REAL TIME SYNCHRONIZATION OF SIGNALS FOR HETEROGENEOUS TRAFFIC

1Nuli Sadguna, Department of Civil Engineering, IIT Bombay, Mumbai, India, e-mail: findguna@gmail.com
2Tom V Mathew, Department of Civil Engineering, IIT Bombay, Mumbai, India, e-mail: tvm@civil.iitb.ac.in

ABSTRACT

Real time control of heterogeneous traffic was always a challenge for efficient and effective traffic management. The solution becomes more complex when the heterogeneity is augmented with limited lane discipline. Adaptive traffic signal controllers offer better signal time management especially when the traffic is not saturated. Traditional methods of upstream vehicle detection (UTOPIA, PRODYN, SCOOT, OPAC, and RHODES) work well with disciplined traffic. However, this approach has several limitations when the drivers do not adhere to their respective lanes. A traffic adaptive control model which uses stop-line detector information is proposed in this paper. The model aims at real-time allocation of optimum cycle time through neuro-fuzzy actor critic reinforcement learning; an approach originated from the machine learning community. This approach has the ability to learn relationships between control actions and their effect on the environment while pursuing a goal is a distinct improvement over pre-specified models of the environment. The Model searches for optimum values by beginning with the initial values in real time. To have efficient learning and reliable decision making the model uses measured values. The performance of the model is tested on a typical four phase three intersections arterial with variable flow is simulated using a traffic micro-simulator (VISSIM) and interfaced with the proposed model. The model performance is compared with the traditional fully-actuated system. The results using this approach shows significant improvement over traditional control, especially for high traffic demand.

Keywords: Adaptive, Agent, Reinforcement Learning, and Signal Coordination

INTRODUCTION

The movement of a platoon of vehicles through several signalized intersections is referred to as progression. By properly coordinating the traffic signals in a network, platoons of vehicles
can keep progressing. Signal coordination can be achieved in off-line or on-line means. Offline coordination is done either by progression-based methods, which maximize the bandwidth of the progression, or dis-utility based methods, which minimize a performance measure such as the overall delay and stops. Bandwidth optimization techniques, such as MAXBAND (Little et al. 1981), MULTIBAND-96 (Stamatiadis and Gartner 1996), and PASSER (Venglar et al. 2000) use traffic volumes, signal spacing, and desired travel speed to determine the optimum width of the progression band that can be accommodated on an arterial. Whereas dis-utility based models such as TRANSYT (Robertson 1969), SYNCHRO (Berkeley, California 1996) generally attempt to find a common cycle length that minimizes the amount of overall delay in the system and then compute the offset required for progression. Off-line models are well suited in a situation where the traffic fluctuation is minimum (V/C > 1) and coordination direction is fixed. However, when the flow is dynamic and direction of coordination requires changing frequently these models perform badly/not applicable. To address this on-line coordination was proposed.

The models can be broadly classified into heuristic, centralized/hierarchical control, and multi agent control techniques. Well known heuristic systems employ a library of pre-stored signal control plans, which are developed off-line on the basis of historical traffic data. Plans are selected on the basis of the time of day and the day of the week, directly by the operator, or by matching from an existing library a plan best suitable for recently measured traffic conditions (Katwijk et al. 2006). Although the solution provided by heuristic methods is very quick, it is not truly adaptive with respect to horizon time or future traffic pattern due to its reactive nature of control.

To have traffic signal control in true adaptive nature numerous models such as SCATS, SCOOT, OPAC, UTOPIA, PRODYN, MOTION, RHODES, and ACS Lite were proposed (Lowrie 1982, Robertson 1969, Robertson and Bretherton 1991, Liao 1998, Mauro and Taranto 1989, Henry 1989, Kruse 1998, Busch and Kruse 2001, Mirchandani and Head 2001, FHWA 2006). The models use various detection configurations, optimization techniques, and prediction models to obtain an optimal solution. The operational characteristics and working principles of most widely used models are shown in Table 1. It clearly shows that the models are pro-active based on the predictions made using advance/upstream detection information. Further, the models require communication to/from central server for efficient functioning.

To eliminate communication requirement to/from central server and to have distributed control few researchers applied agent control techniques to model the traffic signal control problem. Isolated intersection control is usually modelled as a single agent decision making problem. For a system that has more than one intersection such as an arterial traffic control should be modelled as multi agent system (MAS) (Hu and Wellman 2003, Littman 1994, Hu and Wellman 1998, Tan 1993). In the independent agent control each agent treats all other agents as part of the environment. One potential problem of this method is that the existence of other agents may affect the environment and invalidate the Markov property assumption (Hu and Wellman 2003). To overcome this limitation researchers have applied two-agent
zero-sum SG and general sum SG theory to model urban arterial. For arterial traffic signal control, the gain of one control agent does not necessarily mean the loss of other agents. Therefore, the zero-sum SG framework is not suitable for modelling arterial traffic control problems. This limitation was addressed by general-sum SG theory formulation using multi agent Q-learning algorithm, where different agents can increase their gains simultaneously. The major difficulty in applying multi agent Q-learning is the number of state variables will become very large and make the learning process extremely slow (Littman 1994, Hu and Wellman 1998).

MARL based on the SG framework is theoretically sound. However, it is not suitable for real world control applications due to its complexity and large state space. Therefore, to address this limitation a new approach of information sharing with each other agents using cooperative - agent theory has been applied. However, in the case of arterial control different intersections may have different geometric settings; their environments are most likely different. Under this circumstance, sharing experience among different agents may not be useful. In addition, it was shown that sharing experience among agents only expedited the learning process and did not appear to improve the learning results (Tan 1993).

In another instance swarm-intelligence technique was applied for arterial coordination. In swarm-based dynamic coordination vehicles waiting for their next green indication, continuously produce pheromone. Traffic signal agents perceive the pheromone trails and select an appropriate signal plan. In this study when agents are free to decide according to the swarm approach the system behaves almost as a central decision support system. Experiments show that agents achieve synchronization without any central management. However, the time needed to converge to a stable coordination can be high, which is a negative aspect especially in highly dynamic environments (Oliveira et al. 2004).

It is relatively easy to develop traffic signal implementation plan using some predictive models if the traffic is more or less homogeneous and also if lane discipline is followed. However, their implementation in heterogeneous traffic poses greater challenges. The term heterogeneous traffic in this paper refers traffic comprising passenger cars, standard trucks and buses, and non-conventional vehicles like, three wheeled auto rickshaws, and motor bikes, etc. These vehicles have diverse static characteristics (dimensions) and dynamic characteristics like acceleration / deceleration. In addition, such traffic is characterized by low level of lane discipline. Due to the in-adherence of vehicles to the lanes supposed to be followed and moving in zigzag manner, it is difficult to estimate reliable queue lengths associated with various movements. This results in considerable error in the green and delay estimations.

Therefore, properly designing an adaptive control plan to handle highly heterogeneous traffic is a complex task. For such complex situations, the modifications and improvements are required both on the control model and implementation strategy to manage the situation (Ravikumar and Mathew 2011).
From the above discussion, following limitations of adaptive traffic control system for a corridor control are identified: (1) No traffic adaptive control model exists to account for traffic heterogeneity & limited lane discipline. (2) Ability to find optimum cycle for large scale problems such as multi-phase signal control and number of intersections on real time.

To address these limitations proposed model “TRaffic Adaptive Signal Control using Reinforcement learning - Corridor (TRASCR-C)” is exclusively designed for corridor control based on stop line detection information using neuro-fuzzy actor critic reinforcement learning. The advantage of this model is, it does not require any prediction or estimation values in decision making as is required by most of the state-of-the-art adaptive traffic control systems.

**METHODOLOGY**

The basic idea of proposed control algorithm is to bring adaptive feature to the stop line based vehicle actuated controller through reinforcement learning. Controller terminates target phase according to maximum green time determined by proposed model TRASCR-C, whereas other phases would be terminated as in stop line based vehicle actuated (VA) control. In order to look at the whole corridor in time & space, the model uses reinforcement learning. The control begins with the supply of various control parameters and initialization of model parameters. If the time is less than simulation/control period, it checks whether the active phase is target phase or not. The active phase is checked for maximum green time criteria if the phase is target phase else VA termination criteria. If the elapsed time is equal to cycle update interval, it gets new state and reward from the system i.e., latest greens of critical intersection and increase in discharge from previous control action. The TRASCR-C model then determines new cycle time. The model learns about these actions through reinforcement learning. If the active phase is target phase, $g_{\text{max}}$ and $g_{\text{min}}$ values are updated as in equations 1 and 2 to achieve coordination.

\[
g_{\text{max}} = C - \sum g_i
\]

if \((g_{\text{max}} - g > \text{trans\_increment})\)

\[g_{\text{max}} = g + \text{trans\_increment} \quad (1)\]

\[
g_{\text{min}} = \text{Offset} - (T - T_{n-1})
\]

if \((g_{\text{min}} - g > \text{trans\_increment})\)

\[g_{\text{min}} = g + \text{trans\_increment} \quad (2)\]
Table 1: Operational Characteristics of ATCS (Source: NCHRP Synthesis 403, 2010)

<table>
<thead>
<tr>
<th>ATCS</th>
<th>ACS Lite</th>
<th>BALANCE</th>
<th>Insync</th>
<th>LA ATCS</th>
<th>MOTION</th>
<th>OPAC</th>
<th>RHODES</th>
<th>SCATS</th>
<th>SCOOT</th>
<th>UTOPIA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adjustment</td>
<td>DCO</td>
<td>TCO</td>
<td>DCO</td>
<td>RA, TCO, DCO</td>
<td>TCO</td>
<td>TCO</td>
<td>RA</td>
<td>DCO</td>
<td>TCO</td>
<td></td>
</tr>
<tr>
<td>Time Frame</td>
<td>5-10 min</td>
<td>5 min</td>
<td>phase/cycle/15 min</td>
<td>5-15 min</td>
<td>phase/cycle/15 min</td>
<td>sec by sec</td>
<td>cycle</td>
<td>cycle/5 min</td>
<td>cycle</td>
<td></td>
</tr>
<tr>
<td>Level</td>
<td>C/L</td>
<td>C/L</td>
<td>C/L</td>
<td>C/L</td>
<td>C/L</td>
<td>C/L</td>
<td>C/L</td>
<td>C/L</td>
<td>C/L</td>
<td>C/L</td>
</tr>
<tr>
<td>Model</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Flexi Region</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Vehicle Actuated</td>
<td>Yes</td>
<td>yes</td>
<td>No</td>
<td>Yes</td>
<td>yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>TSP</td>
<td>No</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

Detection: SL = stop-line, NSL = near stop line, MB = mid-block, US = upstream
Action: P = proactive, R = reactive
Adjustment: RA = rule base, DCO = domain constrained optimization, TCO = time constrained optimization
Level: L = local, C = central
Timings: S = split, C = cycle, O = offset, PS = phase sequence
Reinforcement learning (RL) is an area of machine learning concerned with how an agent ought to take actions in an environment so as to maximize some notion of cumulative reward. In reinforcement learning agent interacts with its environment in discrete time steps as shown in Fig. 1. At each time \( t \), the agent receives information about the state of the environment \( s_t \), which typically includes reward \( r_t \). It then chooses an action \( a_t \) from the set of actions available, which is subsequently sent to the environment. The environment moves to a new state \( s_{t+1} \) and the reward \( r_{t+1} \) associated with the transition \((s_t, a_t, s_{t+1})\) is measured. The goal of a reinforcement learning agent is to collect as much reward as possible. The agent can choose any action as a function of the history and it can even randomize its action selection.

**TRASCR-C Model**

To address the problem of large state space (multiphase signal control) associated with the conventional reinforcement learning techniques such as Q-learning and SARSA, proposed model uses neuro-fuzzy actor-critic reinforcement learning to learn about the relation between various states and corresponding actions to be taken. The design elements of TRASCR-C in terms of the typical RL structure (i.e., state, action, and reward) are discussed next;

**State**

As the queue length cannot be determined in stop line based vehicle actuated control and termination is based on threshold gap/maximum green criteria, the state of the system is represented in a novel way as latest green times of the critical intersection of the corridor. Hence, state definition is represented by a vector of \( F \) components that are the actual greens associated with each phase.
Action

The action of the control agent is to assign common cycle for the corridor. The action space is represented as in the neighbourhood of the previous cycle as given below

\[ C_{\text{new}} = C + n^*5 \quad \text{where } n = -1, 0, \text{ and } 1 \]  \hspace{1cm} (3)

Reward

To have efficient control at saturated/over saturated traffic conditions maximizing total discharge is more rational than minimizing delay and/or no of stops. Therefore, the reward is defined as the increase in the total discharge, i.e., the difference between total discharges between two successive cycle updates. If the reward has a positive value, this means that the total discharge was increased by this value after executing the action. However, a negative reward value indicates that the action results in decreased discharge.

Model

Similar to neural networks, the TRASCR-C model has four layers as shown in Fig. 2. The first 3 layers together represent state of the system and the action space is represented by fourth layer. The learning experience is stored in connection weights i.e., model parameters \( \lambda^{k}_{j} \) and \( w^{k}_{q} \) between 3rd and 4th layers. The data flow across each layer is explained as follows. The first layer is the input layer. It receives state variable values and sends them to different fuzzy membership functions in the second layer. Each node in the first layer represents an input variable. Each node in the second layer is a fuzzy set with a fuzzy membership function associated with it. The inputs to the second layer are the state variable values, and the outputs of the second layer are fuzzy membership function values. The inputs and fuzzy sets of the second layer constitute many linguistic terms such as “Green is Short” and “Green is Long”. Thus, the outputs of the second layer can be considered as degrees of membership values associated with the short green & long green fuzzy variables as given in equations 4 and 5.

\[
\mu_{s}^{i} = \begin{cases} 
1 & \text{if } g = 0 \\
\frac{a - g}{a} & \text{if } 0 < g \leq a \\
0 & \text{if } g > a 
\end{cases} \quad (4)
\]

\[
\mu_{l}^{i} = \begin{cases} 
0 & \text{if } g = 0 \\
\frac{g}{a} & \text{if } 0 < g \leq a \\
1 & \text{if } g > a 
\end{cases} \quad (5)
\]
The third layer corresponds to antecedent part of fuzzy rules in a fuzzy logic controller. For example, a sample fuzzy rule in case of 4-phase signal control can be represented as

$$\text{IF} \{g_1 \text{ is long AND } g_2 \text{ is long AND } g_3 \text{ is long AND } g_4 \text{ is short}\} \text{ THEN} \{\text{increment } g_{\text{max}}\}$$

Therefore, outputs of the third layer can be considered as firing strengths. The fourth layer is a collection of nodes representing consequences part of fuzzy rules. The first node stands for the state value, and its output value shows how good the current state value is i.e., in the present case it represents discharge. The remaining nodes correspond to the available actions that can be taken, and their output values are the preferences to choose each action given the current state inputs.

**Action Selection Method**
To facilitate the agent to search overall state-action space, a well-known ε-greedy method is used for selecting a particular action. The ε-greedy learner selects greedy action most of the time except for a small amount ε of time, it selects a random action uniformly (El-Tantawy and Abdulhai 2010). The value of ε is chosen to decrease gradually with iterations (from 0.9 to 0.1). This will result in more exploration at the beginning of the learning process which enables the agent to search the overall state-action space and gradually emphasizes exploitation as the agent converges to the optimal policy.

**Computation of State & Action values**

In order to select a particular action according to ε-greedy method the model needs to estimate state and action values as discussed below. In the model, the input and fuzzification parts are the same as the typical fuzzy logic control. Given the latest greens of the critical intersection as inputs, the short & long fuzzy variables (equations 4 and 5) generate membership values and these values are fed into the third layer of TRASCR-C.

Assuming the jth fuzzy rule has the following F antecedents

\[ g_i \in \mu_{ij}^m \quad \text{where} \quad i = 1, \ldots, F \]  

(6)

Then the firing strength of the jth fuzzy rule is

\[ R_j^k = \prod_{i=1}^{F} \mu_{ij}^m \]  

(7)

The state value \( V_s^k \) is defined as follows

\[ V_s^k = \sum_{j=1}^{N} R_j^k \lambda_j^k \]  

(8)

Where, \( \lambda_j^k \) is weight connecting the jth fuzzy rule and the state value in the kth iteration; Similarly, the preference of choosing each action \( A^q \) is defined as follows

\[ A^q = \sum_{j=1}^{N} R_j^k w_{qj}^k \]  

(9)

Where, \( w_{qj}^k \) is weight connecting the jth fuzzy rule and the qth action output in the kth iteration;

**Training of TRASCR-C**

The training of TRASCR-C model i.e., the process of tuning of critic and action weights is done based on the well-known temporal difference learning technique. The advantage of
temporal-difference (TD) learning methods is it can learn directly from raw experience without a model of the environment’s dynamics (Sutton and Barto 1998). The TD methods update estimates based in part on other learned estimates, without waiting for a final outcome.

Therefore, a TD error $\delta$ in estimating the state value $V_s^k$ is given as

$$\delta = (V^k - V^{k-1}) + \gamma V_s^k - V_s^{k-1}$$

(10)

Where, $V^k$ is the interval discharge in the $k^{th}$ iteration; $V_s^k$ is the state value in the $k^{th}$ iteration; and the new weights are updated as given below.

$$\lambda_j^k = \lambda_j^{k-1} + \alpha \delta s^{k-1}$$

(11)

$$w_j^{qk} = w_j^{q(k-1)} + \alpha \delta s^{k-1}$$

(12)

**MODEL TESTING**

Any new or modified traffic control system is accepted, only when it satisfies a goal or set of goals. The goal here for the proposed corridor model TRASCR-C is to improve the travel speed along the corridor (west-east). The efficiency of a proposed traffic control system and its strategies towards desired goals is determined quantitatively by means measures of effectiveness (MOEs). To evaluate system efficiency following MOEs is selected: Travel time and throughput along the corridor. The performance of the widely used vehicle actuated control is used as a benchmark and is compared to the proposed model.

The proposed real-time adaptive signal control model is tested using a scalable, high performance microscopic simulation package, Vissim 5.10. It has been widely used in the testing of various algorithms and evaluation of various Intelligent Transportation Systems (ITS) strategies because of its powerful Application Programming Interfaces (API), through which users can access the core models to customize and extend many features of the underlying simulation model such as signal control without having to deal with the underlying proprietary source codes. The proposed model is developed as a Vissim plug-in through API programming.

**Typical Urban Corridor**

A typical urban corridor consists of three intersections as shown in Fig. 3 was selected to test the performance of the model. To account for the effect of heterogeneity and limited lane discipline, the Vissim driving behaviour was fixed according to the calibrated parameters and guidelines given by authors (Mathew and Radakrishnan 2010). To test the robustness of the
Figure 3 Typical Urban Corridor

Heterogeneous Volume (I/P)

Figure 4 Shows typical daily variation of traffic flow
model diverging flow patterns (heavy volume on west and east approaches & low volume on south and north approaches) with typical day time variations and heterogeneous vehicle composition were given as shown in Fig. 4.

<table>
<thead>
<tr>
<th>Table 1 Corridor Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Section</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Section_1 (911 m)</td>
</tr>
<tr>
<td>Section_2 (1328 m)</td>
</tr>
<tr>
<td>Section_3 (1836 m)</td>
</tr>
</tbody>
</table>

The corridor performance of the proposed TRASCR-C model is compared against VA control in Table 1. It clearly shows that the travel time was improved along the coordinated East-West direction for all the measured sections. Further, the impact of the proposed coordination on the overall system is summarised in Table 2. It shows that the discharge from the coordinated phase as well as whole intersections is improved. Therefore, it can be inferred that the model is able to coordinate the desired direction without penalizing overall system.

<table>
<thead>
<tr>
<th>Table 2 Intersection Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Green</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Junction-1</td>
</tr>
<tr>
<td>Phase-1</td>
</tr>
<tr>
<td>Phase-2</td>
</tr>
<tr>
<td>Phase-3</td>
</tr>
<tr>
<td>Phase-4</td>
</tr>
<tr>
<td>Junction-1</td>
</tr>
<tr>
<td>Junction-2</td>
</tr>
<tr>
<td>Phase-1</td>
</tr>
<tr>
<td>Phase-2</td>
</tr>
<tr>
<td>Phase-3</td>
</tr>
<tr>
<td>Phase-4</td>
</tr>
<tr>
<td>Junction-2</td>
</tr>
<tr>
<td>Junction-3</td>
</tr>
<tr>
<td>Phase-1</td>
</tr>
<tr>
<td>Phase-2</td>
</tr>
<tr>
<td>Phase-3</td>
</tr>
<tr>
<td>Phase-4</td>
</tr>
<tr>
<td>Junction-3</td>
</tr>
</tbody>
</table>
CONCLUSION

This paper presented the framework and evaluation of “TRaffic Adaptive Signal Control using Reinforcement learning-Corridor (TRASCR-C)”. As proposed the model is able to brought adaptive feature to the stop line based vehicle actuated controller through neuro-fuzzy actor critic reinforcement learning. The model also addresses the weaknesses of operational adaptive control models by having the following attributes: (1). The information used for decision-making processing is more reliable because it is not dependent on forecasting model as in other state-of-the-art models. (2). Intersection utilization is used explicitly in the objective function. (3). More efficient with respect to field implementation as it requires only stop line detectors information.

As it is very difficult or impossible to include traffic heterogeneity and limited lane discipline into control model (field implementation point of view), TRASCR-C model addressed the issue by moving detector to stop line. The model is also addressed the issue of scalability by integrating fuzzy logic and neural network into reinforcement learning.
NOTATION

The notations used in the model are described below.

\( \phi_i \) = signal state of the \( i \)th phase \{GREEN or RED\}

\( \alpha \) = learning rate

\( \gamma \) = discount rate

\( \delta \) = temporal difference error

\( \lambda_{kj} \) = weight connecting the \( j \)th fuzzy rule and the state value in the \( k \)th update

\( \mu_{ij}^{m} \) = \( m \)th membership function for the \( i \)th input variable in the \( j \)th fuzzy rule

\{\( m=1, \ldots, M \), \( j=1, \ldots, N \} \)

\( A^q \) = preference of choosing \( q \)th action \{\( q=1, \ldots, Q \} \)

\( F \) = total number of input variables/phases

\( M \) = total number of fuzzy membership functions

\( N \) = total number of fuzzy rules

\( Q \) = total number of actions

\( R_{ij}^{k} \) = firing strength of \( j \)th fuzzy rule in the \( k \)th update; \( j = 1, \ldots, N \)

\( V_{s}^{k} \) = state value for a given state \( s \) in the \( k \)th update;

\( V^{k} \) = discharge in the \( k \)th update;

\( g_i \) = actual green for the \( i \)th phase

\( g_{ini} \) = initial green

\( g_{max} \) = maximum green

\( g_{min} \) = minimum green

\( h_{th} \) = threshold gap

\( i \) = phase

\( k \) = update counter

\( t \) = current simulation time

\( w_{ij}^{q} \) = weight connecting the \( j \)th fuzzy rule and the \( q \)th action output in the \( k \)th update;

\{\( q=1, \ldots, Q \}, \{j=1, \ldots, N \} \)

---

13th WCTR, July 15-18, 2013 – Rio de Janeiro, Brazil
REFERENCES


