

Model-based Design of Coordinated Traffic Controllers

Roopak Sinha^a, Partha Roop^b, Prakash Ranjitkar^c, Junbo Zeng^d, Xingchen Zhu^e

^aLecturer, ^{b,c}Senior Lecturer, ^{d,e}Student

^{a,b,c,d,e}Faculty of Engineering, University of Auckland, Private Bag 92019, Auckland, New Zealand, Phone: +64-9-3737599

^aEmail: r.sinha@auckland.ac.nz

ABSTRACT

Modern transportation systems optimize traffic flows in road networks by allowing intersection-level traffic controllers to communicate and synchronize. Typically they are distributed systems, with multiple nodes (intersections) communicating with each other in real time. This paper proposes a design process for traffic coordination systems using model driven engineering (MDE), a paradigm used to design complex embedded systems in automotive and aerospace. The approach combines a conventional micro-simulator (AIMSUN) with a state of the art control modelling tool (Simulink) following a *model-view-controller* approach. The MDE approach allows for both micro-simulation (on a PC) and emulation (using embedded controllers). Once the model is validated, automatic code generation can be used to generate the implementation of the system on embedded devices. As a case study, we designed a SCATS-inspired coordinated intersection control system. To our knowledge, the proposed approach is the first, fully automated approach for the design of complex ITS systems.

Keywords: Model-based design, Micro-simulation, model-driven engineering.

1. INTRODUCTION

In recent years, the rapid increase in the number of private car ownership in developed and developing countries has caused severe congestion in road networks as road infrastructure (supply) cannot keep up with the increasing demand for travel [1].

Adaptive coordinated traffic control systems, such as Sydney Coordinated Adaptive Traffic System (SCATS) [2] and the Split Cycle Offset Optimization Technique (SCOOT) [3] are widely used in different part of the world to coordinate and dynamically adjust traffic flows between multiple intersections. It has been shown that these systems can dramatically improve traffic conditions, reducing wait times by up to 30%. It requires sophisticated tools to design, test, and deploy these systems. Currently, the design approach is almost exclusively manual. Designers first create a model of the road network to be controlled using an appropriate micro-simulator [4], model various parameters such as traffic flow patterns based on measured values, and then test the system's effectiveness in the micro-simulation environment. However, this is a time consuming process, and key issues such as functional safety and 100% coverage are not tested.

In this article, we propose an alternative design process, which uses model-driven engineering (MDE) [5] to design adaptive coordinated traffic control systems. MDE is used to design complex distributed embedded and industrial control systems [6]. Traffic coordination can be seen as a distributed system with multiple intersections interacting with each other to achieve optimal traffic flows.

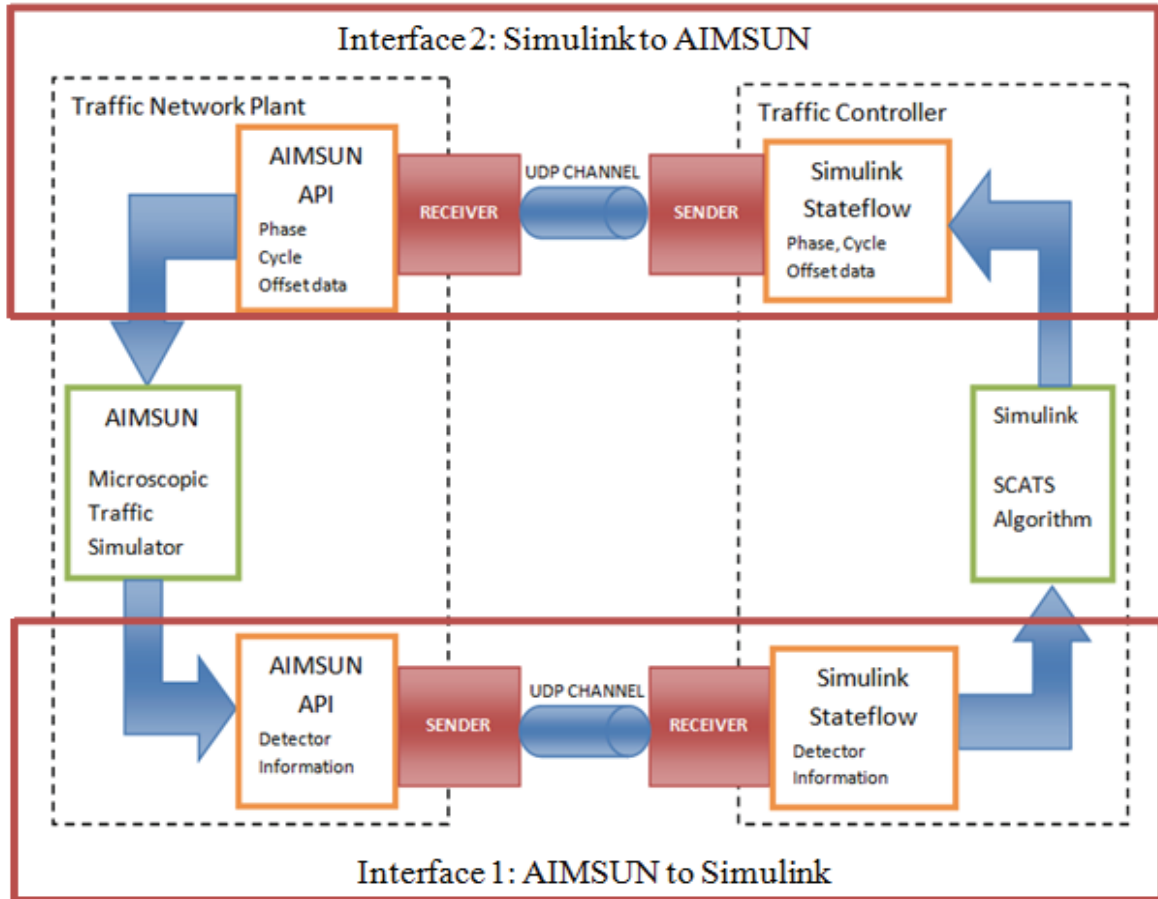


Figure 1: Model-based engineering of Coordinated traffic control using Simulink and AIMSUN

We focus on the Model-View-Controller design pattern [7], where designers build a model of the plant being controlled (road, traffic, etc.), along with the traffic controller (SCATS/SCOOT), and a view to visualize the interaction of the plant model and the controller. More specifically, as shown in Figure 1, we use Simulink [8] to build a SCATS-like controller and use the AIMSUN [4] micro-simulator to model a multi-intersection road model. The interaction of these two models is enabled using network connections between Simulink and AIMSUN (using the AIMSUN API extensions), and their interaction can be viewed during micro-simulation. We report the results of modelling a coordinated traffic controller using this methodology to show that the proposed process can serve as a foundation for building, testing and deploying traffic controllers.

The rest of this paper is organized as follows. Section 2 describes the plant modelling using AIMSUN. Section 3 first gives a brief background of SCATS and then describes how a SCATS-inspired controller is developed using Simulink. Section 4 briefly describes the interfacing between Simulink and AIMSUN. We present experimental results in Section 5 and concluding remarks appear in Section 6.

2. PLANT MODELLING

AIMSUN is a micro simulator that provides a demonstration environment to prove the feasibility and benefits of a proposed system before implementation on the actual traffic network. In addition, it can import actual traffic network data, provide 2D and 3D animation outputs, contains accurate traffic models, and provides an application programming interface (API) to achieve adaptive traffic control during simulation.

In this paper, we use AIMSUN to develop a model of a busy intersection in Auckland. This intersection, where two major arterial routes Balmoral Road and Dominion Road intersect, was also chosen because measured traffic volume data for this intersection was provided to us by Auckland Transport. This data, in the form of 15 minute traffic count, was measured in mid-November, 2008, and was obtained over a period of 14 hours (from 6:00 am to 8:00 pm). The demand data for buses are approximated to be 15 vehicles/ hour during normal off-peak period and 25 vehicles/ hour during peak hours.

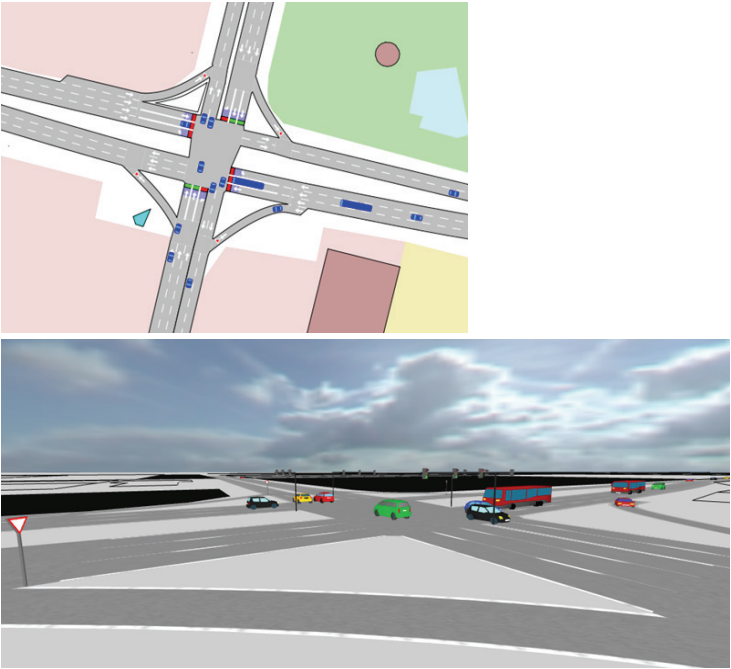


Figure 2: 2D and 3D view of the Balmoral Road - Dominion Road Intersection in AIMSUN

The following steps were used to create this model. The map for this intersection in the Open Street Map XML format [9] was imported into AIMSUN. Then road sections within AIMSUN were created by using the map as an accurate guide and then interconnected to create the intersection. Next, loop detectors and traffic lights (artefacts provided within AIMSUN) were placed on the intersection. Loop detectors, capable of count (number of vehicles passing over them) and occupancy (amount of time the detector was occupied), are used to relay important information to the Simulink controller (this aspect is explained later in Sec. 3.1). Similarly, traffic light phases were created to mimic the operation of the real intersection, which contains 4 phases, as shown in Tab. 1.

Table 1: Phase Arrangement

Phase Number	1	2	3	4
Lane Group				

The traffic lights are controlled by the Simulink controller (described in Sec. 3) using the interfacing between Simulink and AIMSUN (described later in Sec. 4).

3. CONTROLLER MODELLING

3.1. SCATS

SCATS [2] links multiple signalized intersections together to optimize traffic flow. At a single signalised intersection, traffic lights at each intersection operate in cycles, repeating the sequence at the end of each cycle, this is known as the *cycle time*. Each cycle time is further broken down into multiple *phases*; each phase consists of a collection of non-conflicting vehicle movements (approaches). Figure 3 details a standard four phase signalized intersection. During a cycle, only one of the four phases will allow vehicle movement at any given point of time.

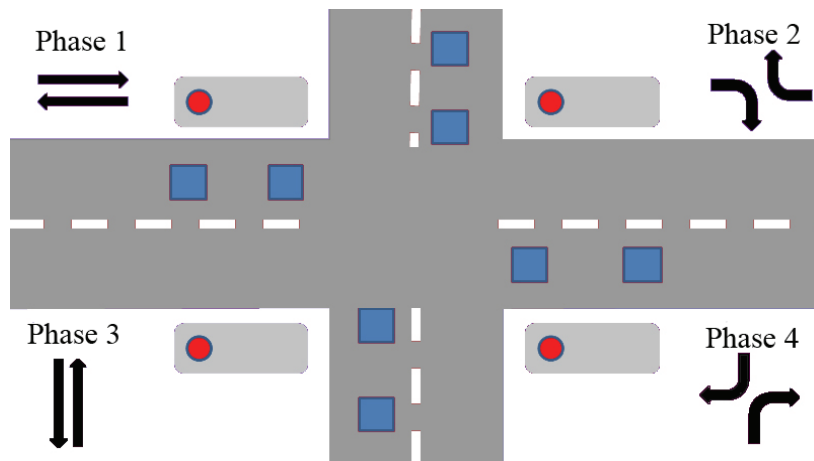


Figure 3: A visual diagram depicting different phases

Each phase is expressed as a percentage of cycle length. It is apparent that it is possible to have infinite combinations of different phase percentages to make up an entire cycle length. Each of these individual combinations is known as a *split plan*, and is one of the key parameters that SCATS optimises to improve traffic flow.

Within each phase, there are four main components, as indicated by the phase time diagram shown in Figure 4. The phase must remain green at least for the *minimum green* duration. Under certain conditions, the green time may be extended for a phase (*extension time*). Finally, *yellow* and all *red* times follow before moving on to the next phase.



Figure 4: Components of a phase

A SCATS controller can vary the cycle time and/or individual phases depending on traffic conditions. The controller reads occupancy information from *loop detectors* placed at every approach. This data can be converted into a degree of saturation value, which is calculated using the formula:

$$DS = K[g - (T - tn)] / (g + r) \quad (1)$$

where, n is the number of spaces counted, T is the total non-occupancy time, t is standard space time, r is the remaining phase time, K is a bias factor (with default value 1), and g is the green time.

SCATS plans are used to control the overall behaviour of an intersection. For example, at nights when the traffic is low, a plan that has low cycle lengths may be used, while different plans may be required for high traffic conditions in mornings and evenings. A plan does not fix the cycle length and phase times, but serves as a starting point which can be varied, depending on traffic conditions.

While coordinating multiple intersections, SCATS looks at creating “green corridors” in order to increase the likelihood of vehicles travelling through two consecutive intersections to experience no waiting at the second intersection. This is done by introducing an *offset*, which ensures that the second intersection turns green on a particular phase so that vehicles released from the first intersection can go through without stopping (as shown in Figure 5)

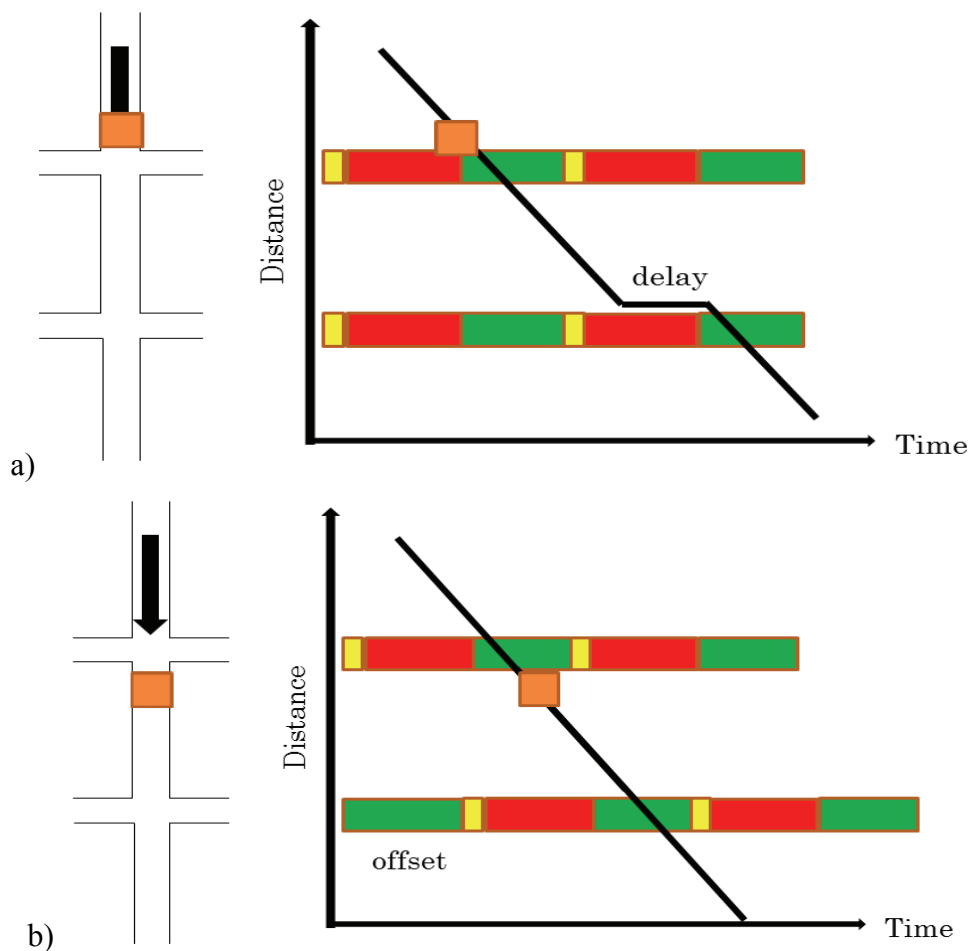


Figure 5: Offsets. (a) Two intersections with no offset, (b) two intersections with offset

3.2. Modelling in Simulink

Model-based design uses visual techniques with an underlying mathematical model to address complex design problems [6]. The main advantages of a model based approach lies within its flexibility and configurability. For example, changing from the control of two to twenty intersections requires only the insertion of new controllers, and making the necessary connections. Matlab/Simulink was selected as the target development platform for developing the traffic controller, because Simulink [8] is a model-based development

platform that is widely used throughout the automotive industry, and supports both continuous time semantics and discrete time signals.

The controller, loosely based on SCATS, has two main levels of control: *strategic* and *tactile* control. Strategic control is based on a centralized server making adjustments to the cycle, phase, and offsets for subsystems (groups of intersections) and intersections. Tactile control is achieved on local controllers placed at intersections and focuses on optimizing cycle and phase times, based on degree of saturation values of the various approaches.

3.2.1 Strategic control

At the strategic level, a subsystem is the basic unit of control. It consists of a group of signalized intersections that share the same cycle length, compatible split plans and offsets. A subsystem usually consists of a group of intersections that are found within 800 meters of each other and show significant traffic correlations [2]. The logic of grouping such intersections together is that they operate on compatible parameters, increasing overall traffic performance of the system. The Figure 6 shows the top level architecture of a two subsystem SCATS model that was implemented.

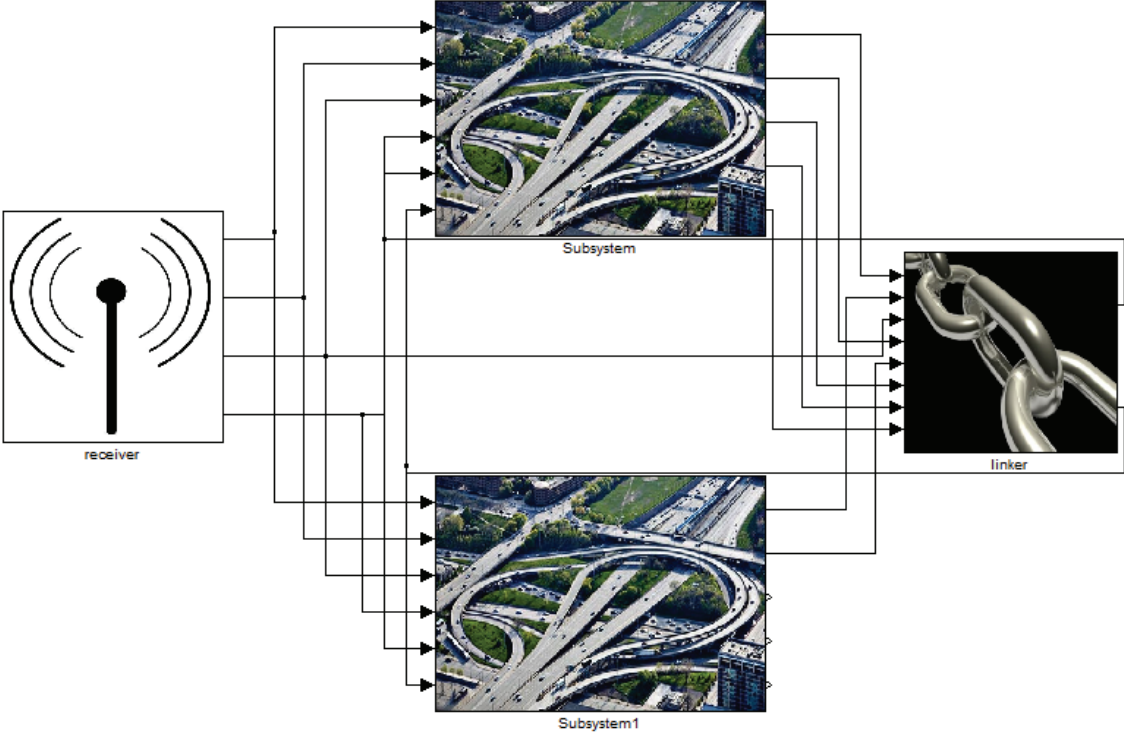


Figure 6: Top level architecture

SCATS optimises the cycle, phase and offsets within each subsystem individually. To account for potential traffic correlations between subsystems, both SCATS, and the implemented solution propose to use a subsystem marriage method. Subsystem marriages are determined within in the linker module shown in Figure 6. The algorithm to compute marriages is shown in Figure 7, a Simulink Stateflow diagram. Stateflow [10] are hierarchical and concurrent state machines supported by Simulink. Basically, if the difference in cycle lengths between the two subsystems is within nine seconds for 4 consecutive sampling times, then a marriage occurs. When two subsystems are married, they share the same cycle time and an offset is introduced between them. Similarly, if the cycle times of two married subsystems differ by more than 9 seconds for 4 consecutive sampling times, a *divorce* occurs, where they are controlled individually.

For each subsystem, the system employs a specific offset plan at any time. A voting algorithm is used to choose the best offset plan (out of 4) depending on varying traffic conditions. The decision is based essentially on SCATS timing plans currently being used, the flow of traffic in both directions (between the two subsystems) and the nominated phase to be coordinated between the two subsystems.

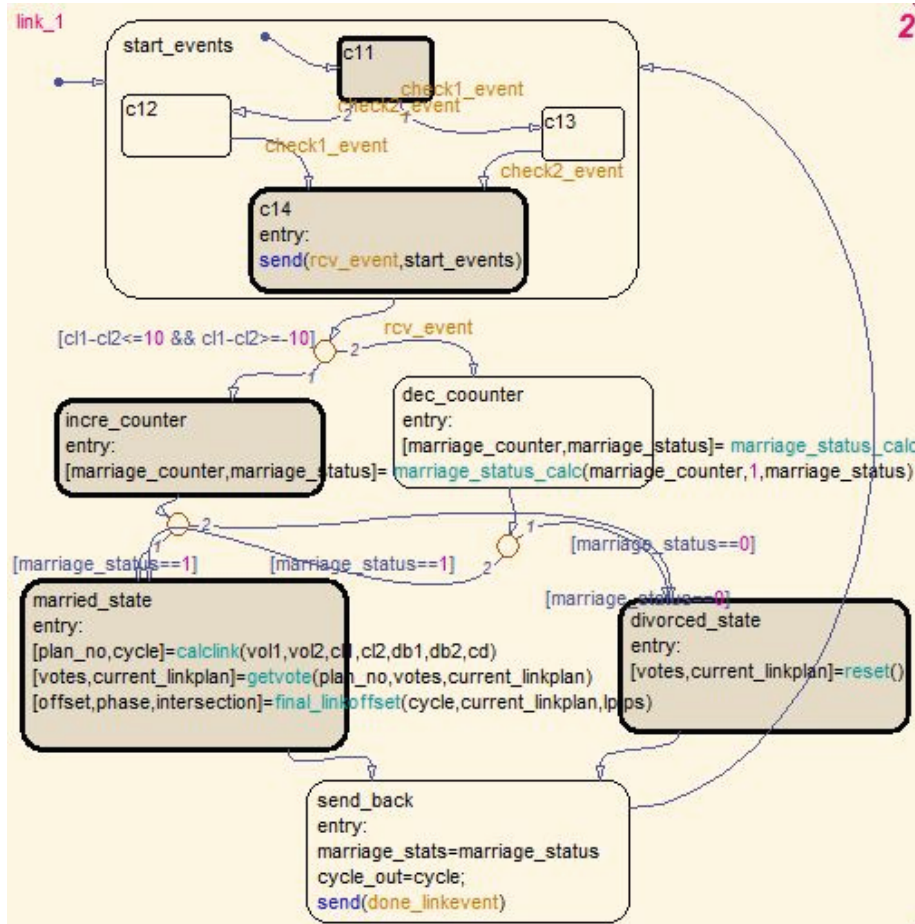


Figure 7: Marriages State Chart

3.2.2 Tactile control

Within a single subsystem, there are two types of intersections: a single *critical* intersection (normally the one with the maximum amount of traffic flow), and other *non-critical* intersections. The cycle length and split plans for every intersection in the subsystem are based on data measured by the detectors at the critical intersection. Non-critical can have their own offsets relative to the critical intersection. Within each intersection, a Simulink Stateflow chart runs continuously, updating cycle length, split phases and offsets when all the detectors data are received from the receiver module at the end of each cycle.

For the critical intersection, the degree of saturation is computed based on the measurements performed by the loop detectors, at the end of every cycle length. The intersection controller then computes the next cycle length based on this value. The controller decides on the following parameters for the next cycle: *the minimum cycle length (mincl)*, *maximum cycle length(maxcl)*, *two stopper cycle lengths (scl1 and scl2)* and *two sz values (sz1 and sz2)*. These new values are combined with weighted versions from the previous two cycle lengths, forming a moving average. This avoid rapid changes due to short pulses of change in traffic. The cycle length is not adjusted by more than 6-9 seconds between two cycles.

Finally, a decision is also made on the split plan to be used for the intersection. Usually there are 4 to 8 split plans. A voting algorithm computes the degree of saturation at (pre-determined) strategic approaches, and then chooses the plan with the most balanced degrees of saturation among competing approaches, thus minimizing delay.

4. INTERFACING PLANT AND CONTROLLER

Intuitively, the operation of the overall system is achieved by executing the plant (AIMSUN model) together with the controller (the Simulink model). Each of them (plant and controller) must also be able to exchange meaningful information at regular intervals with the other in order to enable meaningful interactions. This information exchange must happen over a well-defined interface between the plant and controller. In our case, we use network connections to connect the plant model in AIMSUN with the controller model in Simulink. We employ User Datagram Protocol (UDP) [REF], which involves minimal set-up and handshaking overheads.

In AIMSUN, we enable UDP communication using the Application Programming Interface (API) which interacts with the micro-simulator and allows the plant to send and receive information during simulation. At the end of every simulation cycle ends, the API is used to send loop detector information obtained from the simulator over UDP to the controller. The controller uses standard UDP components available in Simulink to receive this information, and then computes the phase splits, cycle length, and offset time for the intersection for the next cycle. This information is sent to the plant model via UDP, where the API receives it and modifies the simulation parameters appropriately.

5. RESULTS

Two experiments were carried out to test the improvement in traffic flow by using the Simulink controller based on SCATS. Firstly, a generic two intersection network was modelled in AIMSUN. The performance of the Simulink controller was compared with that of an optimised fixed sequence traffic model (with fixed cycle time, phases, and offsets). As Figure 8 shows, the Simulink controller improved delay time, stop time, and travel times by 30-40%.

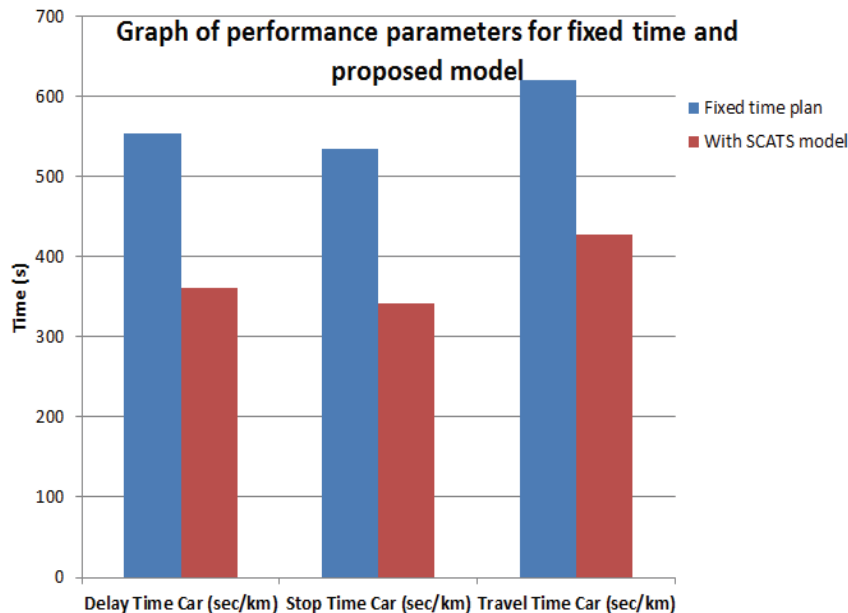


Figure 8: Performance comparison for simulated network (lower is better)

The next experiment involved the Balmoral-Dominion road intersection described in Sec. 2. This model was tested using measured traffic data for this intersection obtained from Auckland Transport. While the controller model does not faithfully model SCATS due to unavailability of several details about the SCATS algorithm (and also the unavailability of SCATS split plans used during measurements), as Figure 9 shows, the Simulink controller had a strong correlation with the SCATS controller at the intersection.

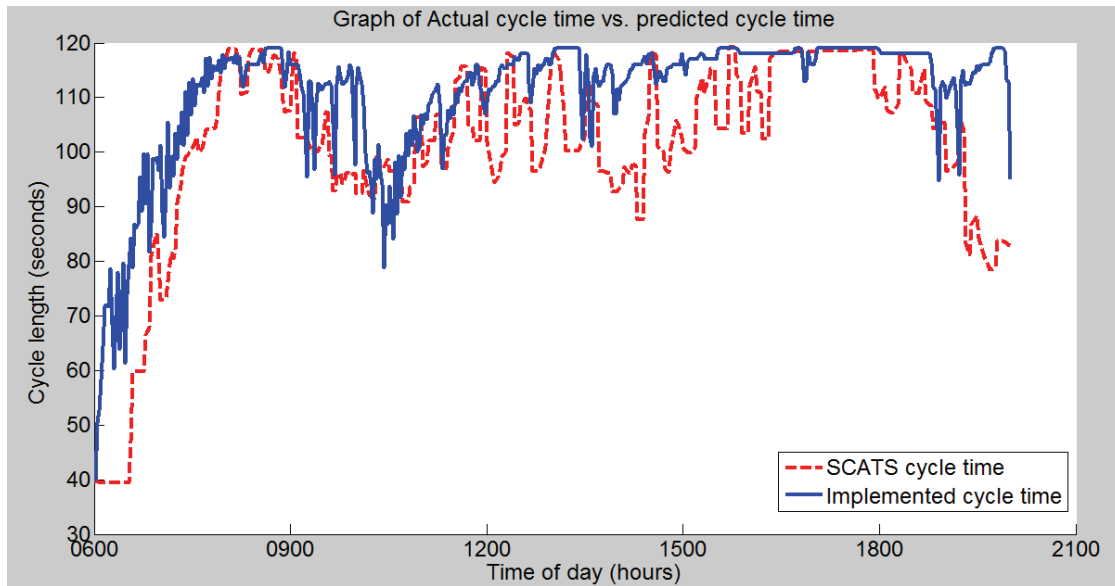


Figure 9: Comparison of SCATS and Simulink controllers for Balmoral-Dominion Roads intersections

A number of interesting observations can be made from the above results. Firstly, they clearly show that tools with completely different capabilities such as AIMSUN and Simulink, can be used to model different aspects of the system (plant and controller respectively) and can be unified using standard network interfaces to seamlessly create the complete system. Secondly, the model-driven engineering approach allows us to successively refine both the plant model as well as the controller model individually to improve the overall system. Thirdly, after validation, the controller model can be turned into C code directly that can be deployed on embedded traffic controllers.

6. CONCLUSIONS

We presented a model-driven engineering method to develop traffic controllers for coordinated intersections. In this process, we built a model of the road network using AIMSUN micro-simulator, and a model of the traffic controller using Simulink. These models are then connected using standard network connections so that the controller model can be tested and validated. We show the usability of this process by building a simple road network model in AIMSUN with a SCATS-inspired controller in Simulink Stateflow. Using standard UDP connections, we were able to test the effectiveness of the controller and compare it with a fixed sequence traffic control model, as well as with the measured performance of a SCATS controller. This framework can be used to model distributed controllers with ease, and can be used in the future to model and validate more sophisticated Intelligent Transportation Systems of the future.

7. REFERENCES

- [1] UNFCCC. (2012, 10 March). *United Nations Framework Convention on Climate Change*. Available: <http://www.unfccc.int>
- [2] P. Lowrie, "The Sydney coordinated adaptive traffic system-principles, methodology, algorithms," in *International Conference on Road Traffic Signalling, 1982, London, United Kingdom, 1982*.

- [3] D. I. Robertson and R. D. Bretherton, "Optimizing networks of traffic signals in real time-the SCOOT method," *Vehicular Technology, IEEE Transactions on*, vol. 40, pp. 11-15, 1991.
- [4] J. Barceló and J. Casas, "Dynamic network simulation with AIMSUN," *Simulation Approaches in Transportation Analysis*, pp. 57-98, 2005.
- [5] D. C. Schmidt, "Model-driven engineering," *COMPUTER-IEEE COMPUTER SOCIETY-*, vol. 39, p. 25, 2006.
- [6] R. France and B. Rumpe, "Model-driven development of complex software: A research roadmap," in *2007 Future of Software Engineering*, 2007, pp. 37-54.
- [7] J. Deacon, "Model-view-controller (mvc) architecture," *Online*[Citado em: 10 de março de 2006.] <http://www.jdl.co.uk/briefings/MVC.pdf>, 2009.
- [8] M. Simulink and M. Natick, "The Mathworks," *Inc., Natick, MA*, 1993.
- [9] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *Pervasive Computing, IEEE*, vol. 7, pp. 12-18, 2008.
- [10] S. Neema, "Simulink and Stateflow data model," *see www.isis.vanderbilt.edu*, 2001.