RECURSIVE APPROACHES FOR ONLINE CONSISTENCY CHECKING AND O-D DEMAND UPDATING FOR REAL-TIME DYNAMIC TRAFFIC ASSIGNMENT OPERATION

Xuesong Zhou
xszhou@umd.edu

and

Hani S. Mahmassani*
masmah@umd.edu
Corresponding author

Maryland Transportation Initiative
Department of Civil & Environmental Engineering
University of Maryland
1173 Glenn L. Martin Hall
College Park, MD 20742
Phone: 301-405-7768
FAX: 301-405-2585

July 28, 2004
Revised on November 12, 2004

Total words: 5500
+ Figures: 6
+ Table: 1
= 7250

Submitted for Presentation at the 84th Annual Meeting of the Transportation Research Board, January 2005, Washington, D.C. and Publication in Transportation Research Record
Recursive Approaches for Online Consistency Checking and O-D Demand Updating for Real-Time Dynamic Traffic Assignment Operation

Xuesong Zhou and Hani S. Mahmassani

Abstract

To maintain an internally consistent representation with actual traffic conditions, this paper presents an OD demand consistency checking and updating model for on-line dynamic traffic assignment operation. Both predictive and reactive approaches are proposed to minimize (1) the deviations between simulated states and real-world observations and (2) OD demand adjustment magnitude. The two objectives are combined into a weighted linear quadratic function to construct a guaranteed overdetermined optimization problem. Alternative recursive solution algorithms are presented to design an efficient feedback controller that regulates the demand input for the real-time dynamic traffic assignment simulator. The proposed model is tested using field data from the Irvine test bed network.

Keywords: Real-time Dynamic Traffic Assignment; Traffic Estimation and Prediction; Intelligent Transportation Systems; Traffic Management Systems
INTRODUCTION

Substantial research over the last two decades has been devoted to the Dynamic Traffic Assignment (DTA) problem (1-3). In order to avoid mathematical intractability, simulation-based dynamic traffic assignment is intended to capture dynamic tripmaker decisions and complex traffic processes in the practical deployment for realistic networks. The traffic simulation-assignment framework has been further extended into the design of real-time DTA systems in order to provide traffic state estimation and prediction for ITS network applications. In particular, the simulator is embedded to describe how traffic flow patterns develop spatially and temporally according to specific user behavior and system assumptions, given on-line estimated and predicted OD demand trip desires. The internal traffic network representation of the real-time DTA simulator forms an important basis for evaluating the effectiveness of alternative management decisions.

Doan, Ziliaskopoulos and Mahmassani (4), Kang (5) and Peeta and Ziliaskopoulos (1) enumerated possible error sources that could cause the potential divergence of the predicted system states from the actual traffic conditions unfolding on-line. These error sources include (i) incorrect prediction of dynamic OD demand, (ii) incorrect route choice predictions, (iii) incorrect traffic flow modeling, (iv) incorrect assumptions on driver behavior and/or response to information provided, (v) unpredicted incidents, as well as (vi) incorrect assumptions on system related parameters. Without correcting OD demand prediction errors in the DTA simulator, the inconsistency in OD flows would accumulate and propagate in the traffic simulation process, making the network state estimation and prediction highly unreliable. This critical operational issue in the deployment of real-time DTA systems has motivated the development of OD demand consistency checking and updating models, which aim to minimize the deviations between the real-world measurements and the simulated states by adjusting OD demand input into the DTA simulator.

Kang (5) designed a diagnostic architecture for checking system consistency in a real-time DTA system. Based on a proportional-integral-derivative (PID) feedback control framework, Mahmassani et al. (6) and Kang (5) further proposed a real-time long term consistency updating (LTCC) module, which heuristically adjusts the demand level according to the discrepancy between simulated and observed link density. In addition, they also developed a greedy short-term consistency checking (STCC) algorithm to correct traffic flow modeling errors by regulating traffic flow propagation in the DTA simulator. Along this line, a dynamic programming approach was further proposed by Zhou and Mahmassani (7) for on-line freeway flow propagation adjustment. Peeta and Bulusu (8) proposed a mathematical programming approach for ensuring on-line consistency, which seeks to minimize deviations between real-time traffic measurements and predicted network states. The OD demand adjustment is calculated by iteratively solving a deterministic DTA problem and a least squares problem within a stage-based rolling horizon framework. The resulting rank deficient least squares problem is tackled by a generalized singular value decomposition strategy, leading to intensive computational requirements.

To date, the theoretical and algorithmic aspects of the dynamic OD demand consistency checking and updating problem are still relatively undeveloped. In order to meet the operational requirements for real-time DTA systems, two critical issues need to be addressed, namely, (1) how to construct an optimization formulation that results in stable adjustment solutions, and (2) how to develop efficient solution algorithms suitable for large-scale real-time applications. Additionally, numerical experiments in the previous studies are based on simulated data sets in relatively small networks, and real-world traffic surveillance data have not been fully utilized to test and compare alternative algorithms and implementation strategies.

The following sections begin with an overview of the recursive demand prediction-correction procedure in the real-time DTA system. Predictive and reactive adjustment models are then presented to correct demand estimation errors in the DTA simulator. After presenting several solution algorithms, this paper concludes with numerical experiments using field data from the Irvine test bed network.

RECURSIVE DEMAND PREDICTION-CORRECTION MECHANISM
A real-time DTA system, for example, DYNASMART-X developed by Mahmassani et al.\(^2\), seeks to perform the following functional capabilities: (i) estimate the current traffic states in the network, (ii) provide future network traffic states for a pre-defined horizon in response to various control and information dissemination strategies, (iii) estimate dynamic OD demand in the current stage, (iv) forecast dynamic OD demand in future stages, and (v) ensure the consistency between simulated states and actual conditions.

This section uses a Kalman filtering framework to illustrate the prediction-correction methodology in real-time DTA and to classify possible error sources in the OD demand estimation and prediction models\(^5\),\(^10\),\(^11\). First, let \( k \) be a subscript for OD demand estimation stages, and let \( X_k \) and \( Y_k \) be the state variable vector and measurement vector at stage \( k \), respectively. In addition, \( H_k \) represents the measurement matrix that relates measurement \( Y_k \) and state \( X_k \), and \( A_k \) represents the transition matrix that maps the state variables from stage \( k \) to \( k+1 \). The Kalman filter used in OD demand estimation and prediction relies on a transition equation to describe demand evolution dynamics

\[
X_{k+1} = A_k X_k + w_k
\]  
(1)

and a measurement equation to link OD demand states to traffic measurements

\[
Y_k = H_k X_k + v_k
\]  
(2)

where \( w_k \sim \text{Normal}(0, W_k) \) and \( v_k \sim \text{Normal}(0, V_k) \) are process noise and measurement noise at stage \( k \), respectively.

Based on the demand transition matrix, the dynamic OD demand state at stage \( k \) is predicted from stage \( k-1 \) by propagating the mean

\[
\hat{X}_{k,k-1} = A_k \hat{X}_{k-1,k-1}
\]  
(3)

and covariance estimates

\[
P_{k,k-1} = A_k P_{k-1,k-1} A_k^T + W_k
\]  
(4)

where

\( \hat{X}_{k,k-1} \) = prediction of \( X_k \) using observations up to stage \( k-1 \),

\( \hat{X}_{k,k} \) = estimation of \( X_k \) using observations up to stage \( k \),

\( P_{k,k-1} \) = predicted state variance covariance matrix of \( X_k \) at stage \( k-1 \), i.e. \( \text{Var}(X_k - \hat{X}_{k,k-1}) \),

\( P_{k,k} \) = estimated state variance covariance matrix of \( X_k \) at stage \( k \), i.e. \( \text{Var}(X_k - \hat{X}_{k,k}) \).

After receiving new traffic observations obtained at estimation stage \( k \), the Kalman filter adds new information consisting of measurement \( Y_k \) into the \textit{a posteriori} estimates of OD trip demand. Specifically, the mean of OD demand is updated by adding a correction term in proportion to deviations between measurement \( Y_k \) and predicted system output \( H_k \hat{X}_{k,k-1} \), i.e.,

\[
\hat{X}_{k,k} = \hat{X}_{k,k-1} + K_k (Y_k - H_k \hat{X}_{k,k-1}),
\]  
(5)

where the weighting matrix is

\[
K_k = P_{k,k-1} H_k^T (H_k P_{k,k-1} H_k^T + V_k)^{-1}
\]  
(6)

The variance of the OD demand estimate is updated according to

\[
P_{k,k} = P_{k,k-1} - K_k H_k P_{k,k-1}.
\]  
(7)

Equation (7) clearly indicates, from the \textit{a priori} demand estimate to the \textit{a posteriori} demand estimate, the uncertainty in the OD demand estimator is reduced by \( K_k H_k P_{k,k-1} \).

When the real-time DTA simulator starts at stage \( k \), it can only use available OD prediction \( \hat{X}_{k,k-1} \) as the demand input, indicating that there exists demand prediction errors \( \hat{X}_{k,k} - \hat{X}_{k,k-1} \). In order to correct the errors in the \textit{a priori} estimate, a demand updating formula equivalent to Equation (5) should be applied to the real-time DTA simulator, after measurement \( Y_k \) is collected from the traffic surveillance system. Without this correction step, the demand prediction errors would propagate to the subsequent network state prediction through the simulation-assignment process. Since the real-time OD demand
estimator relies on the DTA simulator to provide measurement matrix \( H_k \) (i.e. link proportions) in updating equations (5) and (7), an inconsistent internal representation of the DTA simulator could in turn make the OD estimator and predictor gradually diverge from the real-world demand process.

One brute force strategy of maintaining system consistency is to backtrack the simulator to the start time of stage \( k \), and to re-simulate the network traffic conditions using the \textit{a posteriori} estimate \( \hat{X}_{k,k} \). If the state variable vector \( X_k \) also covers the demand flows departing from previous stages, the DTA simulator should be further backtracked to an earlier stage so as to truly reflect the demand update \( \hat{X}_{k,k} - \hat{X}_{k,k-1} \) for lagged OD demand. Obviously, this computationally intensive strategy is not suitable for a real-time DTA implementation, so the following analysis focuses on an OD demand adjustment method that does not require state resetting.

**PREDICTIVE OD DEMAND ADJUSTMENT MODEL**

As shown in Figure 1, OD demand consistency checking and updating in this study is modeled as a feedback controller. For the DTA traffic simulator, i.e. the plant to be controlled, the control input is the adjusted demand from the consistency checking and updating module; and the system output is the simulated network flow pattern. Taking the real-world measurements as the control reference, the controller seeks to (1) reduce deviations between the simulated states and the real-world observations, and (2) keep the adjustment magnitude as small as possible. Similar to the commonly used minimum energy criterion in the optimal control, the second objective is intended to avoid unsteady responses of the DTA simulator due to dramatic demand adjustments.

The controller can be formulated using a 1-step look-ahead linear quadratic control model. The state space representation of the DTA simulator is simplified using a liquid tank model as shown in Figure 2. Let \( t \) be subscript for departure time intervals. Let \( c_t \) and \( c'_t \) represent total numbers of vehicles in the real-world system and in the traffic simulator, respectively, at departure time interval \( t \). Furthermore, let \( d'_t \) and \( u'_t \) denote total predicted demand volume and total demand adjustment, respectively, at departure time interval \( t \). In the following linear quadratic tracking model, we consider the total number of vehicles in the simulation system as the state variable.

\[
J_t = (c_{t+1} - c'_{t+1})^2 + r(u'_t)^2
\]

where \( r \) = weight for the demand adjustment magnitude.

The transition equation is given by

\[
c'_{t+1} = c'_t + d'_t + u'_t - o'_t
\]

where \( o'_t \) = total number of vehicles exiting from the traffic simulator at departure time interval \( t \).

Performance index \( J_t \) simultaneously minimizes the state deviation at the next departure time interval and adjustment magnitude. In the above simplified single-input single-output system, the system transition equation shows that the total number of vehicles remaining in the simulator at departure time interval \( t+1 \) changes by the addition of the demand input (i.e. predicted demand \( d'_t \) plus demand adjustment \( u'_t \)) minus the total exit flow \( o'_t \) at departure time interval \( t \). This state space representation explicitly considers the intermediate effects of the adjustment implemented at departure time interval \( t \) on the future state at departure time interval \( t+1 \). As a result, the above predictive control strategy can compute manipulated variable adjustments to optimize the future performance of the plant (i.e. DTA simulator). However, this predictive control procedure has one critical shortcoming that prevents its application in real-time settings. That is, the future state of the real-world traffic system \( c_{t+1} \) has not been observed at departure time interval \( t \), so the adjustment \( u'_t \) cannot be determined as a function of \( c_{t+1} \). An alternative strategy is to use a predicted value of \( c_{t+1} \) to generate the reference point for this controller, but the resulting prediction errors might dramatically degrade the actual adjustment performance.

**REACTIVE OD DEMAND ADJUSTMENT MODEL**
To circumvent the above-mentioned difficulty in the predictive model, a reactive approach is adopted in this study to optimize OD demand adjustments for prevailing state deviations. Let $e_t$ denote the deviation between the total number of vehicles in the real-world system and that in the simulator at departure time interval $t$. The reactive adjustment model assumes that

$$e_{t+1} = e_t - u'_t, \quad (10)$$

meaning that the state deviation at next time interval $t+1$ is changed by the addition of the demand adjustment at departure time interval $t$. Accordingly, we have a new performance index

$$\min_{u'_t} (e_t - u'_t)^2 + r(u'_t)^2 \quad (11)$$

with a closed form solution as

$$u'_t = \frac{1}{1+r} e_t. \quad (12)$$

The weighting factor $r$ of the objective in the OD consistency checking formulation can be calibrated using the multi-objective programming techniques discussed in reference (13). Without considering the impact of predicted demand $d'_t$ and existing flow $o'_t$, the system transition equation (10) focuses on system state changes due to demand adjustment $u'_t$, so calculated adjustments are only suitable for a short time interval. Essentially, the reactive control strategy shown in Equation (12) follows the closed loop feedback law: if the current number of vehicles in the simulator is lower (higher) than the observed measures, corresponding to the positive (negative) deviation term $e_t$, then positive (negative) adjustment $u'_t$ is applied to the demand input.

As the above univariate liquid tank model is only an extremely simplified representation of the complex DTA traffic simulation system, we further propose a detailed demand prediction feedback adjustment model to deal with a realistic traffic network, which consists of multiple origins and destinations as well as a set of nodes connected by a set of directed links. If both origin and destination of each vehicle are observed by point-to-point sensors, then the unique state of OD demand flows is determined and the corresponding errors in the demand representation can be explicitly identified. However, for most applications, only limited point detectors are available on a subset of links, resulting in inability to reveal the true origin destination states. Alternatively, the number of vehicles on links is selected as the reference measure in the proposed model. A complete discussion on the selection of state variables in the real-time consistency checking and updating model can be found in the paper by Doan et al. (4).

Using link proportions to describe relationships between OD demand adjustments and changes in the number of vehicles on links, the proposed optimization problem seeks to minimize (1) deviations between the number of simulated vehicles and the number of real-world vehicles and (2) demand adjustment magnitude.

$$\min_{u'_t} \sum_{l,j} \left( e_{(t,j)} - \sum_{i,j} \left( \hat{p}_{(t,l,i,j)} u'_{(i,j)} \right) \right)^2 + r(u'_{(i,j)})^2 \quad (13)$$

Subject to

$$d'_{(i,j)} + u'_{(i,j)} \geq 0. \quad (14)$$

where

- $l$ = subscript for link with traffic measurements, $l=1, \ldots, m$,
- $t$ = subscript for observation time interval in an OD adjustment period, $t=1, 2, \ldots, T$,
- $(i, j)$ = subscript for origin-destination pair, $(i, j) = 1, \ldots, n_{od}$,
- $c_{(t,l)}$ = number of vehicles measured on link $l$ during observation interval $t$,
- $c'_{(t,l)}$ = number of vehicles simulated on link $l$ during observation interval $t$,
- $e_{(t,l)}$ = deviation between $c_{(t,l)}$ and $c'_{(t,l)}$,
- $\hat{p}_{(t,l,i,j)}$ = link proportion in the DTA simulator, i.e. the proportion of vehicular demand flows from origin $i$ to destination $j$, contributing to the number of vehicles on link $l$ during observation interval $t$,
\( u'(i,j) \) = demand adjustment from origin \( i \) to destination \( j \) to be applied in the DTA simulator,
\( d'(i,j) \) = demand volume of OD pair \( (i, j) \) during the current adjustment period in the DTA simulator,
\( r_{(i,j)} \) = positive weight for adjustment magnitude of OD pair \( (i, j) \).

The above optimization problem is a linear quadratic model with inequality constraints, and the positive link proportion matrix and positive diagonal weighting matrix \([r_{ij}]\) can guarantee the positive definiteness of the optimization problem and the existence of a unique solution. In other words, the combined weighting objective function successfully overcomes a possible rank deficient problem posed by limited traffic measurements during an adjustment time interval. If ignoring the nonnegativity constraint, the unconstrained problem has a closed form solution:

\[
\begin{align*}
  u' &= \left( P^TP + R \right)^{-1} P^T e \\
  &= \left( P^TP + R \right)^{-1} P^T (d - d' + u')
\end{align*}
\]  

where

\( R = \) the weighting matrix on demand adjustment, i.e. \( n_{od} \times n_{od} \) diagonal matrix consisting of elements \( r_{(i,j)} \),
\( P = \) link proportion matrix consisting of elements \( \hat{p}(i,j) \),
\( e = \) deviation vector of the number of vehicles,
\( u' = \) demand adjustment vector.

The regulation term in the objective function limits the magnitude of adjustment and consequently reduces the chance of violating the nonnegativity constraint. The constrained problem can be solved by the Lagrangian method. Alternatively, we can first solve the unconstrained problem and ensure the nonnegativity constraints by resetting \( u'(i,j) = -d'(i,j) \) for any demand pair such that \( d'(i,j) + u'(i,j) < 0 \). In essence, the proposed OD demand updating function (15) and the OD demand estimation function (5) in the Kalman filter share a similar feedback function form, that is, the adjustment magnitude \( u' \) and \( \hat{X}_{k,k} - \hat{X}_{k,k-1} \) are proportional to gain factors and error terms, respectively. On the other hand, the proposed OD demand adjustment model can be viewed as a simple least squares optimization program, where the Kalman filter is a least squares estimator that fully considers the second order noise statistics.

EFFICIENT ALGORITHMS AND IMPLEMENTATION ISSUES

Integrated in the real-time DTA system, the OD demand adjustment module operates at every updating time interval as follows.

1. Receive real-world observations from the surveillance system,
2. Measure deviations between the real-world observations and the internal states of the DTA simulator,
3. Obtain link proportions from the DTA simulator,
4. Solve the proposed optimization problem and calculate appropriate demand adjustments,
5. Feed the demand adjustment into the DTA simulator.

The subsequent discussion is devoted to the development of tractable and computationally efficient solution algorithms. The closed form solution (15) requires inverting a \( n_{od} \times n_{od} \) matrix, and the complexity of a direct matrix inversion is \( O(n_{od}^3) \). For a short updating period, the number of observations \( m \times T \) is typically less than the number of OD pairs to be updated. For example, at a 5-minute updating time interval, the ratio of the number of observations over the number of OD demand pairs is only 14.6% in the Irvine test bed network. In this case, a recursive updating scheme without involving direct matrix inversions can reduce the computational time dramatically. We first present the recursive least squares algorithm, which has been widely used in the field of Kalman filtering (12). Its each iteration requires \( 2(n_{od})^2 + 3(n_{od}) \) multiplications, leading to the time complexity of the algorithm is \( O(n_{od})^2 \times m \times T \). In addition, storing the variance matrix \( \Phi(n) \) requires \( O(n_{od})^2 \) memory space.

**Notation**

- \( n \) = observation index at each adjustment period, \( n = 1, 2, \ldots, m \times T \)
- \( e(n) \) = deviation in terms of the number of vehicles based on observation \( n \)
\( u'(n) \) = vector of demand adjustment taking into account observations 1, 2, \ldots, n
\( p(n) \) = vector of link proportions related to observation \( n \)
\( g(n) \) = gain vector related to observation \( n \)
\( \Phi(n) \) = error correlation matrix considering observations 1, 2, \ldots, n
\( \alpha(n) \) = step size scalar related to observation \( n \)

Note that, \( u'(n) \), \( p(n) \) and \( g(n) \) are column vectors with \( n_{od} \) elements.

**Algorithm 1: Recursive least squares algorithm**

Initialization

\[ n=0; \ u'(n)=0 \]

\[ \Phi(n)= \text{diag}\left[ \frac{1}{r(1)}, \frac{1}{r(2)}, \ldots, \frac{1}{r(n_{od})} \right] \]

Main loop

For each measurement \( n = 1 \) to \( m \times T \),

1. Compute the gain vector
   \[ g(n) = \frac{\Phi(n-1)p(n)}{1+\phi(n-1)p^T(n)p(n)} \] (16)

2. Update the estimate of the adjustment vector
   \[ u'(n) = u'(n-1) - g(n)[p^T(n)u'(n-1) - e(n)] \] (17)

3. Update the error correlation matrix
   \[ \Phi(n) = \Phi(n-1) - g(n)p^T(n)\Phi(n-1) \] (18)

End Loop

Although the recursive least squares algorithm is able to reduce the complexity from \( O(n_{od})^3 \) to \( O(n_{od})^2 \), calculating and maintaining a huge variance covariance matrix \( \Phi(n) \) is both time and memory demanding, especially for a real-time feedback controller with hundreds of state variables. A simple sub-optimal algorithm with stable performance is preferred in this context. By simplifying time-varying matrix \( \Phi(n) \) to a constant scalar \( \phi(n) = \frac{1}{r} \), we have

\[ g(n) = \frac{\phi(n-1)p(n)}{1+\phi(n-1)p^T(n)p(n)} = \frac{p(n)}{\frac{1}{r} + p^T(n)p(n)} = \frac{p(n)}{r + p^T(n)p(n)}. \] (19)

This leads to the following normalized incremental gradient algorithm, which is extensively used in the field of adaptive filtering and neural networks. Each iteration of this algorithm only requires \( 3 \times n_{od} \) multiplications and \( n_{od} \) memory space. Note that, a positive step size term should be added for this gradient algorithm, as the above gain factor is an approximation for the optimal search direction.

**Algorithm 2: Normalized incremental gradient algorithm**

Initialization

\[ n=0; \ u'(n)=0 \]

**Main Loop**

For each measurement \( n = 1 \) to \( m \times T \),

1. Compute the gain vector
   \[ g(n) = \frac{p(n)}{r + p^T(n)p(n)} \] (20)
(2) Update the estimate of the adjustment vector
\[ u'(n) = u'(n-1) - \alpha(n)g(n)[p^T(n)u'(n-1) - e(n)] \]  
(21)
End Loop

In the above algorithm, the demand adjustment for each OD pair is proportional to \( p^T(n)p(n) \). If ignoring this product term, then we obtain a standard incremental gradient algorithm, where \( g(n) = \frac{1}{r} p(n) \). This simplified algorithm only requires \( 2 \times n_{ad} \times m \times T \) multiplications and \( n_{ad} \) memory space. Compared to the optimal recursive least squares algorithm, both sub-optimal algorithms are substantially faster with less memory requirements. Principally, the normalized gain factor is able to recognize the impact of overlapping OD pairs in the OD demand adjustment process. Its advantage can be illustrated using a simplified network shown in Figure 3, where vehicular flows along OD pair \( a \) pass through link 1 and vehicular flows along OD pairs \( b \) and \( c \) pass through link 2. We can further assume the link proportions from OD pair \( a \) to link 1, from OD pair \( b \) to link 2, and from OD pair \( c \) to link 2 are all 100%. If the same deviations are observed on links 1 and 2, i.e. \( e(1)=e(2) \), the standard incremental gradient algorithm would suggest that the OD demand adjustments for three OD pairs have the same magnitude, leading to possible over-adjustments for traffic states on link 2. Since product terms \( p^T(1)p(1)=1 \) and \( p^T(2)p(2)=2 \), according to the normalized incremental algorithm, the individual demand correction for overlapping OD pairs \( b \) and \( c \) would be only half of the correction for independent OD pair \( a \). As shown by Bertsekas (14), the above three algorithms are variants of the incremental gradient method to solve the sequential linear quadratic problem. These algorithms process the data blocks in sequence, and they mainly differ in terms of the calculation related to the gain matrix (factor). In addition, the recursive least squares algorithm only needs a single pass through the entire data to reach the optimum, where two sub-optimal algorithms might require multiple passes to find the minimum, although a single pass typically produces a very dramatic decrease in the value of the objective function. Because the link proportion matrix in the adjustment formulation is still a simplified representation of the input-output relation for the DTA simulator, it is difficult to make the simulator reach completely the desired state even using an “optimal” adjustment based on the proposed mathematical model. In fact, computational efficiency is a more important consideration. A desirable real-time demand regulator should require substantially small computation effort while maintaining adjustment performance. In addition, for a stable feedback controller in a complex and dynamic real-time environment, the weighting matrix (factor) \( R \) must be properly tuned to obtain a desired response to a given disturbance. A small \( R \) indicates a large gain, quick response and small errors when a disturbance occurs. On the contrary, the choice of a large \( R \) will result in great errors under disturbances, but tends to be stable when the process structure changes. Experiments are also needed to select an appropriate demand updating frequency, which is jointly determined by the execution time for retrieving link proportions from the DTA simulator and computational efficiency of the demand adjustment algorithm.

**NUMERICAL EXPERIMENTS**

The Irvine test bed network (9) is used to evaluate the performance of different OD demand adjustment algorithms. As shown in Figure 4, this network includes 61 OD demand zones, 326 nodes and 626 directed links, where traffic counts are measured at 30-second intervals on 19 freeway links and at 5-minute intervals on 28 arterial links. The root mean squares error in terms of number of vehicles on links with observations is selected as the consistency measure.

\[ RMSE = \sqrt{\frac{\sum_{t} \sum_{i} (c_{i,t} - \hat{c}_{i,t})^2}{mT}} \]  
(22)

In the following experiments, the roll period of network state estimation is set to 0.5 minute, and the execution cycle and prediction horizon for network state prediction are 5 minutes and 20 minutes,
respectively. The OD demand estimation time length is 5 minutes, and OD prediction model executes every 10 minutes and forecasts further OD demand for next 45 minutes. To specifically examine the impact of demand consistency checking on the system performance, the short-term (flow propagation) consistency checking module is not activated in the following experiments. The archived data on the first day are used to construct the \textit{a priori} estimate of the regular demand pattern \cite{11} by using an off-line OD estimation method. Real-world observations on the second day are used to calibrate system parameters in the real-time OD estimation and prediction, and consistency checking models. Data from the third day are used to validate the proposed algorithms. The time of interest in the following experiments is the morning peak period (4:00 AM – 10:00 AM).

\textbf{(1) Computational performance}

The first task is to compare the computational performance of the three consistency solution algorithms. Table 1 lists average execution time and estimation errors for these three algorithms at a 1-min updating interval. Obviously, the efficient sub-optimal approach is suitable for real-time operation, and the exact solution algorithm does not satisfy the real-time response constraint for the medium-scale network considered in the study. However, the optimal adjustment solution result can still serve as a useful benchmark for comparing the performance of sub-optimal algorithms. The percentage improvements shown in Table 1 are the relative error reduction compared to the do-nothing case (i.e. without OD demand adjustment). Overall, both sub-optimal algorithms can dramatically reduce the estimation errors, and the normalized incremental algorithm outperforms the standard algorithm, because the former can capture the overlapping effect in OD demand adjustments.

Figure 5 details the time-varying performance for different consistency checking algorithms. From 4:00 AM to 6:30 AM, the benefit of all three adjustment algorithms is insignificant, as the overall OD demand level remains steady and OD prediction errors are relatively small. After 6:30 AM, the estimation errors in the do-nothing case increase sharply, which can be attributed to high OD prediction errors during the peak period. Between 6:30 AM and 7:30 AM, all three algorithms are able to reduce the system inconsistency, and the optimal solution procedure clearly generates a lower bound of error reduction relative to the other two sub-optimal regulators. During the off-peak period (7:30 AM to 8:30 AM), the deviations between the real-time simulator and real-world system cannot be significantly reduced by any OD demand consistency checking algorithm. By carefully comparing the simulation results with the real-world observations on a link-by-link basis, we find that the remaining large state deviations are more likely due to incorrect route choice prediction or incorrect traffic flow modeling. As only the OD demand consistency checking module is activated in this study, the system might attribute all the state inconsistency to the OD demand prediction errors, leading to slightly worse results compared to the do-nothing case. This observation underscores the need for a joint consistency checking system, which should correctly recognize and correct system inconsistency caused by different error sources. After 9:00 AM, the adjustment algorithms again produce significant error reductions.

\textbf{(2) Step size in sub-optimal algorithms}

The following experiments are designed to tune the step size in the two sub-optimal algorithms. The minimum estimation error for the normalized incremental gradient algorithm is obtained for a step size of 0.5. The standard incremental gradient algorithm gains the best performance when the step size is around 0.05. The dramatic difference between these two step size parameters is due to the existence of the term \( p^T(n)p(n) \) in the normalization algorithm. As expected, when the step size parameter in both sub-optimal algorithms decreases, the experiment results show that the OD demand correction magnitude tends asymptotically toward zero, and the performance improvement due to demand consistency checking and updating becomes gradually negligible.

\textbf{(3) Updating time interval}

With different updating time intervals, Figure 6 depicts the performance of the normalized incremental gradient algorithm in terms of network state estimation and prediction errors. Under all three updating intervals, the sub-optimal algorithm produces dramatic error reductions for the real-time traffic state estimation. Moreover, the short (i.e. 1-min and 2.5-min) updating interval can control the near-term prediction errors within a certain range, compared to a sharp prediction error increase in the do-nothing case. This indicates that frequent updating is preferred in order to rapidly respond to possible OD demand prediction disturbances. In comparison, a longer updating interval corresponds to less frequent updates,
implicitly allowing more time for OD demand prediction errors propagating in the DTA simulator. The figure clearly illustrates that, at a 5-min updating interval, the influence of error propagation becomes very significant when the prediction horizon extends from 10 minutes to 20 minutes. In this case, OD demand consistency checking only provides marginal quality improvement for medium-term network state prediction.

**CONCLUDING COMMENTS**

To maintain the consistency between the internal representation of the simulator and real-world traffic observations, this paper deals with OD demand consistency checking and updating for on-line dynamic traffic assignment operation. Predictive and reactive formulations are developed to minimize (1) the deviations between simulated states and real-world observations, and (2) OD demand adjustment magnitude. These two objectives are combined into a linear quadratic function that guarantees convexity and uniqueness in the resulting model. Alternative efficient solution algorithms and implementation strategies are proposed to design a robust real-time demand consistency checking and updating module. The proposed model is tested using field data from the Irvine test bed network. The experiment results show that the normalized incremental algorithm and the standard algorithm are able to generate on-line OD correction solutions for the real-time DTA simulator, and the former sub-optimal adjustment algorithm makes significant reductions in both estimation and prediction errors, especially with a short updating interval.

A promising direction in dynamic OD demand consistency checking and updating is how to utilize other emerging ITS data sources, such as point-to-point AVI measurements, to uncover the true state of OD flows and identify the OD demand prediction errors. It should be also noticed that, the proposed models rely on accurate link proportion matrices to optimize OD demand adjustments, and other error sources such as incorrect route choice predictions and incorrect traffic flow modeling can significantly impact the accuracy of link proportions and further make the network state prediction become highly unreliable. Thus, on-line DTA applications, especially in network with complex route choice alternatives, call for the further development of consistency adjustment models for identifying and reducing route choice and traffic flow propagation errors, as well as a comprehensive and integrated framework for ensuring system-wide consistency at different levels.

**ACKNOWLEDGEMENTS**

This paper is based in part on research funded through the US Department of Transportation. The work has benefited from the contributions of several current and former graduate students, including Xiao Qin, Robert H. Mahfoud, Ying Kang and Chung-Cheng Jason Lu. The authors are of course responsible for all results and opinions expressed in this paper.
REFERENCES


TABLE 1 Computational performance of consistency solution algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Avg. execution time (sec)</th>
<th>Avg. RMSE</th>
<th>Percentage improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recursive least squares algorithm</td>
<td>192.83</td>
<td>18.17</td>
<td>44.8%</td>
</tr>
<tr>
<td>Normalized incremental gradient algorithm</td>
<td>1.13</td>
<td>23.99</td>
<td>28.4%</td>
</tr>
<tr>
<td>Standard incremental gradient algorithm</td>
<td>0.82</td>
<td>27.12</td>
<td>19.1%</td>
</tr>
</tbody>
</table>

FIGURE 1 Feedback control model for OD demand adjustment.

FIGURE 2 Conceptual state space representation of traffic simulator.

FIGURE 3 Idealized network for illustrating the advantage of the normalization.
FIGURE 4 Detailed Irvine network.
FIGURE 5 Performance of three consistency checking algorithms.

FIGURE 6 Estimation and prediction errors under different updating time intervals.