Network-based strategies for signalised traffic intersections

Magdi S. Mahmoud*, Faisal A. Al-Nasser and Fouad M. Al-Sunni

Systems Engineering Department,
King Fahd University of Petroleum and Minerals,
P.O. Box 5067, Dhahran 31261, Saudi Arabia
E-mail: msmahmoud @kfupm.edu.sa
E-mail: faisal_1997_n@yahoo.com
E-mail: alsunni@kfupm.edu.sa
*Corresponding author

Abstract: In this paper, different traffic light control structures over communication links, including the decentralised, quasi-decentralised, distributed and hierarchical networked structures, are considered. These structures used for coordinating multiple intersections, which could be a great application of networked control problem control for the signalised traffic light intersections that will help the designer to achieve certain objectives. Some of these objectives are, minimise the waiting time during the red light period and perform better control in the next green cycle, maximise the flow between consecutive intersections which will minimise the number of stops, minimise the average waiting time and more will be highlighted in this paper. A quick literature about all models used for traffic control problem is done. A generic state space model of traffic dynamics under these different control structures is proposed that takes into account many effects of lossy communication links such as networked induced delays, packet dropout and varying sample interval. Also, a sufficient condition for system stability is provided based on LMI. Finally, comparison of different types of networked control systems was done using MATLAB simulation.

Keywords: intelligent traffic control; packet dropout; decentralised networked control; quasi-decentralised networked control; distributed networked control.

Reference to this paper should be made as follows: Mahmoud, M.S., Al-Nasser, F.A. and Al-Sunni, F.M. (2013) ‘Network-based strategies for signalised traffic intersections’, *Int. J. Systems, Control and Communications*, Vol. 5, No. 1, pp.15–48.

Biographical notes: Magdi S. Mahmoud has been a Professor of Systems Engineering since 1984. He is now a Distinguished University Professor at KFUPM, Saudi Arabia. He has been actively engaged in teaching and research in the development of modern methodologies to distributed control and filtering, switched time-delay systems and fault-tolerant systems. He is the principal author of 30 books, inclusive book-chapters and the author/co-author of more than 490 peer-reviewed papers. He is the recipient of two national, one regional and two university prizes for outstanding research in engineering. He is a Fellow of the IEE, a senior member of the IEEE, the CEI (UK), and a registered Consultant Engineer of information engineering and systems (Egypt).

Faisal A. Al-Nasser finished a double major in Engineering at King Fahd University of Petroleum and Minerals, (KFUPM), BSc in Systems Engineering with Automation and Control and the 2nd BSc in Computer Engineering, both finished in 2003. He completed his MSc and PhD in Systems Engineering in 2007 and 2012, respectively. His areas of interest include control networks, control systems, multimedia communications, WLAN, WSN and more.

Fouad M. Al-Sunni is a Professor of Control and Automation and the Head of the Department of Systems Engineering. His areas of research include optimal control, robust control and identification of dynamic systems. He published in the areas of fuzzy logic, neural networks, optimal control, identification, time-delay systems, model predictive control and control and engineering education. He is a senior member of the IEEE and IIE, and a member of INFORMS and ISA.

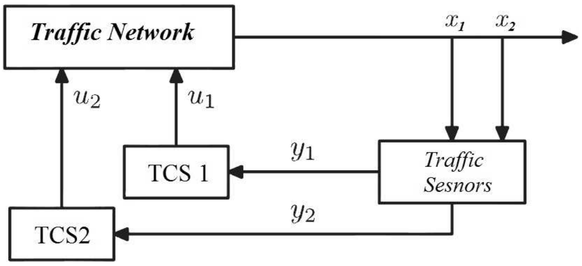
1 Introduction

Applications of systems engineering concepts and techniques have been the subject of numerous research investigations. In particular, the general problem of dynamic assignment of sensors to local fusion centres (LFCs) in a distributed tracking framework was considered in Tharmarasa et al. (2009) with particular emphasis on the frequency channel limitation and the advantage of variable transmitting power. In multi-agent systems, agents directly interact via a form of message passing. Information about these interactions can be analysed in an online or offline way to identify clusters of agents that are related. A dynamic model was proposed (Kubalik et al., 2010) for agent clustering and experimental results that demonstrate applicability of the approach. The effects of network delay among interconnecting elements of a distributed computer network control system were analysed in Martins and Jota (2010). Experimental tests were performed to show the influence of the combined effects of the network delay (between sensor/controller and controller/actuator) on the overall performance of a feedback control system. An architecture to avoid unwanted downtimes to a maximum extent in distributed industrial process measurement and control systems (IPMCS) was developed in Strasser and Froschauer (2012). The proposed concept facilitates the exchange of hardware components including automation and control devices with no need for extra control.

On another research front, in modern urban areas, the number of vehicles is gowning larger and larger and the requirements for travelling by vehicles are becoming more demanding than ever. Many large and sound traffic networks (freeways and roads) are already constructed, traffic congestion still cannot be avoided efficiently and it is time and money consuming to build more common transportation infrastructures or reconstruct the already existing one. So, traffic jams may occur frequently and it will lead to severe impacts, especially when people need to use the common infrastructures with limited capacity at the same time, during rush hours (Al-Nasser and Rowaihy, 2011). Traffic congestion will cause traffic delays, economic losses, traffic pollution, driver misbehaviour and so on. Therefore, effective traffic control methods are necessary to reduce traffic jams. Several traffic control strategies (Mehta, 2008; Zhang et al., 2007; Zhao et al., 2009; Narayanan et al., 2003; Abdulhai, 2003; Li and Wang, 2008) were proposed and implemented in the field, like fuzzy control, PID control, MPC control, to name a few. However, these algorithms are mainly focusing on controlling a single intersection or a single traffic control measure without global scope, and have limited control effect for the whole traffic network. In reality, each intersection is affected by situation in other intersections and a traffic jam may happen in one intersection because of abnormal case in another intersection in the same traffic network. So, it is necessary to understand the traffic networks behaviour, and to investigate the coordinated control approaches that can better coordinate and control traffic networks.

From control engineering view, each intersection is a control loop and the controllers can be designed to work in a decentralised fashion (see Figure 1). The two traffic control systems (for example, TCS 1 and TCS 2) are designed based on two different continuously-sampled outputs, y_1 and y_2, of the system. The two controllers do not exchange information and operate in a decentralised fashion which makes each intersection isolated from the others. Similar system is shown in Figure 2 but here it is over communication links, so it will be called decentralised networked control systems (DecNCS), and the red dashed lines represents the real-time network links. Communication networks make the transmission of data much easier, provide a higher degree of freedom in the configuration of control systems, allows for easy modification of the control strategy by rerouting signals, having redundant systems that can be activated automatically when component failure occurs, and in general, it allows having a high-level supervisory control over the entire system. In the context of networked control systems, key issues that need to be carefully handled at the control system design level such as data losses due to field interference, time delays due to network traffic as well as due to the potentially heterogeneous nature of the additional measurements, transmission constraints and more issues. As a result, the controller will do a control over network not through network and design shall be robust to all previously mentioned issues (Wei, 2008; Schendel et al., 2010; Nesic and Liberzon, 2007; Singh et al., 2011).

Figure 1 Traditional traffic control system with two control loops



To coordinate multiple intersections across a long road, we may need to consider other strategies, namely, the quasi-decentralised and distributed control. The quasi approach will allow minimum required information to be exchanged to do the required coordination, e.g., queues length, phase selected (a phase is any period in a cycle where non-conflicting traffic movements may run), while in distributed more information will be exchanged, e.g., cycle length (a cycle is the time to complete all phases in a timing plan), time splits (the amount of time allocated to a phase in a cycle), offsets (green signals at adjacent intersections are set to occur at a given time, relative to that at a reference intersection), …, etc. The use of communication network with previous strategies will introduce the quasi decentralised networked control system (quasi-DecNCS) (Sun and El-Farra, 2009a, 2009b) and distributed networked control (DNCS) (Oh and Sastry, 2006, 2007) (see Figure 3). The hierarchical control structure (Rohloff et al., 2006) (see Figure 4) can be used also in case if we are trying to control very large-scale traffic network with large number of intersections. Instead of giving all the control authority to local controllers, the hierarchical control structure divides the control problem into multiple control problems or zones at multiple levels.

Figure 2 Decentralised networked control system (see online version for colours)

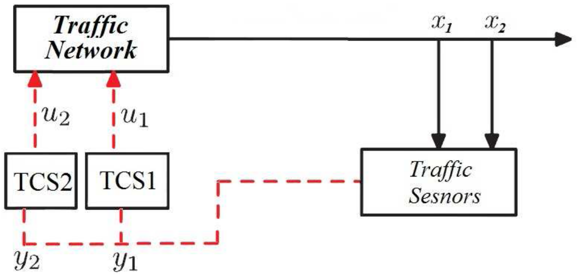
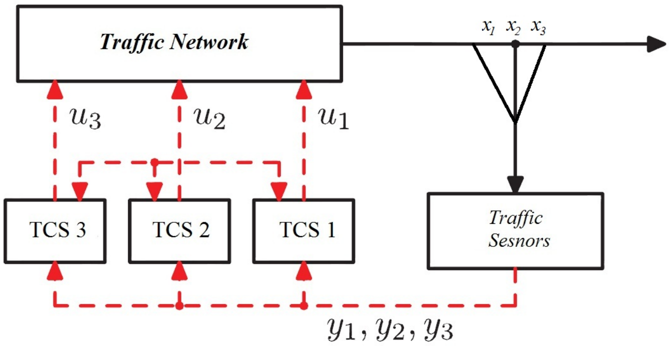


Figure 3 Distributed networked control system (see online version for colours)



All of these models can be used for the traffic signal intersection control to achieve wide range of objective functions (Mehta, 2008; Wen et al., 2007; Tubaishat et al., 2007; Albagul et al., 2006; Hirankitti et al., 2007; Zhao-Sheng et al., 2005; Martin and Radiocomun, 2012) such as:

1 minimise overall delay to vehicles

2 minimise the waiting time at the signal

3 maximise the service time for each signal

4 minimise delays to public transport

5 minimise delays to emergency services

6 minimise delays to pedestrians

7 equitable distribution of delays between competing traffic

8 maximise reliability

9 maximise network capacity

10 minimise accident potential for all users

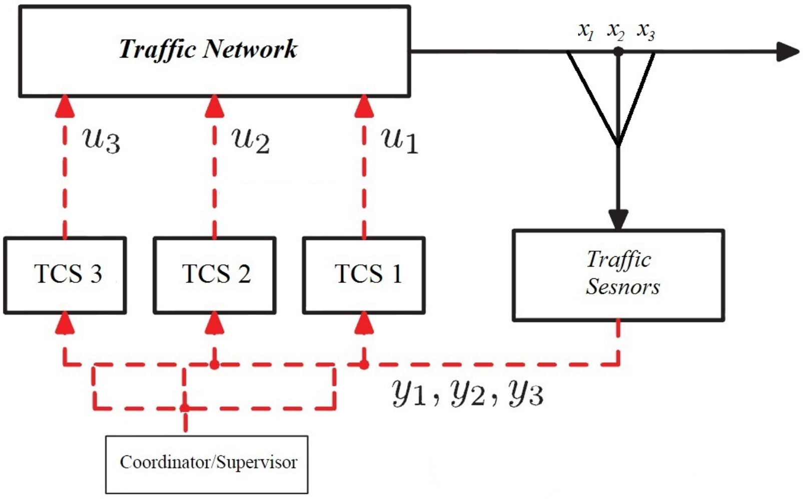
11 minimise environmental impact of vehicular traffic (noise, atmospheric pollution, visual intrusion)

12 energy efficiency

13 handling of the red light crossing violations

14 indication to the police traffic control room.

Figure 4 Hierarchical networked control system (see online version for colours)



It is important to note that some of the objectives do conflict and a compromise may have to be made in the selection of objectives. However, some objectives can be met in tandem, for example, minimising delay to vehicles would also help to minimise fuel consumption, atmospheric pollution and increase network throughput.

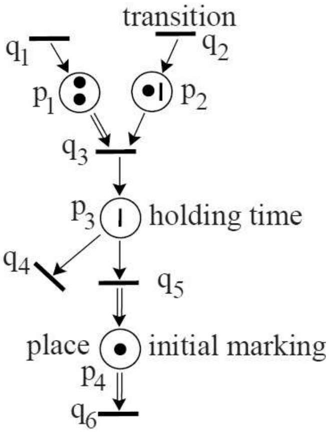
Another important issue for traffic control system is how to collect the traffic data. This can be done by using sensors network (wired/wireless) that will feed the control system with the number of incoming traffic data, passing vehicles and crossing the signals (Wen et al., 2007). An intelligent traffic light system senses the presence or absence of vehicles then it controls the traffic lights accordingly using one of the control approaches we have mentioned. The very obvious idea behind intelligent traffic systems is that drivers will not spend unnecessary time waiting for the traffic lights to change which may lead them to some traffic violations and accidents when some drivers start to lose their patience (Tubaishat et al., 2007; Albagul et al., 2006).

2 Signal traffic control

Traffic light control is a complex problem even for single intersection because there might be no obvious optimal solution. In case of multiple intersections, the problem becomes more complicated, because the situation of one intersection influences the flow of traffic towards many other intersections. Another fact that complicates this problem more, is the flow of traffic is constantly changing, depending on the time of day, the day of the week, and the time of year. More factors will add to the complexity of this type of problems like roadwork, improper roads design and accidents.

In practice, most traffic lights are controlled by fixed-cycle controller in which all traffic gets a green light at some point. The split time determines for how long the lights should stay for each direction. Busy roads can get preference by adjusting the split time and longer cycles lead to better performance for crowded intersections. The offset of a cycle defines the starting time of a cycle relative to other traffic lights and it can be adjusted to let several lights cooperate which will minimise the number of stops and hence the travel time. For pre-timed or fixed cycle controllers, we need to manually fine tune from time to time to perform well. Usually, a table of time-specific settings is used to enable a light to adapt to recurring events like rush hour traffic. This type of controller has several disadvantages: it needs a lot work, regular updates and it will not give any priority to the roads with higher traffic. Traffic may accumulate quickly and traffic jam can happen during few seconds in case the traffic control system is not efficient to properly manage the vehicles accumulation in fast and smart manner.

Figure 5 Petri net example



Many traffic light models were developed in the literature to enhance traffic light performance and efficiency. The following list shows these models:

1 Pre-timed control: all of the control parameters are fixed and preset offline.

2 Queue traffic light model (simple, extended, event driven): The queue length in each lane can be evaluated using different techniques depending on street width and the number of vehicles that are expected at a given time of day (Michal et al., 2006), also the store and forward model (Aboudola et al., 2009).

3 Knowledge-based models: are artificial intelligent tools that work in a narrow domain to provide intelligent decisions with justification (Nan et al., 2003) based on stored data like the detailed response time, incident duration, and lane-blockage conditions, …, etc.

4 Graph-based models (Petri net models): It transforms a real traffic network into a graph in which vertices represent the intersections of roads, and edges represent the road segments (Lefei, 2005; Ganiyu et al., 2011). These models have been used as a tool for various kinds of simulation and control logic. This type of model has some disadvantageous and it is hard to manage.

5 Sensors-based models: In these models, different types of sensors can be used including the wireless sensors. Examples of sensors can be inductive loop detectors, micro-loop probes, IR, LED, motion detectors and pneumatic road tubes (Chien, 2005; Mehta, 2008).

6 Extension neural network (ENN) model: consists of extension theory and a neural network that uses a modified extension distance (ED) to measure the similarity between data and a cluster centre (Kuei-Hsiang et al., 2009; Gartner et al., 2002; Serrano et al., 2005). It is developed to deal with object recognition in outdoor environments.

7 Reinforcement learning (RL) models: use machine learning framework (Are et al., 2010; Silva et al., 2006) to approximate an optimal decision-making policy where an agent, by interacting with and receiving feedback from its environment, attempts to learn an optimal action selection policy in an iterative manner that generates states, subsequent state, evaluation (reward/penalty) and value function (maximise/minimise). A commonly used RL algorithm is Q-learning which is a model-free RL algorithm.

8 Algorithm-based models: the famous algorithm used is the genetic algorithm (Leena et al., 2009) that uses the rules of nature and it is great advantage is that solution through evolution, but this is also the greatest disadvantage because it not necessarily to evolve towards a good solution and may evolve into an evolutionary dead end.

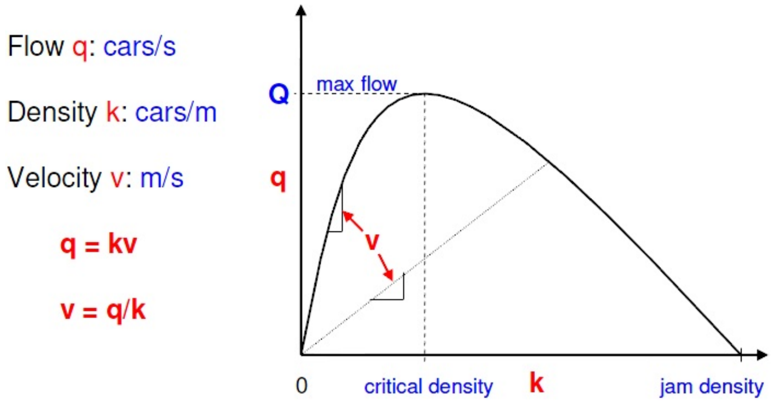
9 Fuzzy logic models: Fuzzy logic (Zhang et al., 2007; Zhao et al., 2009), offers a formal way of handling terms like more, less, longer, …, etc. The controller determines the time that the traffic light should stay in a certain state, before switching to next state but the controller can skip a state if there is no traffic in a certain direction.

10 Vision-based models: Video sensors (Serrano et al., 2005) have become particularly important in traffic applications due to several factors and they have the ability to monitor wide areas. Intelligent systems may use cameras for extraction/detection useful information such as traffic density, vehicle types and moving objects which is very helpful for traffic management especially in the mega cities.

11 Adaptive control models: Collect data in real-time from sensor systems to identify traffic conditions, then evaluate alternative signal timing strategies on a model of traffic behaviour to implement the best strategy according to some performance metric.

Several real-time traffic signal control strategies for urban networks developed in the past few decades. Some of these stratifies have been implemented in real-life conditions while others are still in the research and development stage. The authors in Aboudola et al. (2009) classified the strategies into two principal classes of signal control strategies. In the first class, strategies are only applicable to (or efficient for) networks with undersaturated traffic conditions, whereby all queues at the signalised junctions are served during the next green phase. In the second class, the strategies applicable to networks with oversaturated traffic conditions, whereby queues may grow in some links with an imminent risk of spill back and eventually even of gridlock in network cycles. SCOOT and SCATS are (Gartner et al., 2002; Serrano et al., 2005; Robertson and Bretherton, 1991; Pitu and Fei-Yue, 2005; Freeman et al., 1999) two well-known and widely used coordinated traffic-responsive strategies that function effectively when the traffic conditions in the network are below saturation, but their performance may deteriorate when severe congestion persists during the peak period. Other elaborated model-based traffic-responsive strategies such as PRODYN and RHODES employ dynamic programming while OPAC employs exhaustive enumeration. Due to the exponential complexity of these solution algorithms, the basic optimisation kernel is not real-time feasible for more than one junction.

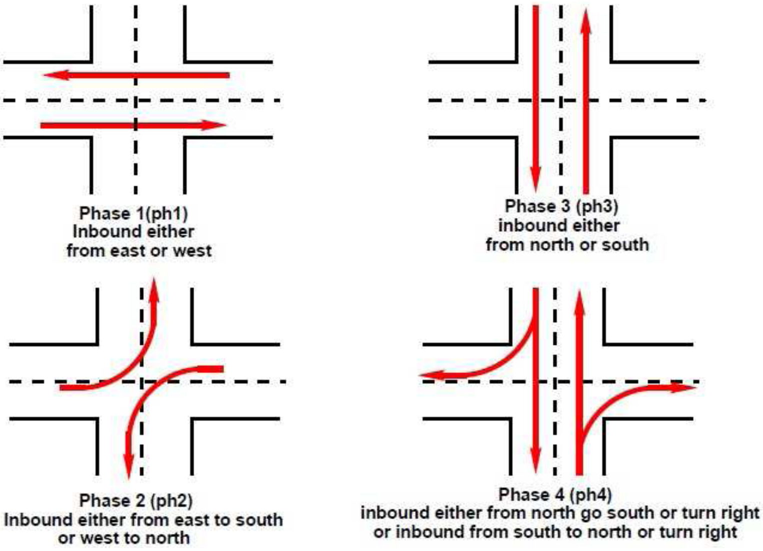
Figure 6 Traffic flow density relation



3 Traffic flow characteristics

Flow characteristics of traffic are fundamental in analysing intersection delay or capacity. Vehicles occupy space and, for safety, require space between them. With vehicles moving continuously in a single lane, the number of vehicles passing a given point over time will depend on the average headway or the average arrival rate per unit time. These factors (interruption of flow, stopping, and starting delay) reduce capacity and increase delay at a signalised intersection as compared to free-flow operations. Vehicles that arrive during a red interval must stop and wait for a green indication and then start and proceed through the intersection. The delay as vehicles start moving is followed by a period of relatively constant flow. The phase or cycle must be bounded to maintain the fairness for all directions in that intersection and shall not push the situation to exceed the saturation level which will lead to intersection traffic jam. Phasing reduces conflicts between traffic movements at signalised intersections. A phase may involve (one or more vehicular movements, a combination of vehicular and pedestrian movements, one or more pedestrian crossing movements). The National Electrical Manufacturers Association (NEMA) has adopted and published precise nomenclature for defining the various signal phases to eliminate misunderstanding between manufacturers and purchasers. Figure 7 illustrates a four-phase sequence separating all vehicular conflicts. Holding the number of phases to a minimum generally improves operations. As the number of phases increases, cycle lengths and delays generally increase to provide sufficient green time to each phase. However, actuated control skips phases with no traffic present and terminates certain movements when their traffic moves into the intersection. This capability produces a variation in the phasing sequence. The phasing options selected may be changed with the signal timing plan.

Figure 7 Example of four phases intersection (see online version for colours)



4 Traffic dynamics and problem definition

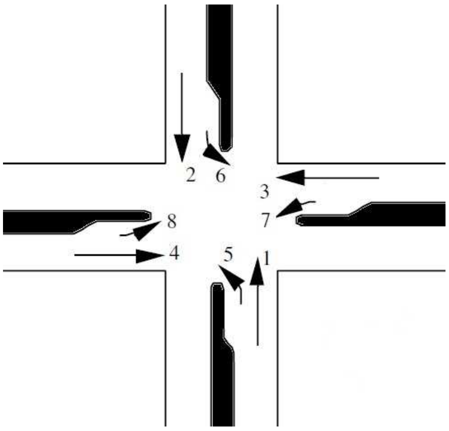
Controlling the traffic light intersection requires a prior knowledge of that intersection and the traffic load to be able set the proper parameters for the control algorithm, especially if the system used is not an intelligent system like time-based traffic control. Basically most of the traffic signals intersections have four directions queues, North (*N*), South (*S*), East (*E*) and West (*W*). The other queues possibilities are North West (*NW*), South East (*SE*), East South (*ES*) and West North (*WN*). The model in Figure 8 simply shows that two directions can be open at the same time, for example, *N* and *S* direction will move then *W* and *E* at the same time because there is no turning in other directions like NW or SE. Then, the other directions *NW*, *SE* and *ES*, *WN* can move at same time. For simplicity, we will give a number for each queue q_i where $i = 1, \ldots, 8$ for the following in order (*N*, *S*, *E*, *W*, *NW*, *SE*, *ES*, *WN*). The intersection consists of four streets with eight possible queues, assuming all right side movements are free and do not require a signal. The state equation for the continuous traffic flow process associated with any movement *i* that is sampled every Δt seconds, where time is indexed with the integer *k*, can be expressed by the current queue $q_i(k)$:

$$\begin{aligned} &q_i(k+1) = q_i(k) + \Delta q_i(k) + \Delta p_i(k), i = 1, 2, ..., 8 \\ &\Delta q_i(k) = q_i^{in}(k) - q_i^{out}(k) \\ &\Delta p_i(k) = p_i^{in} - p_i^{out} \end{aligned} \tag{1}$$

where $q_i^{in}(k)$ is the incoming new vehicles at time interval $[k - 1, k]$ in link or queue *i*, $q_i^{out}(k)$ is the number of vehicles able to pass the intersection during the green signal interval *Tg* from link or queue *i*, also *Tg* can be called as the control interval or green time for a certain phase, $q_i(k - 1)$ is the queue of vehicles waiting for green signal to happen at time *k*, $\Delta p_i(k)$ represents the fluctuation between a parking lot and link *i* or the effects of any non-controlled intersection between any two intersections where p_i^{in} used for vehicles left the parking or came from non-controlled intersection and joined the traffic in the queue *i* and p_i^{out} for vehicles left the queue *i* and went for a parking or went into a sub road or what we call it non-controlled intersection. These disturbing flows can be considered either as disturbance or as known perturbations if they can be well measured or estimated. In case of these uncertainties or perturbations are unknown and cannot be measured, then robust control system is needed. The output $q_{out}(k)$ can further be expressed as a function of the current control of the intersection, $u(k)$, and the current queue, $q(k)$:

$$q_{out}(k) = f_{out}\left(u(k), q(k)\right) \tag{2}$$

Figure 8 Typical traffic signal intersection control



The general discrete LTI state space representation the following (state space model is similar to the one in Aboudola et al. (2009) but in our work we used it differently, add to it more parameters and extended their work to cover the communication over networks):

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + Fd(k) \\ y(k) &= Cx(k) \end{aligned} \tag{3}$$

Figure 9 Queues uncertainties in traffic between intersections

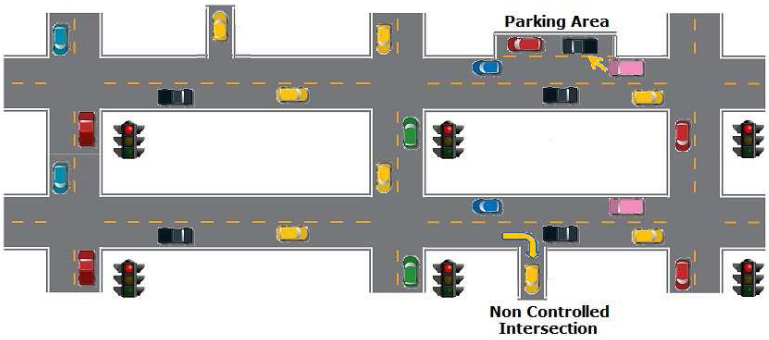
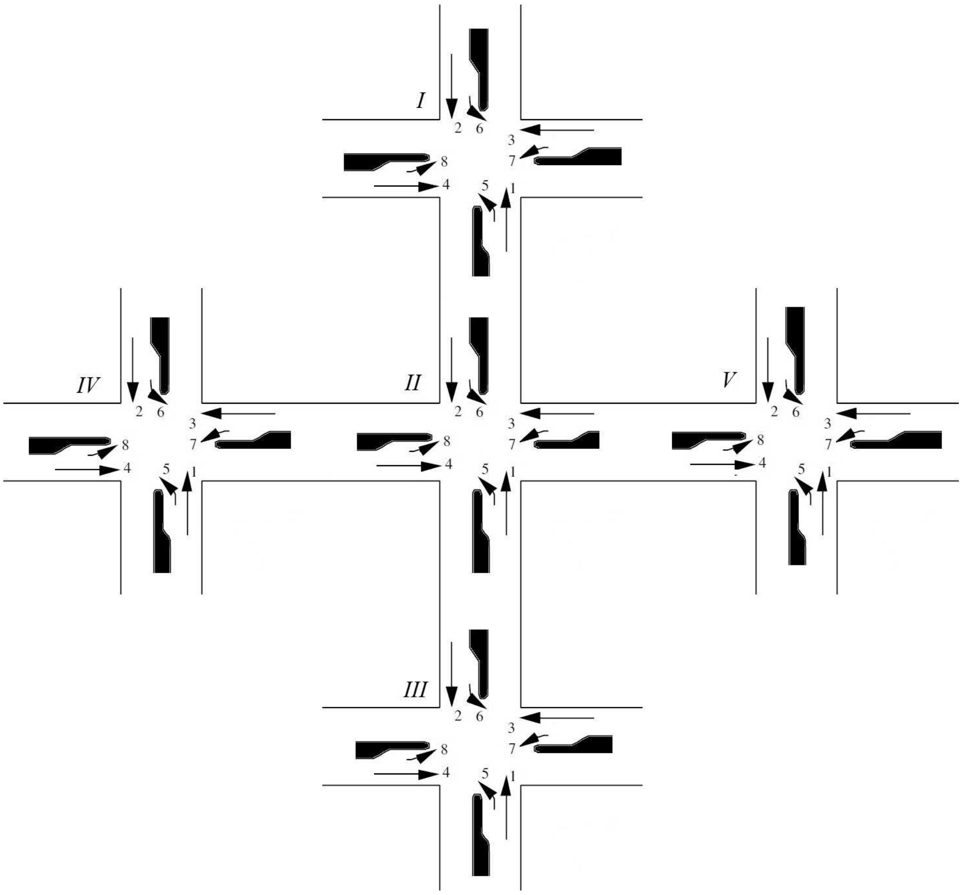


Figure 10 A traffic network with five intersections



Using equation (3), it is possible to describe the dynamics of a traffic network with the following: The state matrix A is set to be as an identity matrix where the elements of the state vector $x(k)$ represent the queue length $q_i(k)$ in link i considering all lanes in the road. The second term of the state equation is the product of input matrix B and control input u where the vector u store the green times Tg_i of all phases. As will be shown in equation (6), the Tg is proportional to the waiting time Tw where one of the major objectives is to *minimise Tw*. Matrix B composed of saturation and turning rates. The diagonal values of B are negative and represents the saturation flow and the product of $B_{ij}u_i$ where $i = j$, diagonal elements shows the outflow from link i. The other elements in B_{ij} where $i \neq j$ contains the turning rates from link i to link j. The assumption here and it is usually there in reality, the number of states is equal to the number of controlled links in the network. The product $Bu(k)$ is shows the difference of in and out flow for the traffic in the link or queue i during the control interval. Each output inside of the network is a measured state (number of vehicles of the link i) that makes the output equation simplified to $y(k) = x(k)$ and $C = I$. Finally, the traffic coming from non-controlled intersections or parking are considered as disturbance to the system in $d(k)$. The equation (3) can be rewritten as:

$$\begin{aligned} Q(k+1) &= AQ(k) + BG(k) + Fd(k) \\ Q^{out}(k) &= CQ(k) \end{aligned} \tag{4}$$

where $Q(k)$ is a vector of queues information for all the eight directions showing in Figure 8, and $G(k)$ contains the green timing for each direction.

$$\begin{aligned} Q(k) &= \left[q_1(k)q_2(k).....q_8(k)\right]'_j \\ G(k) &= \left[Tg_1(k)Tg_2(k).....Tg_8(k)\right]'_j \end{aligned} \tag{5}$$

Following the same manner, we can generalise that to traffic networks with multiple intersections. In a traffic network with n intersections, the order of the dynamic equations is increased to $n \times m$ where m is the number of possible movements in that intersection, for example, in Figure 10 we have $m = 8$ for any intersection. However, any complicated traffic network can be decomposed into a group of small ‘elementary networks’, similar intersections. In this manner, the study of the entire traffic network can be reduced to the analysis of these elementary networks and the inter-connections between them.

5 Problem constraints

There are several constraints which have to be taken into account and these can determined by the geometry of the traffic network and we can list them as the following:

- Queue/link capacity: is defined by the maximum number of vehicles for link i and it can be determined by the length of link between two intersections. So $0 \leq q_i(k) \leq q_{i,\max}(k)$.
- Control constraint: the maximum Tg is the interval of seconds of green time for link or queue i shall not exceed certain value to be fair with other links during the same cycle time. $Tg_{i,\min} \leq Tg_i(k) \leq Tg_{i,\max}$, see Figure 11.

- Waiting time: the time *Tw* spent by vehicles waiting until the signal becomes green. It is very important to *minimise* this time as much as possible by providing good service mechanism at the signalised intersection. This parameter can be calculated for direction *i* at intersection *j* by taking the sum of all other directions green time, or simply the phases because two directions can be in one phase so easier to use the phase *p*, in the same intersection *j*.

  $$\begin{aligned} &Tw_{ij} \geq \sum_{m \neq i} \left[Tg_{mj} \right] \\ &m = 1, 2, 3, \ldots, p \end{aligned} \tag{6}$$

  as example if we have four phases it, the estimated Tw_1 which is same as Tw_2 because both are in phase p_1, it will be:

  $$Tw_{p1} = Tg_{p2} + Tg_{p3} + Tg_{p4} \tag{7}$$

  we can see a direct relation between equation (4) and *Tw*. Theoretically, the total service time *Ts* required for one phase to pass all the cars waiting in a queue q_i is depending on τ_s the service time required to pass one row of cars at the same time and the physical structure of the street, here we mainly focus on the number of lanes. So, $T_s = (\tau_s * q_i) / NoOfLanes$. Also, as part of *Tg* there is an important component which is the startup delay time *Td* where the drivers take few seconds sometimes to realise the green LED is ON.

- Cycle time: the cycle time *Tc* is the time to complete the execution of all phases for the intersection and it shall be bounded by a certain value, $\sum Tc_j \leq Tc_{\max}$. It is also possible to choose the maximum best cycle time of any phase in that cycle as shown in Figure 12. The cycle time may vary due to traffic situation. In case of heavy traffic the best way is to have long cycle times to maximise steady-state flow. In contrast, when the traffic is light the better is the short cycle time to minimise the delays for vehicles. Another important issue for the cycle time selection which is related to the nature of intersection control whether it is for single or multiple intersections. For single intersection, the ratio of $R_i(k) = (flow_{in} / flowout_{\max}) < 1$ shall not reach $R_i(k) = 1$ which is the saturation level and the traffic jam will occur at $R_i(k) = 1$ which is the worst scenario. If this happened, then phase time shall recalculated to have the ratio $R_i(k) / P_i(R_i(k))$. In case of multi-intersections, we need to have careful timing to achieve the best throughput with good platoon management to make the flow of vehicles smooth through several intersections with less delay (green-wave progression or successive green signals) and to minimise overall delay and/or number of stops. This concept is explained in Figure 13 where we can see the platoon of vehicles are moving in the two parallel directions in manner that number of stops are minimised and this because of proper coordination between intersection controllers.

- Dilemma zone: a dilemma zone (Liu et al., 2006) is a range, in which a vehicle approaching the intersection during the yellow phase can neither safely clear the intersection, nor stop comfortably at the stop-line and it is one of the main contributors to signal-related accidents. Note that both the length and the location of a dilemma zone may vary with the speed of the approaching vehicles, driver reaction times, and vehicle acceleration/deceleration rates (this will not be considered in this work).

Figure 11 Green time extension

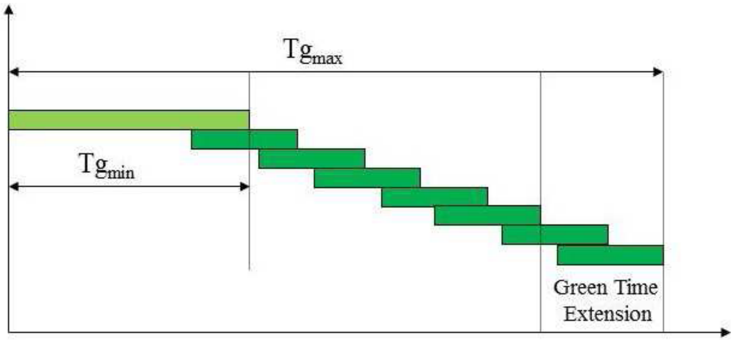


Figure 12 Best cycle time

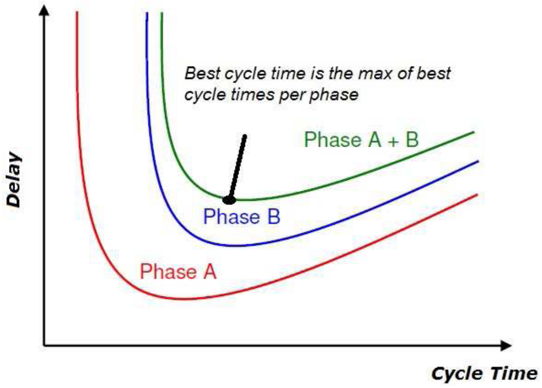


Figure 13 Multi-intersection control timing with coordination

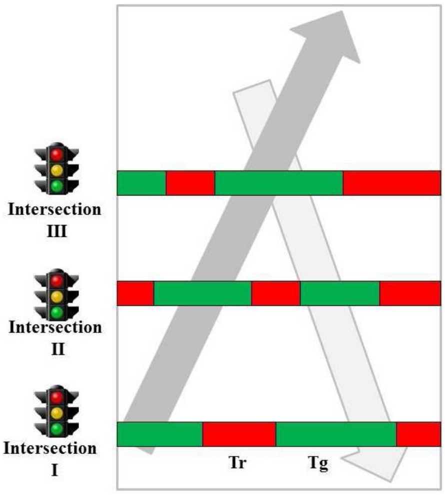
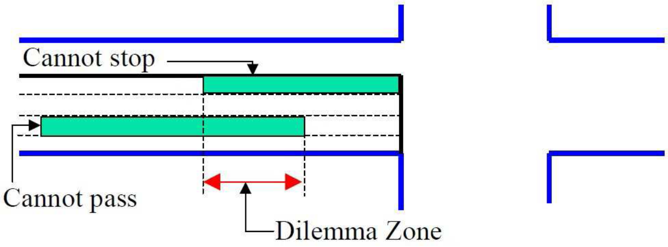


Figure 14 Dilemma zone



6 Communication link impacts

The use of communication link between sensors and intersection controller will introduce some network issues due to the nature of this shared link. The following list summarise in brief the most important effects on the transmitted signals:

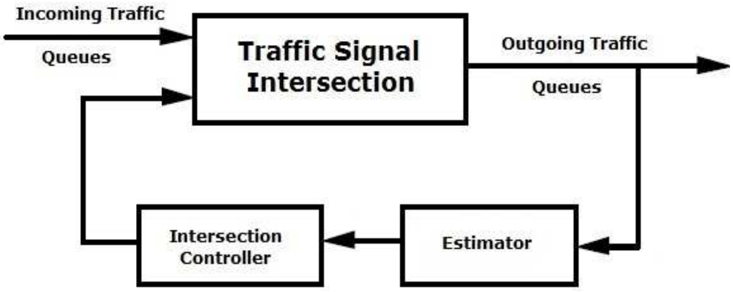
- Sensors packet dropout: if the packet does not arrive before the end of the sample period then packet is lost. So, the intersection controller may use the previous queue data to apply the control value but it should not be less than minimum value of the green time period. $y_k = \beta_k \hat{y}_k + (1+\beta_k) y_{k-1}$ and $\beta_k \in \{0, 1\}$, we used the term $\hat{y}_k$ to indicate that it is the networked version that could be same as the original value or little vary from the original sent value.
- Varying sampling interval: Due to the nature of the network, the actual sampling times are not necessary to equidistant in time. For a constant sampling interval *h* and instead we will use h_k.
- Transmission constraints: Since the controller and sensors are communicating over a communication link, it is possible to have a type of network that allows one node to access the network and transmits its corresponding values at each sampling time. This will add constraints on the transmission of sensors data, and we know that the actual $\hat{y}_k$ is not equal to the plant output y_k due to network effects. In other words, we can say that $\hat{y}_k$ is a networked version of y_k or the noise corrupted signals. $\hat{y}_k^i = \Gamma_{\sigma_k}^y y_k + (1-\Gamma_{\sigma_k}^y)\hat{y}_{k-1}$ where σ_k is used as switched function to determine which node will have the access to transmit, Γ_ℓ^y are diagonal matrices where the jth diagonal value is 1 if output belongs to node and zero otherwise.
- Network induced errors: the network induced error can simply be shown as discrepancies between current and the most recently transmitted input/output values of nodes' signals and it can be used to design dynamic output feedback and communication protocol without the need for knowledge about the controller and plant states. Also, the network induced error can be used for transmission scheduling where the node with highest error will have the higher chance to obtain the network access for transmission. $e_k^u = \hat{u}_{k-1} - u_k$, $e_k^y - \hat{y}_{k-1} - y_k$. We can define threshold levels γ_i^u, γ_i^y for the induced error where $e_k^u < \gamma_i^u$ and $e_k^y < \gamma_i^y$ for each subsystem *i*.

The intersection controller will run based on the pre-timed tables in case of any significant delay or many packets dropout to avoid open loop problem which will lead to huge traffic accumulation and violations. That will continue for one cycle until next data arrived properly otherwise after certain number of similar problems, alarms will be sent to traffic control room operator for maintenance and troubleshooting.

7 Decentralised networked control structure

The computational complexity of a large traffic network can be reduced efficiently by dividing the network into small intersections, and controlling the local intersection controllers separately in a decentralised structure over a communication network. The traffic flow interactions between intersections are cut off (or disconnected) and because of that, the local controllers may not be able to find the real optimal solutions for the intersections. Moreover, since the intersection is completely disconnected, the overall performance of the whole network will be deteriorated when we have a high traffic flow between intersections along that highway.

Figure 15 Traffic controller using estimator at the intersection



By applying this structure, we will have the generalised model for system that has five traffic light intersections as the following:

$$\begin{aligned} &Q(k) = \left[Q_1 Q_2 \ldots Q_j\right],\ j = 1, 2, \ldots, 5 \\ &Q_j(k) = \left[q_{1,j}(k) q_{2,j}(k) \ldots . q_{8,j}(k)\right]', \\ &q_{i,j}(k) = q_{i,j}(k-1) + \Delta q_{i,j}(k),\ i = 1, 1, \ldots, 8, \\ &\Delta q_{i,j}(k) = -q_{i,j}^{out}(k) \end{aligned} \tag{8}$$

Here, $\Delta q_{i,j}(k)$ is (–) because we do not consider the incoming traffic from other intersections, and hence the state space model will be

$$\begin{aligned} &x_j(k+1) = A_j x_j(k) + B_j u_j(k) \\ &y_j(k) = C_j x_j(k) \end{aligned} \tag{9}$$

where j represents the intersection number, as we can see from this structure the queues information between intersections are not known in advance so we may use an estimator to help the intersection controller to perform better by having some estimates about the new queue length considering the outgoing traffic. The assumption in this structure is a full decentralisation where all intersections are fully isolated and each one working independently from each other which is the case in many intersections in several countries. So, we will not discuss the communication link effects for this structure between intersections controllers. However, the data collection for each lane coming to an intersection is communicating via shared link to the intersection controller which will address some of the issues mentioned earlier.

8 Distributed networked control structure

Similar to the DecNCS, distributed networked control (DNCS) also uses independent local controllers for different subsystems. Different from DecNCS, the local controllers exchange information and coordinate between each other. Therefore, each local controller will make its own decisions based on both information from the subsystem itself and the information obtained from other subsystems. The more complete information the local controllers have, the better overall performance of the whole traffic network will be achieved. However, if the amount of information that the local controllers take into consideration of increases, the computational complexity will become very high. By applying this structure we will have the generalised model for system shown in Figure 10 that has five traffic light intersections as the following:

$$\begin{aligned}
&Q(k) = \left[Q_1Q_2....Q_j\right],\ j = 1, 2, ..., 5\\
&Q_j(k) = \left[q_{1,j}(k)q_{2,j}(k)....q_{8,j}(k)\right],\\
&q_{i,j}(k) = q_{i,j}(k-1) + \Delta q_{i,j}(k),\ i = 1, 2, ..., 8,\\
&\Delta q_{i,j}(k) = q_{i,j}^{in}(k) - q_{i,j}^{out}(k)
\end{aligned} \tag{10}$$

as we can see that we consider all the incoming traffic from other intersections where for example, the traffic coming from intersection *III* from queue 6, 3 will affect the queue in intersection *I* in queue 6, 8 and so on. and hence the state space model will be

$$\begin{aligned}
&x_j(k+1) = A_jx_j(k) + B_ju_j(k) + H_j(k),\\
&y_k^j = C_jx_j(k) + W_j(k)
\end{aligned} \tag{11}$$

where $H_j(k) = \sum_{n=1,n\neq j}^{5} A_{n,j}x_n(k)$ that contains the information about the other intersections queues that may help the current intersection in case of long queue there to pro act to minimise the vehicles accumulation in that lane and $W_j(k) = \sum_{n=1,n\neq j}^{N} C_{n,j}x_j(k)$ to show the information about the output queues from other intersections that is exchanged between the controllers. For example, the intersection *I* controller in Figure 10 will be able to know the status of the signal at line 6 from intersection *III* if it is green and also the queue length and the output queue during the green period will be also sent before that to controller at intersection *I*, then there could be several scenarios to minimise the queue length at line 6 in the intersection *I* by extending the *Tg* where $Tg < Tg_{\max}$, (see Figure 11), if it is green or give the priority to this side if the other sides in the intersection *I* has lower queue length or minimise the *Tg* for the other sides if the queue lengths are smaller.

Since we are using a control over a communication network, then we may have some problems due to the use of the share communication link such as delay, packet dropout, varying sample interval and transmission constraints. Each intersection will have an information about the other intersection outgoing queue which will help for better estimation and control for the value of *Tg* and also in case the next traffic signal is too crowded the proceeding intersection controller will try to delay the traffic by using the minimum *Tg* to avoid sending more traffic for that crowded intersection hopefully the jam will be released during the next cycle.

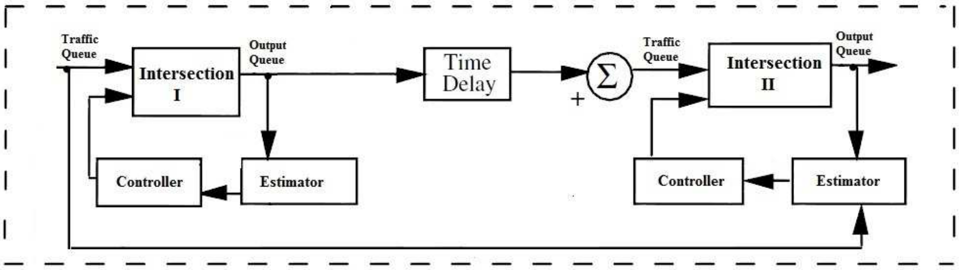
9 Quasi-decentralised networked control structure

To solve the problem where a DecNCS structure cannot provide the required stability and performance properties, and to avoid the complexity and high exchange of information required between controller in DNCS, a quasi-decentralised control (partially decentralised and not fully distributed) strategy with minimum cross communication between the intersections offers a suitable compromise and it provides a way of ensuring partial knowledge of how the local controller is affecting the global system. The term quasi-decentralised control refers to a situation in which most signals used for control are collected and processed locally, although some signals (the total number of which is kept to a minimum) still need to be transferred between local units and controllers to adequately account for the interactions between the different units and minimise the propagation of disturbances and process upsets from one unit to another.

$$\begin{aligned}
&Q(k)=\left[Q_1Q_2....Q_j\right],\\
&Q_j(k)=\left[q_{1,j}(k)q_{2,j}(k)....q_{8,j}(k)\right],\\
&q_{i,j}(k)=q_{i,j}(k-1)+\Delta q_{i,j}(k),\ i=1,2,...,8,\\
&\Delta q_{i,j}(k)=q_{i,j}^{in}(k)-q_{i,j}^{out}(k)
\end{aligned}\tag{12}$$

where $q_{i,j}^{in}(k)$ is the incoming new vehicles at time interval $[k-1, k]$ for intersection *j* for queue line number *i*, $q_{i,j}^{out}(k)$ is the number of vehicles were able to pass the intersection *j* during the green signal interval, *Tg* for the queue line *i* at that intersection and $q_{i,j}(k-1)$ is the queue of vehicles were waiting for green signal to happen at time *k*.

Figure 16 Two intersections traffic controllers



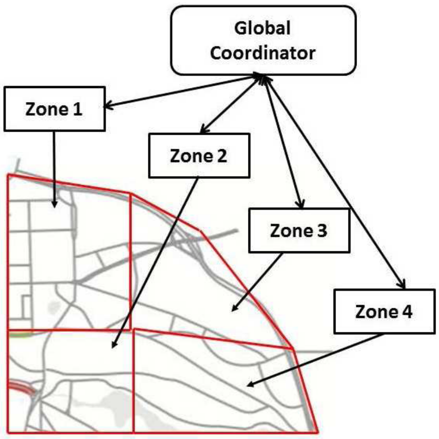
The discrete state space for the generalised model with multiple intersections can be shown as the following:

$$\begin{aligned} x_j(k+1) &= A_j x_j + B_j u_j(k) + H_j(k), \\ y_k^j &= C_j x_j(k) \end{aligned} \tag{13}$$

10 Hierarchal structure

The main aim of this structure is to perform traffic management at a strategic level in urban, interurban or mixed areas. The city or traffic network where traffic has to be supervised is divided in several sections called problem areas or zones. The decomposition of the city into zones allows a better analysis and understanding of the causes and evolution of traffic problems than if performed from a global perspective. Each zone controller will try to manage the coordination between its' local intersection properly achieving certain objectives. A master coordinator will monitor all zones behaviour and react in certain manner to achieve the global objective, e.g., avoid transferring the congestion from zone to another by rerouting the traffic in case of accident. The zone may overlap with surrounding zones sharing, for instance, some signals but using them from different points of view. So, a problem area or zone is a part of a city where traffic behaviour is locally studied and suitable control actions may be defined to improve the traffic state. In general, the zone controller understands traffic problems as an imbalance between capacity and demand which lead to dramatic increase in density that extremely affect the smoothness of traffic flow. In a city network, the problems appear when a queue of vehicles propagates to the surrounding streets, blocking intersections and generating a series of congestions.

Figure 17 Hierarchal traffic control – dividing the network into problem areas or zones (see online version for colours)



11 Closed-loop models

The most common and systematic approach is to use a dynamic output feedback, where the controller (or compensator) has its own dynamics. The simplest form is an observer structure

$$\begin{aligned} &\tilde{x}_j(k+1) = A_i\tilde{x}_j + B_j u_j + L_j\left(y_j - C_j\tilde{x}_j\right) \\ &u_j = -K_j\tilde{x}_j,\ j = 1, ..., 5 \end{aligned} \tag{14}$$

In this simple approach, $\tilde{x}_j$ is an estimate for the actual x for each subsystem i and we need to pick a good observation gain L_i such that $\tilde{x}_j \to x$ as fast as possible. In this work, we will use observer-based controllers in the sense that for each intersection of the traffic network we have one observer-based controller and the controllers exchange information or not based on the selection of the structure from the three we have mentioned previously in this paper. The i^{th} networked observer-based controller is given by considering the network side effects we have discussed in this work:

$$\begin{aligned} &\tilde{x}_j(k+1) = A_j\tilde{x}_j(k) + B_j\hat{u}_j(k) + O_j + H_j \\ &O_j = L_j\Gamma_j^y\left(\hat{y}_j(k) - C_j\tilde{x}_j(k)\right) \\ &H_j = \sum_{j=1, j\neq i}^{N} A_{i,j}\hat{x}_j(k) \\ &\hat{u}_j(k) = -K_j\tilde{x}_j(k) \end{aligned} \tag{15}$$

where $\tilde{x}_j(k+1)$ represents the state estimate at time k +1 for the plant state $x_j(k + 1)$, $B_j = \left[\int_0^{h-\tau_{rt}^k} e^{As} ds\right] B$ when $\tau_{rt}^k \leq h$ where h is the sampling interval. The output-related matrices $L_j(k)$, K_j, $j = 1, \ldots, 5$ are the subsystem gain matrices. The state estimation error is $\psi_j(k) = \tilde{x}_j(k) - x_j(k)$.

To deal with the communication constraints, the observer structure is used where the standard output is applied only when a new measurement is received. The dynamic of all controllers can be shown in discrete model that composed of block diagonal matrices written as follows for the DNCS, DecNCS and quasi-DecNCS

$$\xi_{k,DNCS} : \begin{bmatrix} x_k & \psi_k & e_k^y & H_k & W_k \end{bmatrix}^t \tag{16}$$

$$\xi_{k,DecNCS} := \begin{bmatrix} x_k & \psi_k & e_k^y \end{bmatrix}^t \tag{17}$$

$$\xi_{k,Quasi} := \begin{bmatrix} x_k & \psi_k & e_k^y & H_k \end{bmatrix}^t \tag{18}$$

by combining the foregoing relations, the overall closed-loop dynamics can be expressed as

$$
\begin{aligned}
&\xi_{k+1,DNCS} = \mathcal{A}_{k+1,DNCS}\xi_k \\
&\mathcal{A}_{cl,DNCS} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{bmatrix}
\end{aligned}
\tag{19}
$$

$$
\begin{aligned}
&a_{11} = A_j + B_j K_j,\, a_{12} = -A_j + L_j C_j, \\
&a_{22} = A_j - L_j C_j, \\
&a_{31} = C_j\left(-A_j + \alpha_k K_j B_j + I\right), \\
&a_{32} = \alpha_k K_j B_j C_j, \\
&a_{33} = \left(I - \beta_k \Gamma_j^y\right), \\
&a_{44} = I, \\
&a_{55} = I,
\end{aligned}
\tag{20}
$$

The others non-mentioned elements are zeros.

For the fully decentralised structure in which case the exchange of state information among subsystems is not allowed, then it will be reduced to:

$$
\begin{aligned}
&\zeta_{k,DecNCS} := \begin{bmatrix} x_k & \psi_k & e_k^u \end{bmatrix} \\
&\zeta_{k+1,DecNCS} = \mathcal{A}_{DecNCS}\zeta_k \\
&\mathcal{A}_{DecNCS} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & 0 & 0 \\ a_{31} & a_{32} & a_{33} \end{bmatrix}
\end{aligned}
\tag{21}
$$

and finally, for the QuasiNCS the $\mathcal{A}_{QuasiNCS}$ will be as the following:

$$
\begin{aligned}
&\xi_{k,QuasiNCS} := \begin{bmatrix} x_k & \psi_k & e_k^y & H_k \end{bmatrix}^t \\
&\zeta_{k+1,QuasiNCS} = \mathcal{A}_{QuasiNCS}\zeta_k \\
&\mathcal{A}_{QuasiNCS} = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 \\ a_{21} & 0 & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ 0 & 0 & 0 & a_{44} \end{bmatrix}
\end{aligned}
\tag{22}
$$

To sum up, the foregoing control structures can be cast into the generic form

$$
\begin{aligned}
&\zeta_{k+1} = \mathcal{A}\zeta_k \\
&\mathcal{A}_{cl} = blocking\left\{\mathcal{A}_{1,cl}, \ldots, \mathcal{A}_{N,cl}\right\}
\end{aligned}
\tag{23}
$$

12 Stability analysis

In the sequel, we define a global Lyapunov functional by

$$V = \xi_k^t \mathcal{P} \xi_k, \mathcal{P} = blocking\{\mathcal{P}_1, ..., \mathcal{P}_N\}, \mathcal{P}_j > 0 \tag{24}$$

Evaluating the first difference ΔV along the solutions of (23) yields

$$\Delta V = -\mathcal{P} + \mathcal{A}_{cl}^t \mathcal{P} \mathcal{A}_{cl} \tag{25}$$

According to Laypunov stability theorem, necessary and sufficient condition for stability is $V > 0$, $\Delta V < 0$. The following is a preliminary result

Lemma 12.1: Given the gains K and L, system (19) is said to be asymptotically stable if there exists positive definite matrices $0 < \mathcal{P}_j = \mathcal{P}_j^t \in \Re^{n_i \times n_i}$, $0 < \mathcal{X}_j \in \Re^{n_i \times n_i}$, $0 < \mathcal{Z}_j = \mathcal{Z}_j^t \in \Re^{n_i \times n_i}$, $i = 1, \ldots, NumOfDirections$ such that the following LMIs

$$\begin{bmatrix} -\mathcal{P}_j & \mathcal{A}_{j,cl}^t \mathcal{X}_j \\ \bullet & -\mathcal{X}_j - \mathcal{X}_j^t + \mathcal{Z}_j \end{bmatrix} < 0, j = 1, ..., N \tag{26}$$

have a feasible solution for $j = 1, \ldots, N$, where N is the number of intersections.

Remark 12.1: By looking at the closed-loop matrix (19) to (20) in the distributed-control case, it is instructive to let the matrix X has the following form where the size will be matching with the size of A_{cl} according to the control structure that we have selected:

$$\mathcal{X} = \begin{bmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} \\ 0 & X_{22} & X_{23} & X_{24} & X_{25} \\ 0 & 0 & X_{33} & X_{34} & X_{35} \\ 0 & 0 & 0 & X_{44} & X_{45} \\ 0 & 0 & 0 & 0 & X_{55} \end{bmatrix} \tag{27}$$

Indeed, the decentralised and quasi-decentralised cases can be done in a similar way and X will be smaller in size similar to A_{cl} matrix size for each control structure [see equation (22)].

Proof: That $\mathcal{P}_j > 0$ implies that $V > 0$. Applying Lemma 12.1 to inequality $\Delta V < 0$ using (25) with $\mathcal{M} = \mathcal{P}_j$, $\mathcal{N} = \mathcal{A}_{j,cl}^t$ and invoking Schur complements, we readily obtain inequality (26).

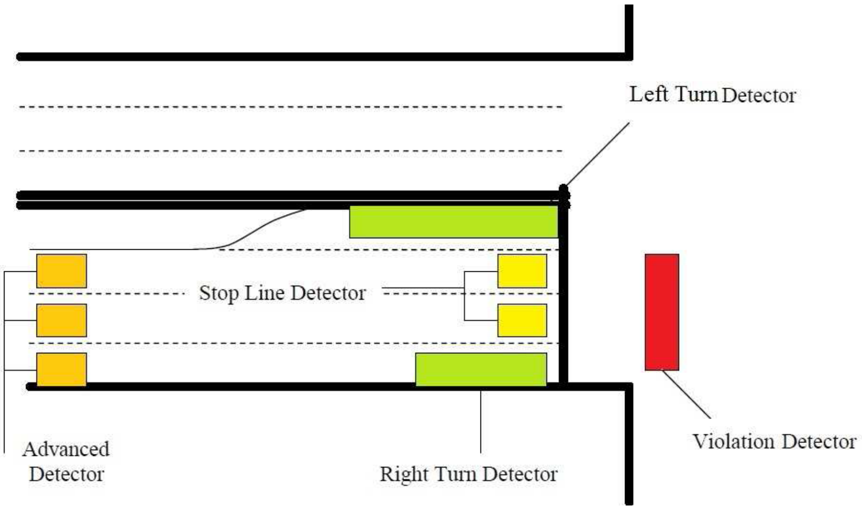
13 Simulation studies

In the simulation, we considered five intersections and we tried to compare the results from the proposed approaches. The simulation was done using MATLAB 2008 on

Laptop with Windows 7 Professional, 2.73 GHz with 8 cores and 8 GB Memory. The following assumptions were used:

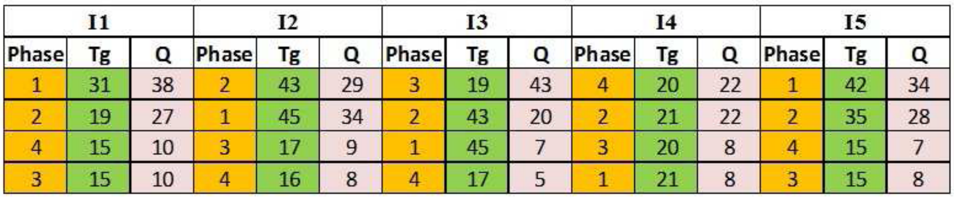
- distance between each intersection is known (1 km)
- average speed is 70 km/h.
- flow of traffic is smooth and no major interruption
- communication between sensors to controller is over a lossy network
- cross-communication between each intersection controller is over lossy network
- detectors (sensors) in each lane at the upstream and downstream direction for counting and event triggering
- left and right lanes have sensors to count the vehicles going in these directions
- estimated time to travel from one intersection to another with 70 Km/h is around 45 sec
- each intersection operates in four phase's mode as default, which means that every two parallel directions will run at the same time to maximise the flow
- the average of arrivals for each parallel direction will be taken as input, and the considered phases (N, S, E, W, NW, SE, ES, WN)
- simulation runs for 30 minutes
- in the simulation we considered five intersections as shown in Figure 10 and we tried to compare the results from the proposed approaches.

Figure 18 Vehicle detectors



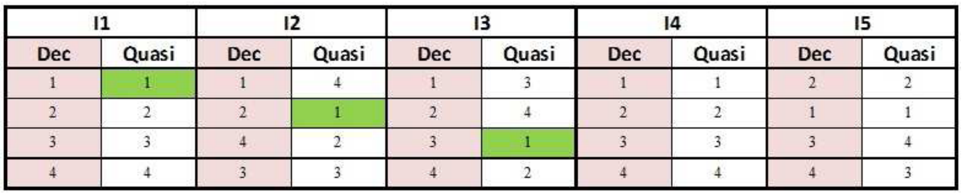
From the estimator side, the simplest approach to model vehicle arrivals is to assume a uniform arrival. This will results in a deterministic, uniform arrival pattern which means constant time headway between all vehicles. However, this assumption is usually unrealistic, as vehicle arrivals typically follow a random process. Thus, a model that represents a random arrival process is needed and the most suitable one is the Poisson distribution with arrival rate of λ. In general, the car arrival is part of the queuing model (e.g., $M/M/1$ or $M/G/1$) which simulates the traffic signal operations. Basically, the queue model is any service station with one or multiple servers, waiting area or buffer.

Figure 19 A sample of car arrivals rate/min (Q)



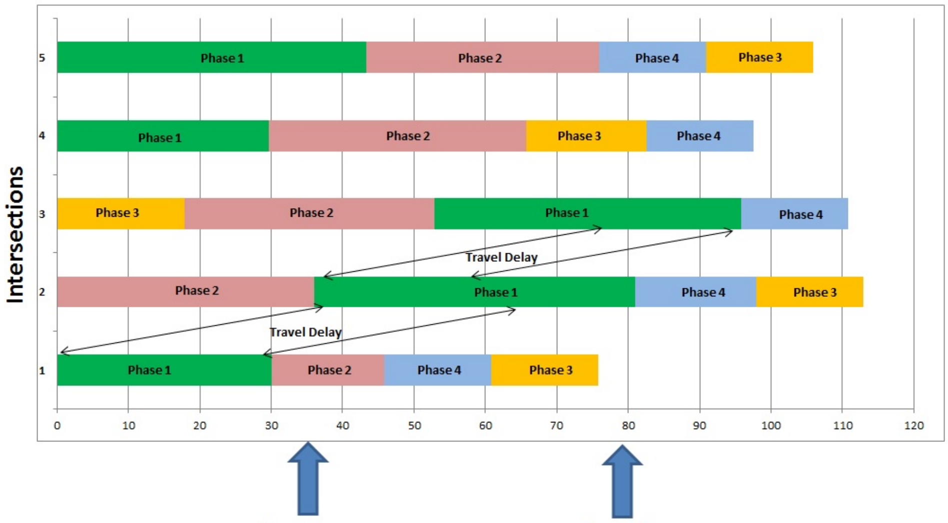
I1			I2			I3			I4			I5		
Phase	Tg	Q	Phase	Tg	Q	Phase	Tg	Q	Phase	Tg	Q	Phase	Tg	Q
1	31	38	2	43	29	3	19	43	4	20	22	1	42	34
2	19	27	1	45	34	2	43	20	2	21	22	2	35	28
4	15	10	3	17	9	1	45	7	3	20	8	4	15	7
3	15	10	4	16	8	4	17	5	1	21	8	3	15	8

Figure 20 Decentralised and quasi phase selection



I1		I2		I3		I4		I5	
Dec	Quasi	Dec	Quasi	Dec	Quasi	Dec	Quasi	Dec	Quasi
1	1	1	4	1	3	1	1	2	2
2	2	2	1	2	4	2	2	1	1
3	3	4	2	3	1	3	3	3	4
4	4	3	3	4	2	4	4	4	3

Figure 21 Phase selection in QuasiNCS for each intersection to maximise the flow from intersection 1 up to 3



Flow coming from **I1** will be able to catch the green

Flow coming from **I2** will be able to catch the green

Figure 22 Phase selection in DecNCS for each intersection

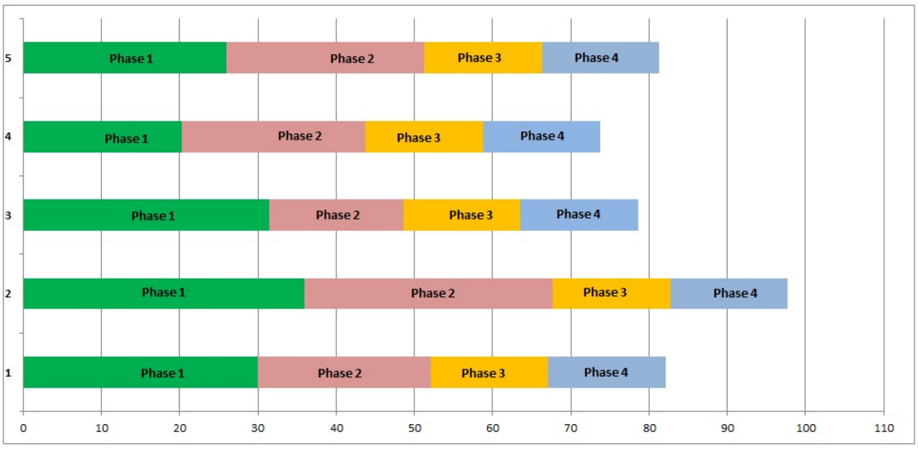
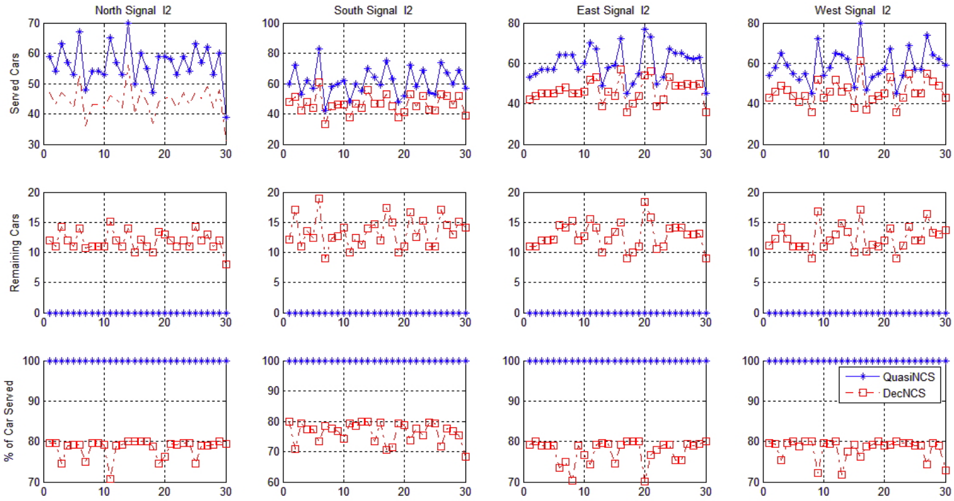


Figure 23 QuasiNCS vs. DecNCS, intersection 2, main directions



The level of information exchanged is shown in Table 1. When we look to Figure 20, we can see that at intersection 1, we started with phase 1, then by the time the flow will reach to intersection 2, which is around 45 seconds, the incoming flow plus the existing flow will move together without stoppage and same will happen at intersection 3, this explanation is shown clearly in Figure 22. That shows the beauty of quasi decentralised approach over the decentralised itself as shown in Figure 21, where in the quasi we have benefited from the limited communication over a network to smooth and maximise the flow in certain direction between intersections. In DecNCS approach, the controller will control each intersection independently from others and the only information sent over lossy link is the arrival traffic via the sensors placed at the beginning of the roads towards that intersection. In the case of QuasiNCS, the information about the phase selection in each intersection is exchanged among the adjacent controllers, the one before and the one after, to allow continuous progression of platoons through successive signals along multiple intersections without stopping because the on/off nature of traffic signals tends to accumulate the vehicles in longer queue. The total trip time in case of DecNCS will be more 150 sec to cross the distance starting from intersections 1 to 3 with 2 stoppages while in QuasiNCS it around 94 sec with zero stoppage. Also, we can observe from Figure 21 that a synchronisation can happen between intersections 2, 4 and 5 where the majority of the traffic between East and West can run smoothly in the successive intersections. In case of the packets delay or dropout or another communication constraints, the controller can depend on the last received data and in case of long stoppage of sensors or physical damage, the controller can depend on either fixed green time (45 sec) or based on average arrival rate from historical data. For the QuasiNCS, the phase selection information will be affected in case of any delay or dropout, it will simply run based on arrival data coming from the sensors. The third approach which is the DNCS that requires more data to be exchanged to perform in the required manner but with more computation time. The simulation of hierarchical approach is not presented in this work.

Table 1 Data exchange in each approach

	Dec	*Quasi*	*Dist*	*Hir*
Traffic arrival	Y	Y	Y	Y
Phase selection($I_i to I_j$)	N	Y	Y	Y
Traffic arrival($I_i to I_j$)	N	N	Y	Y
GreenTime($in I_j$)	N	N	Y	Y
Traffic jam Info	N	N	N	Y
Avg. arrivals speed	N	N	Y	Y

Figure 24 QuasiNCS vs. DecNCS, intersection 2, subdirections

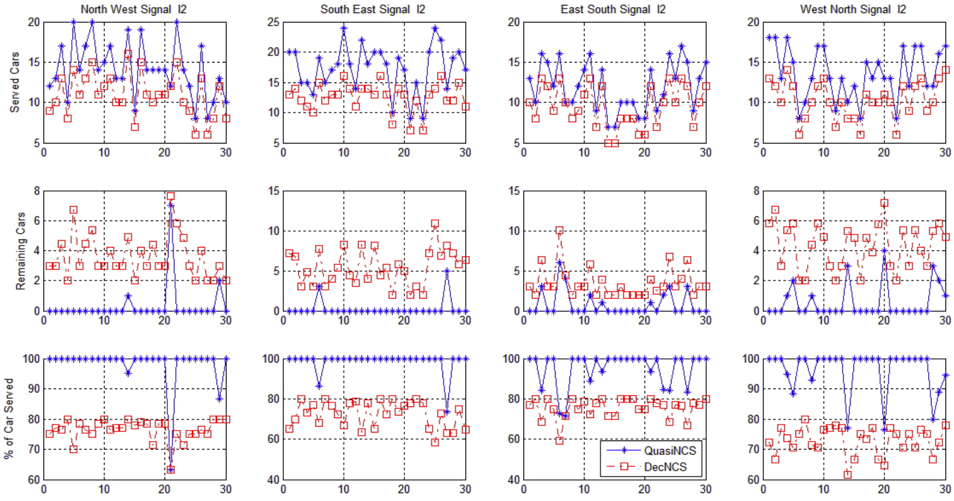


Figure 25 QuasiNCS served vs. arrived cars, intersection 2

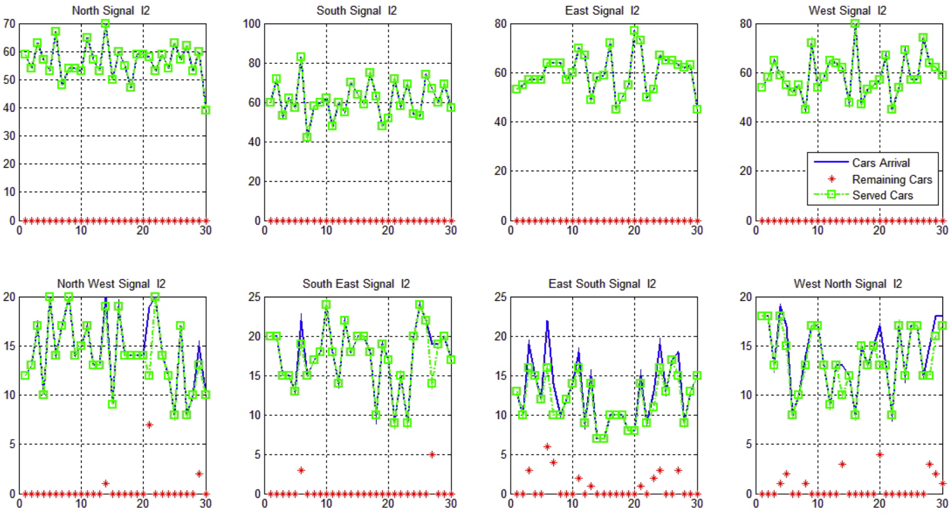
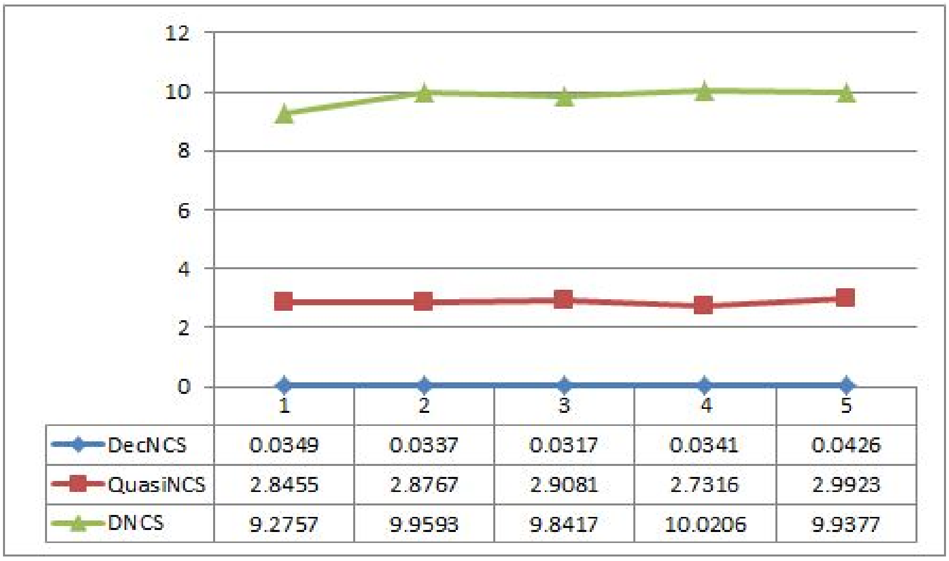


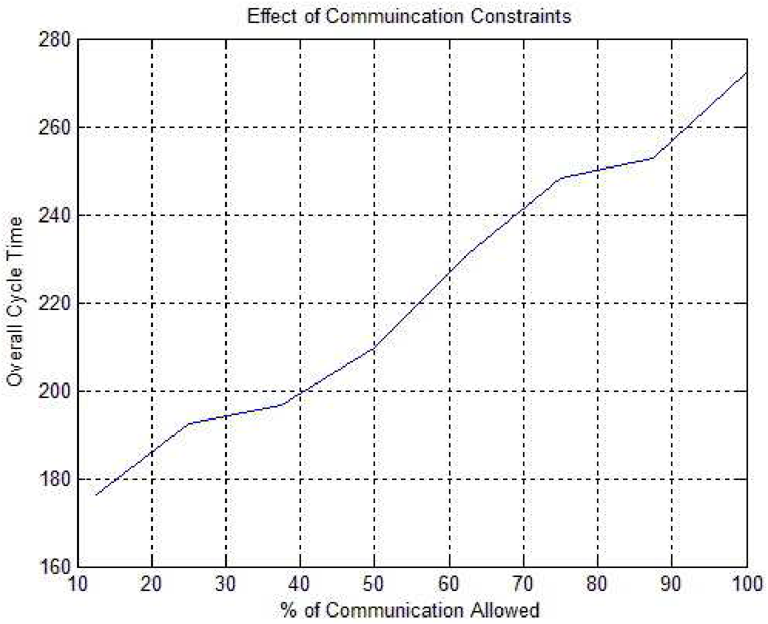
Figure 26 CPU computation time



	1	2	3	4	5
DecNCS	0.0349	0.0337	0.0317	0.0341	0.0426
QuasiNCS	2.8455	2.8767	2.9081	2.7316	2.9923
DNCS	9.2757	9.9593	9.8417	10.0206	9.9377

The CPU computation time, for the main control loop, for each approach is shown in Figure 26 and it is clearly the QuasiNCS approach is providing good solution as was presented in Figure 21 where a good platoon movement between *I*1 and *I*3 while in the same time requires low computation time comparing to DNCS and not far from the DecNCS. Figures 23 to 24 show the difference between QuasiNCS and DecNCS for intersection 2, while Figure 25 shows some more details about QuasiNCS for intersection 2 also. Other intersections results also shows that QuasiNCS perform much better than DecNCS but due to page limitation we cannot show it all.

Figure 27 Communication constraints effect (see online version for colours)



Finally, we can show in the following points the communication effects in brief:

- Communication constraints: the communication is an important factor for the controller to make the proper coordinations with other controllers. In the case of DecNCS, there is no communication between controllers and decision will be made on the intersection data only. From the first look, you may see that the DecNCS giving low cycle time but in reality it is much more because it did not consider the new arrivals. For the QuasiNCS, we have simulate the effect of communication constraints as shown in Figure 27, you can see the more communication we allowed, the cycle time is getting change and this is required for the proper coordination between intersection considering the current and new coming traffic for each intersection.
- Packets drop: another issue we can show also is the effect of β values which will affect more the QuasiNCS as shown in Figure 28, and the drop of sensors packets will reflect on the cycle time but not too much because usually such sensors applications will send few packets (number of cars, time, …, etc.) cyclically, and if the drop is increasing the controller will switch to the local intersection control.
- Computation time: the traffic density is not really an issue for the computation time as we can see from Figure 29.
- Waiting time: this is very important measure for the control system, because the longer waiting time the more the drivers will get frustrated and the potential of violation will be higher. So, in this simulation, we focused on the waiting time behaviour during an incremental traffic by increasing the traffic arrival every cycle by 25% and we stop increasing it when the traffic density exceeds 1 as shown in Figure 30 and the Figure 31 shows the waiting time during normal random arrival.

- Communication delay: the effect of data packets delay from previous intersection controller to the next intersection controller will let the 2nd controller to increase the intersection cycle time to accommodate the incoming traffic up to certain limit then it will not extend. If the packets delay exceeded the maximum allowed limit, then the controller will ignore the delayed packets and start a new control cycle and if this problem continue for certain number of cycles, which mean that the link need longer time to be fixed, the controller will use one of the options we mentioned earlier (historical data, fixed time or behave like DecNCS locally), Figure 32 explains this issue clearly. Also, we can see from same figure in intersection 2 after certain time it will stop doing green time extension because the delay exceeded the limit.

Figure 28 Beta values effect

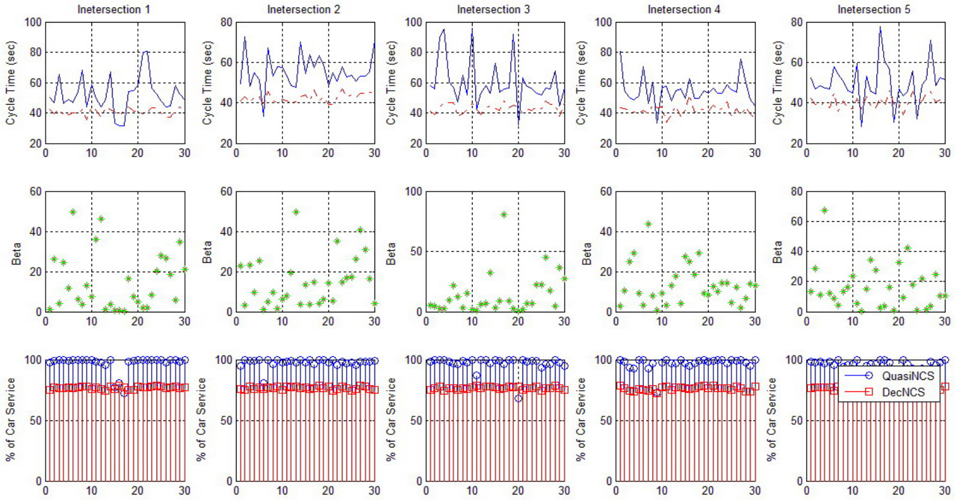


Figure 29 Traffic density effect on control computation time

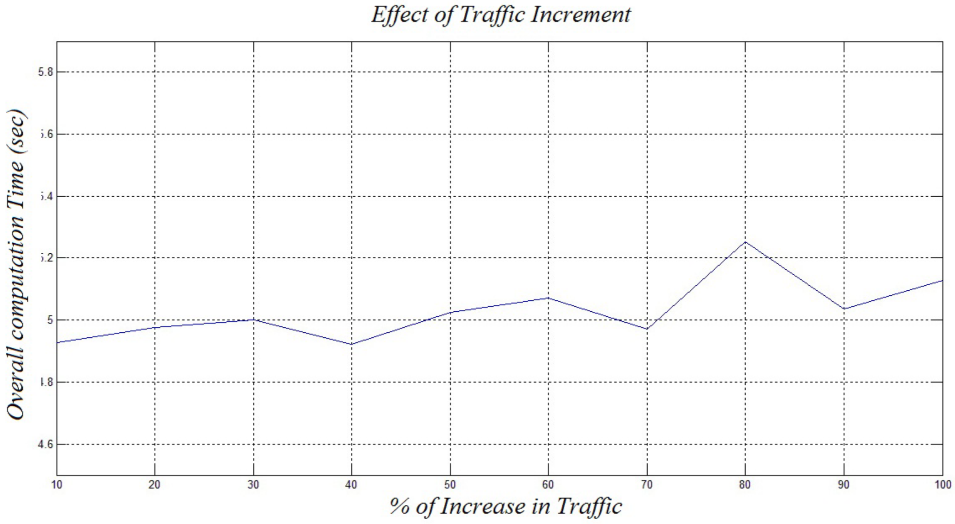


Figure 30 Incremental traffic density effect on waiting time

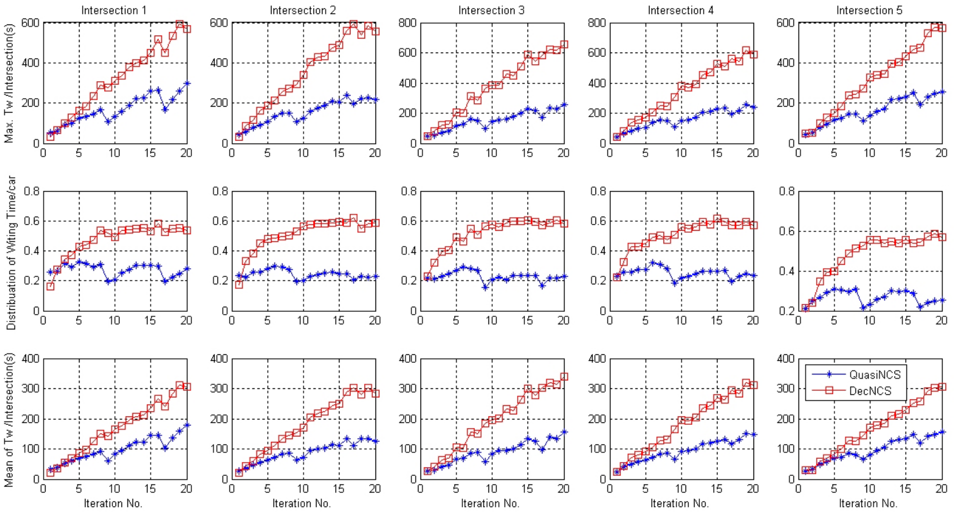


Figure 31 Random traffic density effect on waiting time

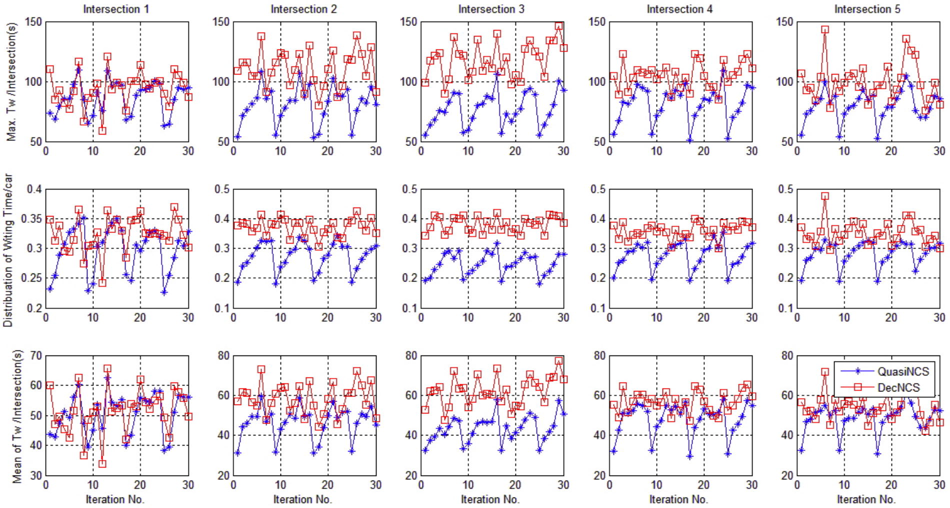
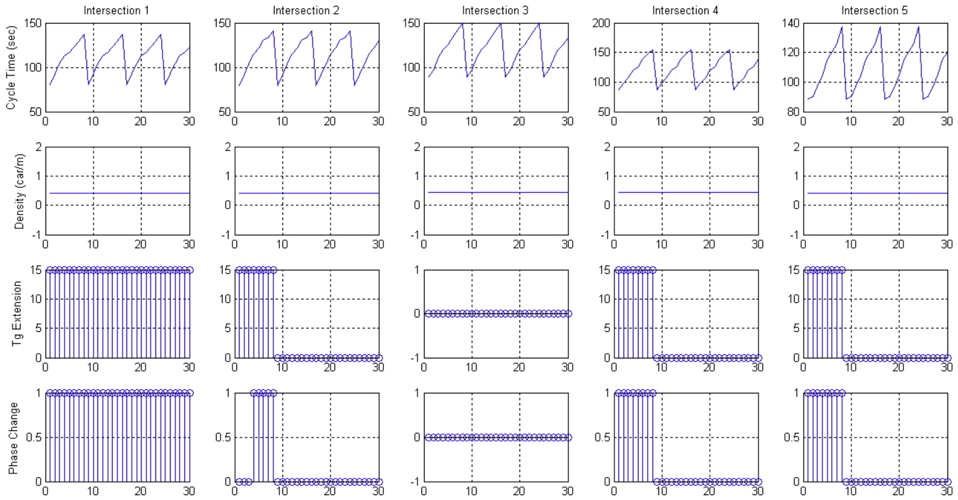


Figure 32 Delay effects on cycle time



14 Conclusions

In this paper, we have carefully examined the decentralised, networked control architectures for interconnected dynamical systems. Also, we have studied different control techniques like quasi-decentralised over network for traffic light intersections and we highlighted the major points about the distributed and hierarchal architectures over communication network. A comprehensive survey has been made about the traffic control methods and techniques including several traffic concepts and fundamentals. The work has discussed the underlying rationale for the individual architectures and illustrated the fields of application and the merits/demerits as reported in the literature. A state-space model system was developed and considered several parameters in the model such as the networked induced delays, packet dropout and varying sample interval. Simulation was performed selected real-life application which is the multi signalised intersections control and coordination. Different traffic signal management approaches were proposed with different control stratifies. The complexity of each approach considering different parameters and network effects also were discussed.

Acknowledgements

This work is supported by the Deanship of Scientific Research (DSR) at KFUPM through research group project no. RG-1105-1.

References

Abdulhai, B. (2003) 'Reinforcement learning for true adaptive traffic signal control', *ASCE Journal of Transportation Engineering*, Vol. 129, No. 3, pp.278–285.

Aboudola, K., Papageorgiou, M. and Kosmatopoulos, E. (2009) 'Store-and-forward based methods for the signal control problem in large-scale congested urban road networks', *Transportation Research Part C-Emerging Technologies*, Vol. 17, No. 1, pp.163–174.

Albagul, A., Hrairi, M., Wahyudi, R.A. and Hidayathullah, M.F. (2006) 'Design and development of sensor based traffic light system', *American Journal of Applied Sciences*, Vol. 56, No. 9, pp.1745–1749.

Al-Nasser, F.A. and Rowaihy, H. (2011) 'Simulation of dynamic traffic control system based on wireless sensor network', *IEEE Symposium on Computers & Informatics (ISCI 2011)*, pp.302–307.

Are, I., Liu, C., Urbanik, T. and Kohls, A.G. (2010) 'Reinforcement learning-based multi-agent system for network traffic signal control', *IET Intelligent Transport Systems*, Vol. 4, No. 2, pp.128–135.

Chien, L.K. (2005) 'A real-time dynamic traffic. Control system based on wireless sensor network', *International Conference on Parallel Processing Workshops – ICPPW.*

Freeman, W., Ho, K. and McChesney, E. (1999) 'An evaluation of intersection system analysis techniques', Presented at the *78th Annual Meeting of the Transportation Research Board*, Washington, DC.

Ganiyu, A., Olabiyisi, O., Omidiora, O., Okediran, O. and Alo, O. (2011) 'Modelling and validation of a multi-phase traffic light controlled cross-type intersection using timed coloured Petri nets', *American Journal of Scientific and Industrial Research*, Vol. 2, No. 5, pp.807–819.

Gartner, N., Poorhan, F. and Andrews, C. (2002) 'Implementations and field testing of the OPAC adaptive control strategy in RT-TRACS', Presented at the *81st Annual Meeting of the Transportation Research Board*, Washington, DC.

Hirankitti, V., Krohkaew, J. and Hogger, C. (2007) 'A multi-agent approach for intelligent traffic-light control', *Proc. of the World Congress on Engineering – WCE*, July, Vol. 1.

Kubalik, J., Tichy, P., Sindelar, R. and Staron, R.J. (2010) 'Clustering methods for agent distribution optimization', *IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Reviews*, January, Vol. 40, No. 1, pp.78–86.

Kuei-Hsiang, C., Ren-Hao, L. and Meng-Hui, W. (2009) 'An intelligent traffic light control based on extension neural network', in *Knowledge-Based Intelligent Information and Engineering Systems*.

Leena, S., Sudhanshu, T. and Himakshi, A. (2009) 'Time optimization for traffic signal control using genetic algorithm', *International Journal of Recent Trends in Engineering*, November, Vol. 2, No. 2, pp.234–245.

Lefei, L. (2005) 'Traffic signal priority/preemption control with colored Petri nets', *Proc. IEEE Intelligent Transportation Systems*, pp.694–699.

Li, L. and Wang, F. (2008) 'Control and communication synthesis in networked control systems', *Int. J. Intelligent Control and Systems*, Vol. 13, No. 1, pp.81–88.

Liu, C., Tao, H. and Tabacek, M. (2006) 'Empirical observations of dynamic dilemma zones at signalized intersections', *Proc. the 86th Meeting of the Transportation Research Board*, July.

Martin, F. and Radiocomun, S. (2012) 'Vehicular traffic surveillance and road lane detection using radar interferometry', *IEEE Trans. Vehicular Technology*, March, Vol. 61, No. 6, pp.959–970.

Martins, E.C. and Jota, F.G. (2010) 'Design of networked control systems with explicit compensation for time-delay variations', *IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Reviews*, May, Vol. 40, No. 3, pp.308–318.

Mehta, S. (2008) 'Fuzzy control system for controlling traffic lights', *Proc. the International Multi Conference of Engineers and Computer Scientists (IMECS)*, 19–21 March, Vol. 1.

Michal, K., Zden, S. and Anton, C. (2006) 'Balancing the waiting times in a simple traffic intersection model', *The 11th IFAC Symposium on Control in Transportation Systems*.

Nan, Z., Pei-Wei, L. and Gang-Len, C. and Saed, R. (2003) 'An integrated knowledge-base/simulation system for real time incident management', Department of Civil and Environmental Engineering, University of Maryland at College Park.

Narayanan, R., Udayakumar, R., Kumar, K. and Subbaraj, L. (2003) 'Quantification of congestion using fuzzy logic and network analysis using GIS', *Proc. of Map India Conference*, New Delhi, India, 28–31 January.

Nesic, D. and Liberzon, D. (2007) 'A unified approach to controller design for systems with quantization and time scheduling', *Proc. 46th IEEE Conference on Decision and Control*, pp.3939–3944.

Oh, S. and Sastry, S. (2006) 'Distributed networked control system with lossy links: state estimation and stabilizing communication control', *45th IEEE International Conference on Decision and Control*, pp.1942–1947.

Oh, S. and Sastry, S. (2007) 'Approximate estimation of distributed networked control systems', *Proc. American Control Conference*, pp.997–1002.

Pitu, M. and Fei-Yue, W. (2005) 'RHODES to intelligent transportation systems', *IEEE Intelligent Systems Journal*, January, Vol. 20, No. 1, pp.54–62.

Robertson, D. and Bretherton, R. (1991) 'Optimizing networks of traffic signals in real time – the SCOOT method', *IEEE Transactions on Vehicular Technology*, February, Vol. 40, No. 1, pp.11–15.

Rohloff, K., Ye, J., Loyall, J. and Schantz, R. (2006) 'A hierarchical control system for dynamic resource management', *IEEE Real-Time and Embedded Technology and Applications Symposium*.

Schendel, J., Donkers, M., Heemels, W. and Wouw, N. (2010) 'On dropout modeling for stability analysis of networked control systems', *Proc. American Control Conference*, pp.555–561.

Serrano, A., Cristina, C., Licesio, R., Raquel, M. and Enrique, C. (2005) 'Computer vision application: real time smart traffic light', in *Computer Aided Systems Theory EUROCAST*.

Silva, C., Oliveira, D., Bazzan, C. and Basso, W. (2006) 'Adaptive traffic control with reinforcement learning', *Proc. the 4th Workshop on Agents in Traffic and Transportation (AAMAS)*, May, pp.80–86.

Singh, J., Patra, B. and Singh, S. (2011) 'An algorithm to reduce the time complexity of earliest deadline first scheduling algorithm in real-time system', *Int. J. of Advanced Computer Science and Applications*, Vol. 2, No. 1, pp.31–37.

Strasser, T. and Froschauer, R. (2012) 'Autonomous application recovery in distributed intelligent automation and control systems', *IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Reviews*, November, Vol. 42, No. 6, pp.1054–1070.

Sun, Y. and El-Farra, N.H. (2009a) 'Quasi-decentralized control of process systems using wireless sensor networks with scheduled sensor transmissions', *American Control Conference*, pp.3390–3396.

Sun, Y. and El-Farra, N.H. (2009b) 'Quasi-decentralized scheduled output feedback control of process systems using wireless sensor networks', *IFAC Symposium on Advanced Control of Chemical Processes*.

Tharmarasa, R., Kirubarajan, T., Peng, J. and Lang, T. (2009) 'Optimization-based dynamic sensor management for distributed multitarget tracking', *IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Reviews*, September, Vol. 39, No. 5, pp.534–546.

Tubaishat, M., Shang, Y. and Shi, H. (2007) 'Adaptive traffic light control with wireless sensor networks', *4th Consumer Communications and Networking Conference – CCNC*.

Wei, J. (2008) 'Stability analysis of decentralized networked control systems', *World Congress on Intelligent Control & Automation*, Vol. 7, No. 4, pp.5477–5482.

Wen, Y., Pan, J.L. and Le, J.F. (2007) 'Survey on application of wireless sensor networks for traffic monitoring', *International Conference on Transportation Engineering – ICTE*.

Zhang, H., Yang, D. and Chai, T. (2007) 'Guaranteed cost networked control for TS fuzzy systems with time delays', *IEEE Trans. Systems, Man, and Cybernetics: Part C*, Vol. 34, No. 2, pp.160–172.

Zhao, Y., Zhang, C. and Gao, H.J. (2009) 'A new approach to guaranteed cost control of T-S fuzzy dynamic systems with interval parameter uncertainties', *IEEE Trans. Systems, Man, and Cybernetics: Part B*, Vol. 39, No. 6, pp.1516–1527.

Zhao-Sheng, Y., Xin, C., Yang-Shan, T. and Jian, P. (2005) 'Intelligent cooperation control of urban traffic networks', *Proc. the 4th International Conference on Machine Learning and Cybernetics*, August, pp.18–21.