This is a pre-publication draft of the paper in Engineering Applications of Artificial Intelligence V. 25(1) pp. 107-118 (2012), doi:10.1016/j.engappai.2011.09.008

An automated signalized junction controller that learns strategies from a human expert

Simon Box and Ben Waterson

Transportation Research Group, Faculty of Engineering and the Environment, University of Southampton, UK, SO17 1BJ.

Abstract

An automated signalized junction control system that can learn strategies from a human expert has been developed. This system applies Machine Learning techniques based on Logistic Regression and Neural Networks to affect a classification of state space using evidence data generated when a human expert controls a simulated junction.

The state space is constructed from a series of bids from agents, which monitor regions of the road network. This builds on earlier work, which has developed the High Bid auctioning agent system to control signalized junctions using localization probe data. For reference the performance of the Machine Learning signal control strategies are compared to that of High Bid and the MOVA system, which uses inductive loop detectors.

Performance is evaluated using simulation experiments on two networks. One is an isolated T-junction and the other is a two junction network modelled on the High Road area of Southampton, UK. The experimental results indicate that Machine Learning junction control strategies, trained by a human expert can outperform High Bid and MOVA both in terms of minimizing average delay and maximizing equitability; where the variance of the distribution over journey times is taken as a quantitative measure of equitability. Further experimental tests indicate that the Machine Learning control strategies are robust to variation in the positioning accuracy of localization probes and to the fraction of vehicles equipped with probes.

Keywords: , Intelligent Transportation Systems, Signal Control, Machine Learning, Neural Network, Traffic Control

1. Introduction

This paper describes the development of a Machine Learning junction control system that employs pattern matching techniques including logistic regression and neural network classification to find statistical trends between the signal control decisions made by the human expert and the state of a simulated network as described by simulated localization probe data.

1.1. Background

In the United Kingdom urban signalized road junctions are usually controlled by one of two systems, MOVA (Vincent and Peirce, 1988) for use at isolated junctions and SCOOT (Hunt et al., 1982), which can coordinate multiple adjacent junctions. Both these systems use sensors, including inductive loops (Sreedevi, 2005) and microwave emitter/detectors (Wood et al., 2006), to detect the presence of vehicles at fixed locations on the roads around the junction. The data from these sensors are used as a descriptor of the state of the network by the control algorithms to inform decisions on which colour to set the traffic lights.

Data collected from counts of vehicles at fixed locations is called *census* data. Previous reviews (e.g. (Rose, 2006)) have suggested that *localization probe* data, that is dynamic position and speed data from on board vehicle sensors, can present a different view of the state of the network. The European Commission has recently invested significant resources in three major studies to look into the benefits of vehicle to infrastructure (V2I) and vehicle to vehicle (V2V) communications (Kompfner, 2008; COOPERS, 2010; SAFESPOT, 2010). Furthermore common European protocols have been set for this type of communication (IEEE 802.11p). This has laid the ground for this technology to become commonplace in Europe in the near future. This technological advance would enable localization probe data to be collected and employed in urban traffic control (UTC) systems.

1.2. Context and Motivation

Early work to investigate the scenario of localization probe data in signal control has employed simulation to develop and evaluate control methods. Box and Waterson (2010a) presents an *auctioning agent* control method. This work showed the auctioning agent approach outperforming MOVA in simulations on an isolated T-junction. In Waterson and Box (2011) the auctioning agent system was subjected to a rigorous quantitative stochastic analysis, which characterized the effects of varying the positioning accuracy and the fraction of vehicles equipped with probes.

The simulation test bed used for the work presented in this paper is described in detail in Box and Waterson (2010a); Waterson and Box (2011). In summarry: it uses S-Paramics microsimulation software to model networks and simulate the movement of individual vehicles through junctions. Built around this are a number of bespoke software modules for simulating localization probe data, making control decisions, and implementing control directly in the simulation.

Box et al. (2010) showed that a human interface layer can be connected to the simulation test bed allowing an expert human to control the signals at simulated junctions. Results indicated that an expert human controller can outperform both MOVA *and* the auctioning agent approach from Box and Waterson (2010a) in terms of delay across the junction.

This motivates the development of Machine Learning junction control systems that can mine the data generated when a human expert controls simulated junctions and emulate human control strategies under automated control. Box et al. (2010) also demonstrated how the auctioning agent method could be adapted, employing the pattern recognition technique of Logistic Regression, to create a *learning junction agent*.

In this paper both the auctioning agent system and the learning agent are developed further. The principal contributions are as follows:

1. An updated structure for the auctioning agent method introducing the *lane agent*.
2. A new learning junction agent, which employs a two layer neural network with back propagation to learn strategies from a human expert.
3. A comparison between the logistic regression and neural network learning junction agents including variation in the resolution of the training data.
4. Simulation tests carried out on a two junction network, which models the High Road area of Southampton, UK.

Other important work where pattern recognition and Machine learning techniques have been applied to junction control include (Choy et al., 2003; Mikami and Kakazu, 1993; Chen and Heydecker, 2009). This work has shown how to use neural networks and other techniques to optimize certain parameters in signal control strategies or to select pre-defined strategies. In the work presented here, Machine learning techniques are used to select signal control decisions by directly classifying state space using evidence data generated by a human expert (Section 3).

2. Signal Control Strategies Overview

Simulation tests were carried out on two network models. Figure 1 shows a view of the first, the Simple T-junction. This is an isolated junction with three signal stages. Figure 2 shows a view of the second, the High Road Junction. This is a model of the High Rd area of Southampton, UK. It consists of two signalized junctions a short distance apart. The westerly junction has four signal stages and the easterly junction has three.

This section presents an overview of the junction control strategies that were investigated using simulations on these junctions. These are: MOVA, Auctioning Agents using the High Bid method and Auctioning Agents using the Learning Junction Agent.

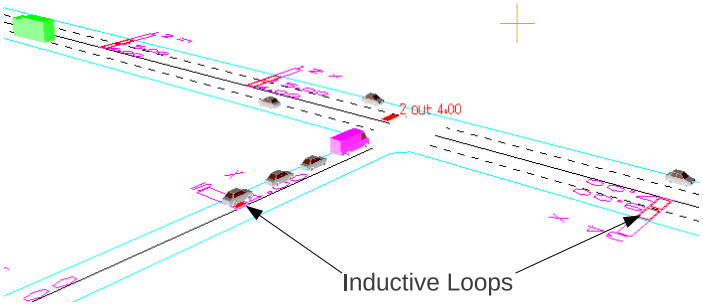

Figure 1: S-Paramics screenshot of the Simple T-Junction simulation model used in simulation tests. Incudctive loop locations are marked by arrows.

2.1. MOVA

The MOVA control strategy (Vincent and Peirce, 1988) is a common strategy that is employed on many isolated junctions in the real world, therefore it is used as a baseline for junction control performance in these tests. The MOVA control strategy was tested on the Simple T-junction only. The S-Paramics simulation models 11 inductive loop detectors (examples are marked in Figure 1), which measure counts of vehicles passing over. The signals from these detectors serve as inputs to the MOVA algorithm. In fact the Simple T-junction model was designed by The Transportation Research Laboratory (TRL) as an exemplar for MOVA control to accompany their MOVA–S-Paramics API. The full details of the MOVA set up for this model are given in TRL (2007).

2.2. Auctioning agents

The auctioning agent system defines the state of the network using *bids*. Sections of road or lane in the network models are monitored by *Lane agents*. A lane agents receives data from vehicles in the simulation whose reported position indicates that they are in that agent's section. The lane agent then uses the vehicle data to generate a *bid*. In simple terms the bid is designed to be a measure of the need for priority at the next junction coming from that section of road. Equation (1) below shows how the bid B is calculated.

$$B = \sum_{c \in C} 1 - \alpha V_c - \beta X_c \tag{1}$$

Here C is the set of all vehicles monitored by the lane agent. V_c is vehicle speed and X_c is the distance of the vehicle from the junction. α and β are coefficients which can be tuned to adjust the relative influence that the number of vehicles, the vehicle speed and the vehicle distance each have on the size of the bid. In previous work (Box and Waterson, 2010b) it has been shown that good values

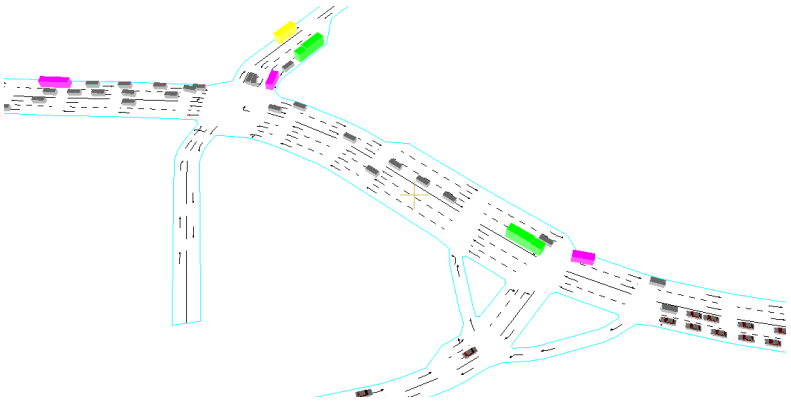

Figure 2: S-Paramics screenshot of the High Road Junction simulation model used in simulation tests

are $\alpha = 0.01$ and $\beta = 0.001$ (assuming SI units for all the terms). These values are adopted in this work.

The lane agents fit into a larger hierarchical agent structure shown in Figure 3. Above the lane agents are stage agents, one for each stage of each junction. Stage agents receive data from any lane agents whose lanes receive the green light from that stage. As an example: Figure 4 shows a simple network containing two junctions, each with the same two-stage structure. Considering stage 1 of the West junction: the lane agents assigned to this stage are labelled (a) and (b). It is possible for a lane agent to be assigned to more than one stage agent (See Figure 3 and Section 4).

High Bid Control. Each stage agent generates an initial bid, which is simply the sum of the bids of all its lane agents. Above the stage agents in the hierarchy is the Junction Agent. The simplest type of control that the junction can employ is *High-Bid* control. In this scenario the Junction Agents requests bids from the stage agents at a pre-set time interval called the auctioning rate (δt). upon receipt of the bids the junction simply gives the green light to the stage with the highest bid.

Coordinated High Bid Control. In the case where two junctions are closely spaced, as in Figure 4, it can be advantageous to coordinate their control. The High Bid control method can be extended to control such junctions with the addition of a zone agent above the Junction agents in the hierarchy (Fig 3). Each junction agent sends the zone agent their highest initial stage bid. The zone agent then picks the junction with the highest overall stage bid to lead the coordination. This means that the winning junction continues as normal and assigns priority to the stage with the highest bid. The zone agent then assigns

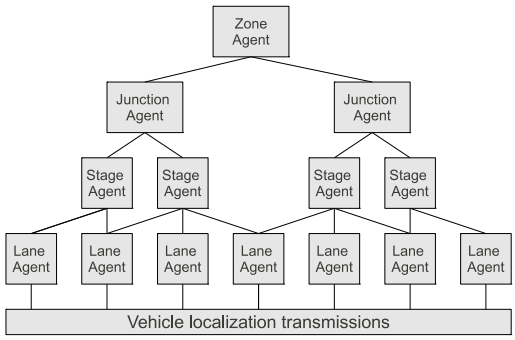

Figure 3: Hierarchy of agents in the auctioning agent control system

some of the lane agents from the winning junction to the stage agents of the losing junction to encourage coordination.

An example of this is shown in Figure 4. In this example The East junction is the winning junction and East stage 1 is the winning stage. If controlled independently then the West junction's stage 1 agent receives bids from lane agent (a) and lane agent (b) (as discussed above). However in the coordinated case lane agent (c) is added to this stage also. In this case lane agent (c) recalculates its bid based on the longer distance to the West junction and the bid is also multiplied by a pre-set coefficient to account for the fact that some of the vehicles in this section will be turning left. The effect of this will be to increase the value of the bid for stage 1 of the West junction to take into account the vehicles that are going to be released by the East junction. That is the principle coordinated high bid control. Detailed descriptions of this control method applied to specific junctions are given in Waterson and Box (2011).

2.3. Machine Learning Auctioning Agents

The auctioning agent system can be adapted, using a machine learning approach, to include a *learning junction agent* that can be trained in control strategies by a human expert. The architecture of the auctioning agent hierarchy (Figure 3) is largely retained although the zone agent is discarded because coordination can be achieved with junction agents operating independently (Figure 5).

Human Interface. The human interface to the simulator is both a junction control method itself and a tool for training the learning junction agent. The human interface consists of a screen and keyboard. The screen displays a realistic 3D scene of the network with simulated vehicles driving around. As with high bid

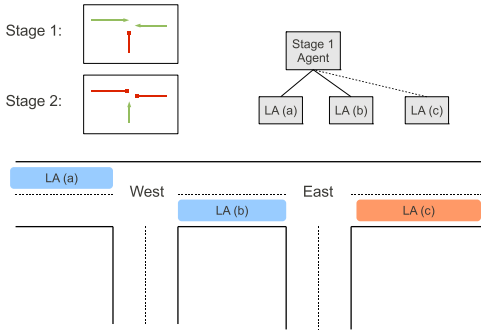

Figure 4: Schematic of a simple twin T-junction indicating the two junction stages (top left), which apply to both the East and West Junctions. Also highlighted are sections of that are monitored by lane agents (bottom). When under coordinated control lane agents can be reassigned from one junction to another (top right).

control, auctions are called at regular intervals (δt). Whenever an auction is called the simulation is paused and the user is prompted on screen to assign priority to one stage for each of the junctions in the simulation. Thus the human user controls the simulated junction. While human control is taking place the lane and stage agents are calculating bids. After each auction the bids and the stage decisions made by the human are stored in a database for offline training that is described in Section 3.

State Descriptor. Unlike high bid control where the stage agent's bids are considered to be proxies for the need for priority, the learning junction agent considers the bids from the stage agents simply as a description of the state of the network. Because of this it is possible for the learning junction agent to consider the bids from the lane agents directly, bypassing the stage agents altogether (see Figure 5). Because there are generally more lane agents than stage agents they can provide a potentially richer description of the state of the network, although this is at the expense of increasing the dimensionality of the problem to be solved (see section 3). In the experiments presented in Section 4 both the stage agent bids and the lane agent bids are used to describe the network. We adopt the terminology of referring to the stage agent bids as the *short* bid data and the lane agent bids as the *long* bid data. Interestingly although the bids are no longer used as proxies for the need for priority the fact that they are designed as such has important implications for the solution of the learning problem that will be described in Section 3.

Junction Coordination. The learning junction agent is not restricted to considering agents that belong only to that junction but can also consider those of neighbouring junctions. If the human expert who trains the junction agents

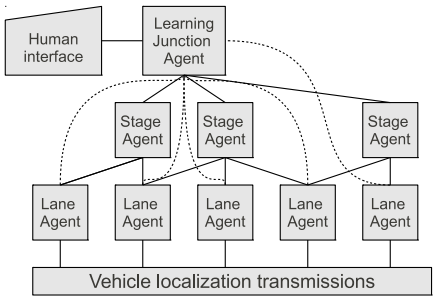

Figure 5: Agent hierarchy for Machine learning auctioning agents

attempts to coordinate them in her decision making then this should result in patterns between bids and coordinated decisions. Therefore, in the experiments presented here, there is no explicit coordination of junction agents, rather they operate independently and if the patterns are captured by the learning junction agents then coordination should emerge under automated control.

3. Learning Agent

There is a number: $(J-1)$ of agents that are used to describe the state of the network for the learning junction agent. The bids from these agents define a $J-1$ dimensional *bid space*. Each new set of bids in the future will define a point in this bid space. In order to be able to assign all possible future combinations of bids to a given signal stage, we need to divide this bid space up into regions that correspond to signal stage decisions.

Such a division of bid space can be done using the data generated by the human expert controller. The expert's decisions represent patterns between points in bid space and signal stage decisions, therefore they can be used as evidence for how the bid space should be divided. In the field of Machine Learning this is a standard classification problem and there are a number of ways to approach it (for examples see Bishop (2006)). In this paper we will be looking at two: The *multi-class logistic regression* and the *two layer neural network with back propagation*. Both approaches are closely related as will be explained below.

3.1. Multi-class Logistic Regression

The approach of multi-class logistic regression is to fit a probability function to the bid space for each of the K possible stage decisions. Thus for each new point in bid space we can determine the probability that this point belongs to stage k. The function used here is the *softmax* function (2)(Bishop, 2006).

$$p(k|\mathbf{b}) = \frac{\exp(a_k)}{\sum_{k \in K} \exp(a_k)} \tag{2}$$

where a_k is a linear function of the vector of bids $\mathbf{b}$.

$$a_k = \mathbf{w}_k^{\mathrm{T}} \boldsymbol{b} \tag{3}$$

where $\boldsymbol{b} = [b_1, b_2, \cdots, b_{j-1}, 1]$ and $\mathbf{w}_k$ is a vector of parameters. The learning task is to find the values of these parameters that minimize the error between the probability surface defined by (2) and the evidence data provided by the human expert. The probability given by (2) will be valid if the distribution of evidence points in bid space for a given stage k conforms to any probability distribution in the exponential family (e.g. Gaussian) (Bishop, 2006).

3.1.1. Learning the parameters

In this section we show how to learn the parameters $\mathbf{W} = \{\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_K\}$ from the expert training data. For each pattern $\mathbf{b}_n$ in the set of N patterns we can define a target vector $\mathbf{t}_n$ which has K elements t_{nk}, which equal 1 if $\mathbf{b}_n$ is associated with stage k and 0 otherwise.

A measure of the *difference* between the probability functions and the evidence data is given by the cross-entropy error function.

$$E(\mathbf{W}) = -\sum_{n \in N} \sum_{k \in K} t_{nk} \ln p(k|\mathbf{b}_n) \tag{4}$$

The values of $\mathbf{W}$ that minimize $E(\mathbf{W})$ are found numerically using the Newton-Raphson update formula.

$$\mathbf{W}^{\mathrm{new}} = \mathbf{W}^{\mathrm{old}} - \nabla\nabla E(\mathbf{W}^{\mathrm{old}})^{-1} \nabla E(\mathbf{W}^{\mathrm{old}}) \tag{5}$$

Where the second and third terms are respectively: the second and first derivatives (Hessian matrix and gradient) of the cross entropy error function. The precise method for calculating these is given in Box et al. (2010). This approach is known as iteratively re-weighted least squares. $\mathbf{W}$ is initialized randomly using the approach recommended by (Nabney, 2002) shown below.

$$w \sim \mathcal{N}\left(0 \left| \frac{1}{\sqrt{N+1}} \right.\right) \tag{6}$$

Typically the above algorithm is run a number of times to avoid a result in a poor local minimum.

3.1.2. Applying the logistic regression

Once the values of the parameters $\mathbf{W}$ have been learned, the probability that a new point in bid space belongs to stage k can be found using (2). Thus a region in bid space where stage k has the highest probability can be defined. This region will be separated from neighbouring regions by $J-2$ dimensional quadratic boundaries where both stages have equal probability. This approach to classification can work well where the evidence points for each stage are well separated into regions that can be partitioned by such boundaries. In fact this is why the bids, which are designed to represent need for priority, are particularly suitable descriptors of the state of the network in this case. If they are at all representative of the need for priority then the points for different stages *will* be well separated in bid space. In cases where the separation is less clear a more powerful approach may be needed, such as the two layer neural network.

3.2. Two-layer Neural Network with back propagation

The neural network approach allows a more flexible division of the bid space where stage k can exist in multiple regions separated from neighbouring regions by $J-2$ hyper-surfaces that can have complex shape. The neural network achieves this by applying a transformation function to the bid space that arranges the evidence points for each stage into positions that are more favourable for a logistic regression. That is to say that after the transformation the softmax function fits the evidence data with lower error that it would if it were applied before the transformation. This transformation function can be parametrized and the parameters learned from the evidence data in the same way as the parameters for the softmax function are learned in the logistic regression.

The standard structure for the neural networks used here is a two layer network with J input units, H hidden units, and K output units. Here J is the number of bids plus one, K is the number of signal stages and the number of hidden units H is a variable that can be tuned to control the complexity of the transformation. A low number of hidden units can limit the complexity of the transformation giving a poor fit to the evidence data. A high number of hidden units allows more complexity in the transformation but can lead to over fitting of the data.

3.2.1. Forward Propagation

The J dimensional bid vector $\mathbf{b}$ is defined as in Section 3.1. For each pattern there is also a K dimensional target vector $\mathbf{t}$ with elements $t_k \in \{0,1\}$, where $\sum_{k\in K} t_k = 1$.

Layer 1. The first layer of the neural network forward propagation transforms the bid vector $\mathbf{b}$ onto the H dimensional *hidden units* vector $\mathbf{z}$ using the following transformation.

$$z_h = \tanh\left(a_h^{(1)}\right) \tag{7}$$

$$\mathbf{a}^{(1)} = \mathbf{W}^{(1)}\mathbf{b} \tag{8}$$

where $\mathbf{W}^{(1)}$ is a $H \times J$ matrix of parameters (to be learned).

Layer 2. The second layer of the neural network forward propagation substitutes the new transformed bid $\mathbf{z}$ into the softmax function.

$$y_k = \frac{\exp(a_k^{(2)})}{\sum_{k \in K} \exp(a_k^{(2)})} \tag{9}$$

$$\mathbf{a}^{(2)} = \mathbf{W}^{(2)}\mathbf{z} \tag{10}$$

where $\mathbf{W}^{(2)}$ is a $K \times H$ matrix of parameters to be learned.

3.2.2. Error back propagation

The error E between the output of the neural network $\mathbf{y}$ and the target vector $\mathbf{t}$ is measured using the cross-entropy error function.

$$E = -\sum_{k \in K} t_k \ln y_k \tag{11}$$

We define the K dimensional error vector $\boldsymbol{\delta}^{(2)}$ using

$$\delta_k^{(2)} = \frac{\partial E}{\partial a_k^{(2)}} = y_k - t_k \tag{12}$$

This error is back propagated to the first layer of the neural network using (13) to give the H dimensional first layer error vector $\boldsymbol{\delta}^{(1)}$.

$$\delta_h^{(1)} = \frac{\partial E}{\partial a_h^{(1)}} = (1 - z_h^2)\mathbf{w}_h^{(2)\mathrm{T}}\boldsymbol{\delta}^{(2)} \tag{13}$$

where $\mathbf{w}_h^{(2)}$ is the h^{th} column of $\mathbf{W}^{(2)}$. The gradient of the error surface with respect to the weights in each layer is given by

$$\nabla E(\mathbf{W}^{(2)}) = \boldsymbol{\delta}^{(2)}\mathbf{z}^{\mathrm{T}} \tag{14}$$

$$\nabla E(\mathbf{W}^{(1)}) = \boldsymbol{\delta}^{(1)}\mathbf{b}^{\mathrm{T}} \tag{15}$$

$$\tag{16}$$

3.2.3. Learning the parameters

The forward propagation and back propagation of error described above relates to a single pattern in the evidence data. To calculate the error and gradient for the whole data set of N patterns we simply sum them.

$$\nabla E(\mathbf{W}) = \sum_{n \in N} \nabla E_n(\mathbf{W}) \tag{17}$$

The values of the parameters that minimize the error E can be learned by gradient descent using the following iteration step.

$$\mathbf{W}^{\mathrm{new}} = \mathbf{W}^{\mathrm{old}} - \eta \nabla E(\mathbf{W}) \tag{18}$$

where η is the *learning rate* that can be adjusted dynamically during the gradient descent to control step size (e.g. reduced if error increases).

The elements of $\mathbf{W}$ are initialised randomly using (6) and in the tests presented in this paper the gradient descent is performed 20 times and the result with the lowest error is selected to avoid results in a poor local minimum.

4. Simulation Experiments

Simulation tests, designed to evaluate the junction control strategies described in Sections 2 and 3, were carried out on the two networks shown in Figures 1 and 2.

All simulation tests covered a simulated four hour period. During the tests the junctions in the simulations were controlled either by MOVA or by one of the auctioning agent algorithms. Auctioning agent tests used either the High Bid method, or the Learning agent method and the auctioning rate was $\delta t = 10\,\mathrm{s}$ in all cases. Learning agents were trained (Section 4.4) using either the Logistic regression (logit) method or the two layer neural network with back propagation (neural net). Finally for each of these training methods either the Stage agent data (short data) were used to describe the network state or the Lane agent data (long data) were used.

4.1. Demand Profile

Each experiment used the same demand profile. Demand is set between origin and destination zones, these are at the end of each modelled road. The demand between zone i and and zone j (D_{ij}, in vehicles per minute) is a function of the basic demand between i and j $d_{i,j}$ and a transient demand multiplier d_t, which is a function of the simulation time (equation (19)).

$$D_{i,j} = d_{i,j} d_t \tag{19}$$

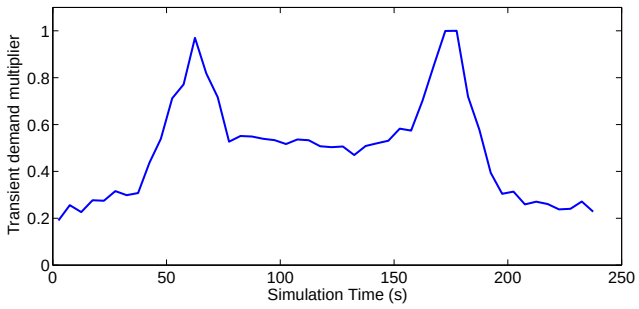

Figure 6: Variation of the transient demand multiplier throughout four hour simulation period

The transient demand multiplier causes the level of demand to vary over the time of the simulation. Under normal operation a junction will usually

experience two peaks in demand over a 24 hour period i.e. in the morning and evening rush hours. To conserve simulation time all simulation tests presented here were carried out over a four hour period. This representes a "compressed day" with peaks in demand at 1 hour and 3 hours. Figure 6 shows the profile of the demand multiplier over the test period.

The basic demands represent the permanent trend in demand between particular origins and destinations. The basic demand matrices for the Simple T-junction and the High Road junction are given in tables 2 and 4 respectively.

4.2. Delay Calculation

During the simulations S-Paramics records detailed information about the journeys of every simulated vehicle. In the analysis presented here the main measurements used are journey time t and delay θ. For a given vehicle p the time it takes to travel from its origin i to its destination j is its journey time t_p. The vehicle's free flow travel time $t_p^{(\mathrm{ff})}$ is the theoretical time that it would take to travel between i and j if it were unimpeded by other vehicles or red signals. The delay for vehicle p is the difference between these two times.

$$\theta_p = t_p - t_p^{(\mathrm{ff})} \tag{20}$$

4.3. Test Junctions

4.3.1. Simple T Junction

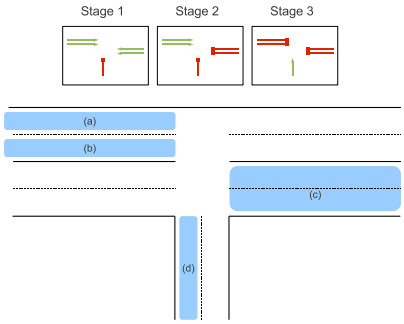

Figure 7: Schematic (*not to scale*) of the Simple T-Junction. Showing the different priorities of the junction's three stages (top) and the sections of road monitored by the four lane agents (a) - (d) (bottom).

Figure 1 shows a view of the model of the Simple T-junction. Figure 7 shows a schematic of the junction indicating the staging and the road sections that are monitored by lane agents.

Agents. Figure 7 shows that there are four lane agents in this network labelled a, b, c and d. Each lane agent section is indicated and these sections cover the entire length of the modelled road. Table 1 shows which lane agents are assigned to which stage agents in the auctioning agent hierarchy.

Stage Agent	Assigned lane agents
1	{a,c}
2	{a,b}
3	{d}

Table 1: Assignment of lane agents to stage agents in the Simple T-junction model

	W	E	S
W	–	18.96	5.0
E	24.01	–	1.26
S	4.05	4.05	–

Table 2: Basic demand matrix (vehicles per minute) for the Simple T-junction network

Demand. The basic level of demand d_{ij} in vehicles per minute between the origin and destination zones in the Simple T-junction model are shown in table 2.

4.3.2. High Rd Junction

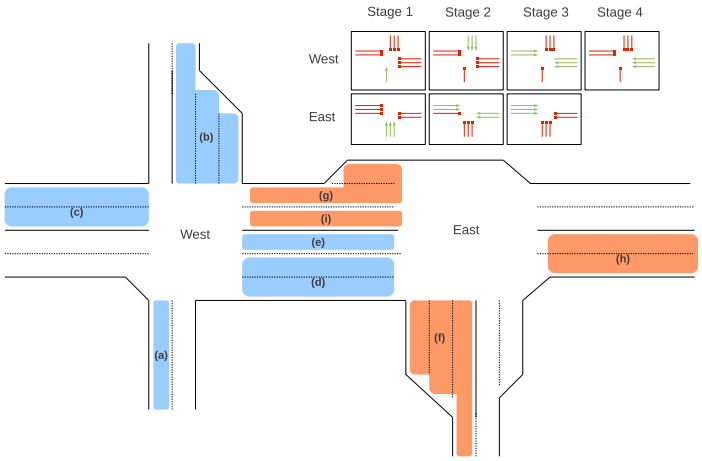

Figure 8: Schematic (*not to scale*) of the High Rd Junction. Showing the different priorities of the East and West junction's stages (top right) and the sections of road monitored by the nine lane agents (a) - (i) (bottom).

Figure 2 shows a view of the model of the High Road junction. Figure 8 shows a schematic of the junction indicating the staging and the road sections that are monitored by lane agents.

Agents. Figure 8 shows that there are nine lane agents in this network labelled a to i. Each lane agent section is indicated and these sections cover the entire length of the modelled road. Table 3 shows which lane agents are assigned to which stage agents in the auctioning agent hierarchy.

In the case of coordinated high bid control agent reassignment takes place following the principle outlined in Section 2. A detailed description of agent re-

assignment for a similar two junction network model can be found in (Waterson and Box, 2011).

In the case of Machine learning control there are two learning junction agents and both of these agents receive data from all of the stage/lane agents in the network.

Junction	Stage Agent	Assigned lane agents
West	1	{a}
	2	{b}
	3	{c,d}
	4	{d,e}
East	1	{f}
	2	{g,h}
	3	{g,i}

Table 3: Assignment of lane agents to stage agents in the High Road junction model

	W	SE	E	N	SW
W	–	4.81	8.76	1.95	0.19
SE	1.95	–	3.89	1.95	0.049
E	8.76	6.89	–	0.97	0.097
N	2.92	0.97	2.92	–	0.097
SW	0.097	0.097	0.097	0.097	–

Table 4: Basic demand matrix (vehicles per minute) for the High Road junction network

Demand. The basic level of demand d_{ij} in vehicles per minute between the origin and destination zones in the High Road junction model are shown in table 4.

4.4. Training Phase

Training of the learning junction agent took place in a separate series of simulations called the *training phase*. These simulations were set up slightly differently from the test simulations described above.

Training sessions were short (30 simulated minutes) with a constant level of demand. For each network (Simple T and High Rd) there were six training sessions, each with a different level of demand. The aim of this approach was to minimize the amount of training time required and make sure that the learning junction agent had equal exposure to data generated at different levels of demand. The total demand between each origin i and destination j was calculated using (21)

$$D_{i,j} = d_{i,j} d_x \tag{21}$$

where d_x is the demand multiplier for test x. Table 5 shows the demand multipliers used in each of the six training sessions.

Number of hidden units. As discussed in Section 3.2, when using the neural network approach to learn strategies the complexity of the fit can be controlled by varying the number of hidden units used in the neural network. Table 6 shows the number of hidden units used for each of the neural networks employed in these tests. These values were selected based on best performance on new simulated data.

Test number (x)	Demand multiplier (d_x)
1	0.2
2	0.4
3	0.6
4	0.8
5	1.0
6	1.2

Table 5: Basic demand matrix (vehicles per minute) for the Simple T-junction network

Network	Agent set	Number of Hidden units
Simple T	Long	7
	Short	5
High Rd	Long	11
	Short	8

Table 6: The number of hidden units used in the various test combinations

5. Test Results

	Delay (s)	
Strategy	Simple-T	High Rd
MOVA	18.64	–
High Bid	17.23	19.45
Logit (Short)	18.10	90.14
Logit (Long)	15.46	83.26
Neural (Short)	15.48	17.86
Neural (Long)	14.12	17.44

Table 7: Delay averaged over all journeys during 4-hour simulation tests on the two test junctions, using various control strategies.

5.1. Simple T-junction

This section presents results from simulation tests carried out on the Simple T-junction model. Five different control strategies were used for the tests presented here: MOVA, High Bid, Logit (using both long and short bid vectors) and Neural network control (using both long and short bid vectors).

The second column of Table 7 compares the measured value of delay for the six strategies tested. Here delay is averaged across all journeys for the duration of the test. This shows that MOVA, the baseline strategy, is outperformed by all other strategies.

The High bid strategy outperforms MOVA, despite its simplicity, due to the fact that the localization probe data provide a richer description of the state of the network than the inductive loop data. Of the Machine Learning strategies tested, the Logit algorithm using the short bid vector has the lowest performance. Previous research has shown this strategy outperforming High Bid on an isolated T-junction and matching human performance (Box et al., 2010). The main difference between the tests presented here and the results

previously published is the varying demand profile. In the previous tests demand was constant during training and tests. Introducing the profiled demand has increased the complexity of the problem leaving the Logit (short) strategy less effective. When using the Long bid vector the logit algorithm acts on a bid space with higher dimensionality ($n = 4$ vs $n = 3$) and is able to capture some of this higher complexity. The Logit (Long) strategy *does* outperform High Bid. The Neural Network strategies can capture more complexity in the training data than the Logit strategies due to greater flexibility in the division of the bid space. Neural network control using the short bid vector matches the performance of Logit (Long), while Neural (Long) is the best performing strategy with a $\sim 25\%$ reduction in delay over the MOVA baseline.

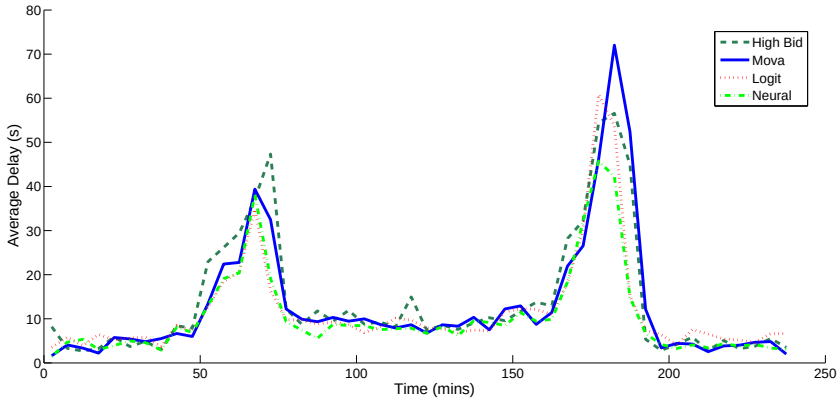

Figure 9: Delay, averaged over successive 5 minute intervals, for the duration of the four hour simulation test on the Simple T-junction. Traces for four strategies are shown: MOVA, High Bid, Logit (Long) and Neural Network (Long).

9 shows the transient delay traces for four strategies (MOVA, High Bid, Logit (Long) and Neural Network (Long)). Here delay is averaged across all journeys that ended within a five minute period, thus indicating the variation in delay over the test period. These results show that the performance of the strategies is similar throughout most of the test, with the exception of the two peaks in demand at 1 hour and 3 hours. This indicates that the algorithms that perform better do so because they deal better with high levels of demand and can postpone or avoid the situation of saturated flow, which leads to high levels of delay.

While delay is the single most informative metric on the performance of the signalized junction it does not tell the full story. It is also important that the junction is equitable. For example a strategy that holds a single vehicle at a red light for 1 hour could still achieve a low average delay if significant numbers of other vehicles are passing through the junction with low delays. However this scenario is not acceptable in real world because it is unfair to the occupants

of the stationary vehicle. This is of particular concern for the High Bid and Machine Learning strategies, which do not explicitly consider temporal effects in their decision making.

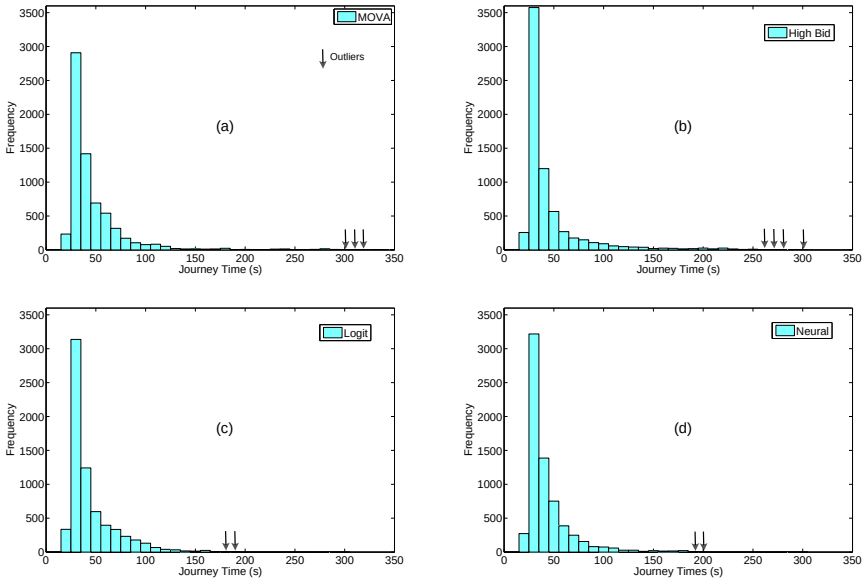

Figure 10: The distribution of journey times over the four hour simulation tests on the Simple T-junction for four control strategies: MOVA, High Bid, Logit (Long) and Neural Network (Long).

Figure 10 shows the journey time distribution for four of the strategies tested (MOVA, High Bid, Logit (Long) and Neural (Long)). The distribution for MOVA shows the peak journey time is in the $30 - 40$ s range with significant numbers of journeys up to 120 s. There are also outlier journeys up to 300 s in length. The distribution for the high bid strategies indicates that it is outperforming MOVA by getting more journeys into the peak $30 - 40$ s range. However the fact that temporal effects are not considered in the High Bid algorithm shows in the fact that there are significant numbers of journeys with times up to 150 s. There are also outliers up to 300 s. The Machine Learning algorithms are not explicitly given any temporal information, however the human expert who generates the training data will be considering temporal effects when making decisions. Therefore this may be captured by the machine learning algorithms if there are patterns relating these decisions to the instantaneous state of the network. The distributions for the Logit and Neural network strategies show that they have less journeys in the peak $30 - 40$ s range than High Bid, but the tails of the distribution are tighter with less outliers and no journeys longer than 190 s. The distributions show that the Machine Learning strategies, which are the best in terms of delay, are also the most equitable strategies.

5.2. High Rd Junction

The following sections present results from tests of the control strategies carried out on the High Road junction model. The third column of Table 7 shows delay, averaged across all journeys, for five tests using different control strategies (High Bid, Logit(short), Logit(Long), Neural (Short) and Neural (Long)). These results show high delays for the strategies using the Logit algorithm, indicating poor junction control. This is because the division of the $J-1$ dimensional bid space into regions separated by $J-2$ quadratic boundaries is insufficient to capture the complexity of the relationship between the network state and the human controller's decisions in the more complex High Road junction system. However the neural network strategies *can* capture some of the complexity of the human decision making process on the High Road junction and these algorithms outperform High Bid. The best performance is achieved by neural network control using the long bid vector, with a reduction in delay of $\sim 10\%$ over High Bid.

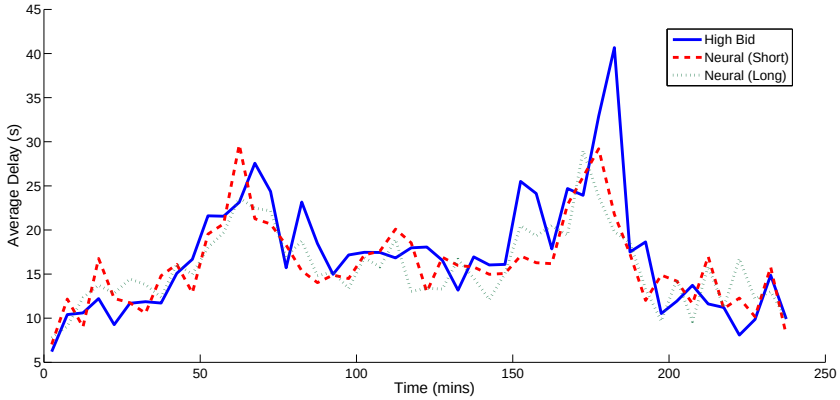

Figure 11: Delay, averaged over successive 5 minute intervals, for the duration of the four hour simulation test on the High Rd junction. Traces for three strategies are shown: High Bid, Logit (Long) and Neural Network (Long).

Figure 11 shows transient delay traces for three strategies (High Bid, Neural (Short) and Neural (Long)). Here the results show the same trend that was seen with the Simple-T results. The better performing strategies show lower delay mainly at the two peaks in demand at 1 hour and 3 hours. This indicates that these strategies achieve their performance advantage by better controlling flow at high demand and avoiding or postponing the saturated flow condition.

Figure 12 shows the distribution over journey times across the test duration for four of strategies (High Bid, Logit (Long), Neural (Short) and Neural (Long)). The distribution for the Logit strategy indicates why its performance is poor. While it has approximately the same amount of trips in the $30-40\,\mathrm{s}$ peak as the High Bid strategy, the tail of the distribution extends to $500\,\mathrm{s}$ with

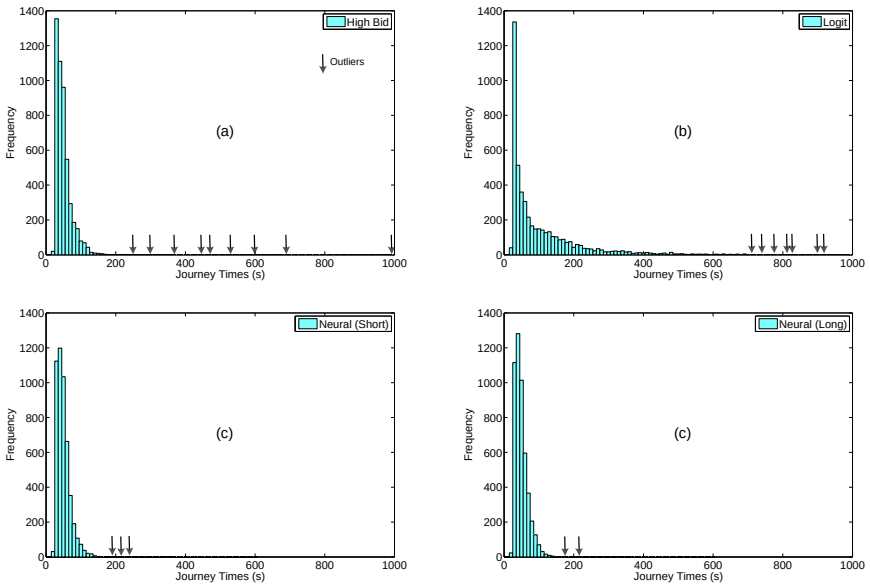

Figure 12: The distribution of journey times over the four hour simulation tests on the High Rd junction for four control strategies: High Bid, Logit (Long), Neural Network (Short) and Neural Network (Long).

outliers beyond 1000 s. Interestingly the Neural Network strategies do not have as many vehicles getting through the junction quickly as the High Bid and Logit strategies with their peaks in the $40-50$ s range. However the distributions have low variance, with almost no trips above 150 s.

5.3. Stranded Vehicles

A potential floor in the auctioning agent method is that it evaluates the state of the network using an instantaneous snapshot of vehicle position and speed and does not consider any temporal effects. A manifestation of this potential floor would be inequality in the delay experienced by different vehicles passing through the network. An example of this would be a single vehicle held at a red light for an indefinite length of time (the *stranded vehicle problem*).

The High Rd junction network in Southampton is particularly apt to testing the stranded vehicle scenario as the southern arm of the West junction leads to a small industrial car park and has very low demand both as an origin and as a destination (see Table 4). In fact this problem does arise under High bid control, while the distribution of journey times presented in Figure 12 is generally tight there are a few outliers with one unlucky vehicle spending 16 minutes in the simulation. However it is a different story under Neural Network control. Here the human controller *is* considering temporal effects and while the learning junction agent does not receive any explicit temporal information the

instantaneous bids may still be informative. The results show that the Neural network strategies are the most equitable strategies that were tested, more so even than MOVA. On the High Rd junction test the longest vehicle journey under neural network (long) control was 3.5 minutes.

5.4. Stochastic Analysis

In all the experiments described so far in this paper perfect information has been assumed. That is to say that it has been assumed that vehicle position is known and accurately communicated by every vehicle in the simulation. Or in the case where inductive loops have been used for MOVA controlled simulations it has been assumed that all detectors are working and detect every vehicle that passes over. In the real world localization sensors in vehicles would be subject to noise and errors making position estimation uncertain. Furthermore it is likely that not all vehicles would have working systems.

This section we examines results of simulation tests where uncertainty in position estimates and reduced fractions of instrumented vehicles has been modelled. The method used to model uncertainty has been described in detail in (Waterson and Box, 2011). To summarize: the position estimates from the simulated localization systems are degraded by the addition of Gaussian noise with 1σ ranging between $2 - 32$m. The number of vehicles in the simulation that are equipped with simulated localization systems is varied between $5\% - 100\%$.

Figure 13 shows results for twelve sets of simulation test on the Simple T-junction controlled by the Neural Network (Long) controller and twelve simulation tests on the High Road junction controlled again by the Neural Network (Long) controller Each test consited of ten simulation runs and the results are averaged. The graph on the left shows results for six pairs of tests where the simulated position of the vehicles has had Gaussian noise with a different variance added to it. The graph shows plots of delay averaged across all vehicles. This shows the expected trend of increasing delay with increase in the uncertainty of position. Significantly higher delays are observed in the case of the Simple T-Junction. The reason for this is that the demand placed on the Simple T-junction in these tests was comparatively higher than that placed on the High road junction (see Tables 2 and 4). This means that if a poor decision is made on the simple T-junction there is a higher chance that this will lead to long queues or even saturated flow. This unstable behaviour at high demands was also observed in similar tests looking at the High Bid algorithm that were published in Waterson and Box (2011).

The plot on the right in Figure 13 shows results for six pairs of tests with varying level of vehicle instrumentation in the simulation. The $1\,\sigma$ accuracy in position is constant at 2 m in all tests. These plots show average delay increasing approximately exponentially with reducing fraction of instrumented vehicles. These results indicate that the performance of the junction controllers is robust to reductions in the percentage of instrumented vehicles down to around 50 % with reasonable performance maintained down to 20 %. The plot for the Simple T-junction experiments show a steeper increase in delay below 20 % instrumented vehicles. This is again due the the instability of operating at

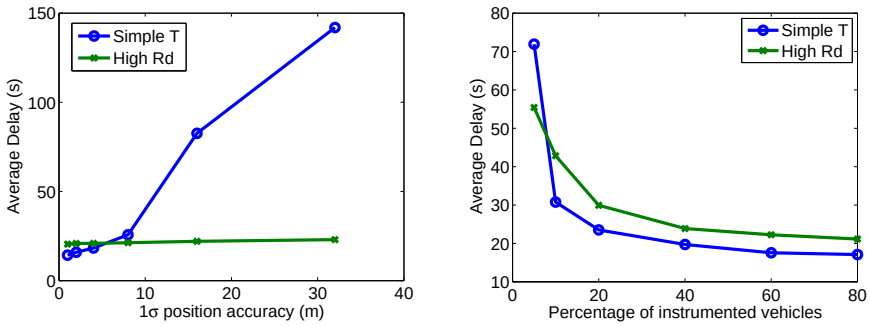

Figure 13: Average delay against (left): positioning accuracy of the simulated localization systems and (right): percentage of instrumented vehicles

higher levels of relative demand as discussed above. The practical significance of this result is that it indicates that even quite a low percentage $(20-40\%)$ of the vehicle fleet instrumneted, this simulation results indicate the low delay is achievable.

6. Conclusions

The principal results of the tests presented in Section 5 are summarized below:

- High Bid control can outperform the MOVA control system on an isolated junction in terms of delay. This is largely due to the fact that the auctioning agent control system uses localization probe data to describe the state of the network. This is a richer source of information than the inductive loop data employed by MOVA.
- Tests using machine learning auctioning agents showed that in all cases the neural network method produced signal control strategies that out performed those produced using the logistic regression method. This was due to the more flexible division of bid space enabled by the neural network approach.
- In all cases the long data set performed better that the short data set. This was due to the fact that there were more lane agents and therefore more information. In principle the number of lane agents in a network can be varied without upper limits. This means it is possible to increase the resolution of information in the state estimate at the cost of increasing the dimensionality of the bid space.
- Transient delay traces indicated that the performance of most of the control strategies was similar when the demand was low and that the main

differences in performance occurred in the peaks of demand. This suggests that the strategies that are performing better are achieving this by postponing or avoiding the condition of saturated flow at high demand.

- Analysis of the journey time distributions showed that neural network control was the most equitable strategy and robust to the *stranded vehicle* problem.

- Stochastic tests indicated that the neural network control strategy is robust to variation in the positioning accuracy of localization probes and to the fraction of vehicles equipped with probes.

This work has demonstrated the potential for applying machine learning techniques to control signalized junctions with expert human data used to train the system. While this supervised learning task has been shown to be useful there are some caveats that need to be attached to this approach. Firstly, by using a human expert trainer the junction controller's performance is effectively limited to being *as good* as the human, but it is possible that better control is achievable. Secondly, this approach implicitly assumes that all decisions made by the human are good ones, where in reality we know the human will make mistakes.

Because the positions and speeds of vehicles passing through the junction are being measured it is possible in principle to evaluate decisions and classify then as good or bad. This raises the possibility of feeding this information back into the learning agent as training data, resulting in a junction control agent that learns from experience. A (*Reinforcement Learning*) approach to this problem by the method of *temporal differences* (Sutton and Barto, 1998) is the focus of the current work in this area.

References

Bishop, C.M., 2006. Pattern Recognition and Machine Learning. Springer.

Box, S., Waterson, B.J., 2010a. Comparison of signalized junction control strategies that use localization probe data, in: The IET Road Transport Information and Control Conference, 25 - 27 May 2010, Dexter House, London, UK.

Box, S., Waterson, B.J., 2010b. Signal control using vehicle localization probe data, in: 42nd Annual UTSG Conference, University of Plymouth,5-7 January 2010. http://eprints.soton.ac.uk/73751/.

Box, S., Waterson, B.J., Hounsell, N., 2010. Comparison of signalized junction control strategies using individual vehicle position data, in: 5th IMA Conference on Mathematics in Transport 12 - 14 April 2010, UCL, London, UK. http://eprints.soton.ac.uk/74658/.

Chen, C., Heydecker, B., 2009. Adaptive traffic signal control using approximate dynamic programming, in: UTSG conference, London, January,2009.

Choy, M.C., Srinivasan, D., Cheu, R., 2003. Cooperative, hybrid agent architecture for real-time traffic signal control. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on 33, 597–607.

COOPERS, 2010. Co-operative systems for intelligent road safety. Online at http://www.coopers-ip.eu/.

Hunt, P., Bretherton, R., Robertson, D., Royal, M., 1982. Scoot on-line traffic signal optimisation technique. Traffic Engineering and Control 23, 190–192.

Kompfner, P., 2008. Cvis – cooperative for mobility. Online at http://www.cvisproject.org/download/cvis_brochure_May2008_Final.pdf.

Mikami, S., Kakazu, Y., 1993. Self-organized control of traffic signals through genetic reinforcement learning, in: Intelligent Vehicles '93 Symposium, pp. 113–118.

Nabney, I.T., 2002. Netlab Algorithms for Pattern Recognition. Springer.

Rose, G., 2006. Mobile phones as traffic probes: Practices, prospects and issues. Transport Reviews 26, 275–291.

SAFESPOT, 2010. Cooperative vehicles and road infrastructure for road safety. Online at http://www.safespot-eu.org/.

Sreedevi, I., 2005. Its decision – services and technologies – loop detectors. Available online at: http://www.calccit.org/itsdecision/serv_and_tech/Traffic_Surveillance/road-based/in-road/loop_summary.html .

Sutton, R.S., Barto, A.G., 1998. Reinforcement Learning An Introduction. Adaptive Computation and Machine Learning, MIT Press.

TRL, 2007. Pcmova user guide. TRL Application Guide AG58 .

Vincent, G., Peirce, J., 1988. 'mova': Traffic responsive, self-optimising signal control for isolated intersections. TRRL Research Report RR170.

Waterson, B.J., Box, S., 2011. Quantifying the impact of probe vehicle localisation data errors on signalised junction control. Accepted for publication in IET Intelligent Transportation Systems .

Wood, K., Crabtree, M., Gutteridge, S., 2006. Pedestrian and vehicular detectors for traffic management and control. TRL Report .