Intelligent crossroads for vehicle platoons reconfiguration

Baudouin Dafflon[1], Franck Gechter[1], Jean-Michel Contet[1] Abdeljalil Abbas-Turki[1], and Pablo Gruer[1]

Laboratoire Systèmes et Transports
Université de Technologie de Belfort-Montbéliard
firstname.lastname@utbm.fr,
http://set.utbm.fr

Abstract. Nowadays, urban environments suffer from recurrent traffic jam with associated side effects. One reason of this is the social implicit priority given to personal cars, which are preferred to public transportation systems the main drawback of which is the time and path rigidity. Recently, some alternative transportation systems have been developed based on size adaptable trains of vehicles called platoon. These approaches still suffer from rigid path planning. The paper presents one possible solution to overcome this drawback. The proposal is based on the use of existing crossroads as hubs able to reconfigure vehicles train while crossing. Thanks to this solution, each train component could have its specific path in the public transportation grid. This paper presents also a comparative study of exposed algorithms relatively to a classical traffic light schedule.

Keywords: active crossroad, platoon, vehicles train, train dynamical reconfiguration

1 Introduction

Nowadays, urban environments suffer from recurrent traffic jam. Consequently, undesirable side effects occur such as greenhouse effect gas (COx NOx,...) and particle emission, noise pollution, transportation time increase... One of the reasons that lead to this situation is the priority choice of a personal car instead of public transportation devices. Moreover, statistical studies established that a high percentage of urban moving cars are occupied by the driver only. However, widely developed, public transportation systems seem to stay unattractive. Many reasons can explain this fact: regular transportation system is rigid (timetable, table, itinerary,...). They are uncomfortable, several transportation systems are required to reach one specific destination,...

To convince people, some research is devoted to create a new transportation service aimed at offering some properties generally associated to personal car. Indeed, these new systems must possess a set of properties such as **adaptability** to the user demand, **autonomy** to be able to act independently to schedule or predetermined path, **reconfigurability** to accommodate a variable number of people to transport. One possible solution, demonstrated in the CRISTAL project[1], is the use of small mobility units that can both be used by a regular driver as a standard electrical vehicle in car sharing and be structured as vehicles immaterial train (i.e. with no material link between vehicles) driven by a professional. To perform This way, each mobility unit is autonomous and proposed service both separately or integrated within a platoon (cf. figure 1).

Fig. 1. Example of vehicle platoon system (CATS European Project)

The key element of such an approach is the platoon function, i.e. the function that makes vehicles able to follow each other without a material link. This function can be considered as the control of the vehicle's inter-distance. This control problem is addressed through two sub problems: longitudinal control and lateral control. The longitudinal control consists in regulating braking and acceleration in order to fix the inter-vehicle distance to a predefined regular distance. By the same, lateral control consists in computing a wheel direction according to the platoon trajectory.

In literature, platoon control can be encountered as global or local approaches. **Global** approaches are composed of a decision-making vehicle, generally integrated in the first vehicle of the train, which computes some reference informations (trajectory points, steering and speed instruction, ...) and send them to each follower vehicle. For example, [1, 2] uses a global positioning system to compute trajectory points and communicate them to the follower vehicles. Another approach [3] consists in broadcasts driving information (steering and speed) to each follower vehicle, as what was made in *Chauffeur* European project. Literature exposes that this approach yield good trajectory matching. However, global positioning sensors or other technologies require road adaptation. Besides, a safe, reliable vehicle-to-vehicle communication network is required. Global control approaches yield adequate results, subject to strong constraints on sensors (high cost), road adaptation and communication reliability between vehicles.
Local approaches are based only on vehicle's perception capabilities. Vehicles

[1] http://projet-cristal.net/

are equipped with low cost sensors, which compute measurements such as inter-vehicle distance vectors. Each vehicle computes the acceleration and wheel direction commands with its own perceptions. In literature, local control strategies proposed within local approaches general use PID controllers [4–6] or other regulation-loop based algorithm [7–9]. Other proposed approaches are based on a physics-inspired, inter-vehicular interaction link from which vehicle's control references can be computed, as in [10] or in [11]. The main interest of local approach is the technical simplicity, require neither expensive road infrastructure, nor reliable inter-vehicle communication and use cheaper and more reliable sensors than the global approaches. However, these approaches suffer from the anticipation error problem. Some recent researches tend to cover this problem. [12] is a local approach which minimizes the anticipation error by taking into account local curve properties with performance levels close to those obtained by global control.

Even with reliable and efficient embedded platoon functionalities, vehicle virtual train transportation systems still have several drawbacks. Among these, the most handicapping is the fixed-line constraints since people who use the platoon are obliged to follow the pre-determined path, even if this was set on the fly. A solution to overcome this problem is to allow a vehicle platoon to merge, split and reconfigure at specific points in order to make users can build their own path using small mobility units that are able to move autonomously hanging on existing vehicles trains. Existing crossroads can be good candidates for these reconfiguration points. The goal of this paper is to present a solution for the reconfiguration of vehicles virtual train on crossroads. This solution is based on a merge of vehicle behaviors and crossroads intelligent priority assignment. In the presented solution, the reconfiguration procedure is totally autonomous. This contrasts with classical methods where each driver has to take the control of its own vehicle on crossroads, following standard circulation rules, making a new train coupling after the crossroad.
The paper is structured as follows:Section 2 presents a state of the art of active crossroads systems. Then, sections 3 and 4 give a detail explanation of the crossroad model and a presentation of the associated experimental results. The paper concludes with an overview of future works (Section 5).

2 Active crossroads: state of the art

Since the vehicle virtual train transportation systems are neither vehicles nor trains, then one of the most important challenges that the concept has to face is the intersection. On the one hand, the traditional traffic lights do not take into consideration long and slow virtual trains of vehicles that cross the intersection. Indeed, the traffic light can change the color without considering whether all the vehicles that belong to the same platoon are released [15]. A similar problem can occur in an intersection with a stop sign. On the other hand, in cities, tramway's signalization considers the previously known number of rail cars. The green time is accordingly fixed to few seconds [16]. However, the core of the proposed system is that the number of virtually attached vehicles is variable. Thus, the initially computed green time may not match the length of the virtual train. Hence, the traffic signalization at intersections deserves a particular attention.

One way to overcome this matter is to use the Traffic-Responsive Strategy based on the vehicle-interval method [17]. A critical interval (CI) is created, during which any detected vehicle leads to a green prolongation that allows a vehicle to cross the intersection. If no vehicle is detected during CI, the strategy proceeds to the next stage. However, this strategy requires some adaptations. Indeed, this strategy is applicable to two-stage intersections and designed for ordinary vehicles [18]. More precisely, a crossroad of virtual trains needs more than two stages, the minimum and the maximum-green durations are not required, the length of the critical interval needs to be accurately estimated and the usual location of the inductive loop detectors (40m upstream of the stop line) is not adapted to our application. Moreover, only the virtual locomotive needs to be informed about whether or not it has the right of way. So, the prolongation of the green light is useless.

From the point of view of the traffic efficiency, the work presented in [19] proposes an interesting approach for building the solution to our system. The authors assume an intersection and vehicles that can wirelessly communicate together. The authors use the well-known Littles formula, for optimizing the traffic. As a result, the stage that releases the greatest number of vehicles is selected. Nevertheless, the use of wireless communication raises the problems of the rates of successful messages and of the communication delay [20]. Hence, an innovative traffic signalization at intersections is required for the vehicle virtual train transportation systems. We will call this signalization: Active Signalization.

3 Crossroads and Vehicle models

3.1 Global overview

In this paper, crossroads are considered to be reconfiguration hubs for incoming platoons (i.e. vehicle trains). Each platoon is composed of several mobility units, which can be autonomous (in train configuration) or human drivable (in single configuration). In a standard traffic light crossroad, this re-configuration requires to turn each vehicle from autonomous mode to driven mode in order to complete merge and split maneuvers. The goal of the approach presented in this paper is to bring to crossroads intelligent routing abilities in order to keep each vehicle in autonomous mode. In this section, vehicle and crossroad models are detailed.

3.2 Vehicle specification

Required functionalities:

The vehicles used are supposed to have got autonomous abilities. To that way, they are equipped by sensors aimed at perceiving surrounded environment.

They also are able to decide, which is the best command to apply in a specific situation. In particular, vehicles have platoon functionality, i.e. they can follow the trajectory of a front perceived object/vehicle. This functionality is used in our case for the train crossroads entrance and exit. Moreover, they also have the capability to perceive the crossroads signals aimed at stopping the vehicle when necessary and giving it drive authorization.

Fig. 2. SeTCAR: one of Système et Transports Laboratory vehicles

Figure 2 represents such a platoon able vehicles. This, called SeTCAR, has been transformed in our laboratory based on standard electrical cars (GEM cars). It was upgraded by integrating direction, speed and breaking control using a Dspace Microautobox[2]. Moreover, SeTCAR is equipped by several sensors such as Laser Range Finders, Lidar, mono-camera, stereo-camera, RTK GPS, Gyroscope, magnetic field detector, ...

Vehicle behavior:
Vehicle behavior can be split into three phases:

- **Crossroad entrance**: Vehicles arrive within a platoon, each vehicle following its predecessor except for head vehicle, which is human driven. When an incoming train is near the crossroad (i.e. when it arrives into crossroad perception range), each vehicle turns into autonomous mode. In this mode, vehicles follow incoming predetermined trajectory thanks to specific sensors/beacons[3]. Moreover, vehicles turning signal lights are also autonomously controlled depending on the goal path of each vehicle.
- **Crossroad control**: When in crossroad, each vehicle follows trajectories enlightened by the crossroad decision process. Decisions are made depending on crossroad configuration and vehicle constraints (see next section for more details. The enlightenment process consists in activating the specific vehicle perceivable beacons which virtually draw authorized trajectories over the

[2] http://www.dspace.de/en/pub/home/products/hw/micautob.cfm
[3] in our case a Laser Range Finder and reflective beacons placed on the road.

road. Each vehicle follows one trajectory in order to reach its destination point. If no beacon is detected vehicle stays at its place waiting for beacon activation.

– **Crossroad exit**: After passing the crossroad, vehicles turn from autonomous mode to regular platoon mode. The driver in the head vehicle takes control of the train.

3.3 Crossroads model

Crossroad decision process is based on a perception rule-based decision loop. This section describes first the perception abilities of the crossroad, and then exposes two proposals for the decision process.

Perception:
Crossroad is able to perceive pending cars and exiting cars. Each incoming road has got a pending car detector. This runs as a simple switch the result of which is equal to 1 when a car is waiting and 0 when there is no car. Moreover, each pending detector is able to count the number of car, which leaves the incoming road to enter the hub area. In simulation, this detector has been made thanks to boxcollider as shown in figure 3 left. In real crossroads, regular magnetic loop can be used.

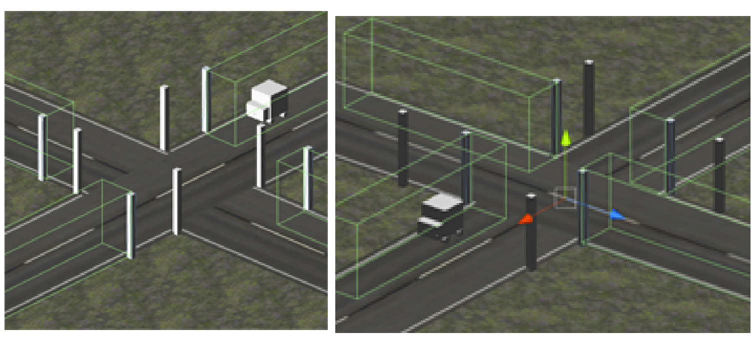


Fig. 3. Pending car detector (left) and output car detector (right) in Vivus Simulator

By the same, each output road is equipped by an output car detector, the goal of which is to count the number of vehicle exiting the hub. Combining this information with the number of vehicles in the hub counted by pending detectors, it is possible to determine if vehicles are still in the hub. These detectors have been also made with boxcollider in simulation (cf. figure 3 right) and can be made with magnetic loop in real road configuration.

Behavior: Regular traffic light. This behavior has been developed in order to obtain a reference in adaptive decision process evaluation. The solution used is based on the simplest possible control system. It uses a timer with a fixed passing time (T_p) whatever the chosen lane (i.e. there is no priority of one lane over the other). Each phase of the signal lasts for a specific duration before the next phase occurs; this pattern repeats itself regardless of traffic. When one light turns to red, traffic lights of the perpendicular lanes stay red for a fixed time called full red time (T_{fr}). This is made to avoid collision between cars that have passed the crossroad on orange and those that start early when light turns green. Older traffic light systems still use this synchronization model.

Behavior: Priority to first incoming vehicle. This first adaptive decision process is based on simple rules:

1. The first vehicle to enter in the perception range of the crossroad has the priority.
2. If more than one vehicle enter the perception at the same time priority is given to right.
3. If four vehicles enter at the same time (i.e. this corresponds to a situation which can not be solved by the two previous rules), priority is given randomly.
4. When all vehicles of one turn left the hub, decision process returns in step 1 until crossroad is empty.

Behavior: Configuration Analysis based decision process The previous hub adaptive behavior can improve the release time of the crossroad. However, there is still one drawback that should be overcome. Indeed, previous behavior processes only one vehicle at each loop. When the processed vehicle leaves the crossroad, hub takes into account next one. In order to optimize the release time, it has been decided to maximize the number of cars entering the hub. The main difficulty is not to introduce more road's conflicts. To that way, all the possible input configuration has been listed taking into account both the presence of the vehicle and the goal of them using turning light perception. Each configuration corresponds to a string of characters. The particularity of this encoding is that it is relative. it does not describe the routes according to the departure and arrival, but in relation to action that will realize the car (go right, go left ...). Table 1 shows the list of usable characters.

R	L	F	X
Vehicle goes to **Right** direction	Vehicle goes to **Left** direction	Vehicle goes **Forward**	There is no vehicle in the lane

Table 1. Usable characters for description of crossroad input configuration

For each configuration, we are now able to generate the corresponding code starting from bottom right and turning counterclockwise. Then the rank of each computed. It corresponds to the number of vehicle that is able to pass the hub at the same time. Table 2 shows all the possible configurations.

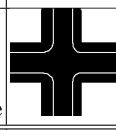
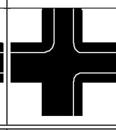
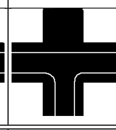
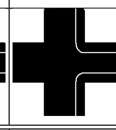
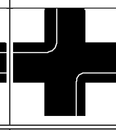
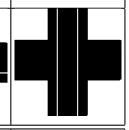
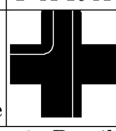
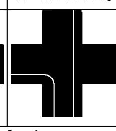
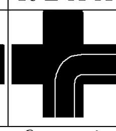
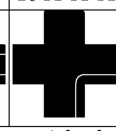
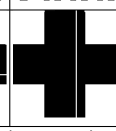
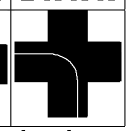
Rank	4	3	3	2	2	2
Coding	R R R R	R R R X	R F X R	R R X X	R X R X	F X F X
Example						
Rank	2	2	2	1	1	1
Coding	F X R X	F X X R	R L X X	R X X X	F X X X	L X X X
Example						

Table 2. Possible input configurations with their associated code

The other possible combinations can be deduced from these by rotation. This corresponds to circular permutation of the code. Once the code of the configuration is determined, the crossroad chooses the one which allows to get pass the maximum of vehicles.

4 Experimental results

4.1 Simulation description

Vivus Simulator:
Many simulators aimed at studying vehicles dynamics are existing. Among the most popular, we can cite Callas and Prosper and widely used in automotive industry. Most of them are focusing on mechanical simulation of the vehicle with a special focus on tyre/road contact. The main drawback of these is the requirement of real vehicle to build a dynamical model and the difficulty to integrate virtual sensors and onboard artificial intelligence abilities.

In this context, System and Transportation Laboratory decided to develop a simulation/prototyping tool, named Virtual Intelligent Vehicle Urban Simulator (VIVUS), aimed at simulating vehicles and sensors, taking into account their physical properties and prototyping artificial intelligence algorithms such as platoon solutions [12] or obstacle avoidance devices [13].

VIVUS was initially based on Java 3D for the 3D graphical part and on PhysX for sensors and vehicle dynamic behaviors [14]. The main problem of this solution is the communication between 3D and Physical part, since they are in different programming languages (Java for the 3D part and C++ for the physical one). Thus, it has been decided to use a simulation environment which can integrate both parts. VIVUS is now based on Unity3D [4]. Unity3D is an integrated authoring tool for creating interactive content such as architectural visualizations or real-time 3D animations. This engine allows to create real-time 3D environment with a real physic interaction between the elements situated in the 3D environment.

In VIVUS, simulating a vehicle consists in physical behaviors, sensors/perception and control board simulations.

Physics 3D model is based on the Physx engine, integrated into Unity3D. This engine is one of the best considering the accuracy and the realistic behavior obtained. In order to obtain simulation results as near as possible from the reality, a complete physical model of vehicles has been made. Models designed for Physx are based on composition of Physx elementary objects. The simulated vehicle is then considered as a rectangular chassis with four engine/wheel components. This choice can be considered to be realist, the chassis being made as a rectangular un-deformable shape. As for the engine/wheel components, vehicle platform owns 4 wheel drive each of them being directly linked to one electrical engine.

Each simulated vehicles are equipped by sensors. VIVUS allows to simulate different sensors :

- *image sensor* produces a bitmap
- *video sensor* produces a sequence of bitmaps
- *geometric sensor* produces information of collisions on a predefined set of rays or between object
- *location sensor* produces the vehicle position and orientation
- *state sensor* produces the state of the vehicle or one of its components (communications, engine, etc.) or the state of the simulated environment's components like weather report

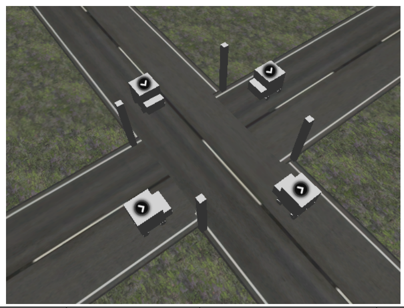
This simulator has been successfully used as a prototyping tool for sensor design/positioning and a development tested for artificial intelligence algorithms.

Experimental configurations and results:
In order to compare the presented approaches, three typical configurations have been chosen:

1. a sequence with one car at each lane with a R L F F configuration cf. figure 4.
2. a sequence with four 3-car trains. The configuration is then FRR, RLL, FFL, and LFL cf. figure 5.
3. a sequence with a mixing between small and big train with the following configuration FRLR, FF, L, FL cf. figure 6.

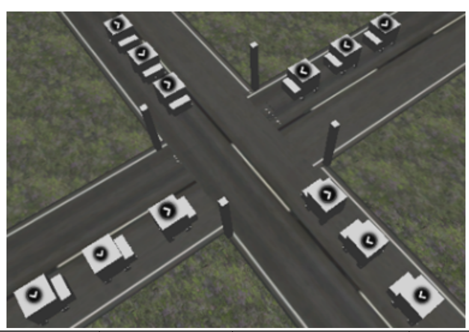
[4] http://unity3d.com/

Associated tables of figures 4, 5 and 6, gives the results obtained these three sequences. Release time, expressed in simulation cycle time, is the time required to process all the incoming vehicles. Mean waiting time is also expressed in simulation cycle time.



Algorithm	Traffic light	First in priority	Configuration Analysis
Release Time	102.70	67.34	60.09
Mean vehicle waiting time	59.75	38.97	33.91

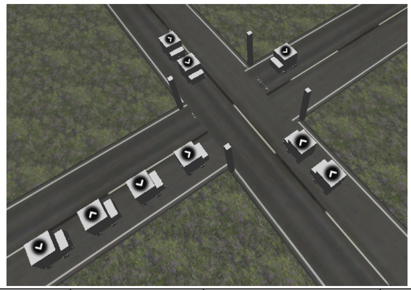
Fig. 4. Small trains reconfiguration and associated results



Algorithm	Traffic light	First in priority	Configuration Analysis
Release Time	264.70	210.34	156.47
Mean vehicle waiting time	151.91	105.56	98.25

Fig. 5. Big train reconfiguration and associated results

Results obtained show that both proposals drastically improve hub release time and mean waiting time as compared to regular traffic lights. The main benefits for the configuration analysis based decision process is linked to the



Algorithm	Traffic light	First in priority	Configuration Analysis
Release Time	145.2703	132.62	98.94
Mean vehicle waiting time	85.88	73.21	51.89

Fig. 6. Example of mixed train and associated results

waiting time since several vehicles can enter the hub at the same time provided they are not in conflict.

5 Conclusion

This paper presented two algorithms aimed at allowing platoon's reconfiguration while optimizing hub release time. These algorithms suppose several constraints on vehicle and crossroad perception abilities. Vehicles must be able to be part of a platoon and to detect specific active beacons placed on road. As for crossroad, it must perceive the presence and the expected direction of the first waiting vehicle for each incoming lane (the number of waiting vehicles is not perceived). It must also perceive if one vehicle exits the hub. This perception can be easily made thanks to magnetic loop detectors. Results obtained show that algorithms fulfilled their function (i.e. platoon reconfiguration) and gives better results than a standard traffic light solution.
From now on, we focus on composition of several crossroads into the grid. The goal of this is both to develop global path planning algorithms aimed at computing the best path in the grid for each mobility unit and to study the behavioral interaction between crossroads in order to improve time travel. We also are trying to adapt this developed solution to other crossroads configurations and to traffic circles.

References

1. P. Martinet, B. Thuilot J. Bom: Autonomous Navigation and Platooning using a Sensory Memory, International IEEE Conference on Intelligent Robots and Systems, IROS'06, Beijing, China, 0ctober 2006, 2006
2. Woo, Myung Jin, and Choi, Jae Weon: A relative navigation system for vehicle platooning, SICE 2001. Proceedings of the 40th SICE Annual Conference. International Session Papers (IEEE Cat. No.01TH8603), 2831, 2001

3. Fritz, Hans: Longitudinal and lateral control of heavy duty trucks for automated vehicle following in mixed traffic: Experimental results from the CHAUFFEUR project, IEEE Conference on Control Applications - Proceedings 2, volume 2, 13481352, 1999,
4. Ioannou, P., and Xu, Z.: Throttle and brake control systems for automatic vehicle following, IVHS Journal 1(4), volume 1, 345, 1994
5. Moskwa, John J., and Hedrick, J. Karl: Nonlinear algorithms for automotive engine control, IEEE Control Systems Magazine 10(3), volume 10, 8893, 1990
6. Daviet, Pascal, and Parent, Michel: Longitudinal and lateral servoing of vehicles in a platoon, IEEE Intelligent Vehicles Symposium, Proceedings, 4146, 1996
7. Sheikholeslam, Shahab, and Desoer, Charles A.: Longitudinal control of a platoon of vehicles with no communication of lead vehicle information: A system level study, IEEE Transactions on Vehicular Technology 42(4), volume 42, 546554, 1993
8. Lee, Hyeongcheol, and Tomizuka, Masayoshi: Adaptive vehicle traction force control for intelligent vehicle highway systems (IVHSs), IEEE Transactions on Industrial Electronics 50(1), volume 50, 3747, 2003
9. Kehtarnavaz, Nasser, Griswold, Norman C., and Lee, Juck S.: Visual control of an autonomous vehicle (BART)The vehicle-following problem, IEEE Transactions on Vehicular Technology 40(3), volume 40, 654662, 1991
10. Gehrig, S.K., and Stein, F.J.: Elastic bands to enhance vehicle following, IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, 597602, 2001
11. Yi, Soo-Yeong, and Chong, Kil-To: Impedance control for a vehicle platoon system, Mechatronics (UK) 15(5), volume 15, 62738, 2005
12. Jean-Michel Contet, Franck Gechter, Pablo Gruer, Abderafiaa Koukam. Reactive Multi-agent approach to local platoon control: stability analysis and experimentations. International Journal of Intelligent Systems Technologies And Application. 2010
13. Franck Gechter, Jean-Michel Contet, Pablo Gruer, Abderafiaa Koukam. Car driving assistance using organization measurement of reactive multi-agent system. Procedia CS1(1): 317-325. 2010
14. Olivier Lamotte, Stephane Galland, Jean-Michel Contet, Franck Gechter, Submicroscopic and Physics Simulation of Autonomous and Intelligent Vehicles in Virtual Reality, The Second International Conference on Advances in System Simulation (SIMUL 2010), August 22-27 - Nice, France, 2010
15. A.J. Miller, A computer control system for traffic networks in Proc. 2nd Int. Symp. Traffic Theory, 1963, pp.200220.
16. W. Ma, X.Yang, A passive transit signal priority approach for bus rapid transit system, IEEE Intelligent Transportation Systems Conference, NO. 413, Seattle, WA, USA, 2007.
17. M.Papageorgiou, C.Diakaki, V.Dinopoulou, A.Kotsialos and Y.Wang, Review of Road Traffic Control Strategies, Proceedings of the IEEE, Vol.91, No.12, December 2003, pp 2043-2067
18. R.A. Vincent and C.P. Young, Self-optimizing traffic signal control using microprocessor: The TRRL MOVA strategy for isolated intersections, Traffic Eng. Control, vol.27, pp.385387, 1986.
19. B.M. Pitu and Z.Ning. Queuing Models for Analysis of Traffic adaptive Signal Control, Intelligent Transportation System, Transaction On, 8(1), pp.50-59. 2008
20. M. Griinewald, C.Rust and U.Witkowski. Using mini robots for prototyping intersection management of vehicles, Proceedings of the 3rd International Symposium on Autonomous Minirobots for research and edutainment. pp. 288-294. 2006.