ELSEVIER

Transportation Research Part C 12 (2004) 73–89

TRANSPORTATION
RESEARCH
PART C

www.elsevier.com/locate/trc

Hardware-in-the-loop simulation

Darcy Bullock [a,*], Brian Johnson [b], Richard B. Wells [b], Michael Kyte [c], Zhen Li [c]

[a] *School of Civil Engineering, Purdue University, West Lafayette, IN 47907, USA*
[b] *Department of Electrical Engineering, University of Idaho, Moscow, ID 83844, USA*
[c] *Department of Civil Engineering, University of Idaho, Moscow, ID 83844, USA*

Received 4 May 2001; accepted 4 October 2002

Abstract

The current generation of macroscopic and microscopic simulation packages do not have control algorithms that incorporate many of the advanced features now available in commercial traffic control hardware. Consequently, there is a need to develop cost effective procedures for evaluating state of the practice traffic signal control equipment so that informed deployment and design decisions can be made. These same procedures can also be used by researchers to develop new algorithms for applications such as transit priority or adaptive control. In order to achieve that objective, this paper presents the motivation for using hardware-in-the-loop simulation procedures. Hardware-in-the-loop simulation presents a new set of challenges for traffic engineering model developers as the "correctness" of a real-time model not only depends upon the numerical computation, but the timeliness with which the simulation model interacts with external control equipment. This paper reviews the state of practice, summarizes the fundamental technologies necessary for implementing such a system, and uses a simple statistical test for assessing the real-time errors introduced into a simulation model.

Keywords: Traffic; Signal; Controller; Simulation; Hardware; Equipment

1. Motivation

Over the past half century, new traffic control procedures have been evaluated using macroscopic models, microscopic simulation models, or structured observation of field deployments.

* Corresponding author. Tel.: +1-765-494-2226; fax: +1-765-494-0395/496-1105.
E-mail address: darcy@purdue.edu (D. Bullock).

0968-090X/$ - see front matter
doi:10.1016/j.trc.2002.10.002

Macroscopic models, such as Transyt and Passer have been enormously useful for quickly evaluating benefits and designing fixed time signal plans (Robertson, 1969; Haenel and Messer, 1974). More recently some packages offer optimization of basic actuated-coordinated controllers (Husch, 2000). However, these models do not consider complex detector logic, shared right of way with light rail, or a myriad of control parameters available on modern traffic signal controllers. This discontinuity between macroscopic models and actual field equipment often leads to performance discrepancies when timing plans developed with macroscopic models are deployed. This discontinuity is particularly evident with emerging systems implementing concepts such as transit priority or adaptive control.

Alternatively, some agencies do not use any models, electing instead to directly modify signal timings in the field and directly observe the results of these changes. Such procedures can be quite effective, particularly for tuning splits and offsets to achieve local optima. These procedures are much less effective for evaluating alternative cycle lengths and the quality of timings plans developed is largely due to the diligence of the technician doing the work. Furthermore, because of the risks of making a big mistake that could lead to gridlock, agencies rarely try creative or innovative timing plans and only find "local optima" with these procedures.

In an attempt to provide a more realistic and uniform modeling procedure for engineers to evaluate alternative timing plans, microscopic simulation models have evolved over the last three decades (Farr et al., 1978) and provide very detailed animation of individual vehicle movements and controller operation. However, because of the competitive market for traffic signal controllers, each vendor uses different procedures and parameters for configuring their traffic control equipment. Consequently, the current generation of microscopic simulation models do not cover the full range of features available in modern traffic signal controllers. The only way to evaluate many of the emerging real-time adaptive control algorithms is to deploy them on the street, and observe their performance. For obvious reasons, it is extremely difficult, and often impossible, to obtain statistically sound before and after comparisons using field observations. Furthermore, direct experimentation with the motoring public requires extreme caution, which precludes trying a variety of innovative control models. The following section further details this problem.

2. Real-time traffic control in the United States

In recent years, there have been two parallel research paths for developing advanced traffic signal systems. Real-time traffic adaptive system research, supported largely by the United States Department of Transportation (USDOT) and smaller scale closed loop systems developed primarily by traffic signal system vendors.

Simulation models have been developed for evaluating USDOT supported real-time control projects, and those results have been reported in the literature (Andrews et al., 1991; Bullen and Memon, 1996; Clark et al., 1997; Head et al., 1992). However, even though those algorithms have undergone rigorous evaluation within the CORSIM model during development, the algorithms inevitably undergo significant revision when they are adapted to run in real-time on field traffic control devices. Consequently, the actual deployed algorithms differ in subtle, but significantly, ways that preclude direct application of a simulation model for operational analysis purposes.

A second class of real-time control systems, vendor-developed closed loop systems, have never had any macroscopic or microscopic model developed. These systems have evolved by vendors developing heuristic rules for selecting cycle, split and offsets. Even though there are several hundred traffic responsive systems deployed, none of the vendor developed closed loop signal systems have undergone rigorous evaluations. This is an area of significant concern because deployment of efficient closed loop signal systems is one of the most cost effective intelligent transportation system (ITS) investment that a small urban area can make. In order to make good deployment decisions, rational quantitative evaluation procedures are required to evaluate feasible alternatives and vendor claims (Bullock and Urbanik, 2000).

3. Evaluation procedures

Several microscopic simulation packages are available that model vehicle movement and basic coordinated-actuated signal logic (Shoup and Bullock, 1999). Ideally, model developers and vendors of traffic signal control equipment would work with developers of simulation models to integrate their control algorithms into simulation models. This has occurred on a limited basis in some European countries. However, due to a combination of vendor concerns with intellectual property and the additional engineering effort that would be required, no similar uniform efforts have been undertaken in the United States. Consequently, there is no generally available package that can be used for either quantitatively evaluating the performance of state-of-the-art algorithms, or to serve as a design tool for "tuning" system parameters prior to deployment.

As a result, the only studies agencies have available to assist in their design and decision-making process are "before–after" studies conducted with probe vehicles or vague performance relationships based upon system detector count and occupancies. Many of these studies use the old system with out dated timings as the "before" case so it is unclear if the benefits are simply associated with the new timings, or the new traffic responsive or traffic adaptive system. Furthermore, because of the natural stochastic variation of traffic, and huge costs associated with systematically collecting system performance data, few if any of the studies present robust statistical comparisons.

Based upon the above discussion, this paper makes the following assumptions regarding the evaluation procedure:

1. Any substantive comparison of traffic control algorithms must be done using the same traffic model to prevent model bias. In other words, Algorithm **A** cannot be evaluated using traffic model **B** and then compared with Algorithm **C** evaluated using traffic model **D**. From a scientific perspective this is quite obvious, but it is frequently overlooked.
2. A macroscopic model is not sufficient for evaluating proposed real-time control models. A microscopic model must be used to evaluate the second by second response of an algorithm to determine how robustly control algorithms reallocate split, cycles, and/or offsets to react to stochastic variations in traffic.
3. A procedure must be developed for integrating a variety of proprietary algorithms into a common traffic model. This would provide a mechanism for evaluating Algorithms **A** and **B** on the same traffic model, **E**.

4. Finally, once the evaluation procedures are implemented, formal assessment procedures must be developed for assessing the statistical significance of errors associated with introducing external control algorithms into the simulation model.

The following sections discuss the equipment and software architectures necessary for conducting the evaluation. Subsequent sections address the assessment.

4. Hardware-in-the-loop simulation concept

The common feature of all responsive traffic control systems is that they use some type of vehicle detection and change the display of signal indications according to some prescribed logic that is designed to optimize certain system measures of effectiveness (MOEs). Since, virtually all of the signal systems in commercial production implement their control logic on unique computing platforms and their algorithms are generally not available to the traffic engineering community for conducting a rigorous scientific evaluation, the only commonly available interface available for testing these algorithms is the discrete inputs and outputs available on traffic signal controller.

To address this systematic evaluation problem, there are several efforts in the United States to integrate microscopic simulation programs with the discrete interfaces on traffic signal control hardware to study the performance of vendor specific algorithms (Balke et al., 2000; Bullock and Catarella, 1998; Bullock et al., 1999; Engelbrecht et al., 1999; Husch, 1999; Koonce et al., 1999; Nelson and Bullock, 2000; Nelson et al., 2000). Fig. 1 depicts the typical configuration of a hardware-in-the-loop simulation for a NEMA closed loop traffic signal. Fig. 2a depicts the abstract links of Figs. 1 and 2b illustrates the three basic components of the interface:

- A controller interface device (CID). This device provides the interface from the traffic controller to the computer running a microscopic simulation. The interface is typically based upon the discrete controller inputs and outputs. These discrete states may be exchanged via the voltage levels used to drive the load switches and monitor loop detectors or perhaps via the SDLC interface used by emerging NEMA TS 2 Type 1 or 2070 controllers (NEMA, 1998).
- A software interface module to provide the linkage between the CID and a microscopic simulation program. Since most traffic simulation software runs under a version of Windows, this software interface is typically implemented in a dynamic link library (DLL) software module. However, alternative interface architectures are quite feasible.
- A microscopic simulation engine that is responsible for moving vehicles through a defined network and tabulating MOEs. The simulation engine does not implement any control logic. Instead, external signal state indications (RED, AMBER, and GREEN) are obtained from actual traffic signal control equipment which is connected to the simulation computer. The traffic signal control equipment is "stimulated" by detector calls (contact closures) placed by the simulation program via the CID.

Since all control equipment ultimately controls load switches and monitors detector calls, this discrete signal interface is the lowest common denominator interface that all controllers must

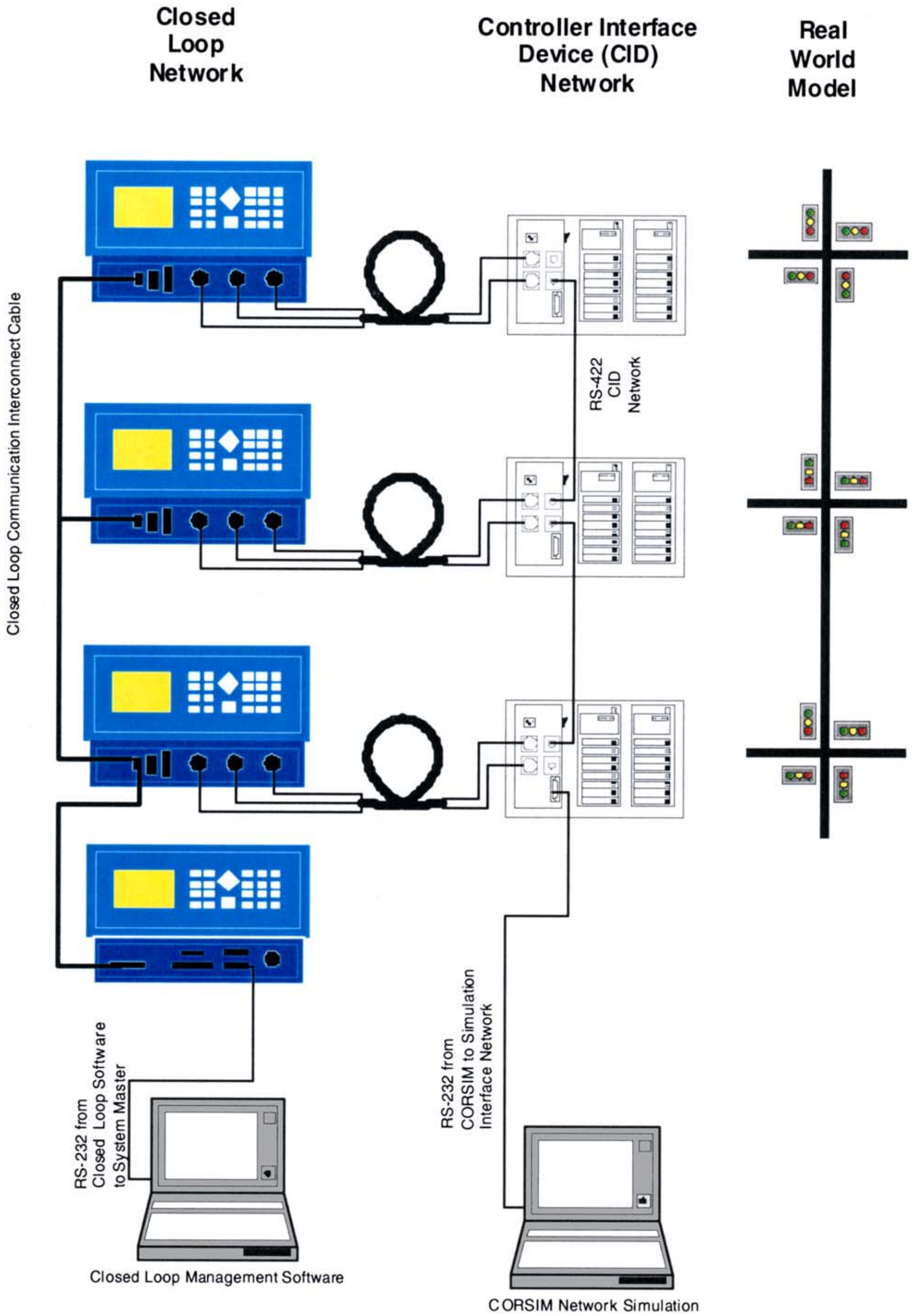


Fig. 1. Hardware-in-the-loop equipment.

have. Consequently, this architecture provides a common evaluation framework that a variety of signal control systems can be connected to for conducting scientifically rigorous and reproducible

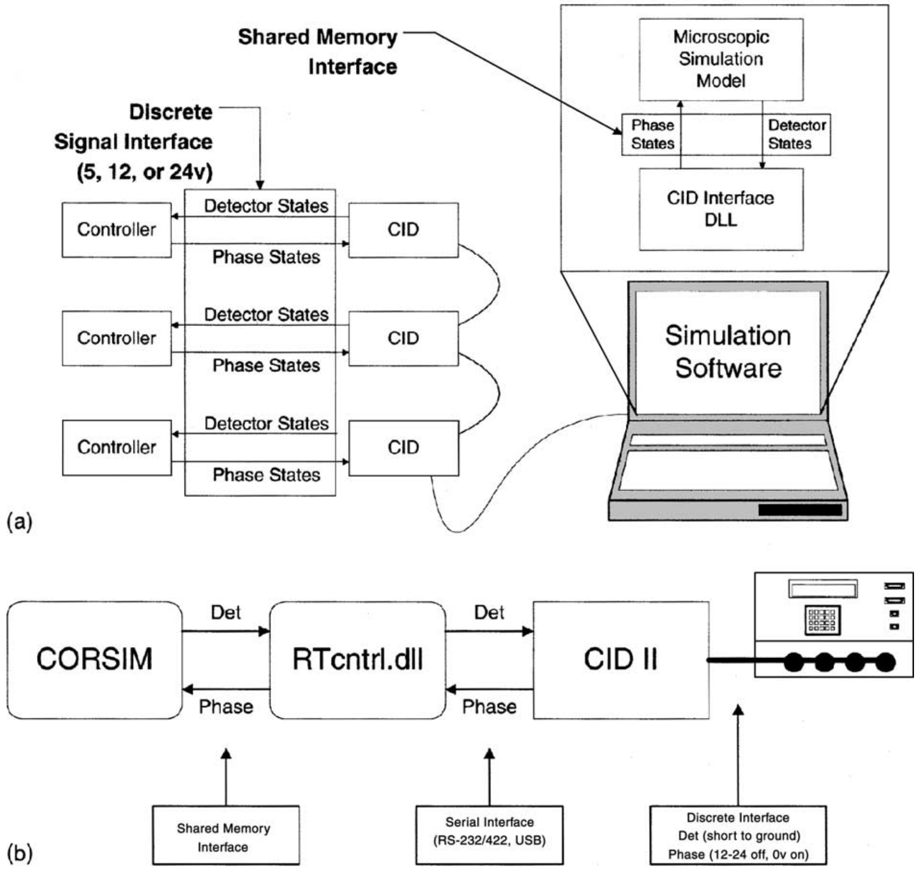


Fig. 2. Schematic diagram of hardware-in-the-loop interfaces. (a) Abstract interface diagram. (b) Interface diagram based upon CORSIM and CORSIM real time extension.

evaluations. Although not shown in Fig. 1, a typical simulation for an adaptive control system would have each controller connected to either a closed loop master or a central control system which would run an algorithm such as RHODES, UTCS, SCOOT, SCATS, or other emerging real-time control procedures (Bullock and Urbanik, 2000). Such a framework addresses the first three points identified in Section 3. Before proceeding to the assessment procedures, the following section describes the fundamentals of real-time computing theory necessary for addressing the third point—integrating algorithms into a common traffic model.

5. Application of real-time microscopic simulation technology

In order to make the evaluation system, shown in Figs. 1 and 2, useful for evaluating alternative control algorithms, it is essential that the CIDs be interfaced with a robust microscopic simulation program. The microscopic simulation is responsible for "moving" all vehicles through a user defined network following prescribed vehicle kinematics. This movement is performed by recalculating the position of each vehicle at a deterministic frequency, typically between 1 and 10 Hz. During each recalculation, vehicle accelerations in the simulation are updated in response to signal indications obtained from the CID and adjacent vehicles in the network. Also during each simulation interval, appropriate detectors states are updated via the CID. To ensure the occupancy calculated by the traffic controllers closely models field conditions, the duration of the presence detectors is inversely proportional to the velocity of the vehicle actuating the detector. This imposes a rather strict timing constraint on the system that can be addressed in one of two ways:

- *Detector durations timed on simulation computer*. There are two problems with this architecture. First, the typical operating systems used to run simulation programs (Windows 98, Windows NT, and UNIX) do not have sufficient real-time capability to effectively implement such timing. Second, even if an operating system did have real-time capability, the asynchronous communication between a CID and the microscopic simulation would not be deterministic. This lack of a deterministic communication link would occur during periods of heavy vehicle traffic and loop occupancy timers in the simulation all expired near or at the same time. This would result in some queuing of communication messages and hence the detector occupancy times would not be accurately timed.
- *Detector durations timed on the CIDs*. This architecture requires that the duration of the detector pulse be calculated by the simulation program and that pulse time is downloaded to the CID every time a detector turns on. The CID implements a precise timing routine that turns off the detector at the appropriate time. This architecture provides both the deterministic communication link and precise detector timing required for a real-time system. This architecture scales very well as the real-time processing is predominantly handled by the CID.

These timing procedures have been addressed both ways. Some have real-time simulations architectures have simply discretized the loop actuations to 100 ms or 1000 ms intervals (Engelbrecht et al., 1999; Husch, 1999). Alternatively, more sophisticated CID electronics can be used which is capable of individually timing detector pulses (Bullock and Catarella, 1998). For simple evaluation of coordinated-actuated control systems, discretizing detector output to 100 ms or 1000 ms resolution is adequate. However, if one want to evaluate the performance of traffic responsive systems (Nelson, 2000) that change cycle lengths based upon system detector occupancy values, the more precise timing of detectors by the CID should be used to ensure that occupancy values are correctly calculated.

6. Task scheduling

The final aspect of understanding how hardware-in-the-loop simulation works requires understanding the task scheduling and sequencing. There are three basic issues required to ensure real-time operation:

1. To ensure synchronization between the external controllers and the simulation, the simulation must be run in real-time.
2. The simulation needs to run in evenly space simulation time steps.

3. The simulation of each time step must run faster than real-time so that the interface software has time to run and wait for the real-time clock to reach the start of the next simulation period.

When a simulation program is being run without hardware connected, the simulation model runs a simulation time interval as fast as it can, without regard to real clock time. However, when hardware is connected the synchronization between the simulation program and traffic control equipment are a very big concern.

If the simulation runs slower then real-time, hardware-in-the-loop simulation is not feasible because it is virtually impossible to modify a traffic controller to run slower then real-time. If a simulation runs faster then real-time, hardware-in-the-loop simulation is feasible. However, when real traffic controllers are connected, the simulation must be slowed down to real-time. In other words, the clock of the simulation must match the real-time clock that the controllers are running to ensure synchronization. For example, if the simulation model has a fixed update period of 1 s, the interface software must also retrieve the phase indications from the controllers and send the detector calls to the controllers at a 1 Hz rate. Intuitively, a 1 Hz simulation rate is about the lower bound on how often the simulation program should run when one is modeling actuated controllers with gap times on the order of 1.5–3.0 s. Faster update rates are more desirable as they will reduce the time lag between the simulation model and real-time control equipment. This results in better synchronization, which reduces errors associated with introducing hardware.

Fig. 3 schematically shows how the real-time interaction works over several simulation periods, P. When the simulation task finishes updating the vehicle positions, it immediately calls an interface function to exchange data with the CID. Since the actual serial communication can be carried out by an asynchronous serial task, the task that actually sends and retrieves serial data to the CIDs is shown as a different task in Fig. 3. The key concept is that the simulation model and the interface tasks must complete execution *every* simulation period. This means the combined execution time should never take more then $1P$, where P is the time step of the simulation program. This can easily be checked having the simulation program keep a log of the actual clock time every time the simulation model is run. For example, this log could then be reviewed to ensure that the simulation for a 15-min period really ran in exactly 15 min. If it did not, the specific time period where problems occurred could be identified.

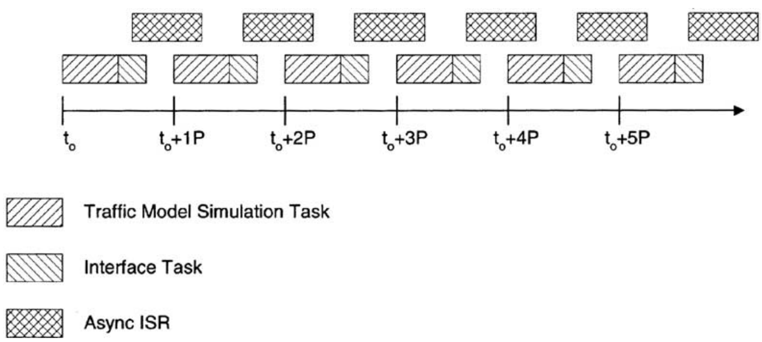


Fig. 3. Task sequence and scheduling for real-time simulation.

7. Assessment of real-time simulation errors

As indicated in the preceding sections, hardware-in-the-loop simulation can provide a very powerful environment for comparing alternative real-time control models. However, the development of real-time interfaces provides new challenges, which if not designed correctly, can introduce significant errors into the simulation. Although not a comprehensive real-time test, a procedure for determining if the hardware-in-the-loop simulation procedure introduces errors in the MOEs tabulated by a simulation model has been proposed (Bullock and Catarella, 1998). In that procedure a basic actuated controller is simulated using both the internal simulation model and hardware-in-the-loop controller. Both simulation runs are replicated five times with the same set of five initial random number seeds. Mean and standard deviation values are tabulated and the means are compared using a standard statistical test to determine if there is evidence to suggest the mean values are different.

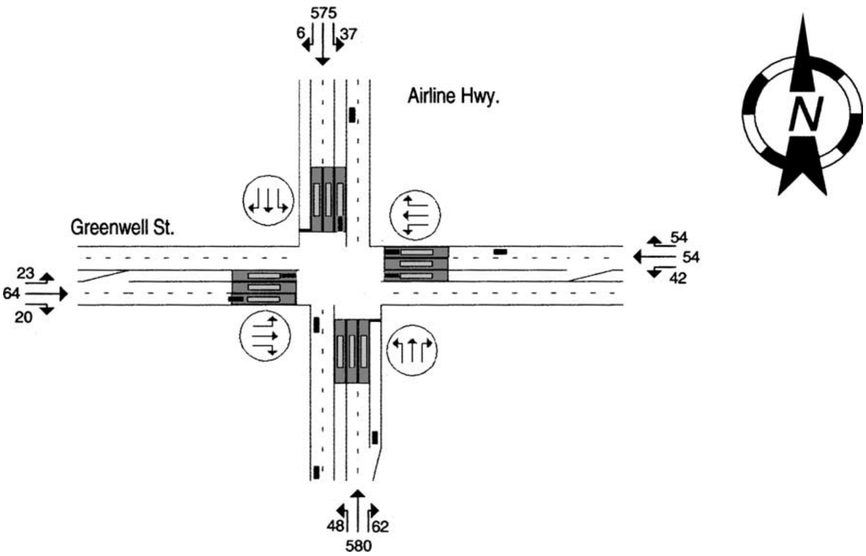
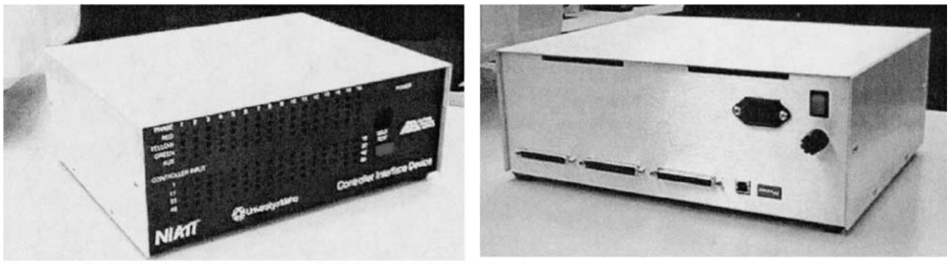


Fig. 4. Geometric layout and demand volumes (vph) for modeled intersection.

Table 1
Phase table entries used by control algorithms

Phase	Interval times							
	1 SBL	2 NBT	3 WBL	4 EBT	5 NBL	6 SBT	7 EBL	8 WBT
MIN GREEN	3	15	3	5	3	15	3	5
GAP, EXT.	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
MAX 1 GREEN	30	99	30	40	30	99	30	40
SPLIT	20	40	20	20	20	40	20	20
YELLOW	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0
RED	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0

For the intersection shown in Fig. 4 and control parameters listed in Table 1, this test was performed for the NIATT CID shown in Fig. 5. The results of that test are shown in Tables 2–4. Table 2 compares the total movement delay and flow rate obtained using both the internal fixed time control model and hardware-in-the-loop simulation with a Peek LMD 8000 running in fixed time. As you can see the numerical difference between the CORSIM simulated controller and the External NEMA controller is very small. Table 3 tabulates the same data, but for actuated control. Table 4 tabulates t-statistics to determine if there is evidence of statistical difference (between a CORSIM simulated controller and one simulated using hardware-in-the-loop) in the mean values shown in Tables 2 and 3. The comparison is very good, with the exception of the flow



(a) (b)

Fig. 5. Photograph of NIATT CID IId production prototype. (a) Front panel with LED display. (b) Backside view of connectors, USB port, and dip switches.

Table 2
MOE mean values for CORSIM algorithm vs. hardware implemented fixed time control algorithm

	CORSIM simulated controller		External NEMA controller	
	Delay (veh-min)	Flow (vph)	Delay (veh-min)	Flow (vph)
NB approach				
Left	9.7	41	10.1	37
Through	42.9	594	41.1	576
Right	2.3	62	2.3	62
SB approach				
Left	7.6	34	10.0	34
Through	40.2	585	42.8	569
Right	0.2	6	0.2	5
EB approach				
Left	5.9	25	6.5	25
Through	10.5	54	11.7	55
Right	1.6	26	1.5	26
WB approach				
Left	7.5	39	10.4	40
Through	11.5	55	12.8	55
Right	3.3	50	3.1	51

Table 3
MOE mean values for CORSIM algorithm vs. hardware implemented actuated control algorithm

	CORSIM simulated controller		External NEMA controller	
	Delay (veh-min)	Flow (vph)	Delay (veh-min)	Flow (vph)
NB approach				
Left	10.4	45	13.4	45
Through	30.6	587	32.1	578
Right	2.4	61	2.5	61
SB approach				
Left	9.0	36	10.6	38
Through	27.9	581	31.6	569
Right	0.2	5	0.2	5
EB approach				
Left	7.1	31	8.5	31
Through	10.6	55	11.4	57
Right	1.5	19	1.2	19
WB approach				
Left	8.1	37	11.2	38
Through	11.2	53	13.4	55
Right	4.2	55	4.4	55

Table 4
Test statistics comparing MOE mean values of vehicle delay and vehicle low for both fixed time and actuated control

	t-Test statistic			
	Fixed time control		Actuated control	
	Delay	Flow	Delay	Flow
NB approach				
Left	0.24	−0.89	1.78	0.00
Through	−0.59	−2.25	0.66	−2.64
Right	0.14	0.00	0.09	0.00
SB approach				
Left	0.67	−0.05	0.68	0.17
Through	1.19	−1.79	1.78	−1.25
Right	−0.02	−0.34	−0.20	0.00
EB approach				
Left	0.35	0.00	0.99	0.12
Through	0.52	0.19	0.52	0.27
Right	−0.13	0.00	−1.29	0.00
WB approach				
Left	1.78	0.28	1.35	0.22
Through	0.90	0.00	1.90	0.16
Right	−0.48	0.20	0.24	0.00

rates for the Northbound through approach. However, when one compares the numerical flow rates (fixed time 594 vs. 576, and actuated 587 vs. 578), they are within 3% and 1%, respectively. Further work is currently underway to determine if the statistical comparison shown in Table 4 is the most appropriate for this hardware.

8. Hardware-in-the-loop simulation diagnostic tools

Configuring a hardware-in-the-loop simulation requires mapping every detector and phase indication used in the simulation to the corresponding detector or phase on the controller. Just as wires are sometimes crossed in the field, this sometimes happened when setting up a hardware-in-the-loop simulation. Consequently, it is important to have a suite of test tools and probes for identifying and fixing these sorts of problems. Fig. 6 illustrates the software equivalent of a "suit case tester" used to identify these sorts of errors.

9. Education and training applications of hardware-in-the-loop-simulation

Once a complete hardware-in-the-loop environment is assembled, a "flight simulator" type experience can be constructed for training personnel new to the profession. Such an environment allows experience-based learning exercises demonstrating various "what-if" scenarios. This type of system has application to a variety of educational efforts including college engineering curricula, continuing professional engineering education, and training of technicians responsible for daily operation and maintenance of the system. Fig. 1 illustrates the concept and Fig. 7 shows example screens.

Fig. 6. Example "suit case tester" screens used to setup hardware-in-the-loop simulation: (a) TS 1 display and (b) TS 2 display.

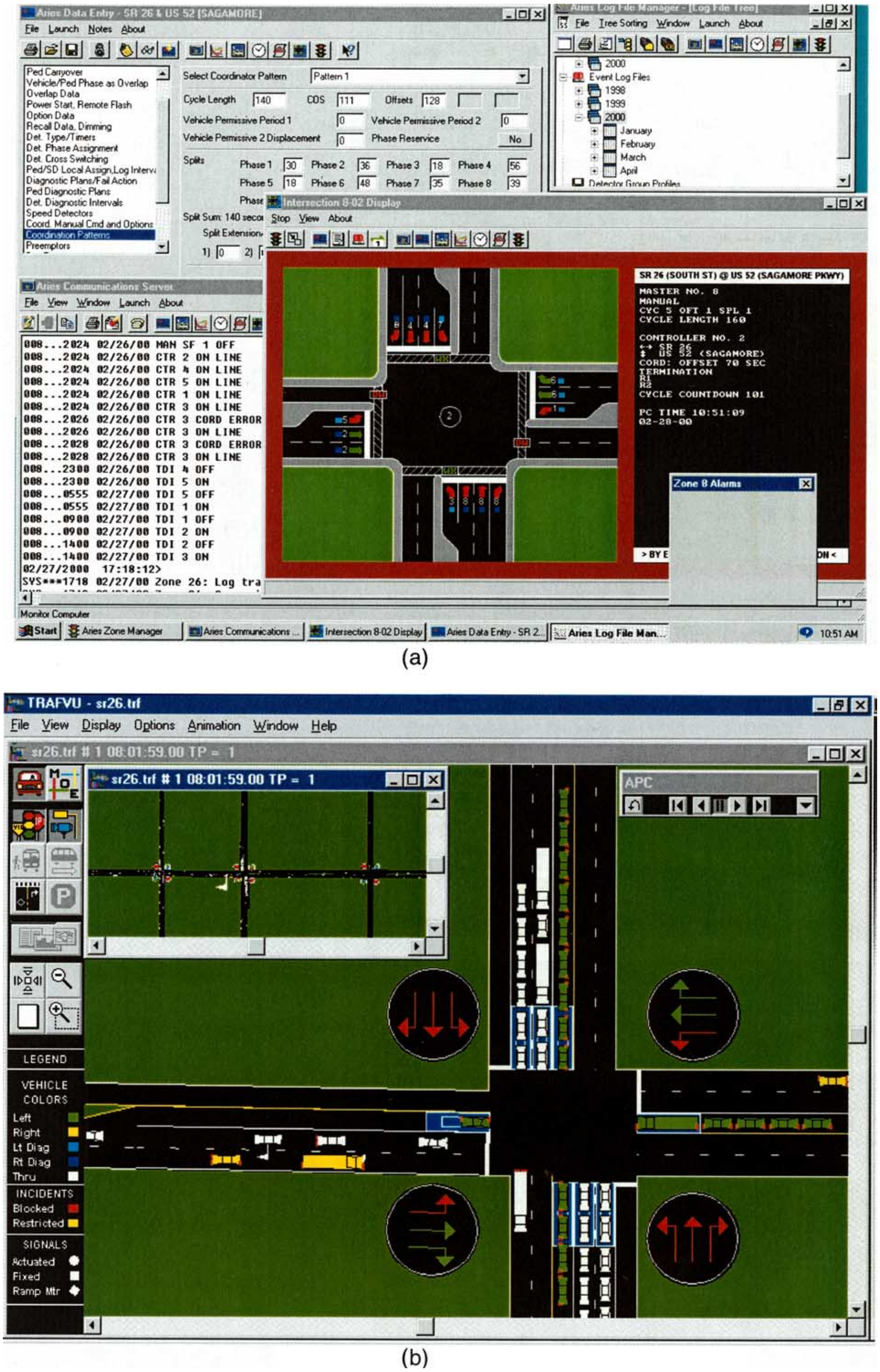


Fig. 7. Hardware-in-the-loop-experience. (a) Example closed loop management software (Aries by Econolite). (b) CORSIM simulation screen.

- Fig. 7a illustrates an operator's view of the signal system being managed. Although there is a lot of information on the screen, there is very little intuitive information that at a glance shows how the network is performing. Such a view is analogous to pilot trying to assess the status of his aircraft without looking out the window.
- Fig. 7b illustrates how the network is behaving in response to the operators actions. Although this screen is visible in real-time, it is often more useful for the student to only view this screen after the training event is over because this view would only be available to an operator if he was in a helicopter looking down on the system.

Such an environment provides an invaluable education tool that permits students to make mistakes, create huge backup traffic backups (in the simulation) and learn from the mistakes. Such practical experience based learning is very difficult to obtain in the field because of the political impact when such mistakes happen.

In addition to the operational training issues illustrated in Fig. 7a and b, this environment is extremely useful for training technicians on the nuances associated with configuring telemetry and the details associated with adjusting a sensitive parameter, implementing a novel controller feature or deploying a complete closed loop system (Koonce et al., 1999; Nelson, 2000; Nelson and Bullock, 2000).

10. Application of CID technology and future challenges

Over the past twenty years, traffic signal systems have evolved from relatively intuitive electromechanical systems to complex distributed control systems. Although these modern systems tend to be less fragile than their predecessors, the additional complexities associated with these new architectures have made it extremely difficult to successfully deploy large systems. These difficulties have not gone unnoticed by our elected officials (GAO, 1994). There is a very urgent need to develop and implement rigorous systematic procedures for testing and evaluating traffic signal systems prior to deployment on the street in order to regain some of the profession's lost credibility.

However, before the traffic engineering profession can adopt more sophisticated testing procedures, it is essential to understand the fundamental real-time control issues that effect hardware-in-the-loop simulation. This paper reviewed several recent efforts that have advanced the development hardware-in-the-loop testing and contrasted alternative computing implementation architectures. Several issues were covered in significant detail, but the fundamental issue the reader should take from this paper is that hardware-in-the-loop simulation is a real-time system where the "correctness" of the system is not only dependent upon a calculation in the simulation program being properly performed, but also the timing with which the simulation program interacts with the equipment being tested. For one simple intersection, the timing constraints are not very restrictive; perhaps detector resolutions within 1 s would not introduce any statistically significant error. However, when more sophisticated systems comprised of a dozen or more intersections each tabulating and reacting to detector occupancy times are considered, the communication protocol and computing architecture are critical to ensure an accurate hardware-in-the-loop simulation.

Fig. 5 shows a photograph of the NIATT CID IId that was recently developed to facilitate large scale hardware-in-the-loop testing. This device is based upon the universal serial bus (USB) interface and has a distributed implementation where the USB protocol communicates detector start time and durations to the CIDs. The CIDs then time the detector pulses to the nearest millisecond and communicate the status of their respective traffic controller back to the simulation program. The CID IId shown in Fig. 5, has discrete interface connectors allowing it to be connected to any controller with discrete interfaces such as NEMA TS 1, NEMA TS 2 Type 2, 170, or 2070 controllers.

Fig. 8 illustrates the vision for how the CID II can be applied in practice. The first four CIDs (ID 1, 2, 3, and 4) are referred to as CID IId because they all have the same discrete input and outputs, but with different cable harnesses for attaching direction to specific controllers or cabinets. The initial deployment of the CID IId is will support a NEMA TS1, NEMA TS2 Type 2, 170, or 2070 controller.

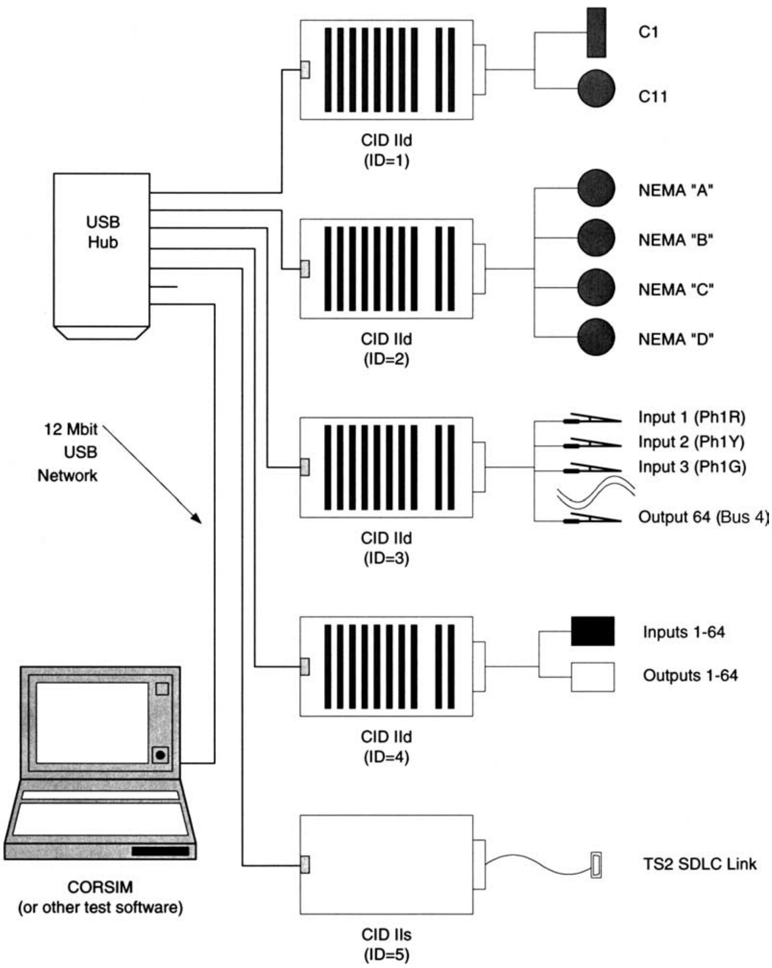


Fig. 8. Vision for family of CID II devices.

The successor the CID IId will be the CID IIs supporting the NEMA TS2 Type 1 SDLC interface. This CID Ms will use the identical USB protocol, but instead of using discrete I/O points it will use an SDLC interface to connect to the controller. In contrast to existing SDLC suit case testers, the NIATT CID will have the unique advantage of having a microprocessor on board managing the timing of each detector output. This will allow the NIATT CID II to scale to the build out of 20 controllers without saturating the communication link and still providing precise detector pulses. Also, since the CIDs will use the same USB protocol, both TS2 Type 1 and Type 2 controllers can be mixed in the same simulation.

11. Conclusion

This paper reviewed both the need and theory of hardware-in-the-loop-simulation. Subsequent sections identified the basic real-time computing issues and statistical tests that should be performed to validate any simulation where new control algorithms are introduced. The paper concludes by summarizing the state of the practice and lays out a framework for including an even more diverse set of traffic control equipment. The intent of this paper is to provide a reference for the application of hardware-in-the-loop simulation so that the profession works to develop procedures that permit rigorous and systematic evaluation of alternative control algorithms using a common microscopic traffic model. As shown in Table 4, the current generation of CIDs still produce a few MOEs that show statistically significant different in Mean values. As these systematic evaluation procedures become more main stream, it is expected that the hardware-in-the-loop simulation will mature and one could imagine vendors of traffic control software developing standardized application program interfaces (APIs) that would allow control models to be directly integrated into a software simulation. However, due to intellectual property issues as well as diverse real-time software and hardware engineering procedures currently used by traffic control vendors, the development of standardized APIs permitting "software-in-the-loop" simulation with a standard simulation program is several years away.

References

Andrews, C.M., Gartner, N.H., Tarnoff, P.J., 1991. Evaluation of optimized policies for adaptive control strategy. Transportation Research Record 1324, 105–114.

Balke, K., Dudek, C., Urbanik, T., 2000. Development and evaluation of an intelligent bus priority concept. Transportation Research Board Annual Meeting, National Research Council, Washington, DC. Preprint #00-1543.

Bullen, A.G.R., Memon, G.Q., 1996. Multivariate optimization strategies for real-time traffic control signals. Transportation Research Record 1554, 36–42.

Bullock, D., Catarella, A., 1998. A real-time simulation environment for evaluating traffic signal systems. Transportation Research Record 1634, 130–135.

Bullock, D., Urbanik, T., 2000. Hardware-in-the-loop evaluation of traffic signal systems. In: Proceedings of the 2000 IEE Conference on Road Transport Information and Control, April 4–6.

Bullock, D., Morales, J.M., Sanderson, B., 1999. Impact of signal preemption on the operation of the Virginia route 7 corridor. In: Proceedings of the 1999 ITS America Conference.

Clark, J., Pooran, F., Tarnoff, P., 1997. A suite of strategies for signal control. Traffic Technology International (June/July), 28–32.

Engelbrecht, R., Poe, C., Balke, K., 1999. Development of a distributed hardware-in-the-loop simulation system for transportation networks. Transportation Research Board Annual Meeting, National Research Council, Washington, DC. Preprint #990599.

Farr, R.M., Gibson, D., Kennedy, W.J., Labrum, W.D., 1978. Analyzing intersection performance with NETSIM. Public Roads 42 (1), 24–29.

GAO/RCED-94-105, 1994. Transportation infrastructure, benefits of traffic control signal systems are not being fully realized. Report to the Chairman, Committee on Energy and Commerce, House of Representatives, United States General Accounting Office, Washington, DC, March.

Haenel, H.E., Messer, C.J., 1974. A report on the user's manual for progression analysis and signal system evaluation routine—PASSER II. Texas Transportation Institute, TTI-218-72-165-14 Intrm Rpt.

Head, K.L., Mirchandani, P.B., Sheppard, D., 1992. Hierarchical framework for real-rime traffic control, Transportation Research Record, #1360, TRB, National Research Council, Washington, DC, pp. 82–88.

Husch, D., 1999. SimTraffic C1 Supplemental Guide. Trafficware, Berkely, CA.

Husch, D., 2000. Synchro Users Manual. Trafficware, Berkely, CA.

Koonce, P., Urbanik, T., Bullock, D., 1999. Evaluation of diamond interchange signal controller settings by using hardware-in-the-loop simulation. Transportation Research Record 1683, 59–66.

Nelson, E., 2000. Quantitative evaluation of closed loop signal systems. Master of Science Thesis, Purdue University.

Nelson, E., Bullock, D., 2000. Impact evaluation of emergency vehicle preemption on signalized corridor operation. Transportation Research Record, #1727, TRB, National Research Council, Washington, DC, pp. 1–11.

Nelson, E., Bullock, D., Urbanik, T., 2000. Actuated control of diamond interchanges. Journal of Transportation Engineering, ASCE 126 (5), 390–395.

NEMA, 1998. National Electrical Manufacturers Association, Standards Publication No. TS 2, Washington, DC.

Robertson, D., 1969. TRANSYT: A Traffic Network Study Tool. Road Research Laboratory, Rrl Bpt Lr 253.

Shoup, G., Bullock, D., 1999. Performance evaluation of coordinated-actuated traffic signal systems. In: Proceedings of 1999 Institute of Transportation Engineers Annual Meeting, Las Vegas, NV.