Jurnal Teknologi, 42(B) Jun. 2005: 29–44

APPLICATION OF LVQ NEURAL NETWORK IN REAL-TIME ADAPTIVE TRAFFIC SIGNAL CONTROL

AGUS PRIYONO[1], MUHAMMAD RIDWAN[2], AHMAD JAIS ALIAS[3], RIZA ATIQ O. K. RAHMAT[4], AZMI HASSAN[5], & MOHD. ALAUDDIN MOHD. ALI[6]

Abstract. Real-time road traffic data analysis is the cornerstone for the modern transport system. The real-time adaptive traffic signal control system is an essential part for the system. This analysis is to describe a traffic scene in a way similar to that of a human reporting the traffic status and the extraction of traffic parameters such as vehicle queue length, traffic volume, lane occupancy and speed measurement. This paper proposed the application of two-stage neural network in real-time adaptive traffic signal control system capable of analysing the traffic scene detected by video camera, processing the data, determining the traffic parameters and using the parameters to decide the control strategies. The two-stage neural network is used to process the traffic scene and decide the traffic control methods: optimum priority or optimum locality. Based on simulation in the traffic laboratory and field testing, the proposed control system is able to recognise the traffic pattern and enhance the traffic parameters, thus easing traffic congestion more effectively than existing control systems.

Keywords: Urban traffic control system, pattern recognition, two-stage neural network, adaptive control system

1.0 INTRODUCTION

The classic problems in urban traffic control system are how to solve the traffic congestion and manage an urban traffic which meets traffic demand. Traditionally, the congestion problem on surface streets was dealt with on the supply side by providing increased capacity, adding more lanes to existing roads or adding new links to the existing transportation network. Such a solution is no longer considered viable because of the prohibitive construction cost and the negative environmental impact. Instead, greater emphasis is placed on traffic management. The management of traffic on surface street networks is achieved primarily via signalised control of intersections [1].

Over the past few years, a number of Intelligent Transportation Systems (ITS) projects have been implemented at the regional and state level, aimed at providing

[1,3&6]Department of Electrical, Electronic and System Engineering,
[2&5] Department of Mechanical and Material Engineering,
[4] Department of Civil and Structural Engineering,
Faculty of Engineering, Universiti Kebangsaan Malaysia, Bangi 43600, Selangor Darul Ehsan, Malaysia.

a systematic approach in making the most effective use of the existing and future transportation facilities, so that any unused capacity can be fully utilised. One of the components of ITS called Advanced Traffic Management Systems (ATMS) focuses on developing more efficient real-time and adaptive traffic control systems [2].

The tasks in ATMS involve signalised intersection control, freeway traffic control, and area-wide integrated traffic management. In developing control strategies for signalised intersections; numbers of vehicle queue length and traffic volumes are essential inputs for both off-line and on-line timing plan optimisation and evaluation.

The first generation of Urban Traffic Control Systems (UTCS-1), determined timing plans off-line based on historical traffic information, and then selected the timing plans using pattern matching techniques. The UTCS-1 system is controlled by centralised system approach. The central computer determines the timing phase, cycle time and control the traffic light. The more sophisticated second generation of UTCS (UTCS-2) was capable of generating and updating timing plans every 5±15 min, based on the predicted demands. The demand predictor in UTCS-2 used current traffic measurements but still relied on historical data as references. Some parts of the system are centrally controlled while other parts are controlled by distributed approach. The third generation of UTCS (UTCS-3) intended for full-responsive, on-line traffic control operation, used current measurements alone to make predictions on cycle-by-cycle basis [2]. Many of the UTCS-3 system are controlled by real-time distributed system approach. In the distributed approach, local computer determines the timing phase, offset and cycle time based on the traffic demands and able to synchronise with another neighbouring local computers in the area.

2.0 URBAN TRAFFIC CONTROL SYSTEM

The traffic control system which is able to synchronise with neighbour controller to manage traffic congestion is the essential part in urban traffic control system [3]. Collecting the traffic data information and controlling the strategies in real-time system becomes the main objective in developing real-time adaptive signal control system.

To provide advanced information, such as queue length and speed, multiple detectors must be used. As more advanced traffic control systems require increasing amounts of information on all approaches to every intersection in a network, the number of loops necessary to meet detection requirements rapidly increases. Video systems are among the most advanced detection in use today in which the video cameras are used as detectors installed in traffic intersection to capture the image of traffic condition. The image carries traffic information such as vehicle queue length, traffic volume, lane occupancy, speed measurement etc.

Real-time adaptive signal control is well recognised as having the potential to increase the operational efficiency and safety of existing roadways [4]. The real-time adaptive control system is developed to collect the real traffic data, interpret the data, determine the timing plans and control the traffic light in real-time and on-line approach. The physical architecture of the real-time adaptive controller contains three components as shown in Figure 1. The first component comprised of four video cameras installed in each intersection. The second consists of local computer (personal computer) as a main controller to process the traffic data, decide the control strategies and determine the split green time, offset and cycle time. The computer is equipped with surveillance video card and communication Ethernet card. The third one is a programmable logic controller (PLC) connected through the traffic light.

In this paper, the scope of discussion is confined to the aspects of traffic data processing and the selection of the control strategy using two-stage neural network using LVQ neural network.

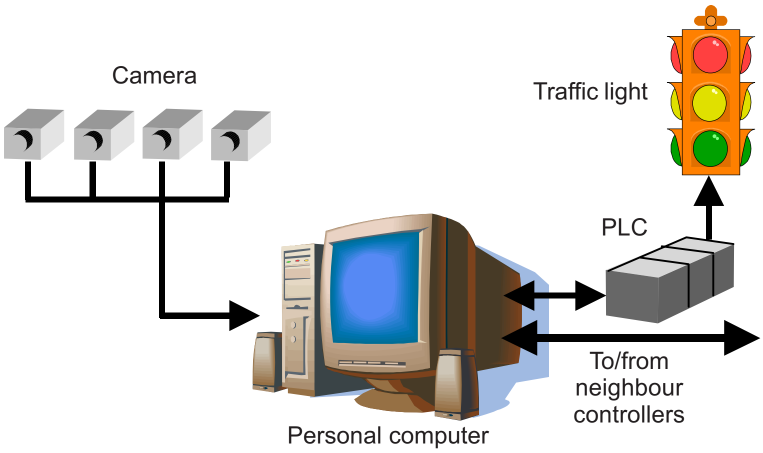
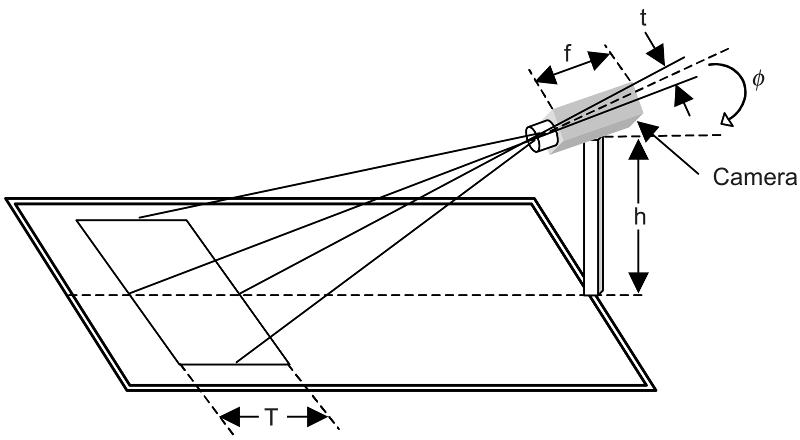


Figure 1 Real-time adaptive traffic control system components

3.0 TRAFFIC PATTERN RECOGNITION BY TWO-STAGE NEURAL NETWORK

The current approach of image processing applied to traffic can be divided into qualitative and quantitative analysis. The qualitative analysis is to describe a traffic scene in a way similar to that of a human reporting the traffic status. In this approach, four smart cameras are used and installed at each intersection using CCTV camera. The smart camera does not only provide means to capture the traffic image, but it also has the capability to analyze the traffic image, determine the number of vehicle passing through the intersection, record speed measurement, classify the vehicle and detect the traffic incidents.

Traffic sensors are very important instrument used in traffic control. These sensors usually include magnetic or inductive loops to evaluate traffic parameters. The most significant disadvantage of these sensors lies in the fact that they can only survey a limited region of the traffic path. To analyse a wider view of the path and evaluate a whole description, video camera is needed. Two major parameters of traffic status are the percentage of road occupied by vehicles and percentage of moving and stationary parts of this occupancy. The algorithm developed for this purpose can automatically divide the scene into a number of blocks based on camera parameters. The camera geometry and real-world block are shown in Figure 2.



Where: h = high camera from earth.
T = length of block.
f = focus
ϕ = degree of camera slope
t = image length in computer in pixel unit

Figure 2 The camera geometry and real-world block

The video images from cameras must be digitised before being processed in personal computer by surveillance system card. The images are digitised frame by frame and created by averaging 25 frames per second in Device Independent Bitmap (DIB) format as shown in Figure 3.

In order to capture a whole description of the traffic scene, the traffic path is divided into several parts where each of them is a rectangular box that includes one and two vehicles each lane. This approach is known as cell-based technique. The status of each box can be related to one of the following conditions: 1) stationary vehicle(s), 2) moving vehicle(s) and 3) no vehicles.

For an accurate decision on existence of vehicles, the effect of background must be removed. Some of the undesired effects of background are stationary, like marks

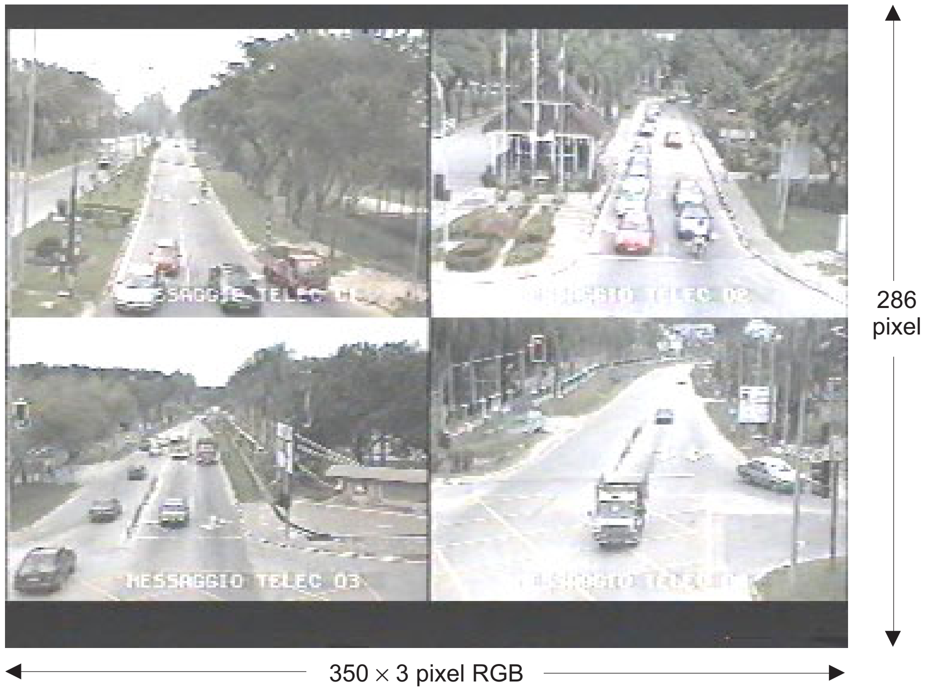


Figure 3 The images are digitizing frame by frame and created in Device Independent Bitmap (DIB) format

on the road, but some of them change their place in time, like shadows of trees, buildings and clouds. In order to have the actual background, it must be updated periodically [5]. The proposed control system is based on the selective-averaging background method to obtain and update the background picture. The flow chart of the proposed algorithm is shown in Figure 4.

In addition to the cell-based technique, the pixel-based technique is also implemented. The RGB color pixel is used separately as compared to manual calculation. Sample of the RGB pixel value at 2 frames per second in the intersection stop line is shown in Figure 4. During experiments, the images are captured at 10 frames per second. In this condition, a vehicle can be read twice [5]. If it captured lower than 10 frames per second, the speeding vehicle will not be detected. Therefore, a vehicle should be determined as one unit if it reads twice in below one second. In normal condition, the vehicle speed passes through the intersection at below 10 meter per second. So, the image captured at more than 2 frames per second is needed [5]. The pixels value can be determined using the following equation:

$$d_{(i,j)} = \begin{cases} 0\, if \left| f_1(i,j) - f_2(i,j) \right| \leq T \\ 0\, if \left| f_1(i,j) - f_2(i,j) \right| > T \end{cases} \tag{1}$$

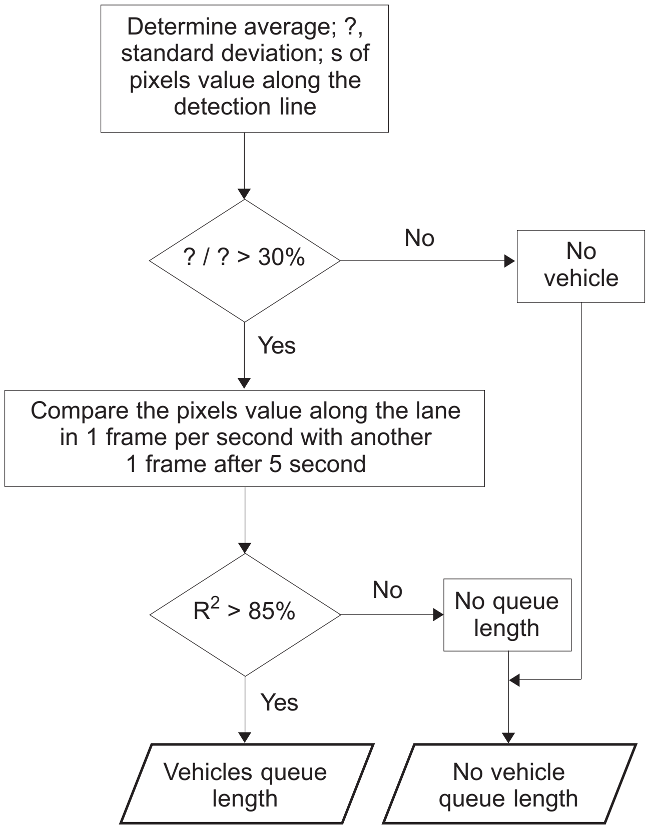


Figure 4 The flow chart of the proposed algorithm (source: Riza Atiq [5])

Where $d_{(i,j)}$ = difference of pixel value
$f_{1(i,j)}$ = pixel at coordinate (i,j) in f_1 frame.
$f_{2(i,j)}$ = pixel at coordinate (i,j) in f_2 frame.
T = random value (normally at 10 % of $f_{1(i,j)}$).

In order, to calculate the image pixels for 10 lines with 10 pixels per line, it is necessary to reduce the captured value to 6 frames per second. In Pentium 3 processor, this condition causes the process to become slow, hence methods using the neural network approach should be implemented to solve the problem.

The proposed sensor system called two-stage neural network is used to recognise the traffic pattern, calculates the traffic parameters and decides the control strategies. The first stage is a Multi Layer Perceptron (MLP) neural network which is used to process the traffic data from video cameras. The output of the first stage neural network is information about traffic volume and vehicle queue length. The second stage is Kohonen's Learning Vector Quantization (LVQ) neural network [6]. The

LVQ network used to decide the control strategies is based on the information from the first stage MLP network.

4.0 FIRST STAGE MLP NETWORK

The MLP network is used to process the pattern recognition problems in the proposed control system. The MLP network is selected because of its capability to recognise the pattern classification [7]. This network has been successfully used in the solution of difficult pattern recognition problems [6]. The MLP is composed of several layers namely, the input layer, the hidden layer and the output layer. The power of the MLP is the non-linear behaviour of its neurons [6].

The structure of the proposed network consists of 20 neurons in the input layer, 20 neurons in the hidden layer and 2 neurons in the output layer. A number of 20 neurons in the input layer is set up to put the pixels in a number of blocks scene that contains 10 pixels at the lane. 10 neurons used to catch pattern-value of pixels and 10 neurons used to catch the different-value of pixels at the same block. One of the neurons in the output layer is used to express the vehicle in the lane while the other one is to detect the vehicle movement. Figure 5 shows the neural network in the proposed system.

The MLP is trained in the traffic laboratory, using real image captured by video cameras in different traffic conditions: no vehicle, few vehicles, more vehicles, traffic congestion, in the morning, afternoon, night, bright, cloudy, and rainy situation. The image is fed forward to the personal computer from the video player. The results confirm that the network is able to provide traffic status in real traffic situation. Table 1 shows the comparison between the pattern recognition result performed by human with that of proposed control system.

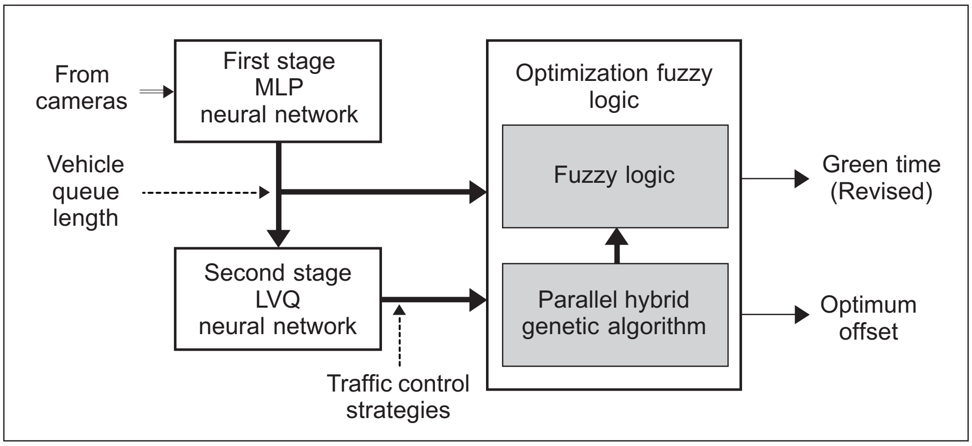


Figure 5 The two-stage neural network in the proposed traffic control system

Table 1 The comparison between the pattern recognition results by human and proposed control system (Riza Atiq [5])

Cycle time	Vehicle Q length	
	Human	Automatic
1	6	6
2	8	8
3	9	9
4	10	9
5	10	10
6	11	10
7	12	10
8	12	11
9	10	10
10	11	10
11	9	9
12	8	8
13	9	8
14	8	8

5.0 SECOND STAGE LVQ NETWORK

An association between SOM and LVQ algorithms, also called Kohonen's LVQ [6] has been applied to a variety of problems encountered in pattern recognition, diagnostics, control systems and monitoring tasks. The unsupervised SOMs are usually defined in metric vector spaces and the supervised-learning algorithms called the LVQ can be used to fine tune the SOM reference vectors for best class separation. The use of Kohonen's SOM and LVQ algorithms to recognize the traffic pattern have been proposed in many papers. Association between SOM and LVQ algorithms have been applied to a variety of problems encountered in pattern recognition, diagnostics, control systems and monitoring tasks [6]. LVQ is a neural network technique that exploits the underlying structure of input vectors for the purpose of data compression [9].

SOM, as well as supervised learning by LVQ can be defined for string variables. Their computation becomes possible when the SOM and LVQ algorithms are expressed as batch versions, and when the average over a list of symbol strings is defined to be the string that has the smallest sum of generalised distance functions from all the other strings. As the unsupervised training of SOM proceeds, neurons in the map will be ordered with respect to the topology of input vectors. If the map is to be used for classification, neurons should be labeled by broadcasting training vectors with known classification and associating to each winner a corresponding label. In the basic LVQ, only the winner node but not its topological neighbours is updated.

Due to these features, SOMs are well suited for clustering, visualisation and abstraction tasks [6]. They can be trained with supervised learning, which means that classification information is included in the input data. The LVQ method can be thought of as a fine-tuning method for the original SOM. Fine-tuning can then be performed with the LVQ algorithm, which separates class borders more clearly.

The data processed in LVQ is used to decide the traffic strategies based on traffic condition. The simple rules are developed as follows:

if traffic condition is normal
    then the control strategy is optimum locality

if traffic condition is congested
    then the control strategy is optimum priority

The proposed LVQ network contains some neurons as many as input data in the input layer and 2 neurons in the output layer. After the learning process, LVQ network will classify an input vector then put the vector in the same class with the output vector, which has the weighted vector nearest to input vector.

The steps of the LVQ learning process are as follows:

1. Set the data weight (w_{nd}).
2. Calculate the weighted value J_{nd} using the following equation:

$$J_{nd} = \sqrt{\left(\sum_{d=1}^{D}\left((D_{nd} - w_{nd}) * 2\right)^2\right)} \quad (1)$$

3. Select the minimum weighted value (J_{nd}*).
4. Update weighted w_{nd} using the following equation:

$$w_{nd+1}{}^* = w_{nd}{}^* + \alpha * (D_n - w_{nd}{}^*) \quad (2)$$

$$\alpha = \alpha - (\theta * \alpha) \quad (3)$$

Where:

θ = momentum.
D = input data.
n = data index.
d = type of data index.
α = Learning weighted.

5. Repeat until maximum epoch.

After the learning process, LVQ is used to decide the control strategy using the following steps:

1. If phase 1 queue length (Q_1), Q_2, Q_3 and Q_4 are all normal, the decided control strategy is optimum locality. At this strategy, phase 1 green time (g_1) is calculated based on Q_1 and Q_2 during phase 4 amber time (y_4) on. And g_2 is calculated based on Q_2 and Q_3, during y_1 is on. In the same way, g_3 and g_4 are calculated until 1 cycle time (C_T) is completed with traffic light sequence as shown in Figure 6.
2. If at least 2 approaches are congested in one intersection, for example, intersection 1 (j_1), the decided control strategy is optimum priority. At this strategy, the controller set up the optimum cycle time based on the maximum queue length. Once the cycle time is set up, the green time is calculated based on the proportions of Q_1, Q_2, Q_3, and Q_4. During the process, neighbour controllers synchorise their cycle time to the j_1 and calculate the self green time, g_1, g_2, g_3, g_4 and offset. The calculated offset time depends on the time a vehicle takes to move from the neighbour intersection, for example intersection 2, and arrive in the end of vehicles queue length in intersection 1, as shown in Figure 7. The strategy is continued until the cycle time is completed and the strategy is called a real-adaptive traffic control system.

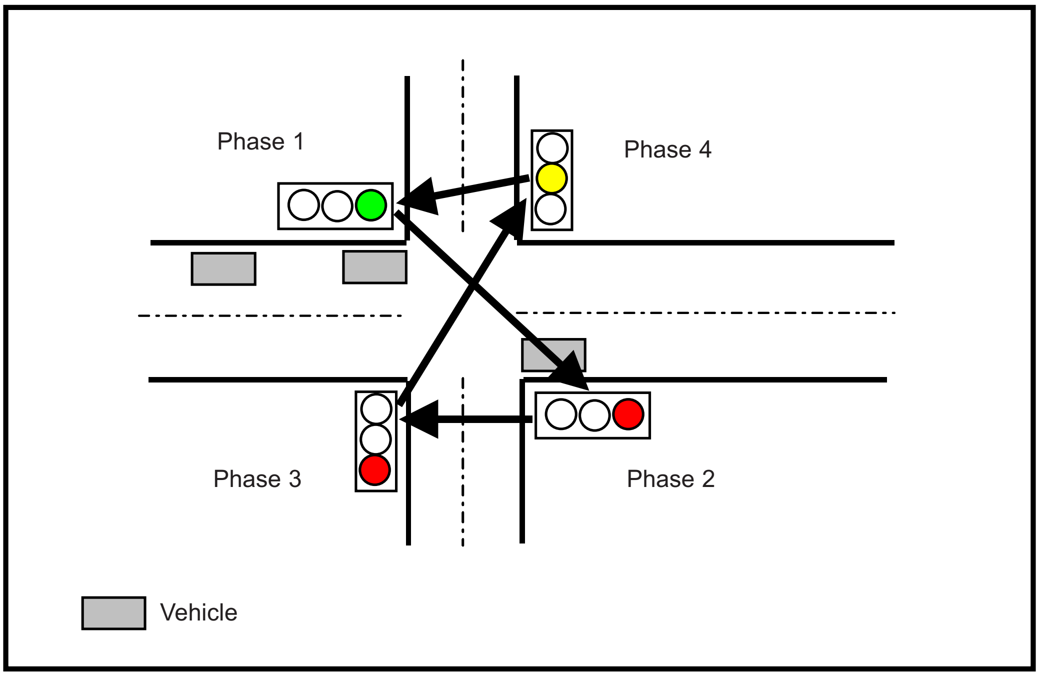


Figure 6 The traffic squence

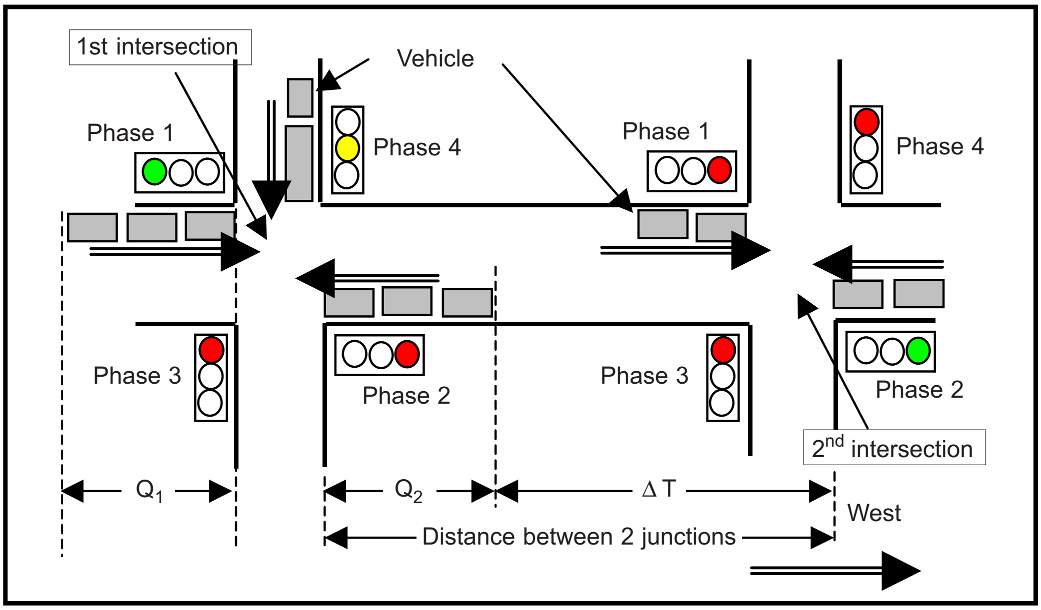


Figure 7 The proposed parameters to calculate the optimum priority

At the controller, the LVQ network is used to classify the traffic condition; normal or congested, as shown in Figure 8. Based on the condition, the LVQ then decides the traffic strategies, i.e. optimum locality or optimum priority. Once the process is completed, the LVQ sends the strategy to the Takagi-Sugeno-Kang (TSK) fuzzy control system. In TSK fuzzy control system, the information is used to calculate the split green time and optimum offset.

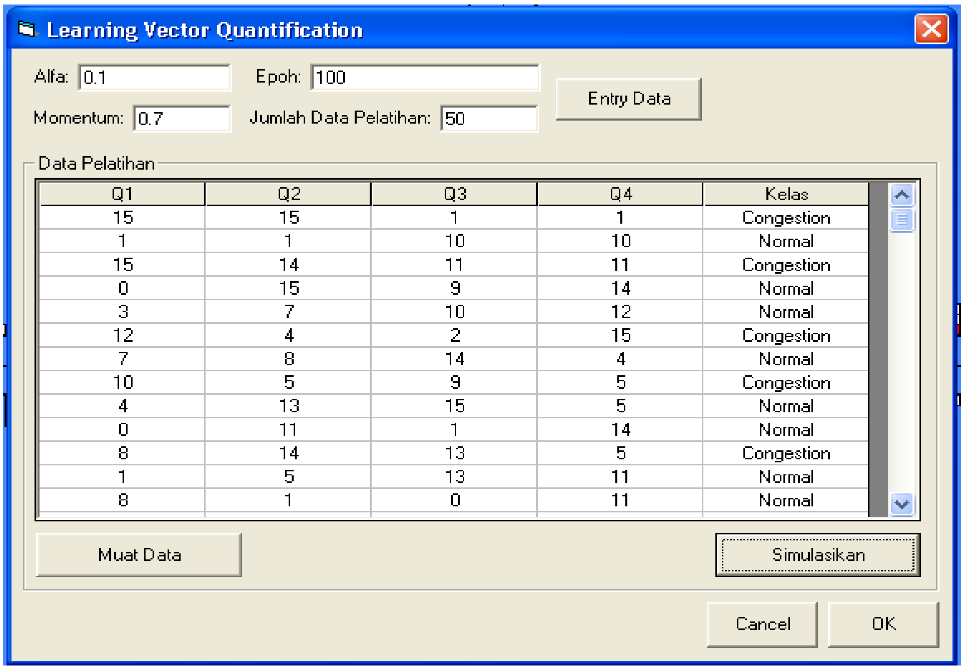
The TSK fuzzy technique is commonly used in prediction and complex control system [10,11]. The TSK fuzzy technique is needed because of its capability to calculate the traffic parameters using mathematical model. The TSK model is composed of IF-THEN rules of the following form:

$$\mathrm{R}(r)\text{: } \textbf{if } x_1 \text{ is } A_r^1 \textbf{ and } x_2 \text{ is } A_r^2 \textbf{ and...and } x_m \text{ is } A_r^m \textbf{ then } y_r \text{ is } f_r(x) \quad (4)$$

where: $f_r(x) = \alpha_r^0 + \alpha_r^1 x_1 + \ldots + \alpha_r^m x_m,$

in which (r = 1,....., n) and x_j ($1 \le j \le m$) are the input variables, y_r is the output variable, A_r^m are fuzzy sets (usually corresponding to linguistic labels), and $f_r(x)$ is a linear function. By this equation, the TSK model of each fuzzy rule describes a local linear behaviour associated to the fuzzy input region characterised by the antecedent of the fuzzy rule.

For any input, say $\hat{x} = \left(x_r^1, x_r^2 \ldots x_r^m\right)$, the inferred value of the TSK fuzzy model is calculated as:



Q1	Q2	Q3	Q4	Kelas
15	15	1	1	Congestion
1	1	10	10	Normal
15	14	11	11	Congestion
0	15	9	14	Normal
3	7	10	12	Normal
12	4	2	15	Congestion
7	8	14	4	Normal
10	5	9	5	Congestion
4	13	15	5	Normal
0	11	1	14	Normal
8	14	13	5	Congestion
1	5	13	11	Normal
8	1	0	11	Normal

Figure 8 LVQ traffic condition classifier

$$\gamma = \frac{\sum_{r=1}^{m} A_r(\hat{x}) * f_r(\hat{x})}{\sum_{r=1}^{m} A_r(\hat{x})} = \frac{\sum_{r=1}^{m} \tau_r * f_r(\hat{x})}{\sum_{r=1}^{m} \tau_r} \tag{5}$$

where:

$$A_r(\hat{x}) = \tau_r = A_r^1(x_r^1) * A_r^2(x_r^2) * \ldots * A_r^m(x_r^m)$$

the τ_r being the level of firing of the r^{th} rule for the current input $\hat{x}$. This algorithm is very appealing as the inference and the defuzzification process are integrated into a single-step procedure [12].

The rule interpretation of TSK fuzzy model depends on the choice of the centre and standard deviation[13]. Only if the premise input space partitioning is performed in an axis-orthogonal manner, the multivariable membership function can be projected to one-dimensional fuzzy sets A. Note that the model output is linear in the weight w but it is non-linear in the centers c and standard deviation σ.

Fuzzy logic inference system is used to collect all of fuzzy rules base to set up the crisp output. Therefore, developing fuzzy inference system must be based on fuzzy rules base. In case of optimization inference system TSK model, the number of desired rules is equal to the number of clustering center. Clustering process produces cluster center value and cluster sigma which will be used to perform the fuzzy logic rules. From this fuzzy rules, the membership of each data on each cluster can also be performed, and the antecedent of each rules can be quantified with union theory [14] as follows:

(a) AND relation use min operation
(b) OR relation use max operation

This quantification process for each rule produces fitness-limit value of each rule. This value is a weight for each fuzzy rule base to set the fuzzy output. In this paper, fuzzy output will be calculated using weight-average method [15] as follows:

$$O = \frac{\sum_{r=1}^{R} (\alpha_r * y_r)}{\sum_{r=1}^{R} \alpha_r} \tag{6}$$

where:

O = fuzzy output
a = fitness-limit value
R = number of rules (equals to number of data cluster)

Generally, the performance of supervised learning process is determined by mean square error (MSE), calculated based on the output split green time which is calculated by human expert. The MSE is determined by the following equation:

$$\begin{aligned} MSE &= \frac{\sum_{i=1}^{n} \left(t_2 - y^r\right)^2}{n}; \forall r \\ y^r &\Rightarrow f(x) \end{aligned} \tag{7}$$

where:

MSE = *mean squared error*
n = number of data

The equation model above needs $y^r \Rightarrow f(x)$ function with fitness-limit value variable, a, which is calculated using genetic algorithm. Genetic algorithm is used to optimize the fuzzy inferences. As shown in Figure 5, the fuzzy logic control system is developed to calculate signal timing and optimum offset in the proposed control system [16].

6.0 CONCLUSION

In the proposed traffic control system, some AI methods such as neural network, fuzzy logic and genetic algorithm are applied to solve traffic control problems. Two-stage neural network model was used to recognise the traffic pattern and then decide the traffic control strategies. A fuzzy–genetic model was used to estimate the objective values in the optimization process with iterative adjustment of signal timings and offset. The proposed method can be applied to an on-line system because it is trained for extensive traffic condition.

ACKNOWLEDGEMENTS

The system described in this paper has been developed at Traffic Laboratory in the Department of Civil and Structural Engineering, Faculty of Engineering, Universiti Kebangsaan Malaysia. The authors wish to acknowledge the Ministry of Science, Technology and Innovation for providing the financial support for this work under IRPA project No: 04-02-02-0047-PR0059/09-01.

REFERENCE

[1] Wey, M. W. 2000. Model Formulation and Solution Algorithm of Traffic Signal Control in an Urban Network. *Computers, Environment and Urban Systems*. 24: 355-377.

[2] Lan, C. J. L., and G. A. Davis. 1999. Real-time Estimation of Turning Movement Proportions from Partial Counts on Urban Networks". *Transportation Research*. Part C 7: 305-327.

[3] DiFebbraro, A., and N. Sacco. 2004. On Modelling Urban Transportation Networks via Hybrid Petri Nets. *Control Engineering Practice*. 12: 1225-1239.

[4] Venglar, S., and T. Urbanik II. 1995. Evolving to Real-Time Adaptive Traffic Signal Control. Proceedings of the Second World Congress Intelligent Transport Systems. Yokohama, Japan.

[5] O. K. Rahmat, Riza Atiq. 2002. Sistem Kawalan Lalu Lintas Ternaju. PhD Thesis. Universiti Kebangsaan Malaysia. Bangi, Selangor.

[6] Kohonen, T. 1990. The Self-Organizing Map. *Proc. IEEE* (*Special Issue on Neural Networks*). 78(9): 1464-1480.

[7] Moshiri, B., and S. A. Jazbi. 1997. Fault Detection and Isolation with RBF Neural Network. Pre-Prints of AIRTC.

[8] Haykin, S. 1999. *Neural Networks, a Comprehensive Foundation*. 2nd Edition. Pearson Education. New Jersey: Prentice Hall.

[9] Gersho, A., and R. M. Grey. 1992. *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer.

[10] Begovich, O., E. N. Sanchez, and M. Maldonado. 2002. Takagi-Sugeno Fuzzy Scheme for Real-Time Trajectory Tracking of an Underactuated Robit. *IEEE Transactions On Control Systems Technology*. 10(1).

[11] Papadakis, S. E., and J. B. Theocharis. 2002. A GA-based Fuzzy Modeling Approach for Generating TSK Models. *Fuzzy Sets and Systems.* 131: 121-152.

[12] Gomez, A. F., M. Delgado, and M. A. Vila. 1999. About the Use of Fuzzy Clustering Techniques for Fuzzy Model Identification. *Fuzzy Set and Systems.* 106: 179-188.

[13] Sugeno, M., and G. T. Kang. 1998. Structure Identification of Fuzzy Model. *Fuzzy Sets and Systems*. 28: 15-33.

[14] Sin, S. K., and D. Figueiredo. 1993. Fuzzy System Designing Through Fuzzy Clustering and Optimal preDefuzzification. *Proc. IEEE International Conference on Fuzzy Systems.* 190-195.

[15] Roger Jang, J.-S., C.-T. Sun, and E. Mizutani. 1997. *Neuro-fuxxy and Soft Computing: A Computational Approach to Learning and Machine Intel ligence*. MATLAB Curriculum Series. Upper Saddle River: Prentice Hall.

[16] Agus, P. et al. 2004. Using Subtractive Clustering to Generate Fuzzy Rules. In ROVISP conf., Malaysia.