INEL-95/00385
CONF-951150--1

AN INTELLIGENT TRAFFIC CONTROLLER

Kishore Kagolanu
Ada County Highway District
Raymond Fink
Herschel Smartt
Robert Powell
Eric Larsen
Lockheed Martin Idaho Technologies
Idaho National Engineering Laboratory

ABSTRACT

A controller with advanced control logic can significantly improve traffic flows at intersections. In this vein, this paper explores fuzzy rules and algorithms to improve the intersection operation by rationalizing phase changes and green times. The fuzzy logic for control is enhanced by the exploration of neural networks for families of membership functions and for ideal cost functions. The concepts of fuzzy logic control are carried forth into the controller architecture. Finally, the architecture and the modules are discussed. In essence, the control logic and architecture of an intelligent controller are explored.

1. INTRODUCTION

As vehicular traffic increases in most cities around the world, increase in the capacity of street networks has not kept pace. This is leading to severe congestion on the street networks. Most of the congestion experienced by traffic occurs at intersections. Therefore, efforts at reducing congestion at the intersection level will improve the performance of the overall street network. In this vein, advanced technologies at intersection controller level are an exciting possibility in reducing congestion.

Most cities use UTCS style control systems with the intersection controllers operating off the system commands. These systems traditionally use a single timing plan for the peak hour and/or a set of timing plans during the day. Such operation neither utilizes the computing capabilities of the controller nor services traffic in an efficient fashion. Existing traffic control systems like SCOOT and SCATS only react to traffic flow and cannot efficiently service traffic.

The advances in computing sciences have not been applied to traffic control. Existing traffic controllers have significant computing power but only react to detection data. Improving the logic in a traffic controller can significantly increase intersection traffic flows. With the above introduction, this paper describes the development of an intelligent traffic controller. The structure of this paper consists of six sections 1) conventional traffic control 2) cost functions 3) fuzzy logic 4) neural network approaches 5) controller architecture and 6) prototype controller.

2. CONVENTIONAL CONTROL

Two decades ago traffic controllers underwent a technological advance by moving from electro-mechanical control to solid state device control. Since then however, controller technology has not significantly improved the control logic to optimize traffic flow. Current advances in computing sciences are not reflected in the traffic control logic. Conventional traffic controllers operate in the following modes: 1) pretimed controllers 2) actuated controllers. The pretimed controllers operate on fixed timing plans. These controllers operate off the master controller sending a synch pulse at regular intervals of one system cycle length. Each controller will dwell at this offset point until it receives the synch pulse from the master controller, by that ensuring the maintenance of the proper offset relationship.

Semi Actuated is similar to pretimed operation, except that the synch pulse is coordinated with a time zero point set on a device called a coordinating unit. Coordination is accomplished by setting a yield point on each coordinating unit corresponding to the end of the main street green or non-actuated phase. This allows the controller to yield control to other phases.

Fully Actuated is similar to semi-actuated operation, except that the main street green phase yield point can be set to any value and the controller will move from the main street green if there is no demand after the set yield time.

Traffic Responsive Mode of operation is

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

available in several systems. These systems use different logic and algorithms using different data inputs, in responding to traffic conditions. There are three main methods: projection, pattern matching, and actuation for traffic responsive operation. However, they have not gained wide acceptance or have proven to be significantly more efficient. For a further discussion on control systems see reference [1].

3. COST FUNCTIONS

An important aspect of a traffic controller is to maximize the ratio of cars exiting the intersection to cars entering the intersection. Also, we want to minimize delay at the intersection. To achieve this, we need a cost function (cost of operation in terms of traffic flow) that incorporates the important variables of traffic flow. Upon developing the cost function, we will maximize traffic flow through the intersection by minimizing the cost function. The input data that are available in real-time to the controller are traffic volumes and delay. Therefore, the cost function will use the volume and delay data to compute the cost of operation. The form of the cost function used in this study is presented in equation 1 [2].

$$Cost=\frac{Delay_{avg}}{k(\frac{Cars_{exit}}{Cars_{enter}})Drive_{avg}} \qquad (1)$$

4. FUZZY CONTROL

In the most elementary sense, fuzzy logic control can be defined as the control of a machine in linguistic terms. A fuzzy controller will operate a machine in a fashion similar to that of a human operator. A set of heuristic control rules, as stated by a human operator, are translated into an automatic control strategy using fuzzy logic. Intersection control is complex and the varying traffic flows within the peak hours lead to difficulties to control automatically. Therefore, the control strategy may be improved by fuzzy rule-based operation.

The fuzzy controller will receive input data from the sensors. These data will include traffic volumes and delay. This input data to the controller is converted to linguistic equivalents and further processed for decision making. A set of rules for making a decision on the state of the system are included in the controller. These rules will output a non-fuzzy decision on the state of the traffic light i.e., change the light to green or continue the red phase. Figure 1 presents the schematic for fuzzy control at a traffic signal. The "Continuous Fuzzy Logic" program developed by INEL will be used for the controller [3]. The program is written in "C" code and will be embedded in the controller.

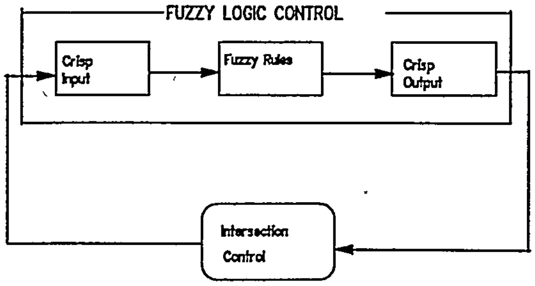

Figure 1 Schematic of Fuzzy Control

4.1. Membership Function For Input Data

An important aspect of the fuzzy control logic used in the signal controller is membership function. A membership value of an element in a set defines the degree to which the element satisfies the condition of the set.

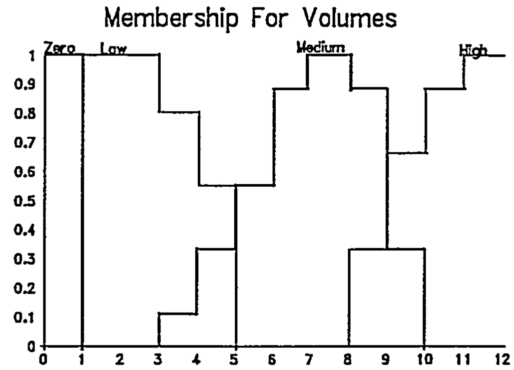

Figure 2 Membership values for volume

The fuzzy controller uses families of membership functions for optimal control. These families are developed as a response to various traffic flows. There are four membership functions (Zero, Low, Medium, High) for the four traffic volumes inputs for all movements at the intersection. The membership functions of the inputs are differently shaped by the overall traffic conditions and the traffic signal phases. Figure 2 presents an example membership function for traffic volumes. Similarly, Figure 3 illustrates the membership function for delay. These membership functions convey the degree to which the different values of delay satisfy the set of small, medium or large delay.

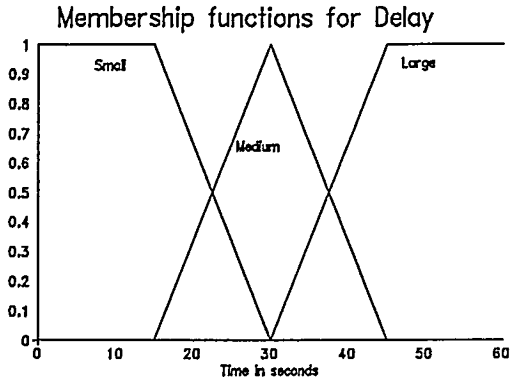

Figure 3 Membership value for delay

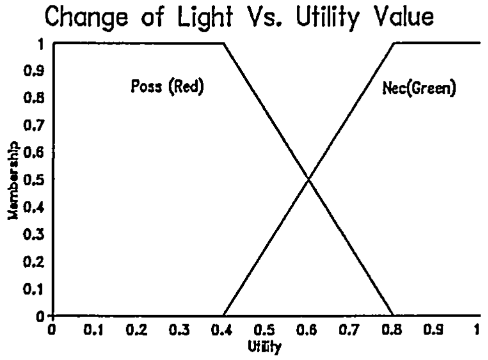

Figure 4 Utility value for phase change

4.2. Membership Function for Utility

The output of the fuzzy controller gives the utility of changing the indication of the traffic light. These utility values for change range from zero to one. These utility values define the possibility of red [Poss (Red)] for the current phase and the necessity of green [Nec(Green)]. These values decide whether to change the existing state of the light (from green east-west to green north-south) or remain in the same state. Figure 4 illustrates the plot of possibility and necessity values for changing the signal phase.

5. NEURAL NETWORK APPROACH

The applicability of neural networks is being investigated in two specific functions of the controller. First, the ideal system function and second, the development of families of membership functions for traffic data inputs. A multi layer network consisting of three completely connected layers, i.e., the input layer, the hidden layer, and the outer layer, is being investigated for the ideal system model. Figure 5 presents the concept on a smaller scale. The learning will be induced via the back propagation algorithm with a sigmoid transfer function. This function is suggested as being appropriate in several similar applications [4-6].

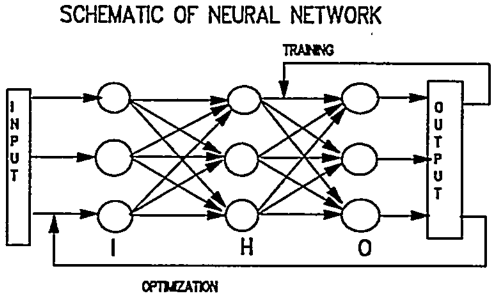

Figure 5 Schematic of neural network

The neural network model describes the relationship between the traffic data inputs and the output variables. That is, traffic volumes, delay, and phase information will be provided into the input layer (I) and outputs will be ideal system values and membership functions at the output layer (O). The input layer serves as the distributor for the hidden layer. The size of the input layer and the output are dependent on the numbers of input and output factors. The hidden layer is the "generalizing layer." It takes input layer neurons and tends to combine them into groups. An understanding of the number of groupings in the application helps in choosing an appropriate number of neurons for the hidden layer. In general, the number of hidden layer neurons is in between the number of input layer neurons and output layer neurons.

Each input variable is normalized to [0,1] by using the ideal system values of the model for the target flow and the values of input data for inflows, respectively. All data will be sampled every cycle, though time sampling in the systems is variable. The output values are system values in the ideal condition and membership values for traffic variables.

The target value estimation algorithm contains two phases. First, the learning phase is used to compute the optimal weights of the neural network for a typical pattern. The second is an adaptive forecasting phase to adapt the weights to the present sensor inputs and output ideal system values.

The training data are system values for traffic flows on a typical business day. The network connecting weights are computed by following the standard back propagation learning rule described below:

$$M_{ji}(n+1)=M_{ji}(n)+\delta M_{ji}(n) \qquad (2)$$

$$\delta M_{ji}(n)=\eta\frac{\partial E}{\partial M_{ji}}+\alpha\delta M_{ji}(n-1) \qquad (3)$$

$$Where E=\frac{1}{2}\sum_{t}(D_t-A_t) \qquad (4)$$

$$D_t=t^{th} desiredoutput \qquad (5)$$

and

$$A_t=t^{th} actualoutput \qquad (6)$$

6. CONTROLLER ARCHITECTURE

The controller is expected to be of a small lunch box size. Architecture of the controller will be designed as an add-on to or a retrofit of existing intersection controllers. The controller will improve traffic flow by providing a continuously variable timing plan. The architecture will accommodate fault-detection, interrupts, and system overrides. Figure 6 presents the architecture of the controller.

The controller architecture consists of five modules 1) Observer 2) Ideal System model 3) Meta-Planner 4) Fault detection and interrupts 5) Timing plan generator. The thick lines in the figure represent the data flows. While thin lines are essentially the global system over ride. The sensor data is the input to the controller and the green times are the output of the timing plan generator.

The observer module is designed to provide information on the state of the existing timing plan. The traffic volumes of on the various approaches are received by the observer module. The observer module will then compute the cost under the existing input data and traffic light indications. This is the actual value of the cost function for current conditions at the intersection; that is, a current measure of performance.

The ideal system module provides information on the optimal solution for the traffic flows. The ideal system model will receive the traffic data from the sensor. Using the sensor data and the cost function, the ideal system model computes the cost of signal operation under ideal conditions. This is intended as an estimator for the minimum cost function value under optimal control.

The sigma function sums the outputs from the observer and the ideal system model. This summation indicates a degree of divergence of the existing operation from the optimal solution. The divergence is termed as error and provides impetus for correction.

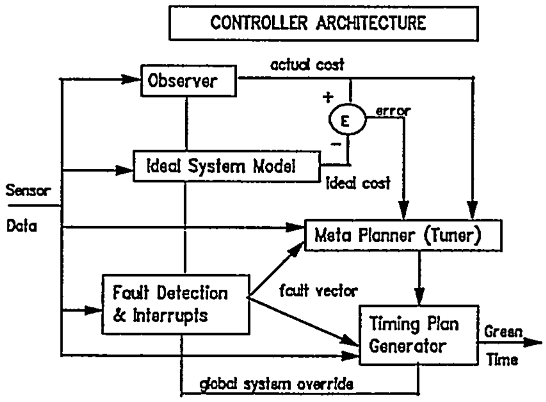

Figure 6 Controller architecture

The Meta-Planner (Tuner) decides whether a change in the timing plan is needed. To make the decision, four inputs are received by the Meta-Planner: 1) actual cost 2) error 3) sensor data and 4) fault vector. The decision will be a rule-based process. The process will accommodate faults and interrupts. The output of the Meta-Planner is a reinforcement vector. The reinforcement vector defines whether a change in the timing is needed.

The timing plan generator is the most complex element of the architecture. The timing plan generator receives inputs from the sensor data, the volumes and delay, the reinforcement vector from the Meta-Planner, the fault vectors and system overrides. These data are used by the timing plan generator to output the green times. The timing plan generator will be rule-based in its elementary form. A neural network may be used to modify the base plan.

6.1. FUZZY RULES FOR CONTROL

The following rules present the elementary rules for initiating signal change. These rules are for an intersection presented in Figure 7. Figure 8 elaborates the phase sequencing process for the controller. The rules will be generalized for an N-approach intersection. Also, rules will be added for turn lanes, platoons of vehicles, and for foot traffic. Provision will be made for emergency vehicles and rail road crossings. The following rules provide an example:

IF EW average is low and NS average is low, THEN bias is equal.
IF EW average is low and NS average is high, THEN bias is low.
IF EW average is high and NS average is low, THEN bias is high.
IF EW average is high and NS average is high, THEN bias is equal.

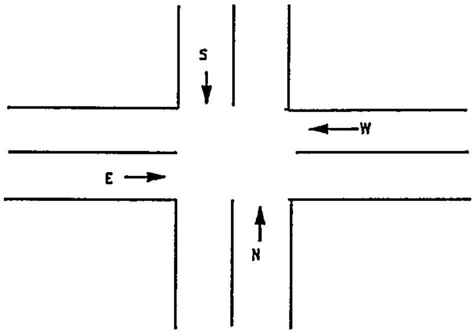

Figure 7 Typical intersection

where average is taken over several cycles.

IF cost is high, THEN frequency is high.
IF cost is low, THEN frequency is low.
IF EW average is low and NS average is low
and
sum EW is high and sum NS is low and state is zero, THEN state is one.
sum EW is high and sum NS is high and state is *, THEN state is *.
sum EW is low and sum NS is high and state is one, THEN state is zero.
sum EW is low and sum NS is low and state is *, THEN state is *.

where sum is cleared as state changes.

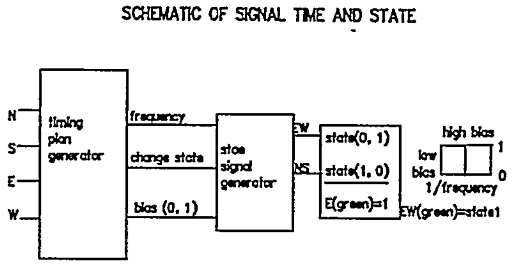

Figure 8 Schematic of time and state change

7. PROTOTYPE

The prototype controller will be specially configured personal computer. The processor board will be a 486 processor with a 32 bit bus. The construction will be a tower case with data acquisition boards and communication boards. The data acquisition boards will be digital boards. Also, the boards will have 64 digital inputs and 64 digital outputs. The boards will be PCI-Bus compatible. The construction will allow hardware interrupts. A sixteen megabyte memory is provided for data buffers. The communication will be either a network or a hi-speed modem. The speed will be INTERNET baud rate or TS/2 baud rate. It should be noted that a production controller will not need all the facilities of a prototype.

8.0 FUTURE RESEARCH

Future research will focus on testing the control logic on more complex intersections and to carry the concepts onto a controller prototype. The controller prototype will be built and field tested at the Ada County Highway District in Boise, Idaho. Refining the control algorithms to work on platforms of interest will require new effort. Also, complementing the findings of related research is expected to accelerate the practicality of the fuzzy controller.

ACKNOWLEDGEMENTS

Work supported through the INEL Laboratory Directed Research and Development Program under DOE Idaho Operations Office Contract DE-AC07-94ID13223.

9. REFERENCES

[1] Kagolanu, K., *A Comparative Study of Traffic Control Systems*, Dallas, ITE Annual Conference, 1994.
[2] Bisset, K.R., Kelsey, R.L., *Simulation of Traffic Flow Using Conventional, Fuzzy, and Adaptive Methods*, York, European Simulation Multiconference, 1992.
[3] Johnson, J. A., Smartt, H.B., *Advantages of an Alternative Form of Fuzzy Logic*, IEEE Transactions on Fuzzy Systems, Vol 3, No 2, 1995.
[4] Nakatsuji, T., Kaku, T., *Development of a Self-Organizing Traffic Control System Using Neural Network Models*, Washington D.C., TRB, 1992.
[5] Canu, S., Sorbal, R., Lengelle R., *Formal Neural Network as an Adaptive Model for Water Demand*, Paris, International Neural Network Conference, July 1990.
[6] Srirengan, S., Looi, C. K., *On Using Back Propagation for Prediction: An Empirical Study*, Singapore, International Joint Conference on Neural Networks, 1991.