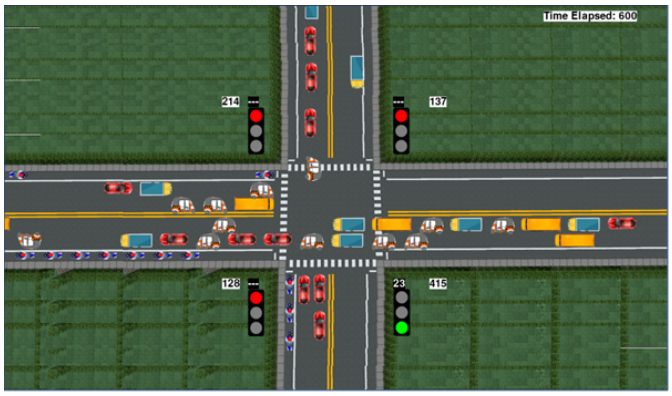

Fig 8. Simulation No 1: Traffic controlled by Adaptive traffic light timer

3 Results and Discussion

3.1 Conventional fixed timer

Six different simulations have been conducted, each for a time span of 10 mins. For the surveyed junction the maximum time given for the fixed time signal is 30 secs. All these data are fed into the simulation software and the results of these simulations are given in Table 2 and efficiency of the Fixed time signal is calculated.

From Table 2, Efficiency of the Fixed time signal = 4649/6690 = 69.4%

Table 2. Number of vehicles passed by conventional timer for 10 min

Simulation No	Lane 1	Lane 2	Lane 3	Lane 4	Total Vehicles
1	196	136	319	126	777
2	182	123	326	135	766
3	182	116	335	155	788
4	185	125	328	136	774
5	188	131	338	110	767
6	203	126	324	124	777
Total Vehicles					4649

3.2 Adaptive traffic light timer

A similar procedure has been adopted to check the efficiency of the proposed adaptive traffic light timer and the results are given below in the Table 3.

Table 3. Traffic controlled using adaptive traffic signal

Simulation No	Lane 1	Lane 2	Lane 3	Lane 4	Total Vehicles
1	214	137	415	128	894
2	194	119	417	143	873
3	234	122	404	139	899
4	206	130	411	132	879
5	209	123	401	141	874
6	207	118	421	131	877
Total vehicles					5296

From Table 3, Efficiency of the adaptive traffic light timer= 5296/6690 = 79.1%

By installing the adaptive traffic light timer, we will be able to

- Reduce the waiting time of vehicles in other lanes.
- Maintain a continuous flow of traffic.
- Improve the traffic infrastructure.
- Reduce the work load on traffic police

4 Conclusion

At present the technology used in Tamil Nadu is fixed time signal which has the capacity to pass 4649 vehicles per hour. According to the experiment done the proposed adaptive traffic signal passed 5296 vehicles per hour. On an average, for every 10 minutes, the adaptive timer was able to pass 107 vehicles more than the former method. The improvement in the flow of traffic is because the adaptive timer reduces the waiting time of the vehicles and increases the no of turns given to vehicles in each lane to cross the signal. Hence this method paves us the way to pass 5136 vehicles in excess than the conventional method for a period of 8 hour. Comparing the two, it's evident that the adaptive traffic light timer outperforms the fixed time signal in terms of capacity, efficiency, and ability to adjust to real-time traffic density. The adaptive timer is able to increase the number of vehicles passing through per hour and reduce waiting times, which leads to better traffic flow. The fixed time signal, on the other hand, has limited capacity and fixed cycle times, making it inefficient during high traffic congestion.

Scope for further study

1. Our initiative is now focused on a single intersection; but, in the future, it may be coordinated to include nearby intersections. The adaptive traffic timer's efficiency can be improved even more by doing so.
2. The idea might be modified to detect emergency vehicles, such as ambulances, and give that lane a green light.

5 Acknowledgement

We thank Our College Management, Principal and Head of the Department, Department of Civil Engineering, R.M.K. Engineering College for their constant support and encouragement.

References

1) Kannan RS, Sundaram. Signals out of sync with Chennai traffic. 2019. Available from: https://timesofindia.indiatimes.com/city/chennai/signals-out-of-sync-with-chennai-traffic/articleshow/71321301.cms.
2) Ha M, Hieu. Smart and adaptive traffic light system. 2020. Available from: https://www.theseus.fi/bitstream/handle/10024/346464/Smart_and_adaptive_traffic_light_system_HieuHa.pdf?sequence=2&isAllowed=y.
3) Stump EJ, Myer DS. Adaptive traffic signals reduce delay, increase safety. Herbert, Rowland & Grubic, Inc. 2017. Available from: https://www.hrg-inc.com/adaptive-traffic-signals-reduce-delay-increase-safety-and-improve-public-satisfaction/.
4) Zaid AA, Suhweil Y, Yaman MA. Smart controlling for traffic light time. *2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*. 2017;p. 1–5. Available from: https://dx.doi.org/10.1109/AEECT.2017.8257768.
5) Wang C, Zhao, Wang H. Self-Similarity based Zebra-crossing Detection for Intelligent Vehicle. 2015. Available from: https://dx.doi.org/10.2174/1874444301507010974.
6) Mihir-M-Gandhi. Adaptive-traffic-signal-timer. 2022. Available from: https://github.com/mihir-m-gandhi/Adaptive-Traffic-Signal-Timer.