Zaria Journal of Electrical Engineering Technology, Department of Electrical Engineering, Ahmadu Bello University, Zaria – Nigeria. Vol. 9 No. 1, March 2020. ISSN: 0261 – 1570.

MODELLING OF AN URBAN TRAFFIC SYSTEM USING ARTIFICIAL INTELLIGENCE

[1]Abdulkarim A. Oloyede, [2]Nasir Faruk, [3]Lukman A. Olawoyin, [4]A. Abdulkarim, [5]Nazmat T. Surajudeen-Bakinde

[1,2,3]Department of Telecommunication Science, University of Ilorin, Ilorin, Nigeria.
[4]Department of Electrical Engineering, Ahmadu Bello University Zaria, Nigeria.
[5]Department of Electrical and Electronics Engineering, University of Ilorin, Ilorin, Nigeria.
oloyede.aa@unilorin.edu.ng

Keywords: –
Vehicle, Artificial Intelligence,
Traffic Control

Article History: –
Received: January, 2020.
Reviewed: February, 2020
Accepted: March, 2020
Published: March, 2020

ABSTRACT

Attributed to a rise in personal vehicle ownership and the ongoing expansion of cities, the highest growth of traffic congestion in recent years has been observed. Advances in the use of artificial intelligence have prompted the question of whether a smart traffic management system could be developed in order to improve the current state of congestion found in urban environments. Increases in traffic congestion in recent years has prompted the need for new and more advanced traffic control solutions. Advances in technology have allowed artificial intelligence (AI) to have an increasing number of applications. This paper investigates the use of AI called reinforcement learning in developing a new autonomous traffic control system based on a realistic traffic system model. This paper provides a critical review of the relevant surrounding literature. A summary of current technology available in traffic control was also investigated. Key simulation design decisions are then discussed such as the level of detail possible on modelling authentic driving behaviour, first required analysing the different ways in which the driver through the control of the vehicle reacts to their environment. Subsumption architecture was then identified as an appropriate method for defining these behaviours. A basic traffic scenario was simulated, and the results show that reinforcement learning can help in Traffic Management.

1. INTRODUCTION

Drivers across the world, we may have noticed that the amount of time they spend waiting in traffic is greater than it has ever been in recent times. Statistics from the United Kingdom (UK) has shown a 2.4% increase in traffic density [1]. Similarly, the National Bureau of Statistics (NBS) estimated Nigeria's vehicle population to 11, 458,370 at of 2017. The report showed that Lagos and FCT produced the highest number of national drivers' licenses while Yobe and Kebbi States had the least. Lagos is one of the most congested cities in the world. 40 per cent of cars in Nigeria are registered in Lagos and fatal accident rate in Lagos is 28 per 100,000 people. This is three times greater than in most European cities. The relative increases in costs of other transport methods have also created a positive trend of car ownership, with many now opting for fuel efficient models with lower running costs. Specifically, urban roads witnessed an average increase of 2%, which has directly resulted in greater levels of congestion [1] [2]. The economic impact of congestion is looking to rise significantly in the future. In a collaborative study between INRIX and the Centre of Economics and Business Research [3] it has been estimated that the annual cost of congestion on the UK will rise 63% by 2030 to £21 Billion". The incurred costs are attributed to factors such as cost of fuel, reduced productivity and rise in business fees for idle assets.

The economy is not the only the factor to suffer as a result of congestion. Traffic has a strong associated link with CO2 emissions [4]. Even with fuel efficiency at the forefront of transportation design, its reduction is small compared to the impact of idle vehicles in congested areas. In addition, the individual impact on the population is one of frustration, whether commuting in a personal vehicle or using public transport. The accumulation of these issues is the driving pressure on developing new traffic management systems to tackle the increases in the amount of congestion. The emergence of new technology in sensing, processing and communication has brought about the race to find the next stage of urban traffic control. A rise in artificial intelligence (AI) capabilities has triggered a buzz of investigation into how an autonomous system could be developed [5]. One growing field of AI is reinforcement learning methods, which has yet to be explored thoroughly for traffic control.

With the success of reinforcement learning application to traditional game playing problems (such as Tesauro's temporal difference algorithm to play backgammon [6]), many started investigating how it could be applied to practical robotics and control [7]. Scenarios such as the transportation of objects [8] and elevator dispatch [9] have deployed reinforcement learning to great effect, matching or surpassing the performance of controller schemes that were in current use.

In the paper, we investigate the use of reinforcement learning methods to an urban traffic simulation. Furthermore, a survey of the relevant literature is given to provide significant background information.

2. LITERATURE REVIEW

1) Traffic Sensors: For the traffic light controller to employ its control scheme correctly it is beneficial for it to be able to sense the environment of the junction. There are two main detection methods currently used in traffic systems: underground inductive loops and over ground radio wave detectors [10]. An inductive loop senses vehicles by detecting a change in the magnetic field induced as they pass over. Using the most up to date technology, Siemens launched their SLD4 Inductive loop detector in June 2013 [11]. Rather than just detecting when a vehicle has passed, this detector also has length based classification in order to distinguish different vehicle types. Inductive loops are installed under the road surface and are used in multiple positions across a junction. They can be placed in the lead up to the junction in order to establish a count of how many vehicles are approaching and, in a multi lane road, will give an indication of their intended vehicle direction. Additionally they can be also placed in the middle of a junction to detect any vehicles that are waiting for a gap in the traffic stream to turn.

In addition to inductive loops there are also over ground sensors that make use of radio wave technology to sense traffic. Siemens also produce a highly accurate over ground sensor using radar technology called Heimdall [12]. Pairing a planar radar antenna system with a digital signal processing engine, Heimdall offers an accurate way of vehicle detection and junction capacity measurement with the additional capability of detecting pedestrians. Older visual sensor technology used to be impeded by external conditions such as weather, however Siemens claim that their newer iterations using radar greatly reduce this effect. Usually installed on the traffic signal heads, over ground sensors are calibrated for the required field of view for that junction.

The type of sensor employed depends on the requirements of the junction. Overground sensors are easier to install but can rendered ineffective if obstructed by the junction environment. Inductive loops, on the other hand, are impervious to environmental factors but maintenance is high as well as risk of damage caused by roadworks. The most robust systems will combine the two sensory methods to ensure full detection ability across the entire junction.

2) Traffic System Communication: Regardless of the type of control scheme used by the junction's traffic controller, the ability to communicate out in the field back to central control is very useful. It allows for rolling updates to the control scheme and the retrieval of traffic data without physically attending the site [13].

Some modern control schemes require real time data in order to function accurately. Therefore, communication links took the form of private optimised telephone circuits [14]. As communication technology has advanced into wireless capability, efforts have been made to incorporate this in addition if not as a replacement to the fixed communication connections. Due to the packet based nature of wireless communication, there can be delays or entire data losses and as a result the control schemes have been adapted to deal with this. As part of a research program on bringing new communication technology to urban traffic control the UG405 protocol [14] was developed to solve this exact problem. UG405 compliant outstations such as the Chameleon from Imtech [15] and Stratos from Siemens are examples where GSM, GPRS and 3G as well as copper and fibre can now be used [16].

3) Traffic Control Schemes: There are several systems deployed to control junctions with varying leveled of complexity and use of current technology. The paper by A. Hamilton et al. based upon the collaboration between the Transportation Research Group, University of Southhampton and Siemens [17] discusses the evolution of these systems and provides a comparison of their advantages and disadvantages. One control scheme, using a fixed time plan, works by manually planning signals in a region so that junctions are synchronised, creating ’green waves’ that should allow traffic to smoothly pass through the area. Multiple schemes can be installed in the controllers to tackle different scenarios such as time of day or a local event. The traffic network study tool (TRANSYT) developed by TRL [18] takes historical data of a real traffic system, produces its own traffic model which it will then use to optimise a set of traffic plans. This has since been extended to allow for on-line plan updates to the controllers in the field if the right communications are available.

To better use available sensing capabilities, vehicle actuated control schemes have been developed. One example of this is Microprocessor Optimised Vehicle Actuation (MOVA) [19]. It works by reacting to the traffic environment at the junction and applies a non-congested mode to minimise delay and a congested mode to maximise capacity depending on the situation. When a vehicle is detected it invokes a demand for a green signal transition. Additional vehicles approaching the junction will increase the length the signal stays green until another demand is received elsewhere.

The most recently developed technology in the UK is a coordinated version of vehicle actuated control called the Split Cycle Offset Optimisation Technique (SCOOT) [20, 21]. Co-developed by Imtech, Siemens and TRL, SCOOT takes in continuous traffic data in a region and makes small changes to traffic plans. Its aim is to provide the ’green wave’ coordination of fixed term plans with the ability to react to traffic fluctuations by altering the split, cycle and offset times. In [17] a discussion of the advantages and disadvantages of each system is given. As the complexity of the control system increases so do the cost of implementation, the reliance on accurate detector implementation and the communication needs. However, the ability to react flexibly to the current traffic situation also increases so the benefits are immediately useful to areas where congestion is of concern.

3. SIMULATION ENVIRONMENT

Simulating a realistic traffic model before implementation is quite important. Figure 1 is a model that is proposed in this paper. There are many aspects to building up an effective model to simulate a traffic environment. The following sections document the research into these areas to help make decisions in designing an appropriate environment for the application of reinforcement learning.

1) Microscopic vs Macroscopic: The level at which a traffic system is modelled determines the level of detail of many aspects of the environment, the inputs it requires and the information that can be retrieved from it. The two main types of model are macroscopic and microscopic simulation. Mott and Beller [22] model the integration of personal rapid transit vehicles at both a macroscopic and microscopic level in separate simulations. Macroscopic simulation is based on flows and volumes as vehicles travel through the environment. They model the stations as nodes and the paths between them as possible flow movements. Visual colour coordination is used to represent the flow capacity along the paths, which will change over

time as the simulated passengers move towards their destination.

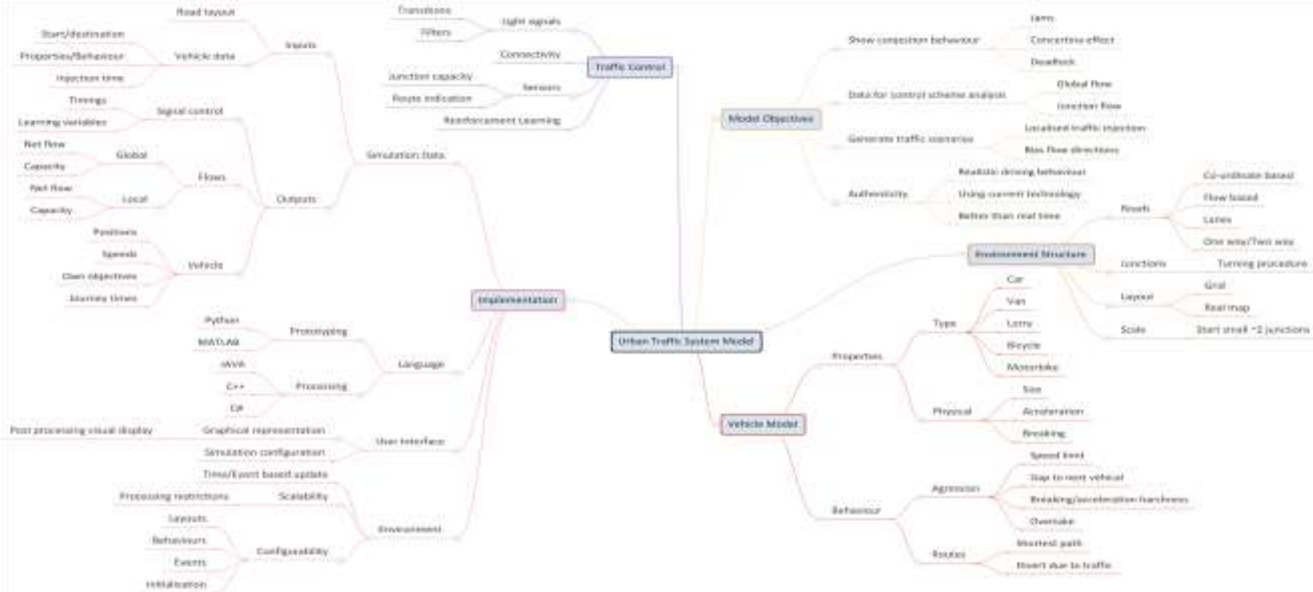

Figure 1. Modelling Urban Traffic Management

There are many aspects to building up an effective model to simulate a traffic environment. The following sections document the research into these areas to help make decisions in designing an appropriate environment for the application of reinforcement learning.

1) Microscopic vs Macroscopic: The level at which a traffic system is modelled determines the level of detail of many aspects of the environment, the inputs it requires and the information that can be retrieved from it. The two main types of model are macroscopic and microscopic simulation. Mott and Beller [22] model the integration of personal rapid transit vehicles at both a macroscopic and microscopic level in separate simulations. Macroscopic simulation is based on flows and volumes as vehicles travel through the environment. They model the stations as nodes and the paths between them as possible flow movements. Visual colour coordination is used to represent the flow capacity along the paths, which will change over time as the simulated passengers move towards their destination.

Microscopic simulation, in comparison, models the passengers and vehicles as individual entities rather than as a flow of movement. The movement of each can be followed and details such as when a vehicle has reached its capacity can be seen. Greater detail such as the infrastructure of the of the travel routes, how the vehicles operate and the way the passengers interact with the environment is required in order for the microscopic simulation to be effective.

In a comparison of the two model types Mott and Beller state macroscopic systems work well for large scale simulations where analysis of overall movement and capacity is important. Microscopic models on the other hand provide operation details where the affect of control methods can be analysed.

The structure of the physical layout requires much more description in microscopic level models than the flow based macroscopic ones. A simple cellular layout model is described in a paper by Nagel and Schreckenberg [23]. In their implementation of a freeway traffic model they represent the road as a one dimensional array where each element is a cellular space that can be occupied by a vehicle. During every update the simulation works out how many cells forward each vehicle travels and if there should be a change in acceleration due to vehicles occupying nearby cells. Tang and Wan [24] took this a step further and implemented a 3D animated traffic system also using a cellular approach.

2) Time vs Event Based Updates: There are two bases for progressing 'time' in simulation, discrete time or discrete events. In their paper on psycho-physical vehicle-following models, Schulze and Fliess [25] describe the functionality of these two forms and a comparison of their effect on running the simulation.

The time based approach works by updating all elements of the environment at steady time intervals. Then if any vehicle is in a situation where it has to change state due to the new position it will update that in the model. This goes for all other model elements such as the controller, sensors and signals of traffic control systems. Using event based methods the model computes an event list using the current disposition of the simulation and the time until the next scheduled event will take place. When the next event is reached, the environment is updated, new events are scheduled and the ordering of events already scheduled is recomputed if they have been affected by the recent change. An example of how the event update would work in car following behaviour can be found in [25]. An event occurs for the leading car as it is approaching a junction to decelerate. The processing of this event consists of calculating the next event time for the that car, scheduling it into the event list, calculating the unexpected event for the following car to also slow down and finally replace the following car's scheduled event with the new one.

They continue by listing the dependencies of both methods on the run time of the simulation. Computation for the time based method increases with a higher frequency of environment updates. Whereas with the event oriented computation the dependency is with the number of possible event cases to calculate. There is an additional limitation with event based methods which is how efficient the event scheduling algorithms, prompting consideration of language and how the simulation is structured.

It is obvious that for both methods the number of dynamic model objects within the system also contributes to the computation time. For this a conflict can be seen between ensuring the scale of the model is large enough to average out traffic flow behaviour across the simulation and minimising computation time.

IV. Simulation Architecture:

There are multiple elements that can make up a simulator programme. One way to structure the simulator is described in [26], based on the authors' work on microscopic traffic simulations. A diagram showing the update flow of the main simulator elements can be seen in Figure 2.

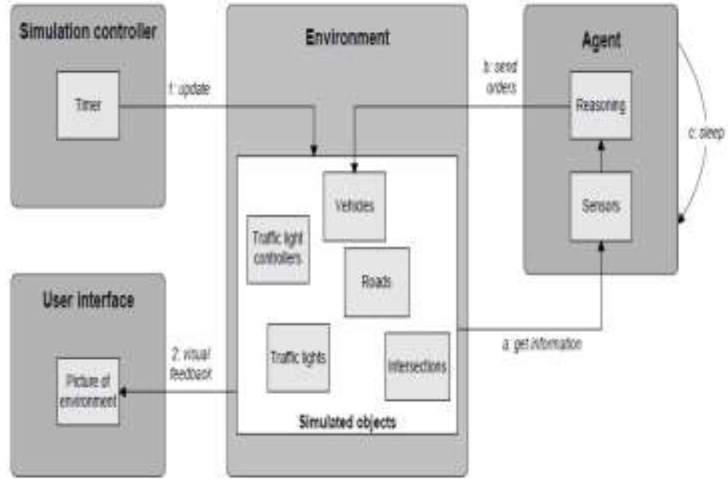

Fig. 2. Simulation and agent update loop [26]

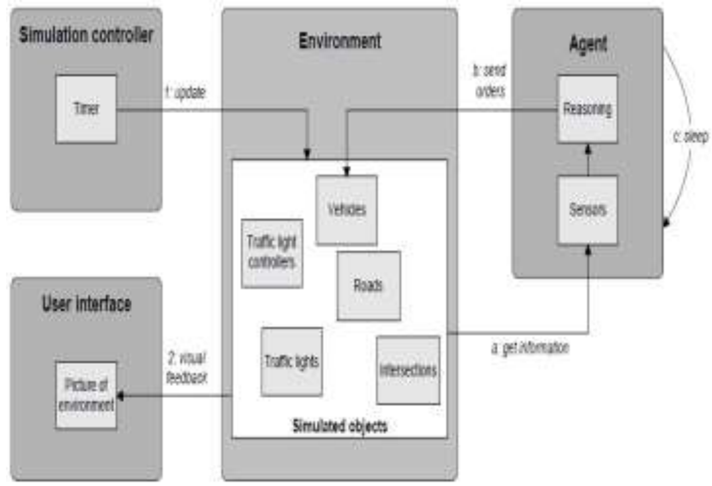
Fig. 2. Simulation and agent update loop [26]

The simulation control triggers updates to the environment using a time based method. On each update the elements of the environment will change depending on their own rules. This system passes information from the environment to the agents which represent the controlled objects in the environment. Finally at every update the output of the simulator is saved. In this case a visual image is produced in real time.

Available Traffic Simulation Software: There are many professional traffic simulation software packages, with most providing microscopic detail. PTV Vissim [27] is a commercial software product and was used in [22] for microscopic simulation. Its claimed features include easy to implement model geometries, active traffic management capabilities and system analysis including emissions modelling. One of the popular open source packages is SUMO

Simulation of Urban MObility [28]. It is another microscopic traffic simulator with a smaller feature set. It allows for multimodal traffic including pedestrians, auto generated traffic light time schedules. Neither SUMO nor PTV Vissim easily allow unconventional control system solutions to be applied to the real time simulations

5. VEHICLE MODELLING

Arguably the behaviour and control of the simulated vehicles themselves are the most important aspect when developing an urban traffic model. The ability to act as real drivers would improve the viability of any control systems that are tested in the simulation. Hence researching driving behaviour and effective ways of modelling it is important. Behaviour: The overall behaviour of a vehicle is the combination of the physical properties of the vehicle and the behaviour of the driver. Different aspects of driving behaviour can be categorised and modelled as a set of parameters used in simulation. One analysis of this breakdown is given in a paper on safety related driving behaviour by Bonsall et al [29]. The following parameters are some of the important behaviours described:

1) Desired speed - most often is the speed limit of the road
2) Desired headway gap - determined as enough time to match deceleration of the leading car
3) Reaction time - used as a measure of how soon after an event is reacted to by the vehicle.
4) Normal acceleration/deceleration - parameters to judge how vehicle speed will change under car following conditions
5) Maximum acceleration/deceleration - parameters to define vehicle physical limits under scenarios such as emergency stop and overtaking
6) Critical gap - defines how big a gap the vehicle desires, to make certain manoeuvres such as turn across a flow of traffic or overtaking

Many of the parameters can be linked to ’aggressiveness’. A more aggressive and impatient driver may occasionally over-speed, leave a smaller headway gap, employ harsher acceleration and require a smaller critical gap whilst making manoeuvres. A more cautious driver may frequently underspeed in busy areas or tight turns, and leave larger gaps in order to prepare to gently decelerate.

6. ROUTE PLANNING:

There are many established route planning algorithms that vary in complexity and efficiency [30]. One of the simplest and commonly used is Dijkstra’s shortest path algorithm and is used in the work of Raney and Nagel [31] for iterative route planning. In their algorithm the path or road weights between junction nodes are based on travel times not distances. The algorithm then iteratively works out the shortest path to an adjacent node from all nodes previously ’solved’ and the node is given a value totalling the weights to get to that node. This happens until the goal node is reached, and the shortest path is found by tracing the weights back to the start.

In another paper [30], Sanders and Schultes investigates algorithms of greater complexity for route planning in a general context. They claimed using simple methods like a bidirectional search using Dijkstra’s algorithm can already speed up the search by applying it in both directions until the same node is reached. In a further paper [32] they go on to introduce their work on the Highway Hierarchies algorithm for very large road networks. It combines the idea of directing the search toward the goal and the idea of having a path hierarchy to filter lesser paths.

A. Agent Based Modelling: Traditional and established approaches to AI use a sensing, planning and action philosophy. In a paper on new approaches to robotics by Brooks [33], he starts by introducing these traditional methods and providing examples of them in practise such as one of the original implementations, ’Shakey the Robot’ [34]. He goes on to criticise these methods for their poor ability to perform in the real world outside of a simulation. One of the ’new’ approaches to robotics outlined in the rest of the paper is subsumption architecture.

Introduced in an earlier paper by Brooks [35], subsumption architecture takes a reactive approach based on a set of agent behaviours. Figure 3 shows how the planning stage in traditional based methods is replaced by a hierarchy of behavioural priorities. The lowest levels are designed to represent the basic necessary functions of the agent and the higher levels become more abstract and goal driven.

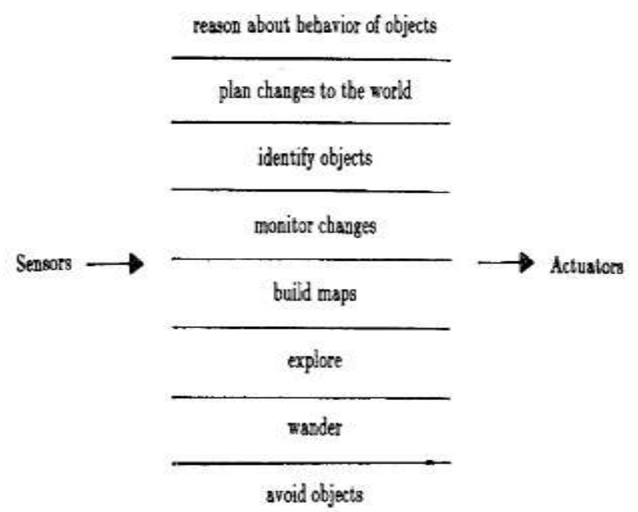

Fig. 3. Example decomposition of a robot using subsumption architecture

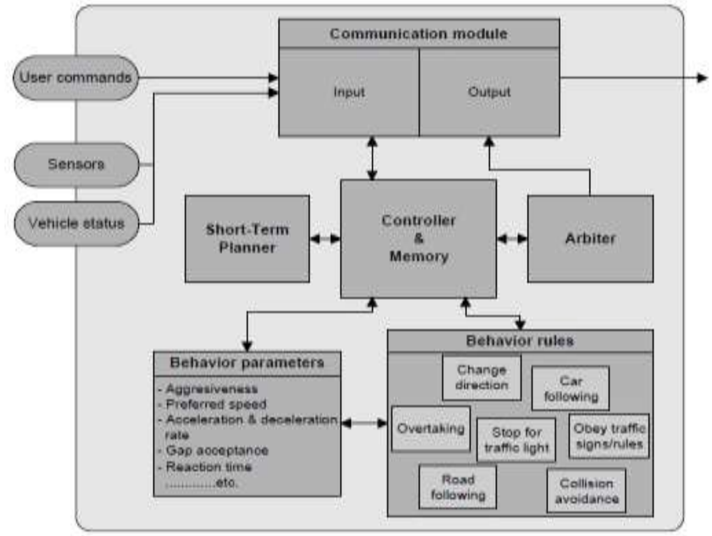

Fig. 4. Driving agent hybrid structure [26]

Each level is made up of connected modules that together define a finite state machine. An example by Brooks is a lowest level used to avoid objects. The input to the level is a sonar sensor, which is feed to a detection module. When an object is detected a runaway module is activated which in turn used turn and forward modules to avoid collision. Higher levels have the ability to suppress and inhibit the output of lower levels in order to action their own functionality. In the example the 'wander' level inhibits the turn and forward signals to act upon the desire to explore. A discussion of traditional methods verses reactive agents using subsumption architecture in a multi agent traffic simulation is given by Ehlert and Rothkrantz [26]. When building their simulation they rejected a full traditional approach, because the complexity of action planning for each agent in that environment would be infeasible to achieve in real time. Reactive agents however have a fast response time, but the lack of any planning ability does impact the accuracy of the model. They decided to implement a hybrid driving agent, which is mostly based on subsumtion architecture with an integrated short term planner and its structure can be seen in Figure 4.

B. Reinforcement Learning

The entire purpose of the research so far has been to aid in creating the most accurate model possible so that the impact of reinforcement learning on traffic control can be reliably measured. The following sections outline the research on the learning algorithms themselves and the considerations when implementing them.

1) ***Basic Philosophy****:* One of the core resources in the research of reinforcement learning is a book written by Sutton and Barto [36]. They introduce the concepts behind the learning algorithms, a coverage of the main algorithms currently being used and thoughts on the many factors that affect them.

One key concept is the idea of states and actions. A system with learning applied it to must have the capacity to be modelled in discrete states, where each state represents one possible configuration status of the system elements. In each state a set of possible actions are available of which one can be selected to transition to a different state. Often the case is that it is uncertain exactly what the new state may be as a result of a particular action. The method by which the action is selected is known as a policy.

Critical to the function of reinforcement learning are rewards and value functions. A reward is given for taking actions which either move towards the goal state or result in a 'better' state than the system was previously in. Value functions are a measure of how good the current state or state action pair is in terms of achieving the overall system goal. It is usually calculated by estimating the expected sum of rewards until the goal criteria is achieved. In this sense, rewards should be designed such that maximising the value function achieves goals.

The learning aspect comes from updating the value functions to represent their ability to achieve goals as a result of interacting with the environment. Over time or after many iterations there should emerge optimal action choices for each state. To help this convergence the magnitude of these updates will decrease overtime until an 'optimal' policy is found.

2) ***Exploration verses Exploitation****:* In a paper which surveys the field of reinforcement learning [7], Kaelbling et al. discuss a well known conflict in most related algorithms, exploration versus exploitation. Exploitation is when a policy (known as a greedy policy) will only select the action which currently results in the highest value function. This may provide reasonable results but runs the risk of being a sub optimal solution to the problem.

Exploration on the other hand promotes not always taking the highest policy so that other states can be tested. This can result in the discovery of new optimal policies from initially uncertain states. Exploration does directly effect the time to convergence of the optimal policy but usually results in an overall better solution.

Balancing exploration and exploitation is one of the main factors to try and optimise and many techniques are used to apply sufficient exploration to achieve optimality within a reasonable time. A common technique is -greedy where it takes a random non highest value action with probability *and* the highest value action the rest of the time. Convergence can then be simulated by slowly reducing. In [7] other techniques are given such as Boltzmann exploration.

3) ***Episodic and Continuous Tasks****:* Some tasks can easily be seen to be episodic in nature, examples of those are a game of chess or getting from point A to point B. In these cases there are clear initial and terminal states which in the case of the former example could be successful or unsuccessful. In comparison tasks such as the classic pole balancing problem could be considered continuous.

In [36], its stated that the value function (or expected return) for a particular state in episodic task learning is the simple sum of the expected rewards until the terminal state. In continuous task learning however the sum is infinite because of a lack of terminal state. The solution given for this issue is discounting. Discounting tapers off the proportion of the reward value of future expected rewards at a rate of γ^k, where k is how far in the future the reward is and γ is the discount rate between 0 and 1. A γ value of zero only accounts for the immediate reward and as it increases further rewards have a greater impact.

4) ***Markov Decision Process****:* As previously mentioned it is uncertain that given a state and an action, which new state will the system transition to. This could be because of environmental factor or otherwise. In [36] Markov decision processes (MDPs) define the probability of moving to a next state from a current state and action, and also the expected reward for a state transition due to a particular action.

These MDPs are used in the Bellman optimality equation, which finds the optimum action-value function of taking action *a* in state *s*:

$$Q^*(s,a) = \sum_{S'} P^a_{ss'}[R^a_{ss'} + \gamma \underbrace{max}_{a} Q^*(s',a')]$$

0

Where *P* is the probability of next state *s* from current state *s* and action *a*, *R* is the reward from that transition, γ is the discount rate and $Q^*(s^0,a^0)$ is the optimum value function for the next state. Whilst the Bellman optimality equation generates the exact optimal policy, its does so by exhaustively computing all future possibilities which even in small networks is unrealistic in implementation. Therefore effective algorithms use approximations to the equation, to achieve near similar results.

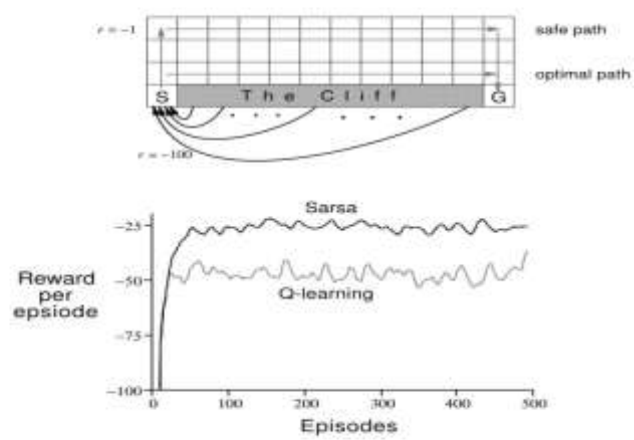

Fig. 5. Cliff-walking task using on-policy and off-policy methods [36]

5) On Policy and Off Policy Methods*:* Temporal difference methods define one of the main sets of reinforcement algorithms [36]. They work by updating the current state value with a fraction of difference between its value and the next state value. This allows states which lead up to high value states to increase in value so that those high value states can be reached. When using an -greedy policy for action selection, if the algorithm uses the chosen next state to update the previous state value, it is known as an on-policy algorithm. However when the algorithm uses the max next state value to update regardless of whether it was chosen by the -greedy policy or not, it is know as an off-policy algorithm.

A comparison example given in [36] can be seen in Figure 5 and highlights the differences between the two methods. The task was to navigate from the bottom left cell of a grid to the bottom right cell whilst avoiding moving into the 'cliff' on the rest of the bottom row. The off-policy method (Q-learning) quickly found the shortest route just above the cliff but due to exploration, often walked into the cliff which suggests it might struggle if implemented 'on-line'. The on-policy method (SARSA) found a longer but 'safer' route due to evaluating the exploration decisions. It is claimed that both methods will converge to the optimal policy as is reduced.

An additional example of an on-policy algorithm is the actor-critic method, and one for off-policy is R-learning.

6) ***Function Approximation****:* If the possible number of states or actions required to model a system are very large or even continuous, calculating value functions becomes extremely difficult [37]. One method used to mitigate this problem is function approximation. It uses approximation vectors to apply more generalised value approximation updates across states with common feature sets. Generalisation methods such as gradient decent or linear coarse coding [38] have been used successfully and even neural networks have been implemented for use as function approximators. These methods can then be paired with both on-policy and off-policy methods to form a full learning algorithm.

7) ***Planning Using Models****:* The probabilities of transitions and expected rewards for each state-action pair can be used to build a model of the current system [36]. Models can then be used to artificially simulate experience simultaneously with real experience from the environment. This can dramatically reduce the amount of time required to converge to an optimal policy.

There are some considerations to be made when using model planning. First the accuracy of the model will impact the effectiveness of the optimal policy if it does not represent the environment well. Sufficient computation has to also be given to backing up the simulated samples appropriately. Having a model can provide unique benefits too such as encouraging the exploration of long untried states by artificially increasing its value function. Depending on the application of reinforcement learning, additional planning through modelling may provide a significant advantage.

VII Reinforcement Learning for Traffic Management

This paper analysing the use of reinforcement learning algorithm for controlling traffic at a single junction. A simulation of a single junction environment was run for 24 virtual hours, using the stochastic scheduler with a car rate of 1 every 5 seconds. The SARSA learning algorithm was chosen with a learning rate, discount factor and exploration probability of 0.1, which is a common initial value for these parameters. Another simulation was run using the same settings but without the learning, which represents a standard set of results for comparison. Using the graph analysis tool developed, graphs were generated for both simulations in order to assess the effectiveness of the learning control system.

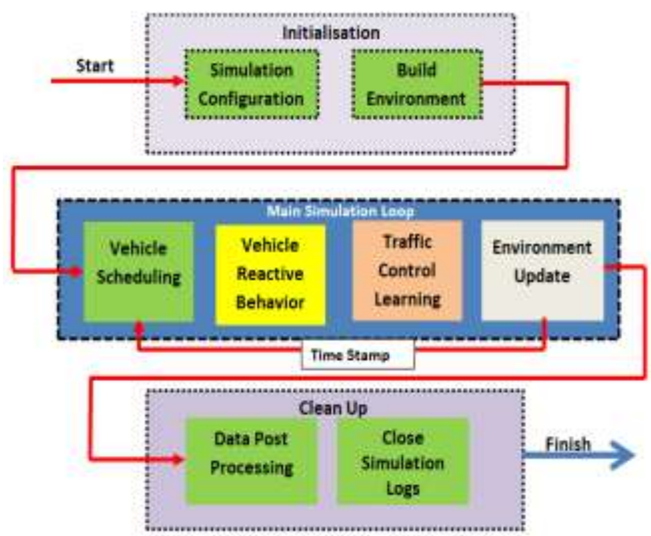

Figure 5: Modelling flow Diagram

Figures 6 and 7 show the average speed of the vehicles in the simulation over the duration for the set plan simulation and the learning control simulation respectively.

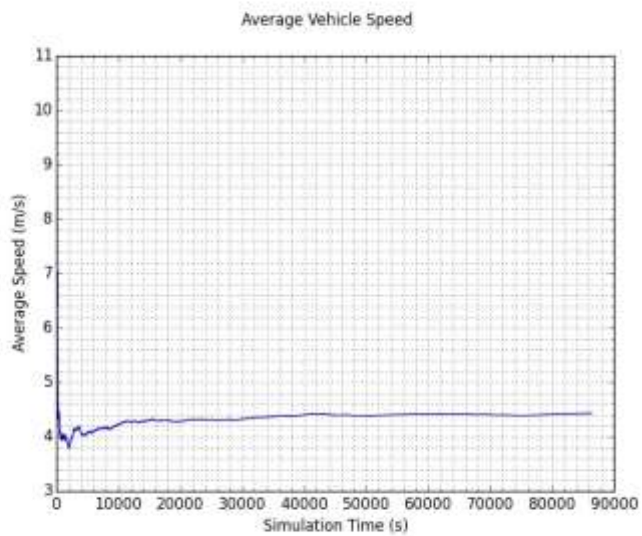

Figure 6: Moving Average Speed Graph for Fixed Plan Control

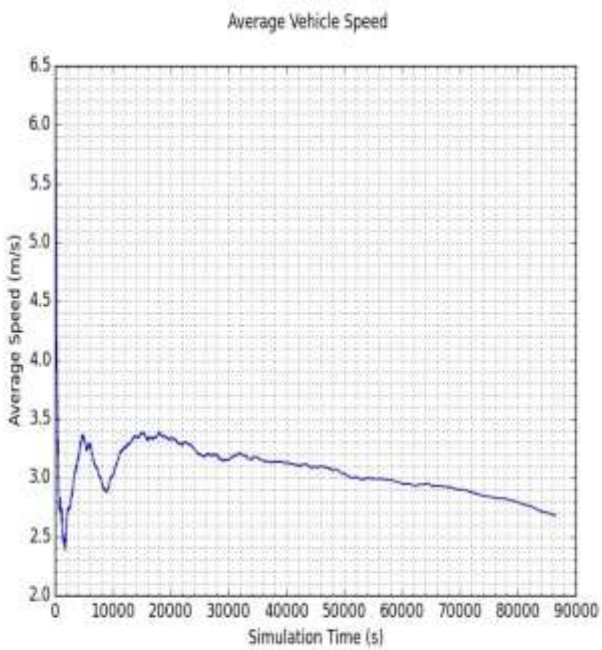

Figure 7: Moving Average Speed Graph for SARSA Control

It can be seen that for the fixed plan simulation, after the initial settling time, the average vehicle speed has settles to around 4.4*m/s*. On the other hand, for the learning control simulation, the average vehicle speed is a fair bit lower and has also not settled, if fact it is in decline. This signifies that the current control scheme has not learnt an optimum control policy or reached convergence by the end of the simulation. This is further confirmed by looking at the total reward graph for the learning control simulation, shown in figure 8.

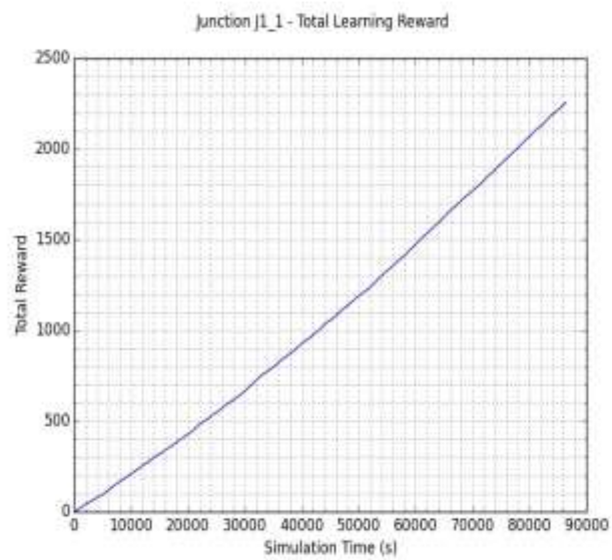

Figure 8: Accumulative Reward Graph for SARSA Control

As the traffic control agent learns, the expectation is that that total reward will exponentially increase over time. This should occur as the set of policies learnt by the agent converge towards an optimum policy, since the performance should increase as a result and should therefore gain a bigger reward each learning step. However, the reward graph for the learning control simulation shows an almost linearly increasing line, with a slight positive curve.

What can be taken from this is that the control agents are still in the early stages of learning at the end of the simulation. In the virtual 24 hour period 5866 learning steps occurred which, considering the number of possible actions (1344) the learning agent could perform across the 576 possible states, is very low. Due to the factors of exploration probability and the chance the agent has of entering every learning state, this problem would require a significantly larger number of learning steps in order to exhaust all possibilities with seeing them a sufficient number of times and for convergence to occur. It is highly probable that improved performance over a traditional fixed plan control system may be possible without full convergence, but a reasonable coverage will still need to be met before a set of policies are considered reliable. A simple solution would be to just run the simulation for longer, however it is worth noting that the simulation with learning took approximately 17 minutes to execute using a personal home computer and generated 256MBytes worth of data in the simulation logs. This paper has shown that whilst reinforcement learning can be applied appropriately to solve the issue of urban traffic management in at least one way, it has highlighted the difficulty in assessing its effectiveness using a realistic traffic simulation model. The dynamics of the traffic model used in the simulation are much faster than the dynamics of the learning agents used to control it and therefore the amount of activity that has to be simulated between learning steps is large. The evidence presented here does signify that reinforcement learning could still provide a novel solution to traffic management in urban environments, however there is a large amount of research that still needs to be conducted in order to find the most optimal implementation.

7. CONCLUSIONS

In this paper a review of the literature has been given in the key areas encompassed in the given project. this paper has developed a method of assessing the efficiency of traffic control systems using a realistic traffic simulation. More specifically it has been used to assess the feasibility of using reinforcement learning techniques to develop an intelligent system which learns to optimise traffic flow through an urban environment. It found that whilst reinforcement learning can be successfully applied to the traffic control problem, the complexity of the problem may require a substantial amount of learning before finding an optimal solution. Due to the realistic dynamics of the simulation being fast compared to the slower learning dynamics of the traffic controller, running a simulation for the length required for a large number of learning steps was infeasible with the processing power and data storage available.

REFERENCES

[1] D. for Transport, *Road Traffic Estimates: Great Britain 2014*, 2015.

[2] INRIX, "Growing economy drives traffic congestion up in over three quarters of uk cities," 2015. [Online]. Available: http://inrix.com/press/scorecard-report-united-kingdom/

[3] ——, "Traffic congestion to cost the uk economy more than 300 billion over the next 16 years," 2014. [Online]. Available: http://inrix.com/press/traffic-congestion-to-cost-the-uk-economy-more-than-300-billion-over/ -the-next-16-years/

[4] M. Barth and K. Boriboonsomsin, "Real-world carbon dioxide impacts of traffic congestion," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2058, pp. 163–171, 2008.

[5] D. A. Roozemond, "Using autonomous intelligent agents for urban traffic control systems," *Proc. 6th World Congr. Intell. Transp. Syst*, 1999.

[6] G. Tesauro, "Temporal difference learning and td-gammon," *Communications of the ACM*, vol. 38, no. 3, pp. 58–68, 1995.

[7] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of artificial intelligence research*, pp. 237– 285, 1996.

[8] M. J. Mataric, "Reward functions for accelerated learning," in *Proceedings of the Eleventh International Conference on Machine Learning*, 1994, pp. 181–189.

[9] A. Barto and R. Crites, "Improving elevator performance using reinforcement learning," *Advances in neural information processing systems*, vol. 8, pp. 1017–1023, 1996.

[10] S. A. , "How are vehicles detected at traffic signals? - vehicle detection techniques," 2015. [Online]. Available: http://www.traffic-signal-design.
com/vehicles detected.htm

[11] H. Griffiths, "Trafficsolutions - sld4 loop detector," 2013. [Online]. Available: http://www.siemens.co.uk/traffic/en/index/
productssolutionsservices/detection/sld4.htm

[12] R. Spence, "Trafficsolutions - heimdall," 2015. [Online]. Available: http://www.siemens.co.uk/traffic/en/index/
productssolutionsservices/detection/heimdall.htm

[13] S. A. , "Urban traffic control (utc)," 2015. [Online]. Available: http://www.traffic-signal-design.com/urban traffic control utc.htm

[14] M. Bodger and K. Manston, "Ug405 in practice," *TEC (Traffic Engineering & Control)*, pp. 440–443, 2009. [Online]. Available: – –
http://www.siemens.co.uk/traffic/pool/documents
/articles papers and presentations/tec nov 09.pdf

[15] Imtech, "Imtech traffic & infra uk — urban - traffic management systems," 2015. [Online]. Available: http://www.imtech.uk.com/ urban-traffic _management systems.html

[16] Siemans, “Traffic solutions - stratos outstation,” 2015. [Online]. Available: http://www.siemens.co.uk/traffic/en/index/ productssolutionsservices/systems/stratosoutstation.htm

[17] A. Hamilton et al., “The evolution of urban traffic control: changing policy and technology,” *Transportation Planning and Technology*, vol. 36, no. 1, pp. 24–43, 2013. [Online]. Available: http://dx.doi.org/10.1080/03081060.2012.745318

[18] TRL, “Transyt online - traffic & control - products - trl software,” 2015. [Online]. Available: https://trlsoftware.co.uk/products/traffic _ control/transyt online

[19] ——, “Mova - traffic & control - products - trl software,” 2015. [Online]. Available: https://trlsoftware.co.uk/products/traffic control/mova

[20] TLR, “Scoot - traffic & control - products - trl software,” 2015. [Online]. Available: https://trlsoftware.co.uk/products/traffic control/scoot

[21] Imtech, Siemens, and TLR, “Scoot,” 2015. [Online]. Available: http://www.scoot-utc.com/

[22] P. Mott and S. Beller, “Macroscopic and microscopic simulation for the evaluation of people mover systems,” in *PRT@LHR*. PTV AG, 2010, pp. 1–15.

[23] K. Nagel and M. Schreckenberg, “A cellular automaton model for freeway traffic,” *J. Phys. I France*, vol. 2, no. 12, pp. 2221–2229, 1992.

[24] W. Tang and T. R. Wan, *Multi-agent Animation Techniques for Traffic Simulation in Urban Environments*. Vaclav Skala, 2015, pp. 161–164.

[25] T. Schulze and T. Fliess, *urban traffic simulation with psycho-physical vehicle-following models*. ACM Press, 1997, pp. 1222 – 1229.

[26] P. A. Ehlert and L. J. Rothkrantz, *Microscopic traffic simulation with reactive driving agents*. IEEE, 2001, pp. 860 – 865.

[27] P. G. , “Ptv vissim,” 2015. [Online]. Available: http://vision-traffic. ptvgroup.com/en-us/products/ptv-vissim/

[28] R. Hilbrich, “Dlr - institute of transportation systems - sumo simulation of urban mobility,” *Dlr.de*, 2015. [Online]. Available: http: //www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931 read-41000/

[29] P. Bonsall, R. Liu, and W. Young, “Modelling safety-related driving behaviourimpact of parameter values,” *Transportation Research Part A: Policy and Practice*, vol. 39, no. 5, pp. 425 – 444, 2005.

[30] P. Sanders and D. Schultes, “Engineering fast route planning algorithms,” in *Experimental Algorithms*, ser. Lecture Notes in Computer Science, C. Demetrescu, Ed. Springer Berlin Heidelberg, 2007, vol. 4525, pp. 23–36.

[31] B. Raney and K. Nagel, “Iterative route planning for large-scale modular transportation simulations,” *Future Generation Computer Systems*, vol. 20, no. 7, pp. 1101 – 1118, 2004, geocomputation.

[32] P. Sanders and D. Schultes, “Engineering highway hierarchies,” in *Algorithms ESA 2006*, ser. Lecture Notes in Computer Science, Y. Azar and T. Erlebach, Eds. Springer Berlin Heidelberg, 2006, vol. 4168, pp. 804–816. [Online]. Available: http://dx.doi.org/10.1007/11841036 71 [33] R. A. BROOKS, “New approaches to robotics,” *Science*, vol. 253, no. 5025, pp. 1227–1232, 1991.

[34] N. J. Nilsson, “Shakey the robot,” DTIC Document, Tech. Rep., 1984.

[35] R. Brooks, “A robust layered control system for a mobile robot,” *Robotics and Automation, IEEE Journal of*, vol. 2, no. 1, pp. 14–23, Mar 1986.

[36] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[37] R. S. Sutton, D. A. McAllester, S. P. Singh, Y. Mansour *et al.*, “Policy gradient methods for reinforcement learning with function approximation.” in *NIPS*, vol. 99. Citeseer, 1999, pp. 1057–1063.

[38] R. S. Sutton, “Generalization in reinforcement learning: Successful examples using sparse coarse coding,” *Advances in neural information processing systems*, pp. 1038–1044, 1996.