PERGAMON

Transportation Research Part C 9 (2001) 415–432

TRANSPORTATION RESEARCH PART C

www.elsevier.com/locate/trc

A real-time traffic signal control system: architecture, algorithms, and analysis☆

Pitu Mirchandani [a], Larry Head [b,*]

[a] *Department of Systems and Industrial Engineering, University of Arizona, Tucson, AZ 85721, USA*
[b] *Gardner Transportation Systems Inc., 6375 East Tanque Verde #170, Tucson, AZ 85715, USA*

Received 1 April 1998; accepted 26 June 2000

Abstract

The paper discusses a real-time traffic-adaptive signal control system referred to as RHODES. The system takes as input detector data for real-time measurement of traffic flow, and "optimally" controls the flow through the network. The system utilizes a control architecture that (1) decomposes the traffic control problem into several subproblems that are interconnected in an hierarchical fashion, (2) predicts traffic flows at appropriate resolution levels (individual vehicles and platoons) to enable pro-active control, (3) allows various optimization modules for solving the hierarchical subproblems, and (4) utilizes a data structure and computer/communication approaches that allow for fast solution of the subproblems, so that each decision can be downloaded in the field appropriately within the given rolling time horizon of the corresponding subproblem. The RHODES architecture, algorithms, and its analysis are presented. Laboratory test results, based on implementation of RHODES on simulation models of actual scenarios, illustrate the effectiveness of the system.

Keywords: Traffic controls; Traffic optimization; Real time systems; Adaptive controls

1. The system

Over the last six years, the authors, with some students, faculty and researchers have proposed and developed a real-time traffic-adaptive signal control system, referred to as RHODES.[1] The

☆ An earlier version of this manuscript was presented at TRISTAN III (Triennial Symposium on Transportation Analysis), June 17–23, 1998 in San Juan, Puerto Rico.

* Corresponding author. Tel.: +1-520-290-8006; fax: +1-520-290-8178.
E-mail address: lhead@gardnersys.com (L. Head).

[1] The authors also acknowledge the significant support received from the Arizona Department of Transportation, City of Tucson and other local agencies in Arizona, and the Federal Highway Administration, without which the authors would not have been able to develop RHODES up to the field testing phases.

0968-090X/01/$ - see front matter
PII: S0968-090X(00)00047-4

system takes input from the surface street detectors (allowing whatever technology that is being utilized: induction loops, video, etc.), predicts the future traffic streams at various hierarchical levels of aggregation, both spatially and temporally, and outputs "optimal" signal control settings that respond to these predictions. The optimization criterion can be any that is provided by the jurisdiction using the system but must be based on traffic measures of effectiveness (average delays, stops, throughput, etc.).

The RHODES architecture for surface streets is depicted in Fig. 1 (Head et al., 1992). At the highest level of RHODES is a "dynamic network loading" model that captures the slow-varying characteristics of traffic. These characteristics pertain to the network geometry (available routes including road closures, construction, etc.) and the typical route selection of travelers. Based on the slow-varying characteristics of the network traffic loads, estimates of the load on each particular link, in terms of *vehicles per hour*, can be calculated. The load estimates then allow RHODES to allocate "green time" for each different demand pattern and each phase (North–South through movement, North–South left turn, East–West left turn, and so on). These decisions are made at the middle level of the hierarchy, referred to as "network flow control". Traffic flow characteristics at this level are measured in terms of *platoons of vehicles* and their speeds. Given the approximate green times, the "intersection control" at the third level selects the appropriate phase change epochs based on observed and predicted arrivals of *individual vehicles* at each intersection.

Essentially, at each level of the hierarchy there is an estimation/prediction component and a control component. These components are discussed in Sections 2 and 3, respectively. Currently, we have not done much model development at the highest level (dynamic network loading) and so we will not include much discussion of level 1 of the RHODES hierarchy.

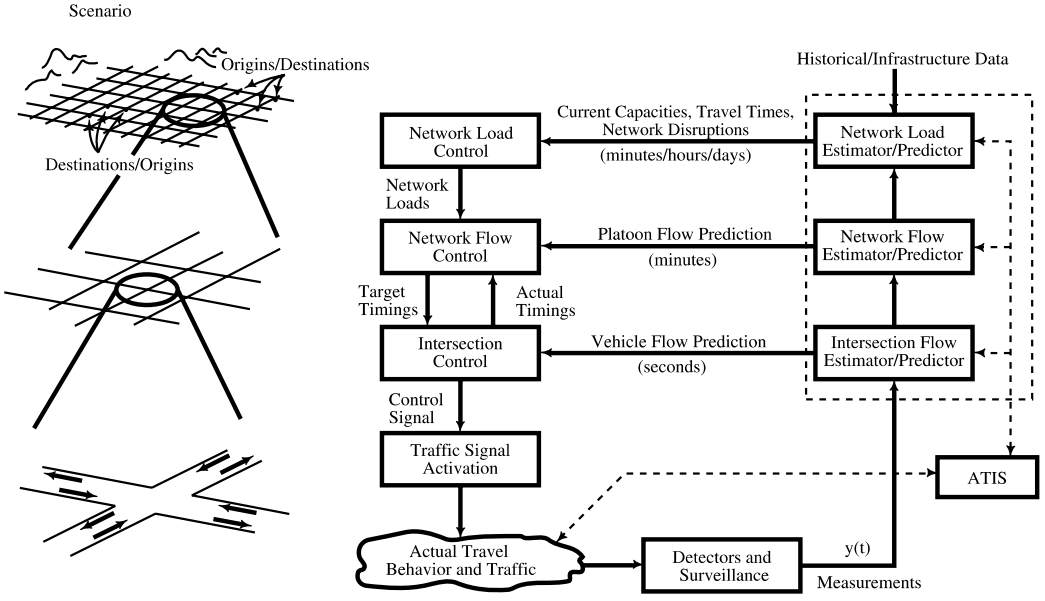


Fig. 1. The RHODES hierarchical architecture.

There are three aspects of the RHODES philosophy that make it a viable and effective systems to adaptively control traffic signals. First, it recognizes that recent technological advances in communication, control, and computation (a) make it possible to move data *quickly* from the street to the computing processors (even now most current systems have communication capabilities that are not utilized to their potential), (b) make processing of this data to algorithmically select optimal signal timings *fast*, and (c) allow the *flexibility* to implement through modern controllers a wide variety of control strategies. Second, RHODES recognizes that there are natural stochastic variations in the traffic flow and therefore one must expect the data to stochastically vary (by simply smoothing the data and working with mean values does not make the actual traffic that the system sees smooth and average as assumed by some real-time traffic control schemes). And third, RHODES pro-actively responds to these variations by explicitly *predicting* individual vehicle arrivals, platoon arrivals and traffic flow rates, for the three corresponding levels of hierarchies described above.

2. The prediction methods

For pro-active traffic control, it is important to predict vehicle arrivals, turning probabilities and queues at intersections, in order to compute phase timings that optimize a given measure of effectiveness (e.g., average delay). To emphasize this importance, consider an intersection with several approaches. Associated with each approach are several possible traffic movements: left turn, right turn and a through movement. Any non-conflicting combination of movements that can share the intersection at any one time can be assigned a signal phase that allows those movements protected use of the intersection. Now consider the signal-timing problem given two possible perfect predictions of arrivals during the planning horizon as depicted in Fig. 2. Each arrival pattern represents the number of vehicles to arrive at the intersection in fixed time intervals. Both arrival patterns are identical until time t_0 when the signal control has to decide whether to serve this approach or to serve another approach. In the top case, the demand occurs

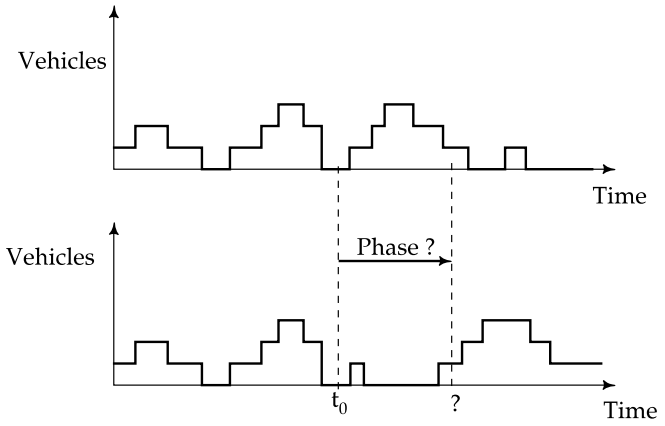


Fig. 2. Graphical depiction of the effect of future arrivals on scheduling phase sequences and durations.

immediately following t_0, whereas in the bottom case there is little demand immediately following t_0 and greater demand in the future. In each case, the total number of vehicle arrivals is equal. However, the optimal signal timings could be significantly different. It is of fundamental importance to know the temporal arrival distribution to build a truly real-time traffic-adaptive signal control logic.

Two issues are important to predict traffic flow: (1) the length of the time horizon, and (2) the number of prediction points per time horizon (called the prediction frequency). The prediction time horizon provides the real-time traffic-adaptive signal-timing control logic with the ability to plan future signal-timing decisions. If the prediction horizon is short, perhaps several seconds, then the signal-timing decisions are restricted. For example, if the predictions are made over a 10 s horizon, the signal-timing logic can only make timing decision that extend or shorten the current phase. On the other hand, if the predictions are made over a longer horizon, the signal-timing decisions can include decisions on phase sequencing and phase duration.

The prediction frequency provides information about the distribution of vehicle arrivals over time. If the predictions are made at a frequency of only one prediction for the decision time horizon, which, say, is 30 s, then the signal-timing logic must assume that the vehicles arrive uniformly during that 30 s. If the predictions are made more frequently, say every second over the prediction horizon, then the signal-timing logic will have a more accurate representation of the distribution of vehicle arrivals over time.

2.1. The PREDICT approach

The PREDICT algorithm (Head, 1995) uses the output of the detectors on the approach of each upstream intersection, together with information on the traffic state and planned phase timings for the upstream signals, to predict future arrivals at the intersection under RHODES control. This approach allows a longer prediction time horizon since the travel distance to the intersection is longer and the delays at the upstream signal are considered. A benefit of this approach is that it includes the effects of the upstream traffic signals in the intersection control optimization problem.

To understand how this approach works, consider the scenario shown in Fig. 3. It is desired to predict the flow approaching intersection A at detector d_A. Making the prediction for the point d_A is important because it is a point on link AB where the actual flow can be measured, hence the quality of the prediction can be assessed in real-time.

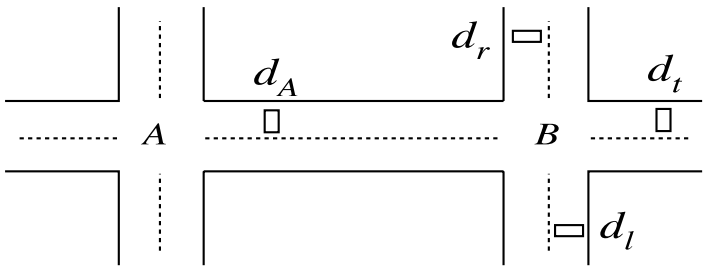


Fig. 3. Prediction scenario based on detectors on the approaches to the upstream intersection (B).

Traffic contributing to the flow at d_A originates from the approaches to intersection B and can be measured at detectors d_l, d_t and d_r representing the flows that will turn left, pass through and turn right, respectively, onto link AB. It is possible to have other traffic that originates at sources between intersections A and B, but this can be considered as unmeasurable "noise". Also, it is possible that vehicles passing over d_l, d_t, and d_r will terminate their trip before arriving at d_A. This can also be considered as "noise" in the prediction. Significant flow volumes that enter or exit the network midblock can be modeled in the predictions as constant bias terms.

When a vehicle passes a detection point, say d_i where $i \in \{l, t, r\}$, several factors affect when it will arrive at d_A including (1) the travel time from d_i to the stop bar at intersection B, (2) the delay due to an existing queue at B, (3) the delay due to the traffic signal at B, and (4) the travel time between B and d_A.

Fig. 4(a)–(d) depicts the delay associated with each of these factors. In Fig. 4(a), the vehicle arrives at detector d_i and passes freely to detector d_A. In Fig. 4(b), the vehicle arrives at detector d_i and is delayed by the signal at intersection B. Hence the travel time from d_i to d_A must account for the travel time from d_i to the stop bar, the delay due to the signal and the travel time from the stop bar to d_A. In Fig. 4(c) the arrival at d_i encounters delay for the signal as well as a standing queue, and has to travel from d_i to the stop bar at B, and from the stop bar to d_A. Fig. 4(d) depicts the case when the arrival at d_i occurs after the signal has begun serving the desired phase, but a standing queue is present. This case is similar to the above, except that the delay due to the

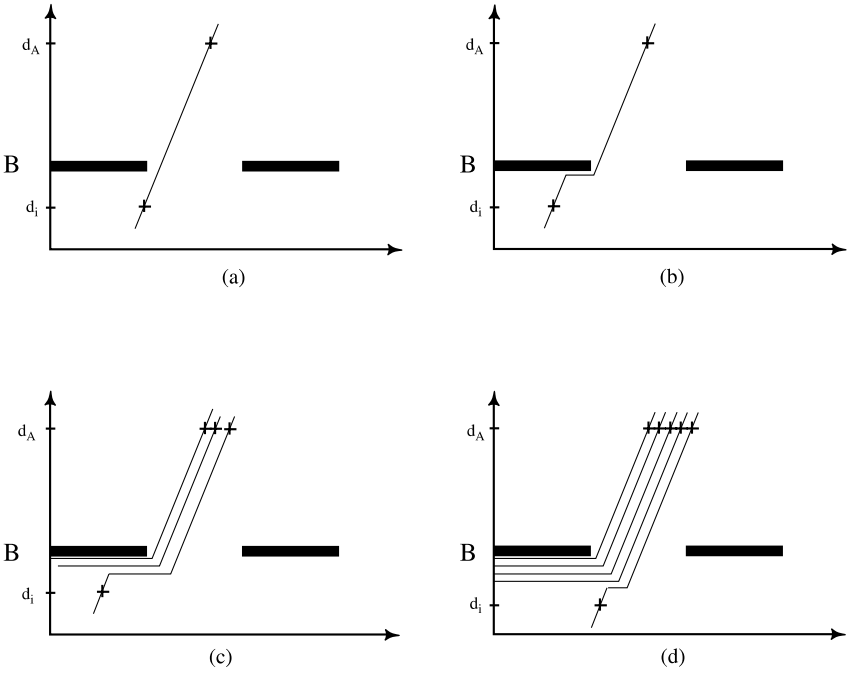


Fig. 4. Delays associated with the prediction of arrivals at the detector d_A: (a) detected vehicle passes freely through intersection; (b) detected vehicle arrives during red signal – signal delay; (c) detected vehicle arrives during red signal and a queue exists – signal and queue delay; (d)detected vehicle arrives during green signal and a queue exists – signal and queue delay.

standing queue must be adjusted based on the amount of time that has elapsed between the onset of the signal and the arrival of the vehicle at d_i and the travel time to the back of the queue.

Once the arrival time at d_A is predicted, the PREDICT model adds a fraction to the current estimate of the expected number of arrivals at that time. For example, if 15% of the vehicles that pass over d_i continue on to d_A, then for each actuation of d_i, 0.15 is added to the current estimate of the expected number of arrivals at the predicted arrival time.

2.2. Estimation of parameters

Observe that to use the PREDICT model, several parameters (given in bold) need to be provided: (1) travel times on links (detector to detector) which depends on the **link free-flow speed** and current traffic volumes, (2) **queue discharge rates** which also depends on volumes (as well as on queue spillbacks and opposing- and cross-traffic volumes), and (3) **turning probabilities**. In addition to these parameters, to estimate arrivals and demand for various phases, we also need to have **estimates of queues** at the intersections and the ramps.

Link free-flow speed can be estimated from historical data and capacity analysis. Link free-flow speeds are needed even in traditional off-line models to optimize fixed phase timings (cycle times, offsets, splits), so to obtain these should pose no major problem.

Through-traffic queue discharge rates are effected by downstream through-traffic volumes, which can be easily measured. Likewise, left-turn queue discharge rates depend on opposing traffic volumes, and right-turn queue discharge rates depend on cross-traffic in that direction. These three discharge rates are initially given from calculated default functions – functions of traffic volumes, but are then adjusted based on how well they predict remaining queues at the stop-bar presence detectors. For example, if the left-turn queue estimate tends to be non-zero when in fact it is zero, then the left-turn discharge rate is adjusted upwards; this procedure will be explained shortly.

TURN algorithm. An assumption for RHODES (as well as current off-line methods to set signal timings) is that some estimates for turn probabilities at the intersection are given. Even the CORSIM[2] model needs information about the turning probabilities when modeling traffic movement. However, from the real-time traffic control perspective, these probabilities are not deterministic; they change stochastically over time. For example, suppose P_{WN} is the probability that a vehicle arriving from the West to some intersection will turn left (North), then it is clear that P_{WN} will depend on the time of the day, the volume of traffic, and the particular mix of the origins/destinations in the group of arrivals being modeled. In other words, P_{WN} is described by a random process.

Our assumptions for P_{WN} are (1) a prior estimate is available whose uncertainty is modeled with a normal distribution with known mean and variance; (2) at any given time, we have measured the percentage of vehicles that have turned left in the last, say, 5 min, as well as percentages that turned right and driven straight through the intersection; and (3) we know the error distributions for these measurements. We had a choice of three turning probabilities models: (1) information minimization/entropy maximization (Mekky, 1979; van Zuylen, 1979; Hauer et al., 1981);

[2] CORSIM is a software package for modeling/simulating traffic on a network; it has been developed by FHWA.

(2) Bayesian (Maher, 1984); and (3) maximum likelihood (Maher, 1984; Nihan and Davis, 1989). The Bayesian model was picked for implementation since the other two models involved a non-deterministic number of iterations based on an error tolerance whereas Bayesian method consisted of exactly seven iterations. In this method, prior variances for the turning volume errors are used along with the prior means. The covariance's of the turning volume errors are assumed to be zero since the traffic detectors are assumed to operate independent from one another.

QUEUE algorithm. There have been a few algorithms that have been reported in the literature that address the problem of estimating queues at an intersection using detector information, most notably that of Baras et al. (1979). However, these are not applicable here because they require excessive computational effort and time that is not available in our real-time prediction scenario. Instead, for our purpose, we developed a simple estimation procedure of accounting for arrivals and estimated departures based on queue discharge rates. Suppose at the beginning of a green phase, say at time t_0, our initial queue estimate at some stop-bar is $q(t_0)$. At the end of the green phase, say at time t_1, the remaining queue $q(t_1)$ is given by

$$q(t_1) = q(t_0) + a(t_1, t_0) - d(t_1, t_0),$$

where $a(t_1, t_0)$ is the number of predicted arrivals between t_1 and t_0, and $d(t_1, t_0)$ the predicted number of departures (using a given queue discharge rate). What allows us to keep biases from creeping into the estimates is that at some epochs we are *certain* that the queue length is zero, specifically when there is no queue at the stop bar as confirmed by the stop-bar presence detector. If the estimated queue is positive while the stop-bar presence detectors indicate no queue, then we effectively decrease our estimate and make it zero. If the estimated queue is zero while the stop-bar indicates a positive queue, then the estimated queue is increased by some positive number such as one.

When the queue discharge rate at the stop bar is estimated well, it would be expected that, on the average, half the time the estimated queues will be greater than the actual queues and half the time less than actual queues. If the estimated queues more often than not tend to be higher than the actual queues (i.e., when there were no vehicles while the queue estimate was non-zero), then we adjust the queue discharge rate upwards by a small amount. If it tends to be less (i.e., when there were vehicles while the estimates were zero), then we adjust it downwards by a small amount. We note that the adjustment of queue discharge rate is only possible when there is no queue spillback into the intersection; in this case queue discharge rate is zero because of blockage and does not depend on traffic volumes. The exact adjustment step size can be configured by the system operator, but typically step sizes of 0.1 vehicles/cycle are sufficient.

It is important to note that the PREDICT model is based on processing arrival data as it becomes available. At any point in time, the predicted arrival flow pattern at d_A accounts for vehicles that have already passed the detectors d_l, d_t and d_r. The benefit of this vehicle-additive process of the predictor is that it constantly provides, for a given prediction horizon, (1) nearly complete information of anticipated vehicle arrivals in the very near future (of those vehicles that have already passed the upstream intersections) and (2) partial information of anticipated vehicles in remaining part of the prediction time horizon (of those vehicles that have not passed the upstream intersections, since some new vehicles may still arrive that will effect the delays in the prediction time horizon). Results of an evaluation study of the PREDICT algorithm for arrivals at an intersection have been reported by Head (1995).

2.3. Network flow prediction

The resolution of traffic at the network flow control level (i.e., level 2 of the RHODES hierarchy) is in platoons. The scope of the prediction is a subnetwork of several intersections (the number of intersections depends on the computational power available but we envision that nine intersections can be controlled by RHODES using only an Intel Pentium processor) with a larger decision time horizon. Typically, RHODES will use a 20–40 s rolling horizon to predict arrivals and queues at each intersection, based on upstream detector data; at the network flow control level, RHODES will use a 200–300 s rolling horizon.

At the subnetwork level, the APRES-NET model (Dell'Olmo and Mirchandani, 1996) is a simplified traffic simulation model based on the same principles as the PREDICT model described above, but instead of propagating a single vehicle at a time from upstream intersections, it propagates platoons of vehicles through a subnetwork of intersections. It is necessarily a simplified model because it is used as an objective function evaluator, or as a network wide performance estimator, for the network control logic.

3. The control algorithms

Fixed control strategies are based on a signal-timing plan defined in terms of operating parameters for traditional signal control, namely *cycle time*, *splits*, and *offsets*. These parameters are generally developed based on traffic studies and standard procedures, such as the Highway Capacity Manual, or signal-timing software such as TRANSYT and PASSER. The traffic studies result in estimates of traffic conditions, link volumes and turning percentages, for specified time periods. Signal-timing *parameters* are developed for each of these time periods and, typically, implemented on a time-of-day basis with no consideration of current actual traffic conditions. In many cases, even the use of standard procedures for the development of signal-timing plans is abandoned and traffic engineers operate in a judgment-based fashion with moderate levels of success. None of these approaches is truly traffic-adaptive or even attempt to actually minimize some measure of traffic performance such as average vehicle-delay.

Currently available traffic responsive systems attempt to address the problem of responding to actual traffic conditions by switching these *parametric* signal-timing plans based on current wide-area traffic conditions rather than time of day. This requires that signal-timing parameters be developed for a variety of possible traffic conditions. Nevertheless, implicit in the usage of *parametric* timing plans is the assumption that for the next several minutes, or even hours, the traffic in the network can be well characterized by the measured *average* flows and parameters. No account is taken of the fact that the second-by-second and minute-by-minute variability of traffic are significant and plans based on averages produce unnecessary delays for some traffic movements when the traffic on conflicting movements is absent, or very small, during some periods.

The RHODES approach is to predict both the short-term and the medium-term fluctuations of the traffic (in terms of individual vehicle arrivals and platoon movements, respectively), and explicitly set phases that maximize a given traffic performance measure. Note that we do not set timing plans in terms of cycle times, splits and offsets, but rather in terms of phase duration for any given phase sequence. (RHODES does not necessarily require a pre-specified phase sequence,

but since many traffic engineers prefer a pre-specified sequence, RHODES has been developed to allow the traffic engineer to specify a desired sequence.) In other words, in the RHODES control strategy, the emphasis shifts from changing timing parameters in *reacting* to traffic conditions just observed to *pro-actively* setting phase duration for predicted traffic conditions.

3.1. Intersection control

At the lowest level of the RHODES hierarchy for a surface street network, i.e., at the intersection control level, RHODES uses a dynamic-programming based algorithm COP (Sen and Head, 1997). There are other signal-timing schemes, which have been experimented, that do not provide parametric timing plans but instead provide phase duration, notably OPAC (Gartner, 1983; Gartner et al., 1991) and PRODYN (Khoudour et al., 1991) and UTOPIA (Mauro and DiTaranto, 1990). In some ways these too use dynamic programming (DP) or related optimization schemes, but, in their current implementations, the underlying models are more approximate and the methods are not as efficient.

Fig. 5 depicts the states of the DP formulation. A rolling horizon approach is used to allow the optimization to take advantage of the most recent predictions and observations. An optimization is started at some time t_0 and considers a time horizon of T s, say 45–60 s. Each stage of the DP is associated with a signal phase. A **phase order** is provided to COP so that a stage corresponds to a phase. The DP state variable s_j is the amount of time that has been allocated to all past phases $1, 2, \ldots, j$. The decision in stage j is to allocate x_j time units to phase j. It should be noted that in general there are more stages in the DP's planning horizon than the number of phases used for control. If there are P phases and N stages ($N > P$) some of the phases may be repeated as stages. If the traffic engineer does not restrict the phases to be in a particular sequence, then this flexibility allows for variable phase sequencing through phase skipping (by effectively allocating zero time for the corresponding stage).

Each decision x_j has an associated value based on a performance measure such as stops or delay. This value is determined by using the predicted vehicle arrivals, the current and prior decisions, and an imbedded traffic flow model that accounts for estimated queues, startup lost time, queue discharge and arrivals, as well as other traffic dynamics that relate the decision to the performance measure.

The DP is completed when each possible decision for each stage has been evaluated in a forward recursion. Then a backward recursion is used to determine the sequence of phases and phase duration that will result in the lowest value of the performance measure over the optimization horizon.

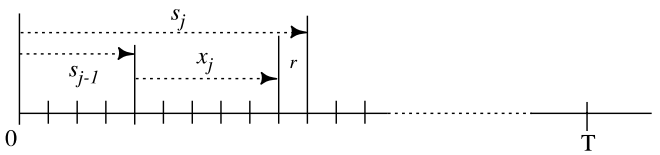


Fig. 5. Stages and states of the COP model.

The decision for the first stage of the optimization is implemented as the desired signal control. Just prior to the end of this first phase, the optimization problem is solved again in a rolling horizon approach. The sequence of phases in the second optimization begins with the current phase, which allow for the phase to be terminated early or extended, based on the re-evaluation with more recent observations and predictions.

3.2. Network flow control

The network flow control logic is based on a model called REALBAND (Dell'Olmo and Mirchandani, 1995) which optimizes the movement of observed platoons in the subnetwork. If minimizing total stops was the measure of network performance, then REALBAND attempts to form progression bands based on actual observed platoons in the network. In general, any delay and/or stops based measure of performance may be optimized.

The basic idea of REALBAND can be understood by considering the following example. Fig. 6 shows a small network with several platoons of vehicles traveling in different directions. Platoons are defined from observed detector data as a flow density above a pre-specified level for some length of time. Each platoon is characterized in terms of size (number of vehicles) and speed.

When two (or more) platoons are predicted to arrive at an intersection and request opposing signal phases, a conflict is said to occur. Fig. 7 depicts the type of conflicts that may occur. A decision tree is built where each branch of the tree represents one possible resolution of a conflict. The decision tree developed is based on the predicted platoon movement over some pre-defined horizon, such as 200–300 s, with node and two out-links for each conflict resolution. Fig. 8 shows the decision tree for the example.

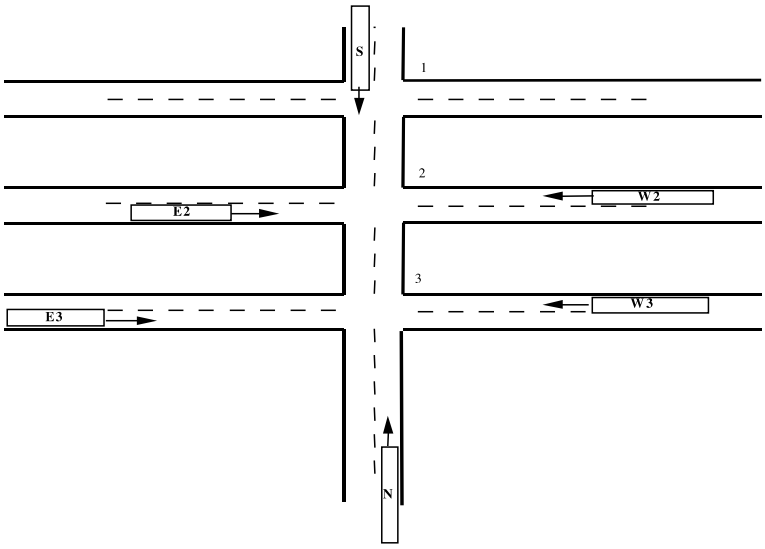


Fig. 6. An example to illustrate REALBAND.

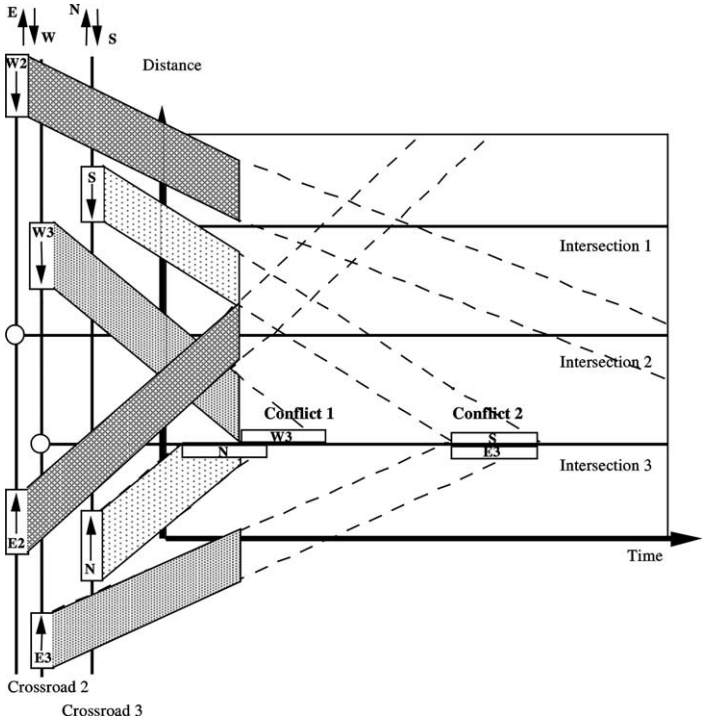


Fig. 7. Realbands for the example of Fig. 6.

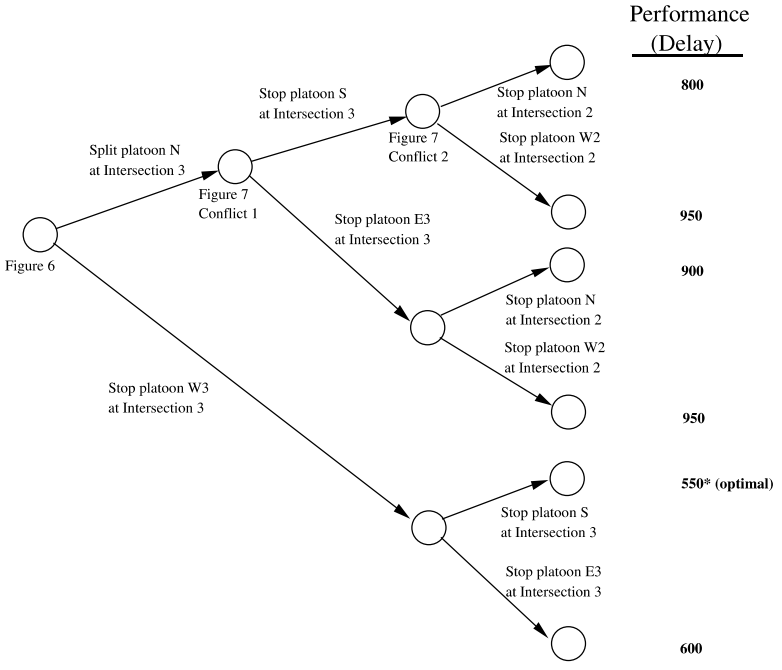


Fig. 8. REALBAND decision tree for the example of Fig. 6

REALBAND evaluates, using APRES-NET, the performance for each branch of the decision tree. When all branches have been explored, a path on the tree (corresponding to a set of conflict resolutions) is chosen with best-estimated performance. For example, in Fig. 8, the path with 550 s of total delay would be selected.

The REALBAND decisions are used as constraints to the intersection control logic (COP). When COP begins it's rolling horizon optimization, a set of decisions on phase durations in the phase order is required to accommodate any constraints that REALBAND conflict resolutions impose, with a relaxation that COP may adjust the phase start and end times based on recent, and more accurate observations of the vehicles in each platoon.

4. The RHODES prototype

The current version of the RHODES prototype logic is depicted in Fig. 9. The prototype consists of five modules: (1) intersection optimization logic, (2) link flow prediction logic, (3) network flow optimization logic, (4) platoon flow prediction logic, and (5) parameter and state estimation logic. The intersection optimization logic and the link flow prediction logic together form the *intersection control logic*. The network flow optimization logic and the platoon flow prediction logic together form the *network control logic*.

The RHODES prototype is currently under evaluation by FHWA as one strategy within the real-time traffic-adaptive signal control system (RT-TRACS) development program. This prototype has been implemented in software and evaluated using FHWA's CORSIM simulation model. This implementation is described below.

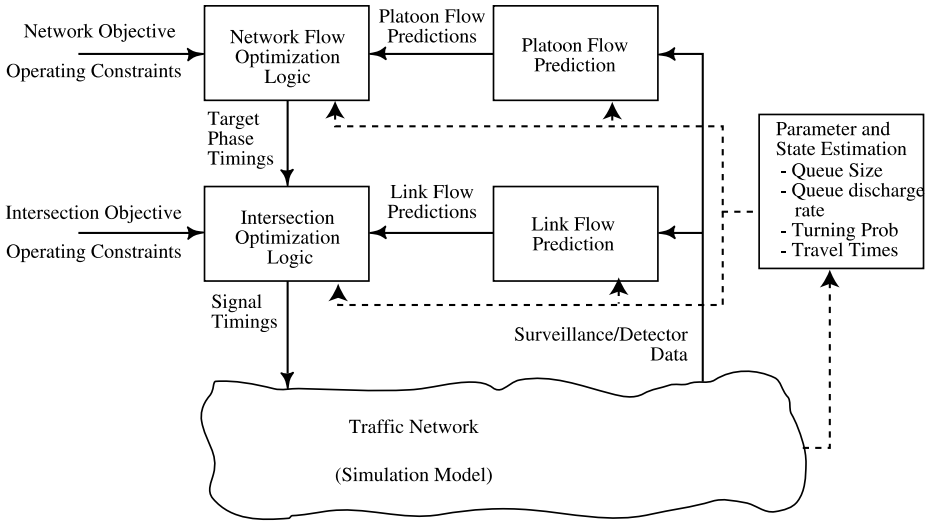


Fig. 9. Components of the RHODES prototype.

4.1. Prototype software design

This software prototype has been developed over a five-year period and was designed as a modular and portable software tool to support research and development of a wide variety of traffic management and control strategies. It should be emphasized that this software is not intended to be an implementable or deployable system, but is a research tool. The software has been developed to interface to the latest version of CORSIM traffic simulation model as is supported by Kaman Sciences Corporation for FHWA.

Fig. 10 presents an overview of the software architecture. The design centers around a simple flat (or blackboard) database and an *executive event controller*. The database contains all relevant network and control information. The *executive event controller* schedules different types of control related events such as updating surveillance information, setting signal control at desired intersections, running network coordination algorithms, running intersection optimization algorithms, and running traffic flow prediction algorithms.

The database contains three types of information: dynamic data, model parameters, and static data. Dynamic data refers primarily to data that changes on a second-by-second basis, such as vehicle detector information, past and planned signal control states, and traffic flow predictions. The time trajectory of dynamic data is important to the control algorithms. Model parameter data refers to information that is either constant or changes slowly over time such that only the current value is relevant and the time-trajectory (past and future) is less relevant. Examples of model parameters include turning percentages, queue departure rates, and average link travel speeds. These values may vary slowly over time and can be estimated by an appropriate algorithm, but for the purpose of modeling the traffic dynamics for intersection control, they are treated as constant parameters. The model parameter data also includes information related to the interface between control algorithm components.

Examples of interface information include signal-timing coordination and signal-timing constraints (e.g., a minimum green time for a phase, or a constraint that requires a phase be served at a given point in time to ensure coordination). Static data includes values that are assumed to

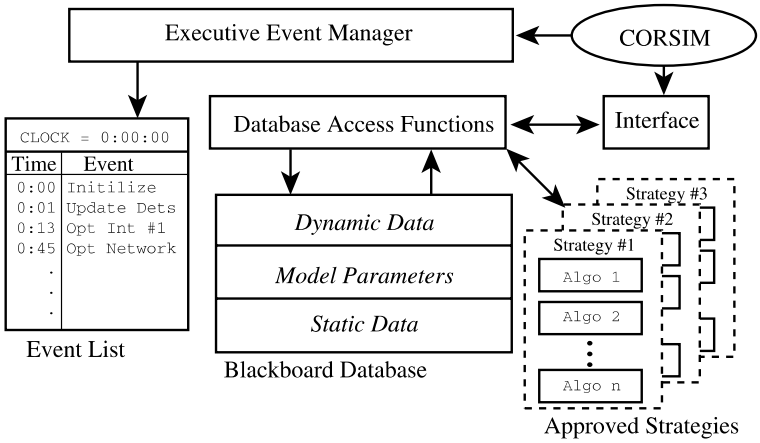


Fig. 10. Software architecture.

remain constant. Network geometrics (node identification numbers, number of lanes on each approach, link lengths, location of detectors, etc.) are the primary types of static data.

When the prototype is initialized at run time, for example, when CORSIM calls the prototype, a disk file is read that initiates and controls the creation of the necessary data structures for the database. The disk file format is very similar to a CORSIM *.trf file. Each record in the file has a card type identifier that specifies the information according to a pre-defined format. Examples of card types include link geometrics, node definitions, detector locations, signal-timing parameters, and optimization logic parameters.

4.2. Analysis using simulation models

It is clear that any type of traffic control algorithm needs to be tested in the "laboratory" before it is implemented and evaluated in the field. The most appropriate method to do this "laboratory" testing is to (1) have a realistic simulation model of traffic flow at an interchange, (2) emulate the detection of the traffic flow, and (3) observe the resulting changes that would come about if the algorithm was implemented in place of the current control system. CORSIM was used for laboratory testing the real-time traffic control algorithms. An actual set of intersections was selected for the simulation model.

There are many factors that must be considered in the evaluation, including

1. the type of network, including issues such as number of traffic signals, spacing between signals,
2. the traffic demand, including time varying demand, vehicle mixtures, bus stations and headways, and pedestrians,
3. statistical issues such as how to characterize the demand, what measures should be examined, how these measures are defined, as well as the number of simulation replications that must be made to support statistical conclusions or statements.

In this discussion, we will present results based on an FHWA test case used by ITT Systems and Sciences Corporation as part of the RT-TRACS prototype evaluation process. The network is based on a section of Tara Blvd. in Atlanta, Georgia. The network consists of nine intersections along a 17.7-km (11-mile) arterial. The traffic volume represents a peak flow that gradually increases and then gradually decreases over a period of 2 h.

The results presented below are based on comparing average delay per vehicle on the subnetwork of links that are directly effected by the RHODES prototype and, on the same set of links, as controlled be semi-actuated logic using timings derived from several optimization packages, including PASSER and TRANSYT. In reporting these results, we draw a useful analogy with traffic on communication networks, as described in Fig. 11.

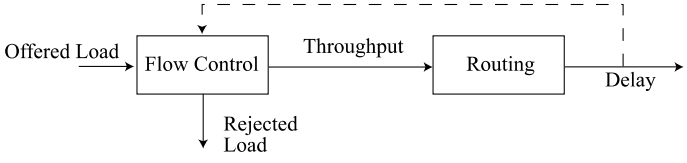


Fig. 11. Offered load, throughput and delay in a communication network.

The evaluation of communication networks, as a system of links and nodes, has received considerable attention in the literature (Bertsekas and Gallager, 1992). Fig. 12 shows the relationship between the offered load, throughput, and delay in the context of routing and flow control. The analogy in traffic control is that the offered load represents the demand, flow control corresponds to signalization (and effective capacity), and routing corresponds the signal timing (which is related to the capacity). The throughput in the communication system is analogous to the trips in a transportation network. The rejected load can be thought of as the unserved demand (spillback queues of vehicles).

Fig. 12 illustrates how good signal control (or good routing) can effect system performance. In periods when the demand does not exceed the effective capacity, the throughput is equal to the offered load. If poor signal control is used, then the effective capacity, in terms of served load, is reduced. Then the throughput is less than the offered load and delay increases considerably. When good signal control is used, the throughput is increased and the delay performance is improved.

This analogy with a communication system suggests that we need to measure both the offered load, the throughput, and the delay when considering the performance of a new signal control strategy. It also suggests the need to consider the performance at various loading levels. The offered load is measured as the number of trips on each link entering the network under study. The throughput is measured as the number of trips on each link exiting the network.

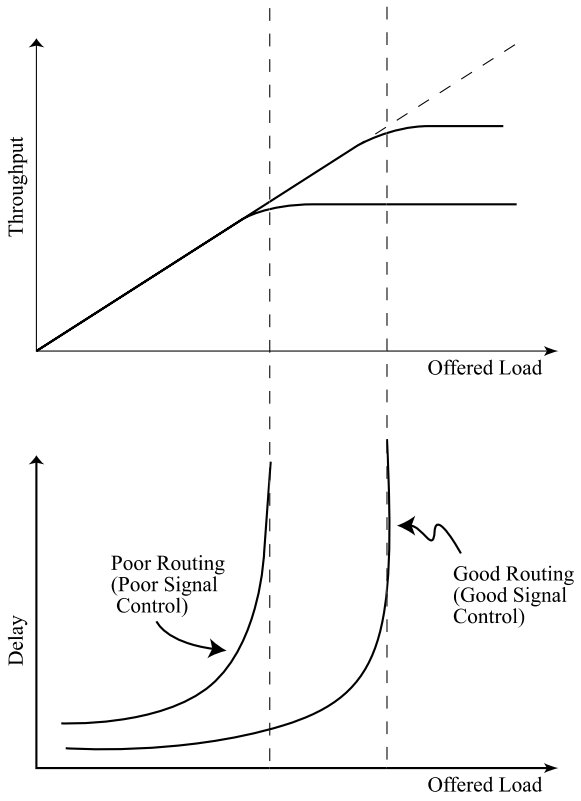
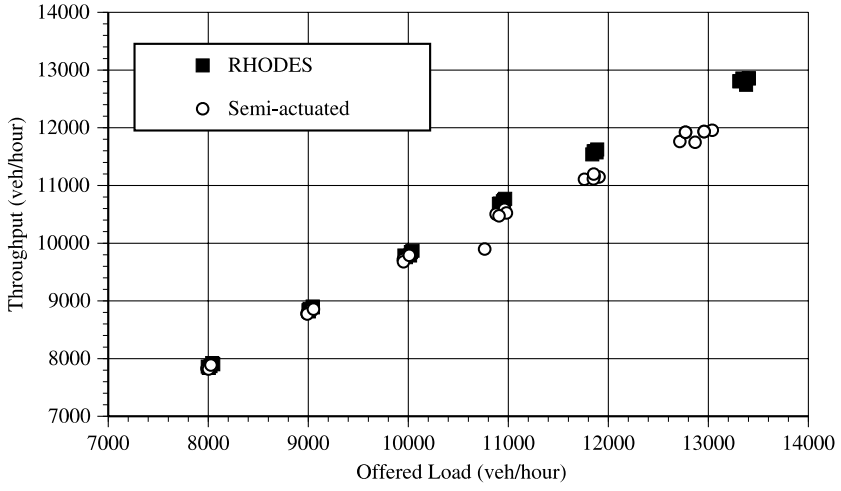


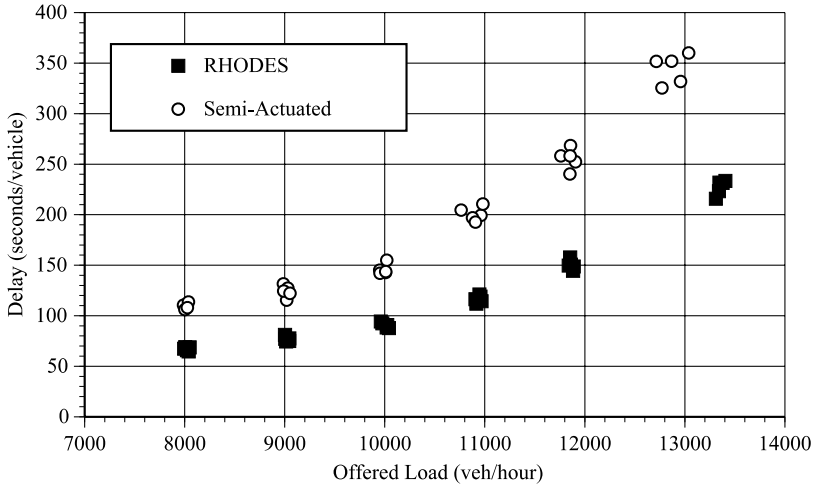
Fig. 12. Relationship between offered load, throughput and delay.

Fig. 13 shows the measured throughput and delay at each of the offered load levels for RHODES and the semi-actuated control for Tara Blvd. in Atlanta. Fig. 13(a) shows, as expected for the congestion levels simulated, throughput is equal to the offered load, implying that the steady state has been achieved. Fig. 13(b) shows a behavior that is consistent with the notion that improved signal control provides less delay. Thus, this simulation study suggests that the RHODES prototype is more efficient in utilizing the capacity of the network.

Observe that average vehicle delays decrease in the range of 50% (for low loads) to 30% (for high loads). In the high load case, not only are the average delays smaller, but also the variance of



(a)



(b)

Fig. 13. (a) Offered load versus delay and; (b) throughput.

the delay is significantly reduced, making the movement through the network more predictable for the driver.

We have implemented RHODES on several CORSIM models of actual transportation networks. Most of these networks have consisted of one single arterial with several cross arterial that carry considerable flow. We have also implemented RHODES at a diamond interchange where two signals are closely spaced and must operate in a highly coordinated manner. The results of all of these tests were remarkably consistent. The RHODES logic appears to take advantage of the natural stochastic variations in traffic flow – as it was designed – in decreasing traffic delay.

5. Conclusions

In this paper, we have presented the RHODES real-time traffic-adaptive signal control system, the software architecture that has been developed, and a simulation-based analysis of the system. The simulation experiments show promising results that encourage future experimentation. Based on these results, three field tests are being planned, one for a 10-intersection arterial segment in Tucson, AZ, (funded by the City of Tucson and FHWA), a 9-intersection arterial segment in Seattle, WA, (funded by FHWA), and a diamond interchange in Tempe, AZ, (funded by ADOT and the City of Tempe).

Acknowledgements

The Federal Highway Administration, the Arizona Department of Transportation, and the Cities of Tucson and Tempe have supported this research. The authors would like to thank and acknowledge Debbie Curtis and Raj Ghaman of FHWA; Jim Decker of the City of Tempe, AZ; Dr. Richard Nassi of the City of Tucson, AZ; Steve Owen and Tim Wolfe of ADOT; Dr. Sarath Joshua of Maricopa Association of Governments, Phoenix, AZ; Charlie Stallard and Larry Owens of ITT Systems and Sciences Corporation; Paolo Dell'Olmo of the University of Roma, "Tor Vergata", Italy; and Suvrajeet Sen, David Lucas, Steve Shelby and Doug Gettman of The University of Arizona.

References

Baras, J.S., Levine, W.S., Lin, T.L., 1979. Discrete-time point processes in urban traffic queue estimation. IEEE Trans. Autom. Control AC-24, 12–27.

Bertsekas, D., Gallager, R., 1992. Data Networks. Prentice-Hall, Upper Saddle River, NJ.

Dell'Olmo, P., Mirchandani, P.B., 1995. REALBAND: an approach for real-time coordination of traffic flows on a network. Transport. Res. Record 1494, 106–116.

Dell'Olmo, P., Mirchandani, P.B., 1996. A model for real-time traffic coordination using simulation based optimization. In: Bianco, L., Toth, P. (Eds.), Advanced Methods in Transportation Analysis. Springer, Germany, pp. 525–546.

Gartner, N.H., 1983. OPAC: a demand-responsive strategy for traffic signal control. Transport. Res. Record 906, 75–81.

Gartner, N.H., Tarnoff, P.J., Andrews, C.M., 1991. Evaluation of the optimized policies for adaptive control (OPAC) strategy. Transport. Res. Record 1324, 105–114.

Hauer, E., Pagitsas, E., Shin, B.T., 1981. Estimation of turning flows from automatic counts. Transport. Res. Record 795, 1–7.

Head, K.L., 1995. An event-based short-term traffic flow prediction model. Transport. Res. Record 1510, 45–52.

Head, K.L., Mirchandani, P.B., Sheppard, D., 1992. Hierarchical framework for real-time traffic control. Transport. Res. Record 1360, 82–88.

Khoudour, L., Lesort, J.-B., Farges, J.-L, 1991 PRODYN – three years of trials in the ZELT experimental zone. Recherche-Transports-Securite (English issue: Special Traffic Management), pp. 89–98.

Maher, M.J., 1984. Estimating the turning flows at a junction: a comparison of three models. Traffic Eng. Control 25, 19–22.

Mauro, V., Di Taranto, D., 1990. UTOPIA. In: Proceedings of the Sixth IFAC/IFIP/IFORS Symposium on Control and Communication in Transportation, Paris, France.

Mekky, A., 1979. On estimating turning flows at road junctions. Traffic Eng. Control 20, 486–487.

Nihan, N.L., Davis, G.A., 1989. Application of prediction-error minimization and maximum likelihood to estimate intersection O–D matrices from traffic counts. Transport. Sci. 23, 77–90.

Sen, S., Head, K.L., 1997. Controlled optimization of phases at an intersection. Transport. Sci. 31, 5–17.

van Zuylen, H.J., 1979. The estimation of turning flows on a junction. Traffic Eng. Control 20, 539–541.