REALTRAN: An Off-Line Emulator for Estimating the Effects of SCOOT

H. RAKHA AND M. VAN AERDE

An off-line emulation tool, entitled REALTRAN (REAL-time TRANsyt), which can emulate the SCOOT version 2.2 signal optimization logic, has been developed. REALTRAN was derived from the TRANSYT-7F model (TRANSYT version 7F) by introduction of various constraints into the optimization logic of TRANSYT-7F. These constraints allow the user to select optimization parameters that enable REALTRAN to operate in a fashion similar to that of the original SCOOT signal optimizer logic. The REALTRAN model is currently intended to serve as an educational tool but can, in the future, serve as a tool for fine tuning the operation of the real-time controls of the SCOOT system in a laboratory environment, where scientific and statistically valid testing and sensitivity analyses of the signal optimization algorithms can be performed. Alternatively, REALTRAN can be utilized to estimate off-line the expected benefits of SCOOT by use of location-specific network and flow data.

Field studies have indicated that various real-time urban traffic control (UTC) systems, including the SCOOT system, are capable of attaining reductions in the range of 10 percent in the network travel time compared with conventional fixed-time signal control (*1*). However, it has been impossible to achieve these reductions consistently. It appears that the main reasons for the lack of more extensive success of these real-time UTC systems are related to the complexity of the problem, the variability of the link flows, and the inaccuracy of the vehicle detector measurements. To address each of the factors there is a need for tools to fine-tune the operation of the real-time controls reliably off-line in a laboratory environment. This environment would allow sensitivity analyses on the settings and signal optimization algorithms of such real-time UTC systems to be performed.

Initially, the structure of the TRANSYT model is reviewed, as this model forms the basis for both the SCOOT system and the REALTRAN model. The SCOOT system is also described in the following section because the REALTRAN model can, depending on user-specified inputs, attempt to replicate the SCOOT signal optimization logic. It must be noted at this point that any reference to TRANSYT in this paper refers to the general optimization logic of TRANSYT including that utilized in the TRANSYT-7F version.

First, the general concept of the REALTRAN model is presented. Subsequently, the details of the REALTRAN hill-climbing procedure and the associated use of high-sensitivity parameters are presented, together with the cycle-length optimization process and a description of how the cycle-length, offset, and phase-split optimization procedures operate together.

Because of the limited space available in this paper, in the following section only a simple example illustration is presented to illustrate briefly how the REALTRAN model makes frequent but minor alterations to the signal settings such that the signal plan evolves incrementally toward the near-optimum signal settings. A more detailed application of the REALTRAN model can be found in the literature (*2*). Finally, a summary and the conclusions of the paper are provided.

Department of Civil Engineering, Queen's University, Kingston, Ontario K7L 3N6, Canada.

BACKGROUND

In this section we describe the more macroscopic concepts of the TRANSYT model, followed by a microscopic description of the TRANSYT hill-climbing procedure. The latter detailed description will provide the reader with an appreciation of the difference between the TRANSYT and SCOOT logic. In addition, this section provides a description of the SCOOT signal optimization logic before describing how REALTRAN can model the SCOOT logic.

Conceptual Description of TRANSYT

The TRANSYT model is perhaps the most widely used off-line signal optimization tool (*3*). The TRANSYT model was developed at the Transport Research Laboratory, and many versions have evolved. One of these TRANSYT versions is TRANSYT-7F, which was developed at the University of Florida (*4*).

TRANSYT is a macroscopic, deterministic simulation and optimization model. The model requires the link flows and link turning proportions as inputs and assumes them to be constant for the entire simulation period. The TRANSYT program simulates the traffic conditions for the duration of one complete cycle length, and these conditions are assumed to be representative of all other cycles.

The TRANSYT model is macroscopic because the traffic module models the flow of vehicles as cyclic flow profiles (CFPs) rather than modeling individual vehicles. Specifically, the cycle length is divided into a number of short time steps, which are typically 1–5 sec long. The CFP records platoons of vehicles as successive steps within the representative cycle, and the shape of the CFP is calculated by the model for each one-way flow in the study area.

TRANSYT Hill-Climbing Procedure

The TRANSYT program carries out a sequence of iterations between the traffic simulation module and the signal setting optimization module. For the initial signal settings the traffic module estimates the performance index (PI) by simulating the traffic as it reaches each intersection in the network. Subsequently, alterations are made to the signal settings by the optimization module. These

signal setting changes are sent to the traffic module, which alters the CFP that leaves each signal, which in turn affects the arrival profile at any downstream signals.

The TRANSYT program searches for the optimum signal settings by using a two-stage procedure. In the first stage the optimum cycle length is found through a search, at user-specified intervals, within a user-specified range of minimum and maximum cycle lengths. Subsequently, in the second stage, the cycle length that produced the lowest PI in the former search is investigated in further detail by a hill-climbing procedure to determine the optimum offsets and phase splits for this cycle length.

As the shape of the PI objective function versus offset, phase split, and cycle length is not always convex, local minima may exist. Thus most conventional derivative methods would fail to find the global minimum and could frequently be caught in local valleys. The offset and phase-split searches verify that the global minimum PI is found by use of a combination of small, medium, and large step sizes that usually move the optimizer away from a local minimum. Although there are no absolute guarantees that the search will find the global minimum, the TRANSYT heuristic has during the past 25 years been found to yield a very practical trade-off between accuracy and efficiency. Considerable work has been conducted to test and evaluate other search methods; however, no major improvements have been made to the TRANSYT search heuristic (*5,6*).

Overview of the SCOOT System

The SCOOT real-time UTC software (*7,8*) uses a traffic simulation model similar to that used by TRANSYT (*9*). This simulation model is used on-line, however, during every cycle by the optimizer to evaluate alternative signal timings and thus find the best signal settings based on the prevailing dynamic traffic conditions. The objective of SCOOT, as in the TRANSYT model, is to minimize the PI. Traffic is also modeled as a CFP in the SCOOT traffic model; however, the time interval is fixed at 4 sec, and each link's inflow CFP is measured directly from the street by detectors, as opposed to being inferred from the turning movements of the upstream intersection.

The SCOOT optimizer updates the traffic signal plan on a cycle-by-cycle basis. In doing this the optimizer uses the previous cycle's signal settings as a seed in the search for new timings and makes minor, but very frequent, alterations to these seed signal settings. The changes to the signal settings are made based on a restricted search for a minimum PI in the immediate vicinity of the seed signal settings, rather than by an exhaustive search for a global minimum PI, as in TRANSYT. The SCOOT signal optimizer effectively uses an elastic coordination plan that stretches and shrinks the coordination scheme to match the latest situation recorded by the real-time cyclic flow profiles. The changes made to the current plan, while minor, are frequent, so that over time the plan evolves considerably without causing major disruptions to traffic.

The three key principles of the SCOOT real-time UTC system that make it different from the TRANSYT model are as follows:

- to measure the cyclic flow profile in real time as opposed to deriving it from upstream turning movements,
- to update an on-line model of queues continuously as opposed to only updating once, and
- to make incremental as opposed to global optimizations to the signal settings.

OVERVIEW OF THE REALTRAN CONCEPT

We developed the REALTRAN model by adding to and altering the TRANSYT-7F optimization logic to perform the following functions: (a) to estimate iteratively the optimum signal timings of a network of traffic signals for a time series of link flows, (b) to evaluate these optimum signal timings, using a second set of link flows, and (c) to allow the user to specify constraints to the standard TRANSYT signal optimization logic.

The REALTRAN model, as does the SCOOT logic, involves the application of the TRANSYT optimization module every minute to an externally specified data stream. Each TRANSYT application is seeded with the signal timings that were found during the previous minute. The search can be constrained, depending on the user-specified parameters, to look only for those new signal timing solutions that are very similar to the previous minute's signal timings. The use of a good seed, plus the constraints on the optimization, can therefore significantly reduce every minute's computational requirements, because the optimizer starts from signal settings that are already very close to the optimum signal settings. Furthermore, as only minor changes are made to the signal settings, this approach can also avoid disruptions to the traffic during signal plan changes in a fashion similar to the SCOOT logic.

Details of the Optimization Procedure

The REALTRAN model can perform standard TRANSYT signal optimization, a restrained SCOOT-like signal optimization, or any user-specified restrained signal optimization, depending on the user-specified input parameters. The REALTRAN model, in simulating the SCOOT logic, makes three main restrictions to the hill-climbing process of the TRANSYT model, as follows: constraining the offset optimizer by allowing changes of only a few seconds for each optimization, constraining the phase-split optimizer to only a few-second changes, and making cycle-length optimizations at intervals of not less than 3 min, using a limited range of potential cycle-length choices.

The first key to the potential success of this effort derives from forcing the TRANSYT optimization to start the search for signal timings for the subsequent minute at the signal timings that were found at the conclusion of the previous minute.

The second key to the practical success of this effort derives from the implementation of a user-specified constraint on the number and size of the optimization steps that can be taken each minute to find improvements on this previous minute's timings.

Although the above two steps toward making a real-time version of TRANSYT satisfy the offset and phase-duration considerations of REALTRAN, a further addition to the logic was required to enable TRANSYT to mimic SCOOT's changes in cycle length. Specifically, at user-specified time intervals (typically every 2–5 min) the model determines whether the overall network PI can be decreased by moving to either a longer or a shorter cycle length. The maximum amount of permitted change in the cycle length is again user specified and cycle-length dependent to replicate SCOOT's different cycle-length increments.

Details of Input Requirements

The REALTRAN simulation program requires four input files, namely, a master file, the standard TRANSYT input file, a link flow

file to be used for signal optimization, and a link flow file to be used for the evaluation of the signal settings.

In the master file the names of the various input and output files are specified. In addition, the cycle-length increment thresholds, cycle-length increments, and the maximum number of steps to be used by the hill-climbing procedure are specified. This provides the user with the flexibility of testing different optimization constraints on the potential traffic signal settings. Furthermore, the "optimization link flow file" identifies the flows to be used by the optimizer to select the new signal settings each minute, and the "evaluation link flow file" identifies the flows to be used in evaluating these new signal settings. In this fashion the model can simulate either a time lag in the optimization procedure or the fact that the link flows input to REALTRAN may be filtered flows and thus differ from the actual flows.

The above input data requirements imply that the present version of the REALTRAN program is intended to be a simulation model that can replicate SCOOT's real-time controls and not a real-time control system that is to be a competitor for SCOOT. What is important, however, is that the user either can specify the cycle-length thresholds, increments, and frequency of full optimization as those used by the actual SCOOT system, making the REALTRAN model simulate control algorithms in a fashion similar to the SCOOT signal optimizer, or vary these parameters to study the impact on the PI.

SPECIFICS OF THE REALTRAN OPTIMIZATION MODULE

The main reason for the success of the modified hill-climbing routine that is described in this section is the elimination of the arbitrariness of the seed solution that is utilized to initiate the search for the next minute's signal timings. In this section we discuss the modified hill-climbing module in further detail and also illustrate how the use of high-sensitivity parameters in the standard TRANSYT input file can help to speed up the optimization process and assist in modeling mini-areas to mimic the SCOOT logic. We also briefly describe the cycle-length search process and the combined cycle-length, phase-split, and offset optimization process.

Modified Hill-Climbing Module

We mimicked the incremental nature of SCOOT in REALTRAN by setting up the modified hill-climbing module so it uses the previous interval's signal timing settings as the seed to initiate the search for the optimum signal settings for the following time period. Figure 1 demonstrates some of the details of how the modified hill-climbing module operates by using a typical two-step constraint, for all possible combinations of the shape of the PI curve, as indicated below.

Case (*1*) illustrates an optimization scenario in which an initial step reduces the PI and therefore leads the algorithm to proceed with a second step. This second step leads to a further reduction of the PI. However, as the limit of a maximum of two steps in a given search direction prevents the algorithm from proceeding any further, the signal settings that are found following the first two steps are retained and are considered to be the new approximation of the global optimum.

In Case (*2*) of Figure 1 the first optimization step is again shown to reduce the PI. However, when the second step is taken in the same direction, the PI is shown to start to increase again. Consequently it appears that the algorithm has found a local minimum, and the algorithm returns to the signal timing settings that were found following the first step as the new-found approximation to the global optimum signal settings.

Case (*3*) illustrates how an attempted shift in the signal timings to the right results in an increase in the PI. Consequently the algorithm reverses its search direction and doubles its step size to make a shift in the signal timings to the left past the initial signal settings and returns the step size to its original value. This move is shown to lead to a reduction in the PI compared with the initial settings. The limit of a maximum of two steps in any direction prevents the algorithm from proceeding any further in this direction. Case (*4*) is similar to Case (*3*) but makes only one step to the left.

Finally, the example in case (*5*) illustrates that a shift in either direction leads to an increase in the PI. Consequently the optimum signal settings are considered to be retained by simply keeping the initial signal settings that existed when the search was initiated.

The above hill-climbing decision logic is used in both the offset and the phase-split optimization processes within REALTRAN, using the minimum resolution if not specified otherwise in card type 4 of the standard TRANSYT input file. Utilizing card 4, one can investigate the effect on SCOOT of different step sizes. In addition, the maximum number of steps utilized by the REALTRAN model in each direction can be set by the user. This value is typically set to two to mimic SCOOT's minimum signal changes while maintaining the ability to escape local minima.

Use of High-Sensitivity Parameters

The sensitivity parameter sets a limit below which any downstream changes to the CFP are neglected. The base TRANSYT model permits the use of a sensitivity parameter card (card type 6) to limit the extent to which the downstream effects, at other intersections, of a change in signal timings at a given intersection will be examined. The setting of these parameters affects the time required for execution of the model. Because the SCOOT optimization logic considers only the flows arriving at the traffic signal in generating the cycle-length duration and phase splits, and considers only the surrounding signals in estimating the optimum offsets, high-sensitivity parameters of 20 percent were utilized as the default in the REALTRAN model.

Search for the Optimum Cycle Length

To minimize any disruptions to traffic within SCOOT, the changes in cycle length are restricted to be very small. This objective is achieved within REALTRAN in three ways: by controlling the number of cycle lengths to be evaluated, by controlling the cycle-length increment, and by controlling the frequency of cycle-length optimization runs.

During each cycle optimization the REALTRAN program uses the previous interval's cycle length as a seed. The minimum cycle length to be considered is then selected as the initial seed cycle length minus the user-specified cycle-length increment. Similarly, the maximum cycle length to be considered is chosen as the seed cycle length plus the cycle-length increment. These optimizations are performed at a user-specified interval and thus can be performed, for example, every 3 min, as is the case with the SCOOT system or at more/less frequent intervals.

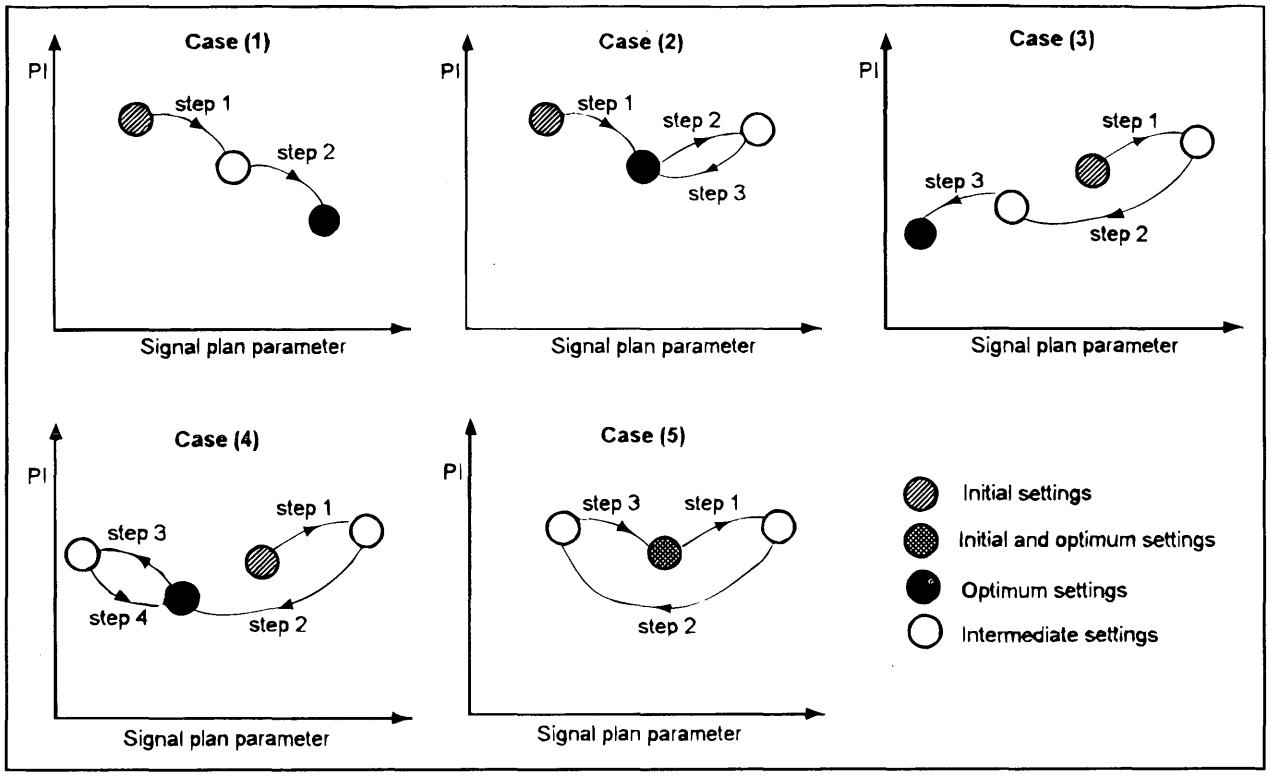

FIGURE 1 Modified hill-climbing mechanism for a two-step constraint.

The REALTRAN model can use as many as three different cycle-length increments, depending on the cycle length. The model can therefore again replicate the SCOOT system's cycle optimization logic to a large extent.

EXAMPLE ILLUSTRATION

To illustrate briefly how the REALTRAN model can examine the impacts of a constrained SCOOT-like optimization for different traffic flow patterns, an extract of the results from a nine-traffic-signal grid network is presented in this section. Because of limited space, only a very brief summary of the results is presented; however, the details of the network and results can be found in the literature (*1,10*).

The network was simulated for a hypothetical sequence of 5 hr within which the flows experienced a peak in the eastbound direction followed by a peak in the westbound direction. Each minute, the REALTRAN optimizer optimized the signal settings, using two optimization scenarios. In the first, the REALTRAN model utilized the standard unconstrained TRANSYT full optimization every minute. Subsequently, in the second scenario the REALTRAN optimizer was constrained to emulate the SCOOT signal optimization logic. To simplify the illustration of the traffic flow pattern, only a sample of the link flows for 10 typical minutes during the simulation period, for the four approaches to the traffic signal located at intersection 5, is provided in Table 1. For these sample arrival link flows Table 2 illustrates the signal settings that were selected by the REALTRAN optimizer for the two scenarios studied, namely, the TRANSYT and SCOOT emulations.

It can be noted from Table 1 that during this time period the flow on the westbound, southbound, and northbound approaches to signal 5 remained constant while the flow on the eastbound approach increased from 525 to 750 vehicles/hr. This change represents

TABLE 1 Link flows arriving at signal 5

Time into Simulation (min)	Approach flows to signal 5 (vehicles/hr)			
	Eastbound	Westbound	Southbound	Northbound
35	525	350	500	250
36	550	350	500	250
37	575	350	500	250
38	600	350	500	250
39	625	350	500	250
40	650	350	500	250
41	675	350	500	250
42	700	350	500	250
43	725	350	500	250
44	750	350	500	250

TABLE 2 Signal settings chosen by the TRANSYT and SCOOT emulations (Signal 5)

Time	TRANSYT Emulation					SCOOT Emulation				
(min)	Cycle	Offset	Phase 1	Phase 2	PI	Cycle	Offset	Phase 1	Phase 2	PI
35	48	0	24	24	80.2	40	12	20	20	78.3
36	44	20	23	21	83.7	40	12	21	19	82.9
37	52	0	27	25	87.7	44	13	23	21	82.0
38	52	3	28	24	91.8	44	11	24	20	92.5
39	52	3	28	24	97.8	44	12	24	20	99.3
40	72	9	40	32	100.1	48	13	26	22	103.8
41	64	6	36	28	106.8	48	9	27	21	115.3
42	68	5	39	29	112.3	48	8	27	21	133.3
43	76	9	44	32	119.9	52	11	29	23	156.9
44	80	4	47	33	129.8	52	9	30	22	173.4

approximately a 50-percent increase in flow in 10 min. Although such an increase within 10 min may not be very likely to occur in practice, the intent was to investigate, by means of such a rapid hypothetical change in flows, the robustness and responsiveness of the SCOOT emulator.

It can be noted from Table 2 that the TRANSYT emulator responded to these drastic changes in flows with major changes in the signal settings. Specifically, the cycle length changed from 52 to 72 sec, which is equivalent to a 35-percent change, at the onset of the 40th min. In contrast, the SCOOT emulator altered the cycle length by only 4 sec from 44 to 48 sec, following which the SCOOT emulator was restricted from performing another cycle-length optimization for 3 min. Also, the TRANSYT emulator made a drastic change in the offset, from an offset of 0 sec to an offset of 20 sec, at the start of the 36th min, for a change in cycle length from 48 to 44 sec, while the SCOOT emulator was restricted to minor alterations. The difference between the maximum and minimum offsets selected by the TRANSYT emulator was 20 sec (0 to 20 sec), as opposed to a 4-sec difference for the SCOOT emulator (13 to 9 sec). The TRANSYT emulator also varied the phase 1 duration from 23 to 47 sec at the onset of the 40th min, which is equivalent to a 14-sec variation, while the SCOOT emulator varied the phase 1 duration only from 24 to 26 sec.

In comparing the PI at each minute (columns 6 and 11 of Table 2), it is evident that initially the PI for the SCOOT emulator was lower. However, as the TRANSYT emulator was not restricted, this trend changed as the link flows varied. However, because the unconstrained optimizer made larger variations to the signal settings, it caused major disruptions to the traffic and thus added inefficiencies that are not accounted for in Table 2. It must be noted also, based on these limited results, that the SCOOT emulator succeeded, albeit with a lag, in following the trend in the variation of cycle length, offset, and phase split, thus allowing the signal timings to evolve over time in a fashion similar to those of the SCOOT signal optimizer.

SUMMARY AND CONCLUSIONS

We believe that in this paper we have made a significant step toward addressing the need for a simulation tool that is capable of emulating the SCOOT optimization logic. The structure of such a model, entitled REALTRAN, has been presented. The REALTRAN model is based on the well-known TRANSYT program, specifically TRANSYT-7F, by constraining the hill-climbing procedure within the TRANSYT model. The user can specify the maximum number of steps allowed in each direction and thus allow REALTRAN to model various constraining conditions that are different from the SCOOT default. The REALTRAN model permits the specification of a separate link flow file that is used to select the optimum signal settings and of an optional link flow file that is used to evaluate these signal settings. Furthermore, the user can specify, through the master file, both the cycle-length increments to be evaluated and the cycle-length thresholds separating various cycle-length increment zones.

It must be noted that the REALTRAN program, like the SCOOT, SCAT, and PRODYN traffic models, is built on the vertical queue model and thus cannot consider in detail the effect of downstream link congestion on the signal output. These models operate well as long as the network is not overly congested. However, they fail to model the effect of downstream congestion on the capacity of upstream intersections during queue spillback. In the case of SCOOT the queuing model is updated by queue measurements from the field. In addition, the REALTRAN model cannot model the rerouting of traffic in response to changes in signal timings.

REFERENCES

1. Boillot, F., J. M. Blosseville, J. B. Lesort, V. Motyka, M. Papageorgiou, and S. Sellam, Optimal Signal Control of Urban Traffic Networks. *Proc., IEE International Conference on Traffic Road Monitoring and Control*, 1992, pp. 75–79.
2. Rakha, H., M. Van Aerde, and E. R. Case, Experiments in Incremental Real-Time Optimization of Phase, Cycle, and Offset Times Using an On-Line Adaptation of TRANSYT-7F. Presented at the Engineering Foundation Conference on Traffic Management: Issues and Techniques, Palm Coast, Florida, April 1991.
3. Robertson, D. I. *TRANSYT: A Traffic Network Study Tool*, RRL Report 253, Crowthorne, UK, 1969.
4. Courage K., and C. Wallace. *TRANSYT-7F Users Guide*. University of Florida, Gainesville, 1991.
5. Timmermans, W. F., P. P. Van Den Bosch, and J. J. Klijnhout. Improved Network Control by Using TRANSYT. *Traffic Engineering and Control*, Vol. 21, 1979, pp. 353–356.
6. Foulds, L. R. TRANSYT Traffic Engineering Program Efficiency Improvement via Fibonacci Search. *Transportation Research*, Vol. 20A, No. 4, 1986, pp. 331–335.
7. Hunt, P. B., D. I. Robertson, R. D. Bretherton, and R. I. Winton. *SCOOT: A Traffic Responsive Method for Coordinating Signals*. TRRL Report LR, 1981, p.1014.
8. Hunt, P. B., D. I. Robertson, R. D. Bretherton, and M. C. Royle. The SCOOT On-Line Traffic Signal Optimization Technique. *Proc., IEE International Conference on Road Traffic Signalling*, Publication No. 207, 1982, pp. 59–62.
9. Robertson, D. I., and R. D. Bretherton. Optimizing Networks of Traffic Signals in Real-Time—The SCOOT Method. *IEEE Transactions on Vehicular Technology*, Vol. 40, No. 1, February, 1991, pp. 11–15.
10. Rakha, H. *A Simulation Approach for Modeling Real-Time Traffic Signal Controls*. Ph.D. dissertation. Queen's University, Kingston, Ontario, Canada, 1993.

Publication of this paper sponsored by the Committee on Traffic Signal Systems.