Incorporating Feedback Loop into FSUTMS for Model Consistency

Final Report (Volume I)

Contract No. BC-791

September 2003

FSUTMS
Version 5.4 (April 2000)
Florida Department of Transportation
Systems Modeling Office
Transportation Planning Package
Under License Agreement with
The Urban Analysis Group

Prepared for

Office of the State Transportation Planner
Systems Planning Office
Florida Department of Transportation

Prepared by

Department of Civil, Architectural, and Environmental Engineering
University of Miami

and

Lehman Center for Transportation Research
Florida International University

Final Report (Volume I)

Incorporating Feedback Loop into FSUTMS for Model Consistency

Contract No. BC-791

Prepared for
Office of the State Transportation Planner
Systems Planning Office
State of Florida Department of Transportation
605 Suwannee Street, MS 19
Tallahassee, FL 32399-0450

Prepared by

Chang-Jen Lan, Ph.D., P.E.
Assistant Professor

Monica Menendez
Research Assistant

Department of Civil, Architectural, and Environmental Engineering
University of Miami
1251 Memorial Dr., 325 McArthur
Coral Gables, Miami, Florida 33146
Phone: (305) 284-6447; Fax: (305) 284-3492
E-mail: clan@miami.edu

and

Albert Gan, Ph.D.
Assistant Professor
Lehman Center for Transportation Research
Department of Civil and Environmental Engineering
Florida International University
10555 W. Flagler Street, EAS 3680
Miami, Florida 33174
Phone: (305) 348-3116; Fax: (305) 348-2802
E-mail: gana@fiu.edu

September 2003

Technical Report Documentation Page

1. Report No.

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle

Incorporating Feedback Loop into FSUTMS for Model Consistency

5. Report Date

September 2003

6. Performing Organization Code

7. Author(s)

Chang-Jen Lan, Monica Menendez, and Albert Gan

8. Performing Organization Report No.

9. Performing Organization Name and Address

Department of Civil, Architectural and Environmental
University of Miami
1251 Memorial Dr., 325 McArthur, Coral Gables, FL 33146

10. Work Unit No. (TRAIS)

11. Contract or Grant No.

BC-791

12. Sponsoring Agency Name and Address

Office of Research and Development
State of Florida Department of Transportation
605 Suwannee Street, MS 30, Tallahassee, FL 32399-0450

13. Type of Report and Period Covered

Final Report (Volume I)
October 2000–September 2003

14. Sponsoring Agency Code

99700-3596-119

15. Supplementary Notes

Supported through a subcontract from the Florida International University. Mr. Shi-Chiang Li, AICP, of the Florida Department of Transportation District 4 Office served as the project manager.

16. Abstract

This project aims to address the well-known problem of inconsistent travel impedances that exists within FSUTMS' four-step traditional demand model by designing and implementing an automated feedback loop in FSUTMS. The direct method and Method of Successive Averages (MSA) method are used for implementing feedback. The results are compared against the preassignment scheme as the benchmark. In terms of the effects on the network performance, the feedback process results in higher speeds, shorter travel times, and lower volume-to-capacity ratios. These changes become more pronounced as the congestion level increases, which are consistent with the findings reported in previous studies, except that the effects of feedback in this study are found to be pronounced at earlier stage, i.e., a medium level of congestion. In terms of Root Mean Square Error (RMSE), the model accuracy is improved marginally due to feedback using the friction factor as determined from the pre-assignment scheme. Feedback with on-line trip length calibration and total VHT (Vehicle Hour Traveled) control through proportional adjustment by iteration (PAI) produces a comparable RMSE and other statistics with the pre-assignment scheme. Overall, the results presented in this report do not provide sufficient evidence to support the significant benefit of feedback process to the model accuracy as claimed by previous research. This can be partially attributed to the fact that the proposed feedback model has not been fully calibrated as in the pre-assignment model. Proper calibration of friction factors governing trip length and other important model parameters seems to bear more significance affecting the model accuracy than the feedback process itself.

17. Key Words

Demand Forecasting, Feedback, Model Consistency, Trip Length Calibration, Method of Successive Averages

18. Distribution Statement

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	118	

ACKNOWLEDGEMENTS

This research was funded by the Research Center of the Florida Department of Transportation (FDOT). The authors wish to thank the project manager, Mr. Shi-Chiang Li, AICP, of the Office of Planning and Environmental Management at the FDOT District 4 Office, for his guidance and advice throughout the course of this research. The following individuals are gratefully acknowledged for their assistance in various capacities: Mr. Xiaofeng Xu, a graduate assistant at the University of Miami, developed the software codes for feedback loop; Dr. Min-Tang Li, a senior research associate at the Florida International University (FIU), provided helpful information and clarification on the execution of TRANPLAN program; and Dr. Jill Strube of the FIU Institute of Government and Metropolitan Center reviewed the entire draft and provided useful comments.

DISCLAIMER

The opinions, findings and conclusions expressed in this publication are those of the authors and not necessarily those of the State of Florida Department of Transportation.

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Like other travel demand forecasting models, the Florida Standard Urban Transportation Model Structure (FSUTMS) is not immune to under- and over-assignments of trips to individual links or screenlines. Various means of model calibration and validation have been developed in an attempt to reduce the degree of assignment errors due to numerous causes and to obtain more accurate trip forecasts. This project aims to address the well-known problem of inconsistent travel impedances within FSUTMS' four-step traditional demand model. In this model, trip generation, trip distribution, mode choice, and trip assignment are calculated consecutively, using the results from the current step as input for the next. The problem of inconsistent travel impedances arises because travel impedances are a function of network congestion, but the level of congestion is not known prior to trip assignment. Consequently, free-flow travel impedances are usually used during the trip distribution and/or modal split steps. For demand forecasts involving congested networks, this problem of inconsistency can have a major impact on the output, as travel impedances, such as travel times and average speeds, are more sensitive at high flow rates. Consequently, the overall pattern of travel flows, such as origin-destination trips and/or their purpose-specific breakdown, may be unrealistic.

The need to achieve consistency in the four-step modeling process is well recognized and has been the subject of research for many years. Theoretically, the combined model was first proposed in 1973 to remedy inconsistencies in travel impedances within the four-step model by solving trip distribution, modal split, and traffic assignment steps simultaneously. The approaches suggested for solving the combined model can generally be categorized into the "direct optimization" and "iterative" methods. Under the direct optimization method, model components from each step are formulated jointly as an optimization problem and solved directly based on a convex combination method like the Frank-Wolfe technique. The iterative method, on the other hand, solves the doubly constrained gravity model and assignment in an iterative manner until convergence is achieved. The well-behaved convergence properties of the iterative method have been theoretically proved.

Practical implementation of a feedback mechanism was not of particular interest until the Annual Meeting of the Transportation Research Board (TRB) in 1993 when officials from the Federal Highway Administration (FHWA) formally addressed the iterative "feedback" procedure in the context of the four-step travel forecasting process in response to concerns about the inconsistency of travel patterns resulting from the traditional four-step procedure. One approach designed to achieve consistent travel impedances provides a feedback loop that iterates through trip distribution, mode split, and traffic assignment several times with or without a successive averaging step until travel impedances stabilize. The feedback procedures with and without successive averaging steps are the Method of Successive Averages (MSA) and the direct method, respectively. The other common approach used in FSUTMS to achieve partial consistency is to perform one round of pre-assignment before the four steps. The free-flow skim times and the congested skim times from the pre-assignment step are weighted and combined into a approximated measure of travel impedance to be used as the input for the formal trip distribution step. All of these methods are performed and compared using a small artificial network and medium real networks in this study. The advantages and disadvantages of each method are also analyzed.

Due to the different levels of congestion, FSUTMS reveals fairly distinctive behaviors of convergence on the small artificial network as expected. Through the direct feedback method, the base and medium scenarios did not reach convergence. In those scenarios, the impedance and link volume oscillate between two states and do not reach convergence. These observations confirm the fact that the direct feedback method can not guarantee convergence. On the other hand, the direct feedback process slowly converges as the network becomes congested. The MSA method differs from the direct method in that it applies the weighted average scheme on results obtained from previous iterations before feeding back to the next iteration. Due to the effect of weighted average, the convergence was significantly accelerated on the small network in all scenarios. The advantage of MSA becomes even more pronounced in the medium and congested scenarios. This demonstrates that the MSA method facilitates the convergence behavior, especially when the network is at a higher level of congestion.

The test results from three scenarios of the small network were used to assess the effects of feedback on the impedance measures and the final link loading. For evaluation purposes, the average change in link-specific measures was used. Under the base scenario, the change in the link loading is significant and the feedback process redistributes the overall trips more evenly over the network. The link-specific travel time and speed are not much affected due to light demand. Under the medium scenario, the change in link-specific volume becomes highly significant, indicating again that the feedback process redistributes the overall trips more evenly. Consequently, the changes in link-specific travel time and speeds become significant. The feedback process results in higher speeds, shorter travel times, and lower volume-to-capacity ratios. As the congestion level further increases in the “congested” scenario, the changes in all of the measures are highly significant, with the effects of feedback being consistent with those revealed from the previous scenarios. Overall, these observations were consistent with the findings reported in previous studies, except that the effects of feedback in this study were found to be pronounced at earlier stage, i.e., a medium level of congestion.

As a comparison for the real networks of medium size, the results indicate that feedback with the method of successive average (MSA) has better convergence behavior than the direct method. Both of the original planning models have performed the pre-assignment scheme to incorporate the effect of ‘composite’ congestion skim time on the final network loading. One of the key elements associated with the pre-assignment scheme is that the friction factors governing the gravity model were calibrated using the trip length distribution compiled from the modeled Origin-Destination (OD) trips, instead of the observed distribution. These OD trips were mapped from the true OD data collected from survey and the corresponding trip lengths were measured based on the free-flow skim times (before pre-assignment) and composite congestion skim times (after pre-assignment). Both means and variances of the modeled trip length for each purpose are about 24% systematically lower than the observed statistics. Although the resultant trip length distributions deviate far from the observed ones, the overall assigned VMT and VHT were controlled to perfectly balance (volume-to-count ratio ≈ 1.0). It should be noted, however, that this perfect control is the consequence of not only the one-time calibrated friction factors, but also many other parameter calibration, such as speed-capacity table, BPR function, etc.

With the identical friction factors used in the pre-assignment scheme, the feedback with MSA results in a slightly lower volume-to-count ratio (≈ 0.97), suggesting that flow loading pattern is affected by the feedback process. One plausible cause is the shortened trip length and distance due to congested skim time. This change is consistent with the previous findings in the small artificial network. Therefore, the friction factors governing the trip length distribution may need to be calibrated on-line during the course of feedback to possibly rectify the systematic discrepancy.

To that end, a simple but effective scheme called the proportional adjustment by iteration, was proposed and implemented along with the feedback process for on-line trip length calibration. The basic assumption of this approach is that every individual trip length is proportionally adjusted upward or downward according to the overall VHT (vehicle-hour-traveled) volume-to-count ratio. This assumption can be justified when there exists difference between the scales of the real and the modeled networks. Using this approach, the authors were able to control the overall VMT volume-to-count ratio to 1.0 and VHT volume-to-count ratio to 1.01 at convergence. The RMSE and other statistics appear reasonably well. The resulting trip length distribution was close to the distribution resulting from the pre-assignment scheme. However, curve shapes of the normalized friction factors by trip purpose are more well-defined than the factors resulted from the pre-assignment scheme.

The results presented in this report do not provide sufficient evidence to support the significant benefit of feedback process to the model accuracy as claimed by previous research overall. This is partially attributed to the fact that the proposed feedback model has not been fully calibrated as in the pre-assignment model. Proper calibration of friction factors governing trip length and other important model parameters seems to bear more significance affecting the model accuracy than the feedback process itself.

CHAPTER 1
INTRODUCTION

1.1. Motivation

Like other travel demand forecasting models, the Florida Standard Urban Transportation Model Structure (FSUTMS) is not immune to under- and over-assignments of trips to individual links. Acceptable percentages of these assignment errors for different facility types are given in *Urban Transportation Planning Model Update-Phase II, Task C, Develop Standard Distribution and Assignment Models* (FDOT, 1981). In fact, some of the errors can be so large that the growth in trips for the forecast year is not sufficient to overcome the magnitude of trip under-assignments, resulting in forecast trips that are lower than those of the base year for some links. This often leaves some modelers to wonder if their models were correct (Li *et al.*, 1999).

Although it is clear that demand models are inherently susceptible to under- and over-assignments, improvements may be made to potentially minimize these assignment errors to obtain more accurate trip forecasts. This research incorporates a feedback loop that aims to minimize assignment errors in FSUTMS. The procedure uses a feedback mechanism that loops selected steps within the FSUTMS model structure in an attempt to achieve better estimates of travel impedances and, thus, more accurate trip assignments.

1.2. Equilibrium Feedback

The equilibrium feedback is to address the well-known problem of inconsistent travel impedances that exists within the FSUTMS four-step traditional demand model. In this model, trip generation, trip distribution, mode choice, and trip assignment are calculated consecutively, using the results from the current step as input to the next. The problem of inconsistent travel impedances arises because travel impedances are a function of network congestion, but the level of congestion is not known until the trip assignment step has been completed. Free-flow travel impedances are usually used during the trip distribution and/or modal split steps. For demand forecasts involving congested networks, the inconsistency problem can have a major impact on the output, as travel impedances such as travel times and average speeds are more sensitive at high flow rates. Consequently, the overall pattern of travel flows, such as origin-destination (OD) trips and/or their mode-specific breakdown, may be unrealistic.

The need to achieve model consistency in the four-step process is well recognized and has been the subject of research for many years. The current approach to achieving consistent travel impedances utilizes a feedback loop that iterates through trip distribution, mode split, and traffic assignment several times until travel impedances stabilize; this approach is the focus of this project. A recent report by the Blue-Ribbon Panel (2002) of the Florida Model Task Force (MTF) identifies feedback modeling as "critical to ensure that, at a minimum, all steps in the modeling process are using the same speeds for modeling travel," and that "it makes no sense to have the trip distribution, mode choice, and traffic assignment steps using different speeds (and therefore travel times) for modeling travel."

1.3. Goal and Objective

The goal of this project is to improve the accuracy of trip assignments in FSUTMS by designing and implementing an automated feedback loop in FSUTMS to achieve consistent travel impedances among trip distribution, mode split, and route assignment.

1.4. Report Organization

The rest of this report is organized as follows. Chapter 2 provides a detailed review of the existing literature on feedback modeling. Chapter 3 describes the process of implementing equilibrium feedback in FSUTMS. Chapter 4 reports findings from the feedback implementation in FSUTMS. Finally, Chapter 5 summarizes the findings and suggests recommendations for further studies.

CHAPTER 2
EQUILIBRIUM FEEDBACK

2.1. Introduction

FSUTMS is based on the traditional four-step travel demand modeling process, feeding the output from one step as an input to the next step. These four basic steps are:

- Trip generation: the process of estimating the number of trips generated by or attracted to traffic analysis zones based on socioeconomic, demographical, land use, and other measures of economic activities.
- Trip distribution: the process of distributing trips originating from one zone to other zones based on measures of accessibility and inter-zonal impedances, such as travel time, distance, or other general disutilities.
- Modal choice: the process of calculating the breakdown of trips by different transportation modes, such as public transportation systems using the transit network and privately owned automobiles using the highway network.
- Traffic assignment: the process of loading trips onto the shortest or minimum travel time paths of highway and transit networks among traffic analysis zones based on various traffic assignment principles.

As explained in Chapter 1, one inherent weakness of this sequential process is that the travel impedances used in trip distribution and modal split are usually not consistent with those estimated from traffic assignment. The negative impact of this inconsistency on congested networks can be especially significant and can lead to an unreasonable overall pattern of travel flows and contribute to the overall assignment errors. The use of a feedback loop that iterates through trip distribution, mode split, and traffic assignment several times until travel impedances stabilize is one common approach to achieving consistent travel impedances.

2.2. Feedback Mechanism and Procedure

Although different in terminology, the feedback mechanism that contemporary researchers and practitioners use actually traces its root to Evans' (1973) iterative method, which later became known as the combined model. The combined model was first proposed to remedy inconsistencies in travel impedances within the four-step model by solving trip distribution, modal split, and traffic assignment steps simultaneously. Since Evans (1973), a number of research studies, including Florian *et al.* (1975), Evans (1976), Frank (1978), Abdulaal and LeBlanc (1979), Sheffi and Daganzo (1980), LeBlanc and Farhangian (1981), have attempted to address such aspects as algorithmic development, convergence property, and parameter calibration of various combined models.

The various approaches suggested for solving the combined models can generally be categorized into the "direct optimization" and "iterative" methods (Lam and Huang, 1992). Under the direct optimization method, model components from each step are formulated jointly as an optimization problem and solved directly based on convex combinations methods such as the Frank-Wolfe technique (1956). To illustrate the underlying optimization problem, let

t_{ij} = set of origin-destination flows, an aggregate result of many individual travel decisions,
O_i = number of trips originating in zone *i*, and
D_j = number of trips with destinations in zone *j*.

Fixing the number of trips leaving origins only, i.e., $\Sigma_j\, t_{ij} = O_i$, results in the singly constrained model. Fixing the number of trips attracted to each destination in addition to those leaving origins, i.e., $\Sigma_i\, t_{ij} = D_j$, results in the doubly constrained model. Finally, if the number of origin-destination (OD) trips is assumed to be proportional to the number of trips produced from the origin and the number of trips attracted to the destination and inversely proportional to the square of the path cost, similar to the gravity model, one could derive the doubly constrained model as

$$t_{ij} = A_i B_j O_i D_j\, exp\, \{\beta c_{ij(min)}\} \tag{2.1}$$

where $c_{ij(min)}$ = minimum path costs between zone *i* and zone *j*,
$A_i = 1/(\Sigma_j B_j D_j \beta\, c_{ij})$,
$B_j = 1/(\Sigma_i A_i O_i \beta\, c_{ij})$, and
β = calibration constant.

Based on the entropy maximization principle, the joint distribution/assignment problem can be formulated as the following minimization program:

$$Z(\mathbf{x},\mathbf{t}) = \Sigma_{kl} \int_0^{x_{kl}} c_{kl}(x)dx \;+ 1/\beta\, \Sigma_i\, \Sigma_j\, t_{ij}\, [1n\{t_{ij}\} - 1] \tag{2.2}$$

where $c_{kl}\,(x)$ = link cost as a function of link volume x.
(I don’t think all terms are defined: Z, x, t are not in any of the other equations, though T is in 1.1)

On the other hand, the iterative method, originally proposed by Evans (1973), solves the doubly constrained gravity model and assignment in an iterative manner until convergence is achieved. The original Evans algorithm can be described in the following steps (let *n* denote the *n*th iteration):

Step 1. For each link (*k,l*), compute the cost c^n_{kl} as a function of $x^n_{kl.}$

Step 2. For each origin zone *i*, find the minimum cost of traveling to each destination zone *j* and determine a minimum cost path from *i* to *j*.

Step 3. Find auxiliary OD trips q^n_{ij} by solving the doubly constrained gravity model of Eq. (2.1).

Step 4. Calculate the auxiliary link flow y^n_{kl} on link (*k,l*) by assigning q^n_{ij} to the minimum path identified in Step 2.

Step 5. Set $(\bar{\mathbf{x}}^n, \bar{\mathbf{t}}^n) = (1 - \lambda)(\mathbf{x}^n, \mathbf{t}^n) + \lambda(\mathbf{y}^n, \mathbf{q}^n)$ and solve the one-dimensional search problem, defined as follows, to obtain the step size $\lambda^* \in [0, 1]$

$$\min_\lambda\; Z(\bar{\mathbf{x}}^n, \bar{\mathbf{t}}^n) = \Sigma_{kl} \int_0^{\bar{x}_{kl}} z_{kl}(x)dx \;+ 1/\beta\, \Sigma_i\, \Sigma_j\, \bar{t}_{ij}\, [1n(\bar{t}_{ij}) - 1]$$

Step 6. Set $(\mathbf{x}^{n+1}, \mathbf{t}^{n+1}) = (\bar{\mathbf{x}}^n, \bar{\mathbf{t}}^n)_{\lambda=\lambda^*}$ and check for convergence. Terminate if the stopping criterion $Z(\mathbf{x}^n, \mathbf{t}^n) - Z(\mathbf{x}^{n+1}, \mathbf{t}^{n+1}) \le \varepsilon$ (where ε is a user-specified threshold) is satisfied; otherwise, set *n=n+1* and go to Step 1.

Evans (1976) provided proofs that the above iterative method converges to the following desired equilibrium conditions (Boyce *et al.*, 1994):

1. The generalized travel costs among each OD pair equal the sums of the individual link costs over the used routes. No other unused route carries a lower travel cost.
2. The number of OD trips depends on the generalized travel costs determined by the link flows resulting from the traffic assignment equilibrium condition.

Note that in Evans' algorithm, both link flows and OD trips from the previous iteration are weighted and averaged with the results from the current iteration using the optimal weight obtained from a one-dimensional (line) search to expedite and ensure convergence. Note also that the above algorithm only applies to the single-purpose and single-mode cases. The extension of Evans' algorithm to multiple purposes and multiple modes is nontrivial (Lam and Huang, 1992; Boyce *et al.*, 1994).

Horowitz (1989) proposed an *ad hoc* modification to Evans' algorithm that requires less computational time and memory. In Evans' algorithm, significant computational time is spent on the second term of Eq. (2.2), while the impact of the term in achieving the final solution decreases with each iteration. Horowitz proposed to eliminate the second term completely to reduce the computational time and storage requirement, claiming that convergence could be achieved without it. Note that only link flows are weighted and averaged in Step 5 of Evans' algorithm. However, Huang and Lam (1992) demonstrated that Horowitz's algorithm does not always converge to the optimal solution. They proposed an alternative modification that ensures convergence. The trade-off of this alternative is that their modified algorithm requires more time than the Horowitz's (but less than the original Evans algorithm).

During the 1993 Annual Meeting of the Transportation Research Board (TRB), officials from the Federal Highway Administration (FHWA) formally addressed the iterative "feedback" procedure in the context of the four-step travel forecasting process in response to concerns on inconsistency of travel pattern resulted from the traditional four-step procedure (Boyce *et al.*, 1994). One approach for such feedback was to directly iterate through trip distribution, mode split, and traffic assignment several times, hoping that the travel impedances will eventually stabilize. However, almost universally, researchers and practitioners have experienced difficulty in achieving convergence in a feedback process when assignment outputs directly convert to trip distribution inputs for the next iteration.

A number of alternatives to the direct iterative method have been proposed to reduce processing time and ensure convergence. For example, the Method of Successive Averages (MSA) averages the volumes and/or other measures from previous iterations with the most recent assignment using the inverse of the weighted current iteration number (Sheffi, 1985). It has been proven both theoretically and empirically that both methods would converge to the same solutions (Evans, 1973, 1976). Evans' algorithm is much more time consuming in each iteration due to additional line search but may converge to the final solution more rapidly in less number of iterations, thanks to its better convergence property.

2.3. Regulation Requirements

The practical use of the equilibrium feedback procedure in transportation planning became increasingly popular after the passage of the Clean Air Act Amendments (CAAA) of 1990. In 1992, the U.S. Environmental Protection Agency (EPA) designated the procedure as a necessary element in the travel forecasting process (COMSIS, 1996). Since then, several empirical studies have been conducted to try to improve the accuracy of the current travel forecasting models.

Stopher (1993) examined the ability of the traditional four-step models to meet the requirements established by the CAAA. He deemed that the major part of the impact of the transportation control measures (TCM) was impossible to model with the traditional methods. He identified the inability of the traditional four-step models to forecast measures affected by the time of day, number of trips, people walking or biking. He called for the development of new models that would account for all these aspects, in addition to auto ownership, location, growth changes, etc. Stopher explained the critical role of the link volumes and average speeds for determining equilibrium. The implementation of models on the Geographic Information System (GIS) platform was recommended. In his conclusion, he recommended running the travel forecasting models through a series of feedback loops throughout the entire planning model, including a land use element, to obtain more accurate values for all the conditions stipulated by the CAAA. Almost at the same time, Harvey and Deakin (1993) published a manual for the National Association of Regional Council on the transportation modeling practice for air quality.

Because of these national regulations, practical implementation of the feedback procedure to achieve equilibrium travel impedances prevails at the Federal, State, and regional levels. The Federal Highway Administration (FHWA) requires that Metropolitan Planning Organizations (MPO) implement equilibrium feedback in their Urban Travel Forecasting Procedure (UTFP). A study done by Miller (1997) is an example of the effort from the Federal agency.

2.4. Implementation Methods

Even though the equilibrium feedback procedure is an inherent part of the widely used Evans' algorithm, several alternative approaches have been developed at different times and different levels. Safwat and Magnanti (1988) developed the Simultaneous Transportation Equilibrium Model (STEM) designed to predict demand and performance levels on large-scale transportation networks simultaneously by balancing the behavioral and computational aspects of equilibrium modeling. Trip distribution was calculated using a logit model (instead of the entropy model), and trip generation was based on the theory of users' behavior in the transportation system (instead of being fixed as in previous models). The globally convergent algorithm they proposed used a "direction finding" step that could be implemented at any iteration. The paper showed that the assumptions regarding the invertability, symmetry, and negative definiteness of the transportation demand functions were not as restrictive as previously thought.

Safwat and Hasan (1989) investigated the computational efficiency of the logit distribution function used in STEM under the impacts of different demand levels and model parameters.

Even though a variety of factors, including trip generation parameters, trip distribution parameters, minimum trip generation, attractiveness measure, network configuration, network size, convergence criterion, accuracy level, and link performance functions, are known to affect the convergence property of the solution algorithm, only the first two are emphasized in their model. The influence of each of these two parameters on the rate of convergence was found to be sensitive to other model constants. In addition, the effect of the trip generation parameters on the rate of convergence was also sensitive to any changes in the trip distribution parameters, and vice versa. However, the effect of the trip distribution parameters was more significant. The compound effect of the two factors on the rate of convergence was significantly larger than the individual effect. Upper bounds were also found for these two parameters, above which the computational time requirement was too long and the algorithm did not reach the desired accuracy level.

Chu (1989) proposed a combined trip distribution and assignment model using dogit destination functions (in contrast with most of the previous literature that focused mainly on the logit and entropy distribution models). Following the same scheme, Chu (1990) incorporated captive travel behavior on his model. Unlike most of the previously proposed methods, it was able to handle situations in which the observed trip patterns were represented by both compulsory (captive) and discretionary (free) travel behavior (Chu, 1989).

Winslow *et al.* (1995) introduced the feedback process into the Urban Transportation Modeling System (UTMS) as a mean to forecast travel demand patterns based on various socioeconomic factors and roadway characteristics. The iterations during feedback allowed for the information from sub-regions to be reused for the calculations for the regional models, improving the accuracy of both models at the local and regional levels.

Metaxatos *et al.* (1995) introduced and implemented the feedback procedure using Evans' algorithm in the destination and route choice model of the EMME/2 system. They proposed several possibilities for future improvements to the model, including:

1. Considering additional costs (insurance, depreciation, fuel, parking, tolls, etc.);
2. Adapting the actual occupancy factor to the specific area as opposed to assuming one person per car to facilitate increase in accuracy in the air quality estimates; and
3. Incorporating mode choices and integrating transit operations into the macro, to allow for the study of other modes of travel, and the impact of different variables, such as waiting time, loading time, and headway.

In addition, Boyce and Zhang (1997) proposed an operational method for executing the four-step forecasting model with feedback, including calibration of the combined model of trip distribution, modal split, and traffic assignment using data collected from the Chicago area.

Abrahamsson and Lundqvist (1999) proposed to use the feedback process for three of the four traditional steps in travel forecasting. They claim that the relative values for the cost-sensitivity parameters obtained by the conventional method are inappropriate. While the traditional method assumes mode choice is conditional based on destination choice, the proposed reverse combined model assumes destination choice is conditional based on mode choice.

In 1999, the USDOT assembled a group of transportation professionals to review the travel demand model used by the Wasatch Front Regional Council (WFRC). The original model was considered unreliable because of the presence of several inconsistencies, and failure to meet some of the basic criteria for model validity. Drawbacks include the underestimation in auto trip lengths, overestimation in the vehicle occupancy and congestion reduction, rough representation of travel times, and ineffective land use scenario simulation and peak spreading effects. In the same year, the direct feedback method was implemented in the WFRC Travel Demand Model, but unsystematic results were found. To overcome some of these drawbacks, one of the recommendations made by the expert panel was the implementation of a feedback mechanism based on the method of successive averages for convergence; this proposal was finally adopted in 2000 (Rodier *et al.*, 2000).

In Florida, several more sophisticated models in the FSUTMS family, such as the Southeast Florida Regional Planning Model (SEFRPM), include a one-iteration feedback loop to produce congested impedances for use in the second (or final) round of distribution and modal split of home-based work trips. Although this one-iteration feedback loop allows the use of congested impedances in trip distribution and modal split, it is not designed to achieve consistency travel impedances in the four-step process.

2.5. Convergence Criteria

Boyce, Zhang, and Lupa (1994) used three fundamental variables as convergence measures: highway link flows, automobile and transit trip tables, and automobile generalized costs (transit generalized costs were fixed for all the models). To achieve convergence, the following two equilibrium conditions concerning link flows, travel times, and OD trips need to be satisfied.

1. The sum of individual costs over the used routes had to equal the generalized cost from each origin to each destination, and
2. No unused routes should have a lower cost.

Also, the total number of trips per hour from each origin-destination pair depends on the generalized travel costs at equilibrium condition.

Winslow *et al.* (1995) used the two following criteria to verify the convergence of the measures utilized by the USDOT:

1. Ground counts versus assigned volumes, and
2. Vehicle-mile-traveled (VMT) calculated from ground counts versus VMT from assignment.

A COMSIS study (1996) recommended the use of more than one criterion in order to determine convergence. A total of five measures were proposed, including:

- Percent change in average speed by functional class and area type,
- Percent of links with less than five percent change in assigned volume,

- Root mean square (RMS) of assigned link volume,
- Percent of person trips with less than ten percent change in origin-destination flows, and
- RMS of origin-destination flows.

Among them, link volumes and average operating speeds on links were considered the two most important criteria. It was also noted that the convergence criteria might vary according to the size of the model. If a particular facility were the focus of the application, then the vehicle miles traveled (VMT) would be a more feasible measure for convergence criterion. On the other hand, should the application be regionally oriented, the use of link volume is more appropriate.

2.6. Comparative Studies

Several studies have attempted to compare the effectiveness of feedback and the different methods of achieving feedback. Based on data from the Baltimore-Washington metropolitan area, Levinson and Kumar (1993) compared a model with feedback that included six components (trip generation, destination choice, mode choice, departure time choice, route assignment, and intersection control) and the traditional model without feedback. They found that the algorithm with feedback obtained more reasonable forecasts and more consistent travel times based on trip generation input and trip assignment output than the traditional model without feedback. Recognizing that the feedback procedure required a considerable amount of computational time and memory, the authors suggest implementing some heuristic averaging or equilibration processes to circumvent this disadvantage.

A study done by Boyce, Zhang, and Lupa (1994) compared three different methods for incorporating feedback into the four-step model:

1. Iterations of trip distribution, modal split, and all-or-nothing (AON) assignment with method of successive averages (MSA).
2. Iterations of trip distribution, modal split, and user-optimal (UE) assignment with MSA.
3. Evans' algorithm.

The results of data from an area of Chicago with approximately 300 zones and 3,000 highway links were compared with those of the four-step process without feedback. In general, these authors showed that Evans' algorithm produced better results than other methods and that repetitive iterations of the four-step model in an *ad hoc* manner would not produce satisfactory results. They noted that Evans' algorithm performed better than the other methods in most, but not all, cases.

Tatineni *et al.* (1994) analyzed the effects of the new transportation policies on travel patterns and compared the results between the combined model and the traditional sequential model. The combined model eliminates the inconsistencies found on the traditional model by performing several iterations until the final equilibrium travel conditions exists among destination, mode, and route choices all together, rather than route choice alone. It also considers a common cost function to model various travel choices including changing factors such as transit fares, fuel costs, and land use (considering the dispersion of employment locations). Results from these three scenarios were then carefully studied to find their impacts on travel patterns and other parameters related to the network. To improve the applicability of the combined model for various policy analyses, they proposed to take into account route choice dispersion in addition to location and mode choice dispersions. Predicting variations of the overall trip rate and average automobile occupancies and the implementation of the model using more user-friendly transportation planning software or any popular transportation planning software such as the EMME/2 system were considered extremely important.

Metaxatos *et al.* (1995) implemented Evans' algorithm as the feedback process for the destination and route choice models in the EMME/2 system. To verify the validity of the new method, a combined model for the City of Winnipeg, Manitoba in Canada was solved; the results showed suitable rates of convergence.

Winslow *et al.* (1995) used data from the State of New Jersey to compare the traditional methodology used by the New Jersey Department of Transportation (NJDOT) and a new methodology that introduced the feedback process into the UTMS. Results showed wide discrepancies in trip distribution tables from the two methodologies, including number of trips from each origin, and the distribution on the type of facility. In terms of total ground counts versus assigned volumes, however, the model with feedback showed only a marginal improvement over the one without it.

Walker and Peng (1995) compared the results from the direct iteration method, the method of successive averages, and the Evans algorithm. They applied data from the regional highway performance monitoring system, 1990 highway traffic counts summarized by screenline, and public transit ridership to the travel simulation model used by the Delaware Valley Regional Planning Commission. It was found that the Evans algorithm required the least amount of time, but did not achieve the same degree of link-level convergence as the simple iteration or the method of successive averages. However, almost all the iterative formulations tended to overestimate the congested highway link speeds, possibly due to the use of data for speed limits rather than actual speeds during congested times.

A major study was conducted by COMSIS Corporation in 1996 to investigate the methodologies and drawbacks from the current feedback models using City of Memphis, Tennessee, and Salt Lake City, Utah as case study sites. Table 2.1 illustrates the distinctive differences in terms of network coding, trip purposes, and models for trip distribution and mode choice in the two sites.

Table 2.1. Attributes and Characteristics of Case Study Sites (COMSIS, 1996).

Attributes	**Memphis**	**Salt Lake City**
Network coding - Highway network - Transit network	 All major roadways Local bus, express bus, and cross-town routes	 Most of major roadways Local and premium bus routes
Trip purposes	HBW, HBO, NHB, truck, taxi, and external trips	HBW, HBO, NHB, HB college, commercial, and external trips
Trip modes	Local bus, express bus, and cross-town routes	Drive alone, carpool, local bus, and premium bus
Trip distribution	Gravity model	Gravity model
Mode choice	Multinomial logit model	Nested logit model

In the study, the following five traffic assignment models were implemented with feedback on the test network mentioned above:

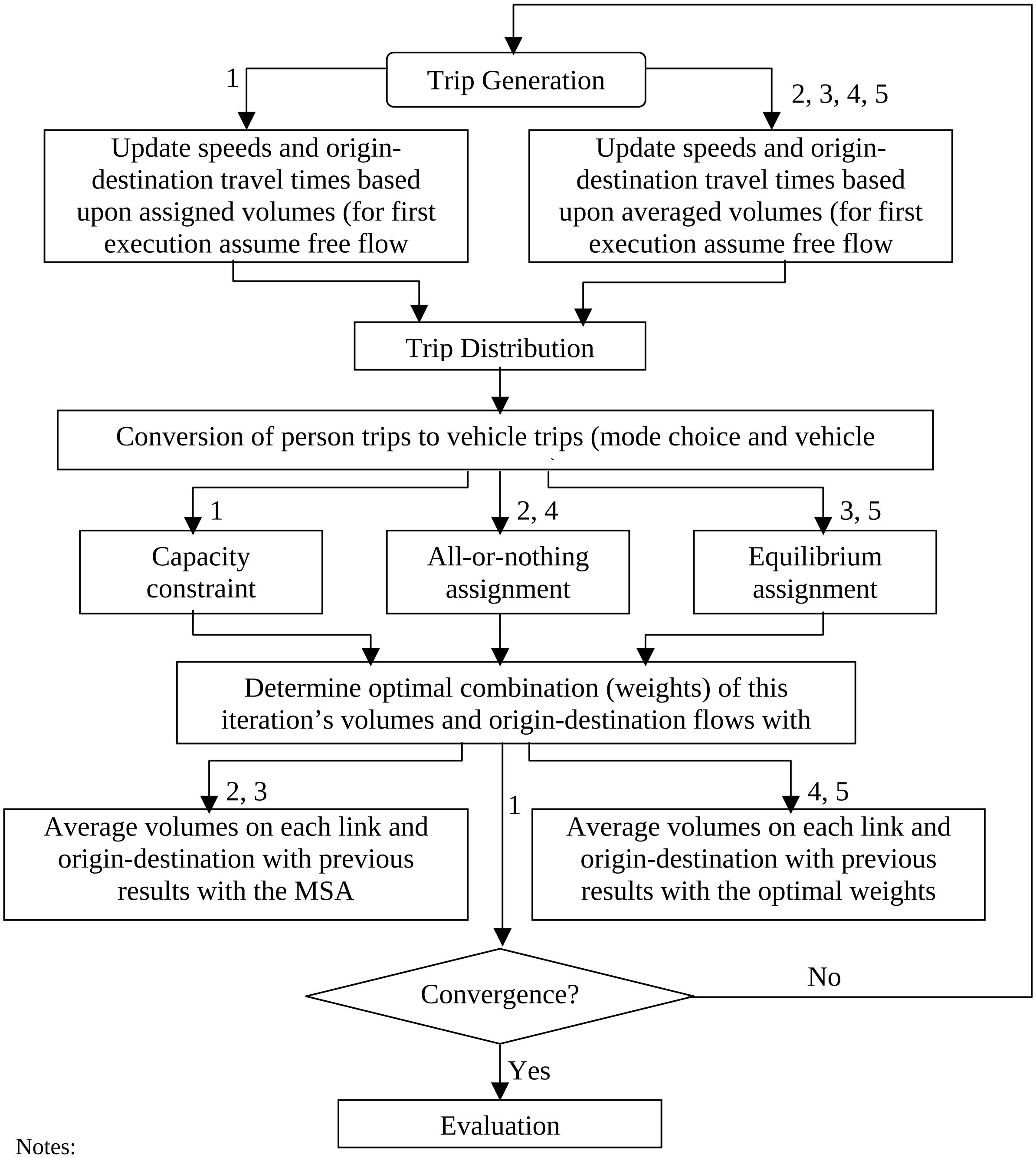
- Direct method,
- Method of successive averages (MSA) with equilibrium assignment,
- Method of successive averages with all-or-nothing (AON) assignment,
- Method of optimal weighting (MOW) with equilibrium assignment, and
- Method of optimal weighting with AON assignment.

Comparisons of the relative advantages and disadvantages among these methods were made using the flowchart shown in Figure 2.1. The first method is the simplest way to apply the feedback, i.e., results from the previous iteration are directly used as input data for the next iteration. The other four methods result from the combination of two assignment algorithms (equilibrium and all-or-nothing assignments) with two averaging schemes of the output measures. In MSA, a weight equal to the inverse of the total number of iterations is used to average the link volumes and origin-destination flows. In MOW, the optimal weight for each iteration is calculated based on Evans' algorithm.

Two options were available regarding the inclusion of the mode choice into the four-step process in this study. The post-feedback mode choice applied the mode choice model after the feedback between trip distribution and assignment is implemented, while the integrated mode choice within feedback implemented the mode choice within each iteration of feedback. The inclusion of the mode choice into the feedback process did not produce any noticeable changes in the number of trips; however, the average system-wide speeds were significantly affected. The increases in this measure by the post-feedback mode choice and integrated mode choice within feedback are 21.3% and 37.0%, respectively.

Among the alternative feedback methods, EQA methods took the least number of iterations to achieve the highest percentage of links with less than 10% change in volume. The AON methods took more than five iterations and did not achieve the same level of accuracy as the EQA methods. In terms of percentage of speed change, the EQA methods converge quickly after a few iterations, while the direct and MSA-AON methods fluctuate and converge slowly. The MOW-AON behaves modestly in this case. In terms of the root mean square of assigned link volume, the EQA methods converge much quickly than the AON methods. Interestingly, the direct method behaves reasonably well in the first and third cases with respect to volume measures.

In terms of execution times, other analyses were made. For the root mean square of the assigned link volumes, the MSA-EQA method converged faster, while the MOW-AON never converged completely. In terms of the percent change in average speed, the MSA-EQA showed the better stability and converged faster, while the MSA-AON varied to a great extent during iterations even though it converged almost at the same time. Finally, in comparing the maximum absolute change in person trip impedance between origin-destination pairs, the MSA-EQA again converged faster than the MOW-EQA as expected. These overall results illustrate the advantages of the averaging methods over the direct feedback model. The averaging methods with equilibrium assignment produced more accurate results and consumed less time than the direct method with equilibrium assignment.


Trip Generation
1
2, 3, 4, 5
Update speeds and origin-destination travel times based upon assigned volumes (for first execution assume free flow
Update speeds and origin-destination travel times based upon averaged volumes (for first execution assume free flow
Trip Distribution
Conversion of person trips to vehicle trips (mode choice and vehicle
1
2, 4
3, 5
Capacity constraint
All-or-nothing assignment
Equilibrium assignment
Determine optimal combination (weights) of this iteration's volumes and origin-destination flows with
2, 3
1
4, 5
Average volumes on each link and origin-destination with previous results with the MSA
Average volumes on each link and origin-destination with previous results with the optimal weights
Convergence?
No
Yes
Evaluation

Notes:

1. Direct Method
2. Method of Successive Averages with All-or-nothing assignment (MSA-AON)
3. Method of Successive Averages with Equilibrium assignment (MSA-EQA)
4. Method of Optimal Weighting with All-or-nothing assignment (MOW-AON)
5. Method of Optimal Weighting with Equilibrium assignment (MOW-EQA)

Figure 2.1. Comparative Study Procedure for Various Feedback Methods (COMSIS, 1996).

The results of the COMSIS study also indicate that the models with feedback are less sensitive than the model without feedback to improvements made on the route in order to reduce congestion. Four stages of the route were taken into consideration: normal, 25% uniform growth, 25% radial growth, and new facility. The following conclusions were made based on the changes that resulted from each model concerning average speed, vehicle miles traveled, and average V/C ratio:

- Average link speed increased,
- Average travel time decreased,
- Average travel distance decreased,
- Average volume/capacity ratio decreased, and
- Total vehicle miles travel decreased.

The compound of these changes suggests that models using the feedback process may offer a more accurate representation of these parameters as well as less deterioration in travel time and average speed in higher level of congestion. On the other hand, models using the feedback process do not provide the same benefit for many projects designed to reduce congestion.

The COMSIS report also detailed the most common problems associated with each process, such as excessive storage requirements, errors due to rounding, and hypersensitivity. Because of the nature of the iterative model and the successive averages method, large errors due to rounding usually accumulate in matrix calculations. To avoid this problem, the report recommends using real numbers for all the trip tables throughout the whole feedback process or using bucket rounding during the averaging procedure, which allows each fractional element be kept and added back in later when accumulating to a whole number. Both of these methods require extra storage space, aggravating the need for real computer capacity.

The COMSIS report concludes that the most efficient way to run the iteration model is to retain all the information from each iteration throughout the whole process. By doing so, however, the process quickly consumes the available storage capacity. In order to reduce this problem, the MSA allows the data from each iteration to be used as it is completed, and then delete the results of the run (COMSIS, 1996).

Hasan and Safwat (2000) compared the traditional demand model currently used by the Texas Department of Transportation (TxDOT) with the Simultaneous Transportation Equilibrium Model (STEM) designed to simultaneously solve the four steps of the sequential method. Both methods were applied to the urban transportation network in Tyler, Texas. These estimates showed 25% improvement over estimates from the existing model when compared to the ground counts.

2.7. Recommendations

As reported by Leblanc and Farhangian (1981), Boyce (1984) and Lam and Huang (1992), the iterative method of Evans' algorithm is more efficient than the direct optimization method in terms of computational time and memory requirements, therefore, more practically useful. In addition, due to various constraints of the FSUTMS software structure, the iterative method is considered more feasible and directly deployable for feedback implementation. The feedback alternatives investigated in this study include the direct feedback method and the Method of Successive Average (MSA). The four-step process with *pre-assignment* method currently used in FSUTMS to possibly achieve consistency will be utilized as baselines to compare with the feedback methodologies proposed in this study.

CHAPTER 3
IMPLEMENTATION OF EQUILIBRIUM FEEDBACK IN FSUTMS

3.1. Introduction

In Chapter 2, different methods of implementing feedback in four-step transportation forecasting models were reviewed in terms of accuracy, convergence criteria, storage requirement, and computational efficiency. Three methods were considered:

- The Method of Successive Averages (MSA)
- The Evans algorithm
- The modified Evans algorithm by Huang and Lam (1992)

Despite the different labels, the Evans algorithm can be equivalently solved by the MSA. The modified Evans' algorithm proposed by Huang and Lam (1992) corrects the inherent non-convergence problem that exists on a theoretical basis in Evans' algorithm, which may or may not exist in practice. Due to the significant degradation of the computational efficiency that results in this modification, the research team decided that MSA is the most appropriate method to investigate the feedback behavior of the planning model.

As pointed out in the previous chapter, the iterative method of Evans' algorithm is more efficient than the direct optimization method in terms of computational time and memory requirements, according to studies by Leblanc and Farhangian (1981), Boyce (1984), and Lam and Huang (1992). In addition, due to various constraints of the current FSUTMS software structure, the iterative method is considered to be more feasible and directly deployable for feedback implementation. Under the iterative category, the direct method, the MSA method, and the optimal weighting scheme resulting from Evans' algorithm have been suggested and implemented in the past. The MSA method performs simple weighting averages between the assigned link volumes (or O-D flows) of the current iteration and the average link volumes (or O-D flows) based on the inverse of the total number of iterations. It has the advantage of saving memory space since the results from previous iterations can be removed as soon as they are used. On the other hand, the optimal weighting scheme requires substantial effort to deal with nonlinear optimization, which is time consuming.

The four-step process with feedback and the direct method were performed to provide baseline data, which were compared against the proposed feedback methodologies. COMSIS (1996) recommended the use of more than one criterion to determine convergence; therefore, a total of five measures were used in this study, including:

- Percent change in average speed by functional class and area type,
- Percent of links with less than five percent change in assigned volume,
- Root mean square error (RMSE) of assigned link volume,
- Percent of person trips with less than ten percent change in origin-destination flows, and
- RMSE of origin-destination flows.

Among these measures, link volumes and average operating speeds on links were considered the two most important criteria. In addition, the convergence criteria were observed to vary according to the size of the model, and were applicable to the analysis of a whole transportation network or the modeling of a regional transportation plan (COMSIS, 1996). If the application were for a particular facility, then the vehicle miles traveled (VMT) would be an appropriate measure for evaluating convergence. On the other hand, if the application were regionally oriented, the link volume would be more appropriate.

3.2. FSUTMS Model Structure

To implement a feedback process in FSUTMS, a thorough understanding of its program structure is required. FSUTMS was customized and adapted for FDOT applications from TRANPLAN, a software developed as part of the URBAN/SYS family of urban planning and related tools to aid transportation planners in analyzing multi-modal transportation systems. A standard procedure and several non-standard procedures are available for running various applications for different urban areas in FSUTMS. The standard procedure in FSUTMS primarily refers to the control files that are included for general purposes. Several options exist within the standard procedure, including:

- Highway network only,
- Single-path transit network,
- Multi-path/single-period transit network, and
- Multi-path/multi-period transit network

These general control procedures are written in files with the extensions *.all and *.hwy for highway networks, and *.all and *.tr1 through *.tr3 for transit networks.

The non-standard procedures were adapted and customized from the standard procedure to serve specific needs and considerations in particular areas. They include those used by planning agencies in Tampa Bay, Miami, Orlando, Turnpike, Jacksonville, and Volusia County, and those that involve multiple planning organizations in their joint effort to model multiple urban areas simultaneously, such as the Southeast regional model and the statewide model.

Despite the different control file structures used in different models and procedures, the basic sub-modules used and the overall process of TRANPLAN are largely similar. Figure 3.1 shows the overall FSUTMS process, which includes ten basic sub-modules:

- EXT (External Model) predicts the travel behavior of trips with at least one end in an external traffic analysis zone.
- GEN (Trip Generation Model) defines de number of trips generated by or attracted to the different traffic analysis zones, according to land use, population, and other measures of economic activity.
- HNET (Highway Network Model) creates a computerized system of nodes and links similar to the highway network, based on all the information provided by the user.
- HPATH (Highway Path-building Model) calculates distances and travel times (skims) between the different traffic analysis zones.
- DISTRIB (Trip Distribution Model) distributes the trips originating in one zone to the

other zones based on some measures of accessibility and economic activity. It uses a gravity model.

- MODE (Mode Choice Model) calculates the number of trips that are going to be carried out by the public transportation systems, and the number of trips that are going to be made using private automobiles. It converts from person trips to vehicle trips.
- HASSIGN (Highway Assignment Model) assigns the trips to the shortest or minimum travel time path between traffic analysis zones.
- HEVAL (Highway Evaluation Model) summarizes and evaluates all the information produced already. Its purpose can be to validate the model or to analyze it.
- EMIS (Mobile Source Emission) estimates area-wide vehicle emissions as required by the Clean Air Act Amendments.
- HPLOT (Highway Plotting Model) produces plot files that graphically represent the entire highway network information produced, validated, and/or analyzed by the software.

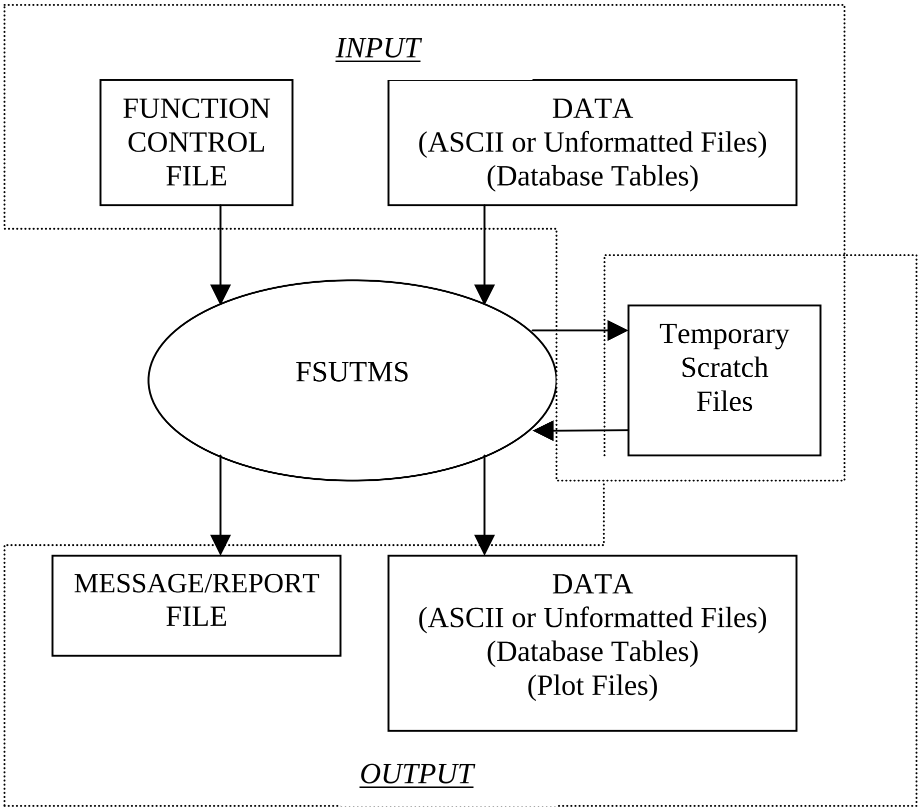

INPUT
FUNCTION CONTROL FILE
DATA (ASCII or Unformatted Files) (Database Tables)
FSUTMS
Temporary Scratch Files
MESSAGE/REPORT FILE
DATA (ASCII or Unformatted Files) (Database Tables) (Plot Files)
OUTPUT

Figure 3.1. FSUTMS Input-Output Structure.

Each basic sub-module can be executed individually and separately from each other, or in combination with others, depending on the purpose of a specific application. Each sub-module uses a set of input and output files. The input files have the extension *xxa*, and output files have the extension *axx*, where *xx* stands for a 2-digit simulation year, and *a* is used to identify different alternatives. Figure 3.1 shows the different types of input and output files used in the program. When FSUTMS is executed, the interface program first generates the *trnplxxx.ins* files in sequence based on the abovementioned control files and all the necessary input files for the specific application (see Figure 3.2). It then calls the TRANPLAN modules to run the process, which produces the final output files from the step as well as the input files for the subsequent sub-module.

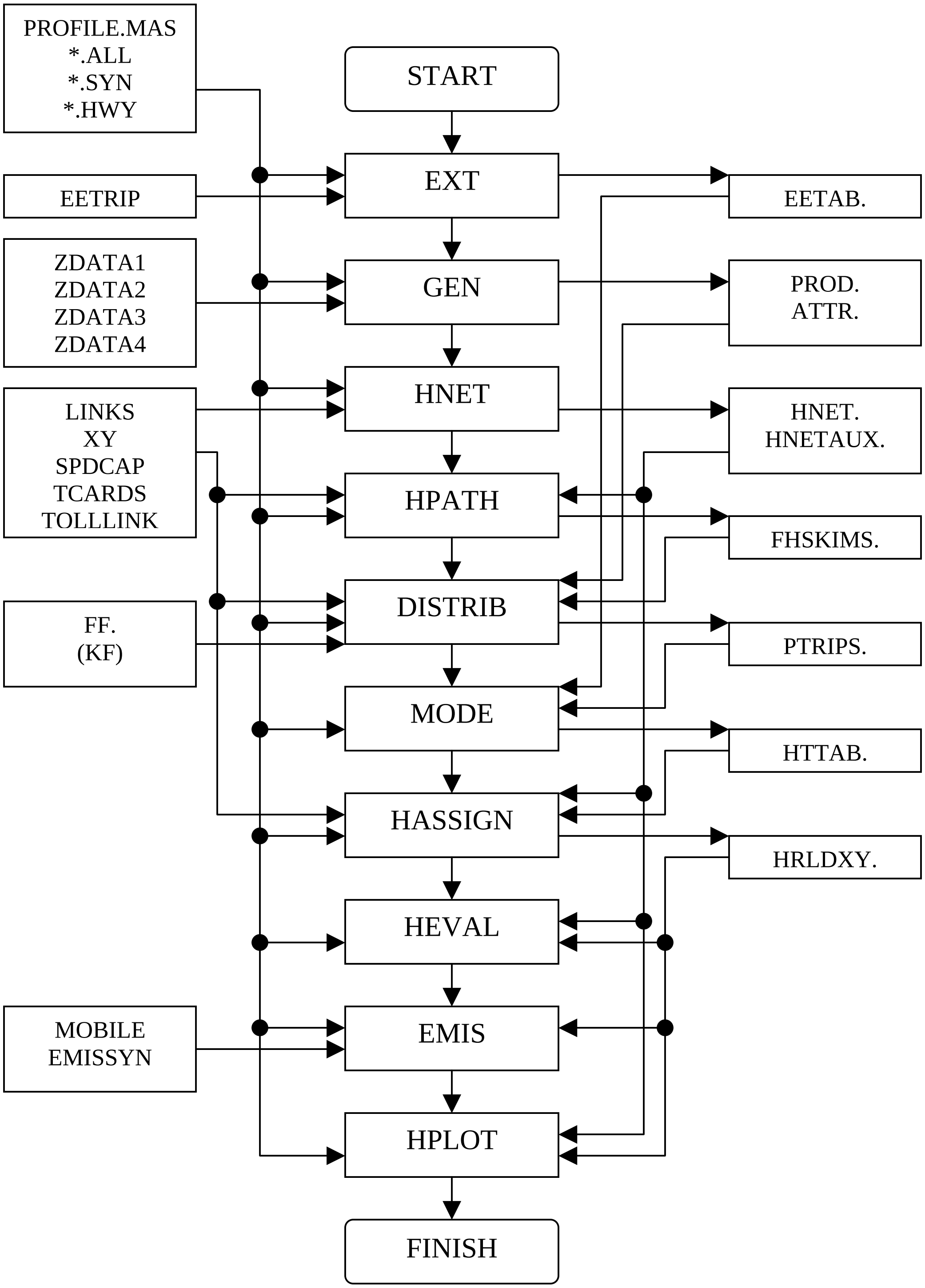


Figure 3.2. FSUTMS Job Process.

The control files are those needed for controlling FSUTMS job processes. They are ASCII (text) files produced by the user to provide the program with instructions for performing one or more functions. The structure of the control files, as follows, is the same for all cases:

```
$function name
$FILES
    INPUT FILE = ftype, USER ID = $fname$
    OUTPUT FILE = ftype, USER ID = $fname$
$HEADERS
    (Up to three header lines)
$OPTIONS
    (List of options)
$PARAMETERS
    (List of parameters)
$DATA
    (Data)
$END TP FUNCTION
```

Each of these instructions is created by the user to provide FSUTMS with all the specifications needed for the execution of each desired function.

- Function Name: The name of the FSUTMS function(s) to be executed and the first line of any control file. It consists of one or more words that uniquely identify the function.
- Files: Specifies the input and output files and the database tables used and/or created with each function. Each file and database table specification must be located in a separate line.
- Headers: Allows up to three lines of project information such as project name and year to be specified. This input does not affect FSUTMS operations.
- Options: Specifies different options for the execution of the above-mentioned function(s). Each function has a set of allowable options.
- Parameters: Specifies all the parameters that must be followed by the program when executing the above-mentioned function. Each function has a set of allowable parameters.
- Data: Provides supplementary input data. Generally, data are specified in either fixed or free format.
- End TP Function: The last entry in all control files. It indicates that FSUTMS has read all of the specifications and data needed to execute the current function.

3.3. Feedback Implementation

To incorporate the feedback process, control files are modified such that they can be executed sequentially (iteratively) and achieve possible convergence. This process, however, could not be done under the current structure of the FSUTMS interface program. An alternative is to use the *fsutmsx.bat* preprocessing batch file provided by FSUTMS to generate the original *trnplxxx.ins* files and then modify these files accordingly. Each time the *fsutmsx.bat* file is executed, in addition to the **.ins* control files, a file named *trnplnx.con* is generated. This file resets the "counter" to the beginning to execute the sequential control file from the path building step.

With the control files and the "counters," the complete process can be executed by a post-processing batch file named *xfsutms.bat,* which is also generated by the preprocessing batch file.

To automate the feedback process, a Visual Basic program called *feedback.exe* is inserted in the end of the post-processing batch file. At the end of each iteration, *feedback.exe* retrieves the travel time outputs from *hrldxy* and compares them with the counterparts from the previous iteration to determine if the convergence criterion is satisfied. If so, the post-processing program terminates and executes the HEVAL module to output the assignment statistics. Otherwise, the program resets the "counter" file to the path building process and reiterates the whole process again until convergence is achieved. The following sections provide detailed implementation logics for both the direct and the MSA method and discuss the effects of feedback implementation on trip length distribution, which illustrate the need for more robust calibration methodology for gravity model.

3.3.1. The Direct Method

The *fsutmsx.bat* file generates 18 *.ins* files, one *trnplnx.con* file, and one *xfsutms.bat* file. The first approach modifies these **.ins*, **.bat,* and **.con* files to incorporate feedback. Since the basic principle of feedback is to replicate the distribution-to-assignment process until possible convergence, the corresponding control files, *trnpl005.ins* through *trnpl018.ins,* were reproduced and renamed to *trnpl019.ins* to *trnpl032.ins,* respectively. The renamed files are placed after the original 18 control files in a sequential order. This new arrangement simulated an additional iteration by repeating all the steps after the assignment step of the first iteration.

The *hrldxy.axx* binary file, which stores the assignment results from first-iteration, was then used as the input file to the highway path building (HPATH) module in the second iteration through the "HIGHWAY SELECTED SUMMATION" function. Since *hrldxy.axx* file uses the same data format as that of the *hnet.axx* file, it can be used to replace the link impedance file from the previous iteration to generate new path information for the next iteration. The *hrldxy.axx* files from different iterations were compared against each other to check for convergence.

Due to the lack of a data format that allows direct retrieval of binary information for comparison purposes, a program named *htp2dbf.exe* was used to convert the output files to the *hrldaxx.dbf* files, which can be imported into a spreadsheet program like Microsoft Excel. After conversion, the *hrldaxx.dbf* files from the first and second iterations were renamed to *hrldaxx1.dbf* and *hrldaxx2.dbf,* respectively. Comparisons were then made by computing the RMSE of each impedance measure between two successive iterations.

Three types of information that are available in *hrldxxx.dbf* can be used to determine convergence. They are congested travel time, speed, and assigned link volumes. This information is found in the last four columns of the *hrldaxx.dbf* file. Table 3.1 provides the description of each data field in *hrldaxx.dbf*. The RMSE of each measure was calculated and the following convergence criterion was used:

$$\frac{RMSE}{Mean \text{ of the previous iteration}} \leq 10^{-3} \tag{3.1}$$

When this criterion is satisfied, the feedback process is terminated; otherwise, the next iteration is executed.

Table 3.1. General Description of the HRLDXY File.

Field	Content
anode	Node number where the trip is generated
Bnode	End node number of the trip
Binkid	Name or number of the link
Altnam	Alternative name of the link
num_lanes	Number of lanes on the link
Cost	Cost associated with each trip
Dist	Length of the links connecting “anode” with “bnode”
speed1	Link free flow speed
speed2	Speed at a certain time, based on link capacity and volume
time1	Link free-flow travel time
time2	Travel time based on link capacity and volume
User	-
Dircod	-
area_type	Type of areas where the trip is located (residential, rural, fringe, etc.)
Facility_type	Type of facility used to travel (freeway, collector, frontage road, etc.)
screenline	Boundary limits of the traffic analysis area
hrlycap	-
Count	Ground count
Twflag	-
concode	-
landuse	Type of land uses of the areas under analysis
Zone	Number of the traffic analysis zones under examination
Geoloc	-
usecode	-
congimp	Congested travel time
congspd	Congested speed
modelcap	Model capacity
Totalvol	Total volume assigned on the link

The *hrldxy.axx* file stores all the link impedance information from each assignment iteration. To obtain consistent congested impedance information used in the distribution step of the next iteration, the parameter IMPEDANCE within the HIGHWAY SELECTED SUMMATION function needs to be specified as LAST ITERATION in order for the program to retrieve link impedance information in the end of the iterative process within the assignment step.

Figure 3.3a shows the modified process of implementing the direct iterative method, in particular, illustrating that the process utilizes the results from the previous iteration as direct inputs to the subsequent iteration without applying the averaging scheme. The **.ins* control files and the *xfsutms.bat* batch file used in the direct method are listed in Appendices A and B, respectively. In addition, the detailed implementation logic of the *feedback.exe* program is presented as a flow chart in Figure 3.3b. The chart shows the data input and output process and the programs used for data conversion. The source code listing for the *feeback.exe* program is provided in Appendix A.

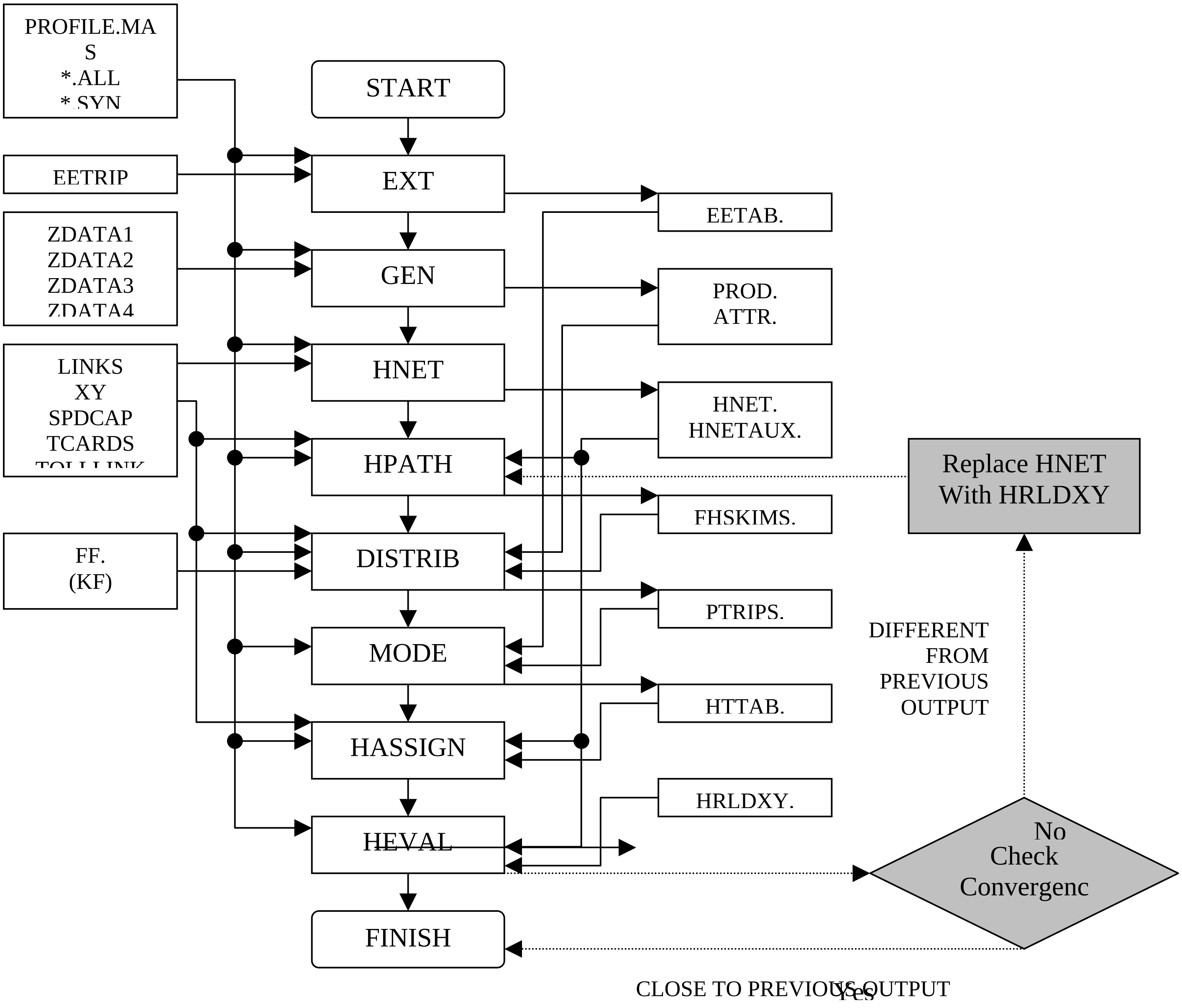


Figure 3.3a. Flow Chart for Feedback with the Direct Iterative Method.

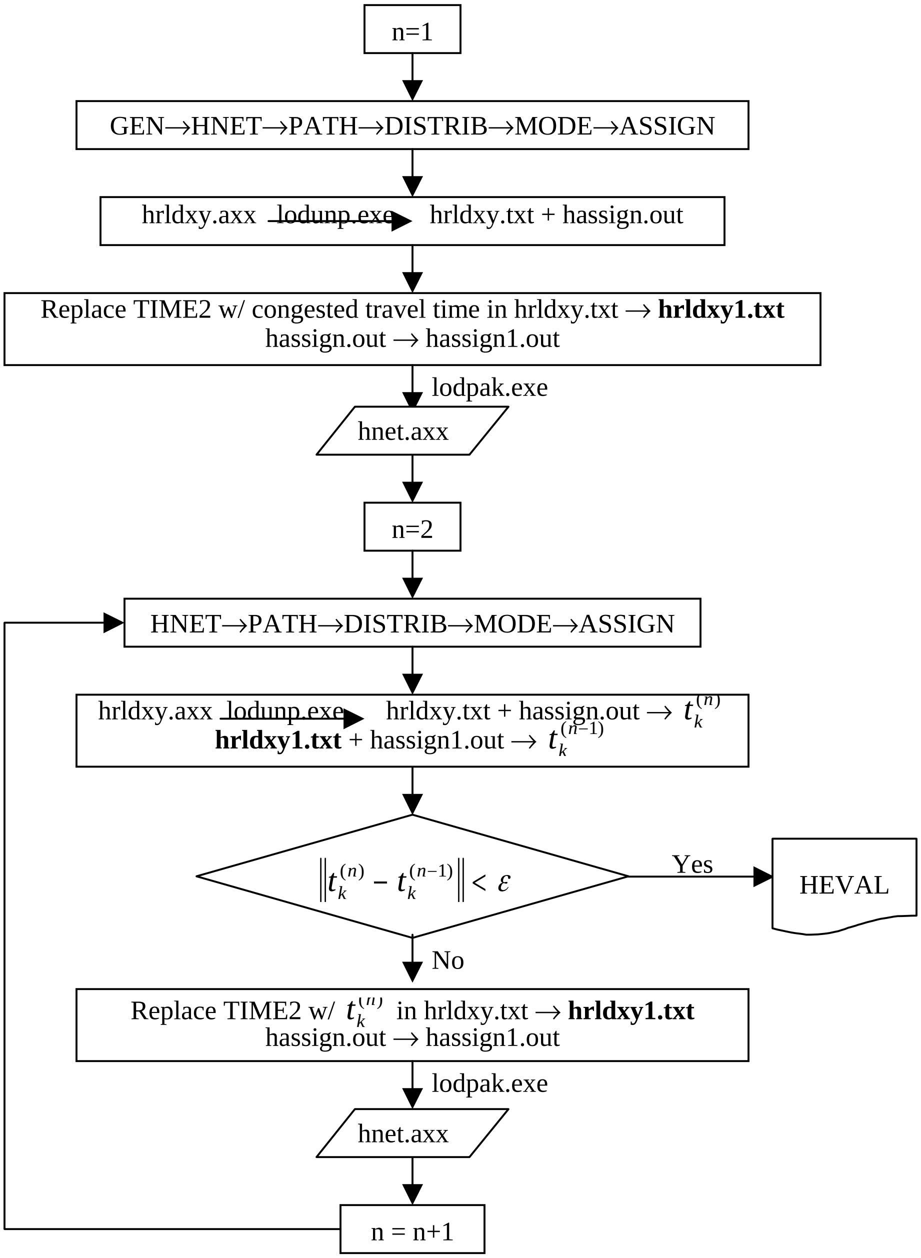

n=1
GEN→HNET→PATH→DISTRIB→MODE→ASSIGN
hrldxy.axx lodunp.exe hrldxy.txt + hassign.out
Replace TIME2 w/ congested travel time in hrldxy.txt → **hrldxy1.txt**
hassign.out → hassign1.out
lodpak.exe
hnet.axx
n=2
HNET→PATH→DISTRIB→MODE→ASSIGN
hrldxy.axx lodunp.exe hrldxy.txt + hassign.out → $t_k^{(n)}$
hrldxy1.txt + hassign1.out → $t_k^{(n-1)}$
$\left\| t_k^{(n)} - t_k^{(n-1)} \right\| < \varepsilon$
Yes
HEVAL
No
Replace TIME2 w/ $t_k^{(n)}$ in hrldxy.txt → **hrldxy1.txt**
hassign.out → hassign1.out
lodpak.exe
hnet.axx
n = n+1

Figure 3.3b. Flow Chart of the Direct Feedback Implementation in FSUTMS.

3.3.2. The MSA Method

The Method of Successive Average (MSA) weighs the results from each iteration in an exponential rate before feeding back to the subsequent iterations. Unlike the direct method, the number of iterations must be specified for the MSA. To that end, another "counter" program is required within the feedback loop executed by *xfsutms.bat*. The users must also define the convergence criteria or maximum number of iterations. Figure 3.4a is a schematic representation of the whole process for the MSA and Figure 3.3b shows the detailed implementation logic of the method in the *feedback.exe* program.

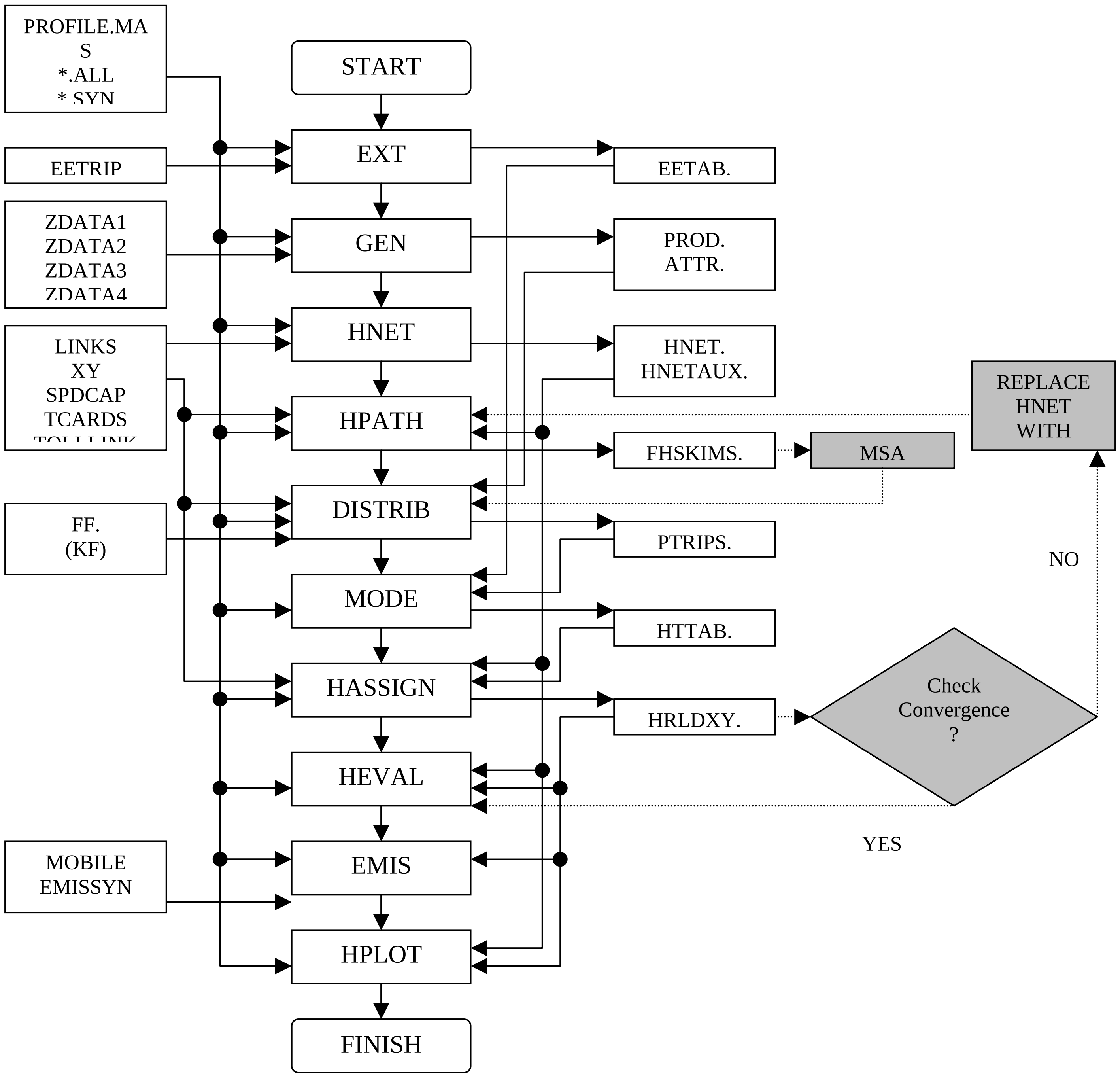


Figure 3.4a. Flow Chart for Feedback with the MSA.

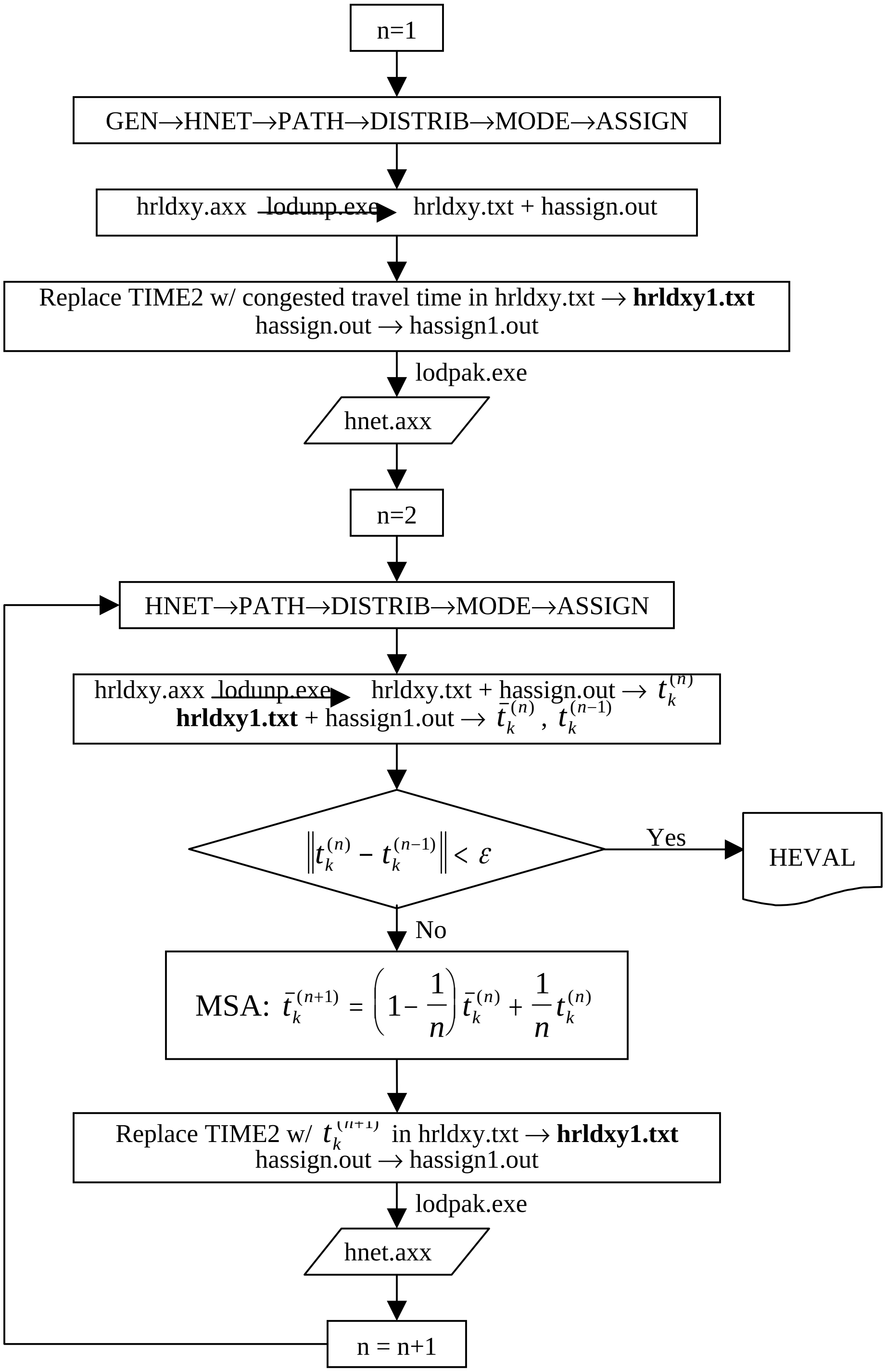

n=1
GEN→HNET→PATH→DISTRIB→MODE→ASSIGN
hrldxy.axx lodunp.exe hrldxy.txt + hassign.out
Replace TIME2 w/ congested travel time in hrldxy.txt → **hrldxy1.txt**
hassign.out → hassign1.out
lodpak.exe
hnet.axx
n=2
HNET→PATH→DISTRIB→MODE→ASSIGN
hrldxy.axx lodunp.exe hrldxy.txt + hassign.out → $t_k^{(n)}$
hrldxy1.txt + hassign1.out → $\bar{t}_k^{(n)}$, $t_k^{(n-1)}$
$\left\| t_k^{(n)} - t_k^{(n-1)} \right\| < \varepsilon$
Yes
HEVAL
No
MSA: $\bar{t}_k^{(n+1)} = \left(1 - \frac{1}{n}\right)\bar{t}_k^{(n)} + \frac{1}{n} t_k^{(n)}$
Replace TIME2 w/ $t_k^{(n+1)}$ in hrldxy.txt → **hrldxy1.txt**
hassign.out → hassign1.out
lodpak.exe
hnet.axx
n = n+1

Figure 3.4b. Flow Chart of the MSA Feedback Implementation in FSUTMS.

In this approach, the output file from *HRLDXY* is compared with the output from the previous iteration. If the difference is greater than a specified convergence threshold, the *HRLDXY* file replaces the *HNET* file in the highway path building process to generate a new skim file to which the MSA weighting factors are applied. Note that, for simplicity, the MSA is applied to the skim files instead of the link (leg) files. The average skim file is then used in the distribution, mode choice, and assignment steps of the next iteration. The complete process repeats itself until the convergence criterion is met. Note also that the MSA is applied to the OD tables as well as from the theoretical standpoint of the combined model.

The basic idea of implementing the MSA feedback procedure in FSUTMS is to continuously feed the average congested link travel times resulted from the current assignments back to HNET until the process converges. In comparison, the output file from the assignment module, *hrldxy.axx*, contains additional data fields for storing congested network conditions, including congested link travel time (CONGIMP), congested speed (CONGSPD), daily capacity (MODELCAP), and assigned link volume (TOTALVOL). Presumably, this information, other than congested network conditions, is interchangeable with the input file to HNET, *hnet.axx*, through program manipulation, provided that the binary format is known. To retrieve and modify the binary *hrldxy.axx* files, the two utility programs, *lodunp.exe* and *lodpak.exe*, provided in TRANPLAN were used to convert the binary format to the text mode, and vice-versa. An example of using these two programs in the DOS mode is given below:

```
>lodunp
 Enter input file name>hrldxy.a00
 Enter output file name>hrldxy.txt
 Delete excessive node coordinates (Y/N)>Y
LODUNP Normal Stop

>lodpak
 Enter input file name>hrldxy.txt
 Enter output file name>hrldxy.a00
LODPAK Normal Stop
```

Note that if the output file exists in either program, the program will stop and generate the following fatal error: “cannot overwrite existing file.” The existing file must be deleted before the programs can function correctly. Although *hrldxy.axx* and *hnet.axx* seem to share a similar format, *lodunp.exe* can only convert *hrldxy.axx*. An attempt to convert *hnet.axx* to the text mode will generate the following error message: “too many bytes read from unformatted record.” Finally, it is important to note that to convert *hrldxy.txt* to *hnet.axx* using *lodpak.exe*, one must change the daily model capacity to the hourly model capacity by dividing it using the congestion factor CONFAC. Errors result when simply replacing the output file name as *hnet.axx* in the above example.

In the feedback procedure, the execution of the these programs is automated in the DOS environment as follows:

```
>lodunp <lan.tmp
```

where *lan.tmp* is a text file containing the following statements:

```
hrldxy.a00
hrldxy.txt
Y
```

Similarly,

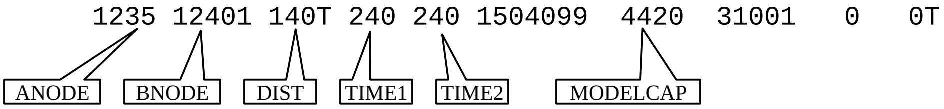
```
      >lodpak <lan1.tmp
```

lan1.tmp contains the following statements

```
      hrldxy.txt
      hrldxy.a00
```

These automatic conversions will be used to minimize unnecessary manual manipulations.

In *hrldxy.txt*, all link attributes, except congested travel times and assigned link volumes, are printed in the first line of the link-specific block, leading by the beginning node (ANODE) and the ending node (BNODE). The data format for the first line is described as follows:



```
       1235 12401 140T 240 240 1504099  4420  31001   0   0T
```

The distance (DIST), TIME1, and TIME2 are displayed in hundredth of mile and minute. To obtain congested link travel times in the end of assignment iterations, the link travel times must be extracted from *hrldxy.txt* for each internal assignment iteration and then weight-averaged by the "FRACTION" associated with each iteration in the "VEHICLE COST-OF-TRAVEL SUMMATIONS" Section of the *hassign.out* file. The congested travel times associated with each assignment iteration are displayed beginning at the second line. Since the default maximum number of iterations allowed in the current version of FSUTMS assignment is 50, no more than ten iterations of travel times can be displayed in each line, and no more than five lines can be used to display congested travel times. The assigned link volumes are displayed in the next one to five lines, depending on number of iterations executed. Finally, the congested speed can be calculated as the distance divided by the congested travel time.

As a default in FSUTMS, the equilibrium assignment consists of a series of all-or-nothing traffic assignments with an adjustment of travel impedance associated with each iteration. During each iteration, FSUTMS calculates the optimal weight, $\lambda^{(j)}$, based on the User-Equilibrium (UE) criterion, and applies the following smoothing scheme on the current link loading (v_k) and the average link loading ($\bar{v}_k$) at the j^{th} iteration to determine the average loading at the $(j+1)^{th}$ iteration on which the calculation of the new link impedances are based.

$$\bar{v}_k^{(j+1)} = (1-\lambda^{(j)})\bar{v}_k^{(j)} + \lambda^{(j)} v_k^{(j)} \tag{3.2}$$

Note that $\lambda^{(0)} = 1$, so that $\bar{v}_k^{(1)}$ is equal to the initial link loading, $v_k^{(0)}$. The "fraction" associated with each iteration can be obtained by deduction as follows.

$$\bar{v}_k^{(j+1)} = \prod_{i=1}^{j}\left(1-\lambda^{(i)}\right)v_k^{(0)} + \lambda^{(1)}\prod_{i=2}^{j}\left(1-\lambda^{(i)}\right)v_k^{(1)} + \lambda^{(2)}\prod_{i=3}^{j}\left(1-\lambda^{(i)}\right)v_k^{(2)} + \Lambda + \lambda^{(k)}v_k^{(k)} \tag{3.3}$$

The time impedance is determined using the travel time function (default as the BPR function) in conjunction with the dampened effect, i.e.,

$$\begin{aligned} t_k &= t_{0k} + DF\left[t_{0k}\left(1 + \alpha\left(\frac{v_k}{UF \cdot c_k} \right)^{\beta} \right) - t_{0k} \right] \\ &= t_{0k}\left(1 + DF \cdot \alpha\left(\frac{v_k}{UF \cdot c_k} \right)^{\beta} \right) \end{aligned} \qquad (3.4)$$

where t_{0k} is the free-flow travel time on link k,
DF is the user-specific damping factor ranged between 0 and 1, and
UF is also the user-specific discounting factor that converts the regulated capacity to the model (practical) capacity.

As shown, the time impedance is "dampened" by a fraction of the time difference between the traditional BPR function and the free-flow travel time. The resultant formulation is equivalent to deflation of the coefficient α (default as 0.15 in FSUTMS) by the damping factor *DF*. For example, if the damping factor is specified as 0.5, the equivalent α value is 0.075. The FSUTMS User's Manual indicates that the damping effect helps reduce the oscillation of link loadings among iterations. Different settings for the BPR model parameters were found to affect convergence in the assignment procedure. In *hrldxy.txt*, the average congested travel time in the end of assignment is internally calculated and displayed as the last iteration of travel times.

Using the above *hrldxy.txt* example, the procedure below demonstrates how to obtain the congested link travel time and assigned volume. A total of three internal iterations were performed and the outputs in *hrldxy.txt* were printed for the initial, the first iteration, and the second iteration link loadings.

Initial	1st Iteration	2nd Iteration	
240	249	246	← Time impedance
3003	2445	2445	← Assigned volume

As stored in *hassign.out*, $\lambda^{(1)} = 0.312$ and $\lambda^{(2)} = 0.883$. The corresponding fractions associated with the initial link loading and the loadings resulted from the first and second iterations are calculated as 0.08057, 0.03662, and 0.88281 based on the equation provided. These link loadings are the results from three all-or-nothing assignment iterations of the Frank-Wolf algorithm. Therefore, the final assigned volume (not shown in *hrldxy.txt*) can be determined as the linear combination of the assigned volumes and the fractions, i.e.,

$$0.08057 \cdot 3003 + 0.03662 \cdot 2445 + 0.88281 \cdot 2445 = 2490$$

Note that the average assigned volume can also be calculated using λ through the recursive equation, i.e.,

$$(1-\lambda^{(1)})\ 3003 + \lambda^{(1)}\ 2445 = 2828.9$$

$(1-\lambda^{(2)})\ 2828.9 + \lambda^{(2)}\ 2445 = 2489.9 \approx 2490$

The average time impedance in the end of assignment iterations is hence equal to (*DF* = 0.5):

$$t_k = t_{0k}\left(1 + DF\cdot\alpha\left(\frac{v_k}{UF\cdot c_k}\right)^{\beta}\right)$$

$$= 240\left(1 + 0.5\cdot 0.15\left(\frac{2490}{0.75\cdot 4420}\right)^{4}\right) = 245.73 \approx 246$$

which matches the time impedance in *hrldxy.txt*.

An alternative source to obtaining the abovementioned information is through the text file named *hrldxy.asc*, which is automatically produced if the user specifies ‘Yes’ for the &ASCII field of the *profile.mas* file. The output looks like this:

```
1235  1240 40 50  1  1  140  240  246   2490   4420
```

3.4. Effects of Feedback on Trip Length Distribution

During the feedback loops, the link travel times change due to the redistribution of link assigned volumes on the network. Consequently, the congested skim times used by the gravity model as a basis for trip distribution among OD pairs will also be changed and affect the OD flow pattern. To maintain consistency between the modeled and observed trip pattern, the OD flow tables need to be calibrated in the trip distribution step if the observed OD flow data are available. However, the OD flow data are difficult to observe and collect. The traditional household travel survey could be utilized to provide some data and establish a broad picture regarding travel patterns, but its applicability is usually restricted to modeling due to a necessarily limited sample size. As an alternative, the trip length distribution data is often used for calibrating the gravity model.

In the calibration process, the deterrence (friction) function parameters are adjusted to reproduce the observed base-year data, such as trip length distribution, OD flows, or traffic counts. Depending on the type of data available, the objective and methodology of the calibration process are distinct but interconnected. First, if the OD flows were directly available, the friction model parameters could have been estimated based on the maximum-likelihood approach by assuming a multinomial distribution for the sampled OD flow frequencies (Hyman, 1969; Evans, 1971), or simply be estimated using the least-squares (LS) methods (Kirby, 1974). Second, if direct observations of OD flows were unavailable, a common approach is to estimate OD flows from traffic counts. Even if no data are available on trip-end data or trip length frequency distribution, calibration of gravity model can be performed to match with the observed traffic counts. A review of estimation algorithms for this purpose can be found in van der Zijpp *et al.* (1998).

If the trip length frequency distribution is available, the iterative procedure suggested by the Federal Highway Administration (1977) for gravity model calibration could possibly be used to reproduce the observed trip length distribution. As described in NCHRP Report 365 (1998), the calibration process iteratively adjusts friction factors using the ratio of the observed and calculated frequencies at each trip length increment, and then uses the adjusted friction factors to fine-tune the functional relationship of the impedance function through the LS method. This iterative process is performed until the calculated trip length distribution sufficiently estimates the observed counterpart. Martin and McGuckin (1998) provide typical friction model parameters by trip purpose that are more applicable and transferable to smaller urban areas. For example, they suggest that the values for impedance exponents and exponential power should be negative and fall within (-1.35, -0.02) and (-0.125, -0.1), respectively. In cases where the distribution of OD trips are not consistent at the interchange level with the observed pattern, further adjustment on socioeconomic factors with resulting friction factors will produce closer estimations between the OD patterns and the trip length distributions (Papacostas and Prevedouros, 2001).

An inherent problem in this iterative process is that the adjusted friction factors may become unstable because the tail areas of the calculated frequency distribution are sensitive to small percentages. As a result, the calibrated parameters in the deterrence function using the LS method could be biased, and the shape of the friction-factor curve sometimes becomes ill-behaved. This problem is further explained and illustrated using an example in the next section. To correct this inherent deficiency and to ensure the robustness of the calibration process, a statistical estimation procedure based on the maximum-likelihood (ML) principle was used. The procedure does not require the calibration procedure to be executed iteratively. The only requirement is the assumption of a probability distribution function for characterizing the observed trip length data.

The following sections review the iterative calibration procedure suggested by FHWA and its potential problems, introduce the proposed estimation methodology, present case studies using the proposed estimation method, and compare their results with those from the traditional calibration procedure adopted in TRANPLAN.

3.4.1. Issues in the Traditional Calibration Procedure

Given trip end data and the observed trip length frequency distribution, the gravity model can be calibrated iteratively in the following steps (FHWA, 1977; Martin and McGuckin, 1998):

1. Assume the initial values for the deterrence function parameters and calculate the friction factors at each time impedance increment (usually 1-min) using the following Gamma function.

   $$F_{ij} = a t_{ij}^{b} e^{c t_{ij}}$$

   where t_{ij} and F_{ij} are time impedance and the corresponding friction factor from origin *i* to destination *j*, and *a*, *b* and *c* are the calibration constants. Alternatively, assume a reasonable set of constant friction factors (say, 100) or values taken from other calibrated study areas.

2. Run the gravity model and derive the trip length frequency distribution. Update the friction factors using the following formula:

$$F_t^{n+1} = F_t^n \frac{f_t^{ob}}{f_t^n} \tag{3.5}$$

where F_t^{n+1} and F_t^n are the friction factor at each time impedance increment *t* for the (n+1)th and nth iterations, respectively, and f_t^{ob} and f_t^n are the observed and calculated frequencies, respectively, at the nth iteration.

3. Regress $\ln\left(F_t^{n+1}\right)$ on dependent variables $\ln t$ and *t* after natural-log transformation to estimate the revised Gamma function coefficients using the ordinary least-squares method. Calculate the new friction factors.

4. Repeat the second and third steps until the calculated trip length frequency distribution is sufficiently closed to the observed one.

The above procedure can be performed for each trip purpose to obtain purpose-specific friction factors. This procedure is easy and straightforward to implement but may be subject to the instability problem because the tail areas of the calculated frequency distribution are sensitive to small percentages. The example below illustrates this problem.

For simplicity, the deterrence function used in this example contains only the exponent parameter, *b*, with the initial value set equal to –2. After several iterations, the above procedure did not seem to converge. As shown in Figure 3.5, the values of friction factors and time impedance in the natural-log scale no longer exhibit the trend of a straight line and the regressed exponent parameter becomes positive (*b* = 1.0 in the case depicted). Meanwhile, the tail of the resulting friction factors continues to increase, possibly due to the small percentage of values that exist in the tail area of the calculated distribution. To verify this, another experiment was conducted. In this experiment, instead of applying to the full range of data, linear regression was applied to variables with values taken from time impedances between the first and 60th minute, dropping all the small percentages in the tail area of the calculated distribution. As a result, the procedure converges at *b* = -1.05 and the friction factor diagram appears much more reasonable than that of the previous case (see Figure 3.6). This example illustrates a potential problem that a planner may encounter during model calibration with the traditional least-squares approach. To address this issue, a more robust statistical estimation procedure is developed and described in the following section.

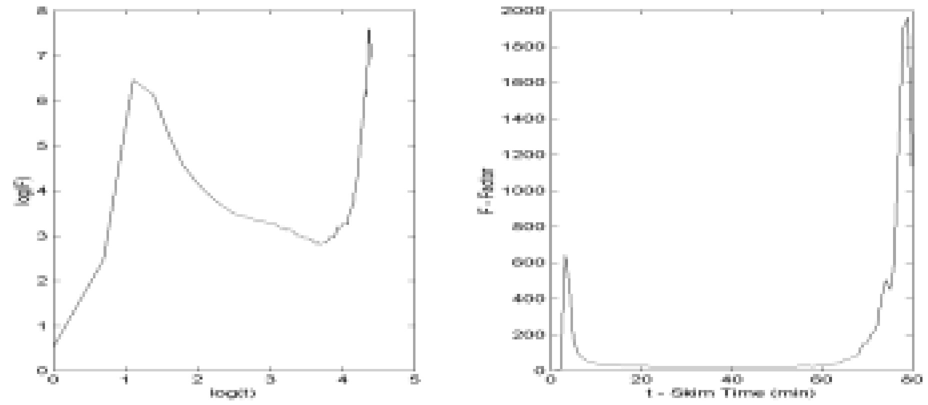


Figure 3.5. Iterative Calibration Procedure Diverges (*b* = 1).

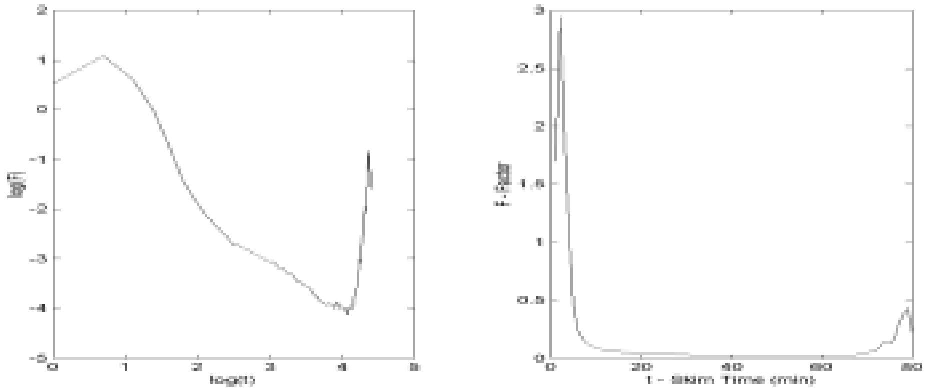


Figure 3.6. Iterative Calibration Procedure Converges at *b* = -1.05.

3.4.2. Proposed Calibration Methodology

As mentioned earlier, the only prerequisite of the proposed estimation method is a representative probability distribution function for sampled trip length data. One probability distribution function commonly accepted for trip lengths is the two-parameter Gamma distribution. Although a more flexible, three-parameter distribution function, such as Pearson-type III, can be used to enhance the estimation results without any loss of generality, the Gamma distribution was selected to demonstrate the feasibility of the proposed estimation method. Let t_{ij} denote the skim time for a given OD pair (*i*,*j*) for a specific trip purpose. Assume t_{ij} follows a Gamma distribution with parameters α and β, i.e.:

$$f_T(T = t_{ij}) = \frac{{t_{ij}}^{\alpha-1} e^{-t_{ij}/\beta}}{\Gamma(\alpha)\beta^{\alpha}} \tag{3.6}$$

Since the mean value $\bar{t} = \alpha\beta$ and variance $\sigma_t^2 = \alpha\beta^2$, the Gamma model parameters can be re-parameterized as:

$$\alpha = \frac{\bar{t}^2}{\sigma_t^2} \text{ and } \beta = \frac{\sigma_t^2}{\bar{t}}. \tag{3.7}$$

Let the estimated skim time for an OD pair be $\tilde{t}_{ij}$, and the estimated OD trips from the gravity model be T_{ij}, the mean and variance of time impedances on network can then be calculated as:

$$\bar{t} = \frac{\sum_i \sum_j T_{ij} \tilde{t}_{ij}}{\sum_i \sum_j T_{ij}} \tag{3.8a}$$

and

$$\sigma_t^2 = \frac{\sum_i \sum_j T_{ij} \left(\tilde{t}_{ij} - \bar{t} \right)^2}{\sum_i \sum_j T_{ij}} \tag{3.8b}$$

Given known trip-end data, the OD flows can be predicted using the following well-known gravity-type formulation, i.e.:

$$T_{ij} = \frac{O_i D_j F_{ij} K_{ij}}{\sum_j A_j K_{ij}} \tag{3.9}$$

where O_i and D_j denote trips originated from zone i and destined into zone j, respectively, and F_{ij} and K_{ij} denote the friction factor and socioeconomic factor, respectively, among an OD pair (i,j). For simplicity, let one assume all ones for the socioeconomic factors, and let F_{ij} carry one of the following general Gamma function, as suggested by FHWA (1977):

$$F_{ij} = a t_{ij}^b e^{c t_{ij}} \tag{3.10}$$

where a, b and c are the calibration constants. Note that the constant a is a scale factor that does not affect the output of the gravity model and will not be treated herein as parameter for calibration purpose. TRANPLAN uses this constant to normalize the reported friction factors to the largest value (999999), in order to fit within the six-byte field in the output file. To calibrate the remaining constants, a maximum-likelihood estimation (MLE) procedure is used. The log-likelihood function can be written as:

$$\ln L(\mathbf{p}) = \sum_i \sum_j \left(\alpha(\mathbf{p}) - 1 \right) \ln t_{ij} - \alpha(\mathbf{p}) \ln \beta(\mathbf{p}) - \frac{t_{ij}}{\beta(\mathbf{p})} - \ln \Gamma\left(\alpha(\mathbf{p}) \right) \tag{3.11}$$

where $\mathbf{p}$ = [b c], ln(.) is the natural logarithm operator, and Γ(.) is the Gamma function. The parameters can be estimated by maximizing the log-likelihood function using an appropriate nonlinear constrained optimization routine, such as *constr* in MATLAB (MathWorks, 1999). Constraints can also be imposed on parameters to produce the desired deterrence function or to avoid undesired functional forms. If constraints are considered, the optimization routine will be required to handle the simple bound constraints or linear constraint equations in this case. The abovementioned MATLAB code is a well-designed nonlinear optimization solver for such purposes.

3.4.2. Feedback Implementation with Self-Calibration

With the calibration procedure for the gravity model embedded in the feedback loops, the feedback implementation is expected to correctly calibrate the deterrence function parameters in the gravity model to produce consistent trip length distribution with the observed counterparts for each trip purpose. As a result, it is also expected to eventually achieve a more reasonable flow distribution on the network that replicates the observed ground counts. When there exists scale difference between real and modeled networks, however, such expectations may not be realized since the modeled trip lengths may not be consistent with the observed counterparts. To resolve this issue, a heuristic scheme called *proportional adjustment by iteration* (PAI) is proposed. The idea is straightforward as follows:

(1) Initialize the trip length distribution based on the Gamma distributional assumption, and the observed means and standard deviations of purpose specific trip length,

(2) Given the current trip length distribution, use GMODEL to calibrate friction factors,

(3) In the end of feedback iteration, calculate the ratio of TOTAL VHT VOLUMES to TOTAL VHT COUNTS, (VHT: Vehicle Hour Traveled)

(4) Divide the means and standard deviations of the resulted trip length distribution by the ratio, and reinitialize the trip length distribution, and

(5) Go back to step (2) and repeat the process along with the feedback process.

The basic assumption of this approach is that every individual trip length is proportionally adjusted upward or downward according to the overall assigned volume-to-counts VHT ratio. This working assumption can be validated when there exists scale difference between real and modeled networks.

CHAPTER 4
CASE STUDIES FOR EQUILIBRIM FEEDBACK IMPLEMENTATION

4.1. Introduction

This chapter presents results from the implementation of feedback in FSUTMS for three networks of different sizes, i.e., small, medium, and large. The small network is based on the standard model, while the medium and large models are based on a non-standard model. Historical ground counts are used to evaluate the effectiveness of the feedback process. The fictitious "FTOWN" sample network that came with FSUTMS was used as a case study involving the "small" study area. Despite its small size, FTOWN is relatively complete, making it especially suitable for procedure design, testing, and fine-tuning. After the feedback process based on the FTWON network was successfully implemented, it was extended to the other two test cases, which are based on the real-world networks of Broward and Palm Beach counties. The following sections discuss the results and findings from implementing the feedback process in FSUTMS in detail for each of the three models, including the calibration of trip length distribution.

4.2. FTOWN (Small) Urban Area Test Case

The FTOWN network consists of 15 internal zones, nine external zones, seven trip purposes, eight roadway facility types, and five land uses. The example has its own control files (*trnplxxx.ins*) written based on the TRANPLAN general control files (with extensions *.all*, *.hwy*, and *.tr*). There are a total of 18 control files created within the FTOWN model. The following sections present the implementation feedback and results using different iterative methods.

4.2.1. Test Scenarios

To facilitate the investigation of the convergence property of feedback, three scenarios corresponding to different congestion levels were created by inflating the population size from original conditions specified in the *zdata1.axx* files for the GEN module, including:

1. Base scenario – Contains the original population
2. Medium scenario – Inflates the population size by a factor of 10
3. Congested scenario – Inflates the population size by a factor of 20

By inflating the original population by a factor of ten and twenty, the corresponding volume-to-capacity (v/c) ratios in the end of traffic assignment were roughly doubled and quadrupled, respectively. Figure 3.5 shows the actual v/c ratios for the three scenarios.

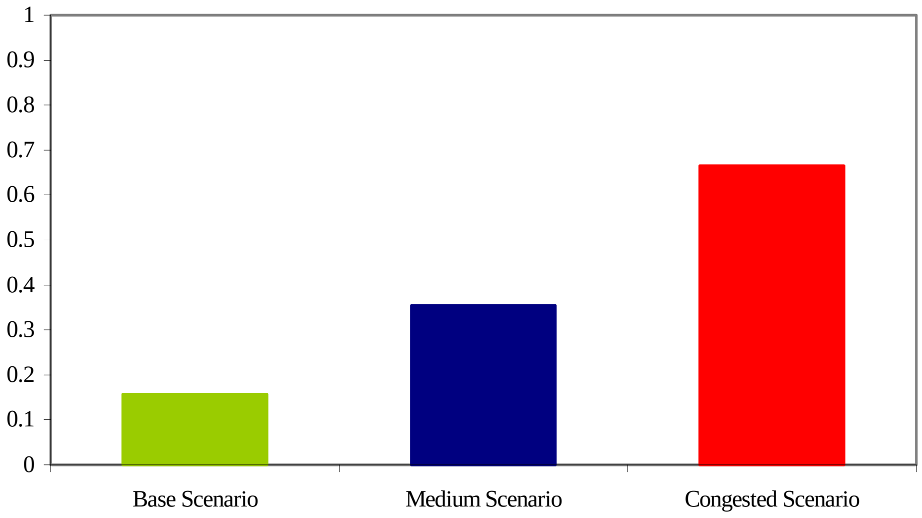


Figure 4.1. Average v/c Ratios for Three Different Scenarios.

4.2.2. Results from Direct Method

As expected, due to the different levels of congestion, FSUTMS reveals fairly distinctive behaviors of convergence. Figures 4.2a-c show the rate of convergence for the base scenario in terms of travel time, speed, and link volume. Figures 4.3a-c and Figures 4.4a-c show the same information for the medium and congested scenarios. As these figures illustrate, the base and medium scenarios did not reach convergence. In the base scenario, the impedances and link volume oscillate between two states after the second iteration and did not converge. The medium scenario took four iterations to reach the oscillation condition. These observations confirm the fact that the direct iterative method can oscillate and the resulting link volumes do not converge. In this case, the convergence failed under light to medium demand conditions. On the other hand, the process converges under the congested scenario. The process converges at a faster rate after the 11th iteration and finally reaches the user-specified convergence criterion (10^{-3}) without any indication of oscillation. Figures 4.5a-c show the final rates of convergence.

4.2.3. Results from MSA Method

As explained previously, the MSA method differs from the direct method in applying the weighted average scheme on results obtained from previous iterations before feeding back to the next iteration. Due to the effect of weighting average, the convergence was significantly accelerated in the FTOWN model in all scenarios. The MSA method took only two iterations to converge for the base scenario (see Figure 4.6). As shown in Figures 4.7 and 4.8, the advantage of MSA becomes much more pronounced in the medium and congested scenarios. All the MSA runs took ten iterations or less to achieve convergence, while the direct method failed to converge to a given convergence criteria (set to 10^{-3} in all measures). This demonstrates that the MSA method facilitates the convergence behavior when the network is at a higher level of congestion.

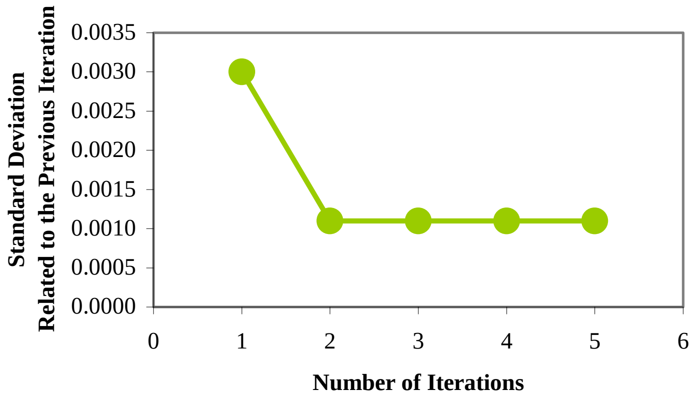


Figure 4.2a. Rate of Convergence for Travel Time (Base Scenario).

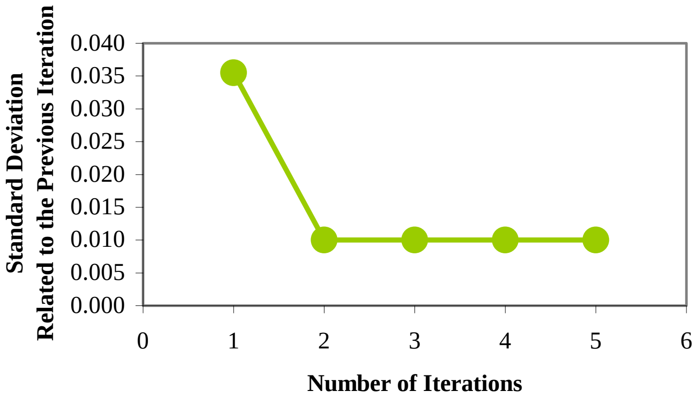


Figure 4.2b. Rate of Convergence for Speed (Base Scenario).

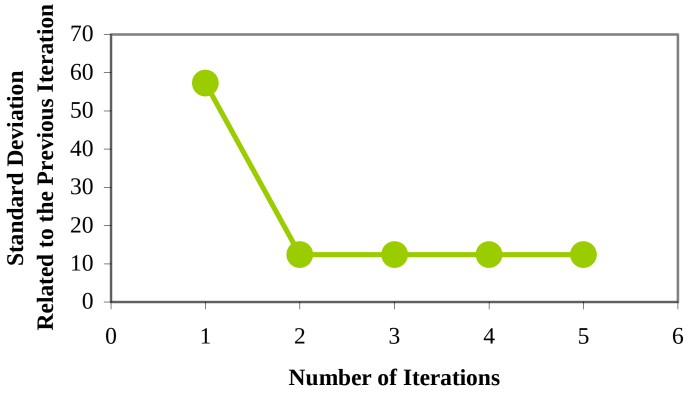


Figure 4.2c. Rate of Convergence for Volume (Base Scenario).

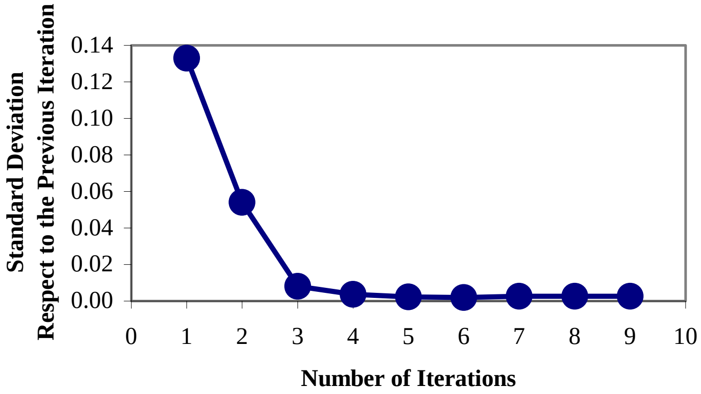


Figure 4.3a. Rate of Convergence for Travel Time (Medium Scenario).

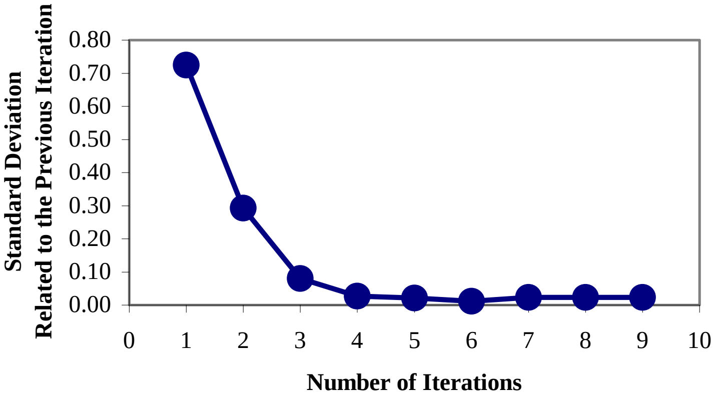


Figure 4.3b. Rate of Convergence for Speed (Medium Scenario).

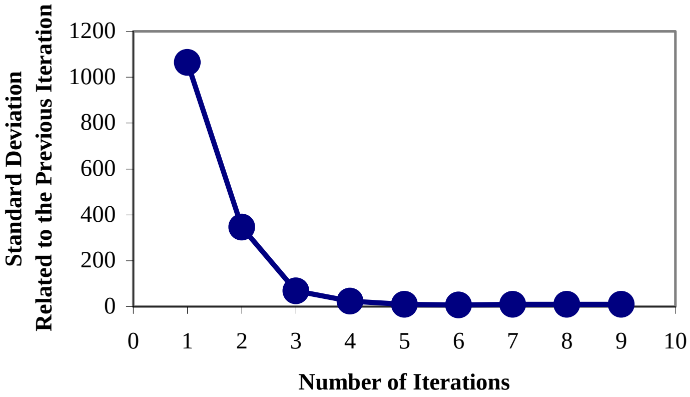


Figure 4.3c. Rate of Convergence for Volume (Medium Scenario).

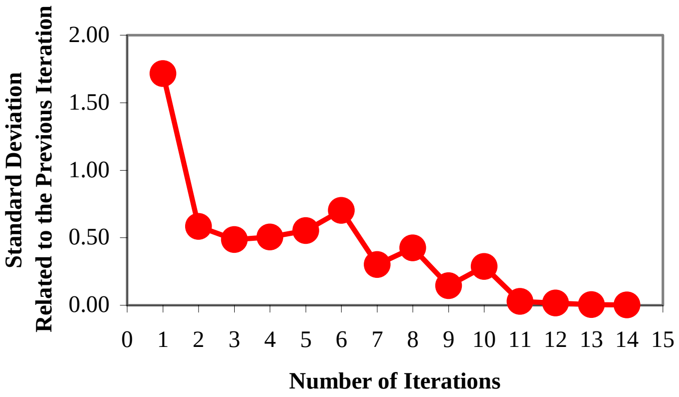


Figure 4.4a. Rate of Convergence for Travel Time (Congested Scenario).

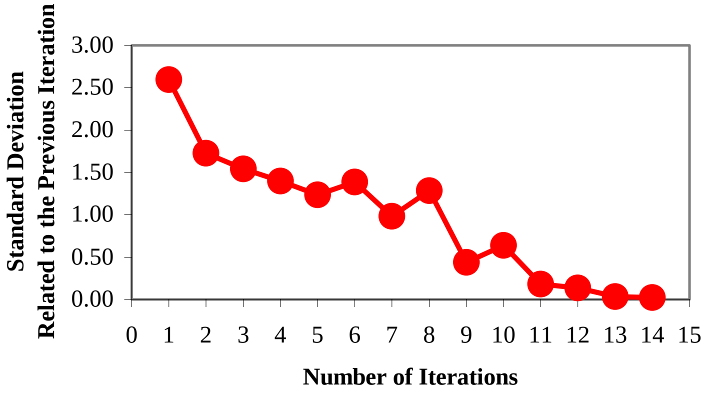


Figure 4.4b. Rate of Convergence for Speed (Congested Scenario).

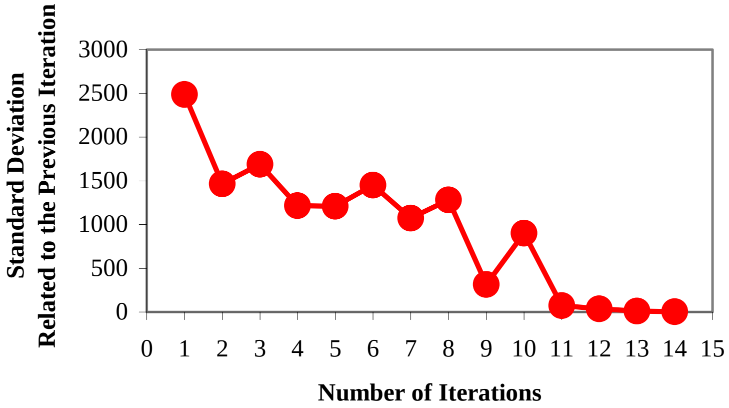


Figure 4.4c. Rate of Convergence for Volume (Congested Scenario).

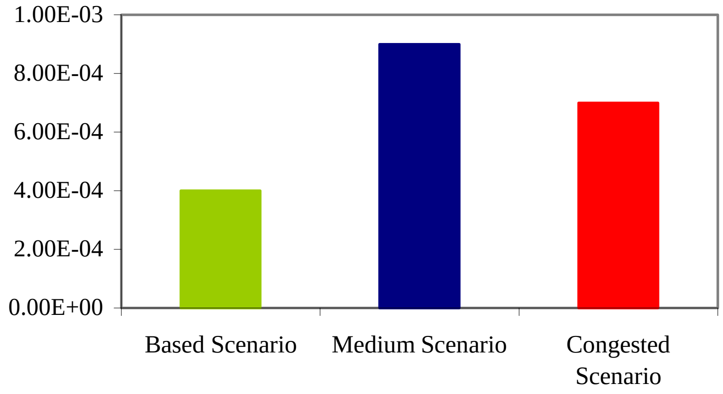


Figure 4.5a. Final Convergence Rate of Travel Time.

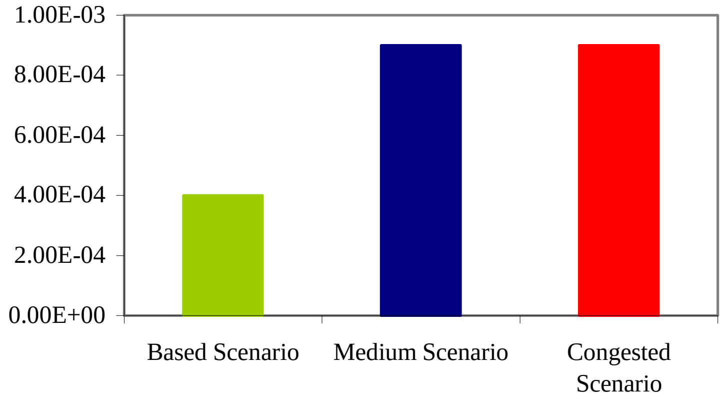


Figure 4.5b. Final Convergence Rate of Speed.

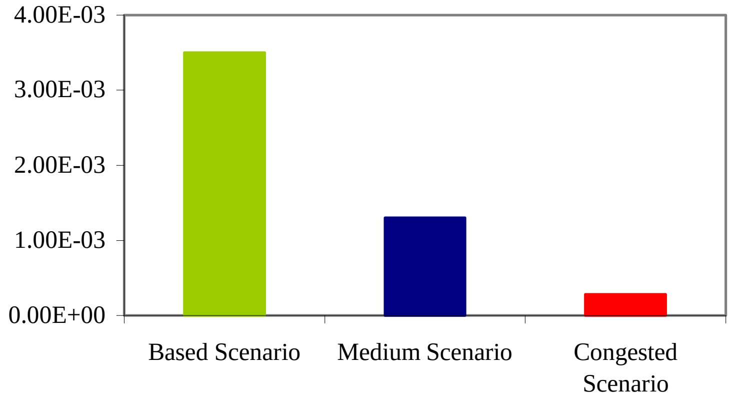


Figure 4.5c. Final Convergence Rate of Link Volumes.

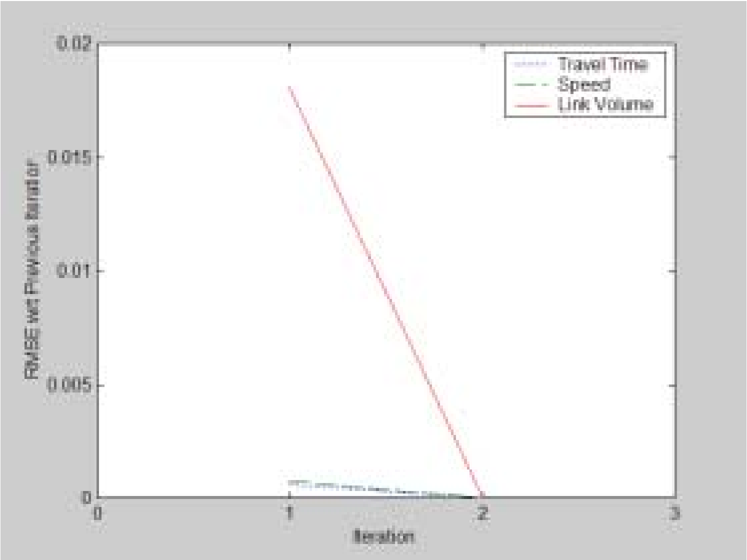


Figure 4.6. Rate of Convergence for All Measures (Base Scenario).

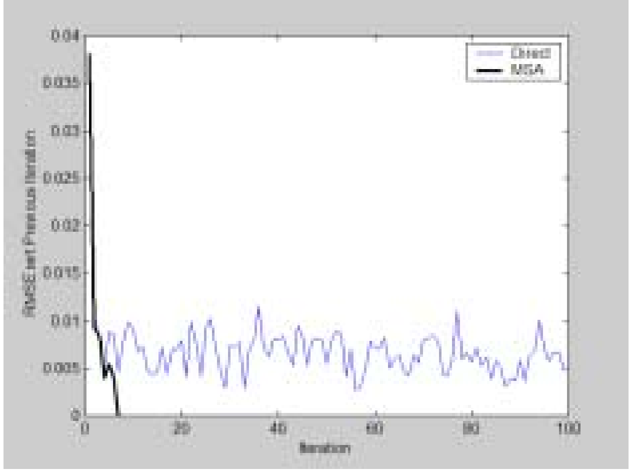


Figure 4.7a. Rate of Convergence for Travel Times (Medium Scenario).

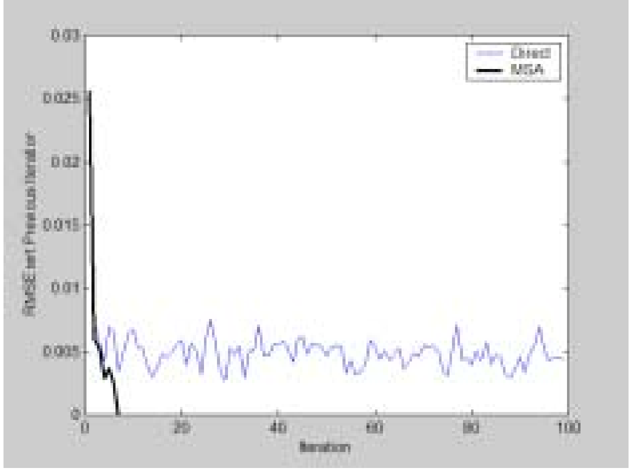


Figure 4.7b. Rate of Convergence for Speeds (Medium Scenario).

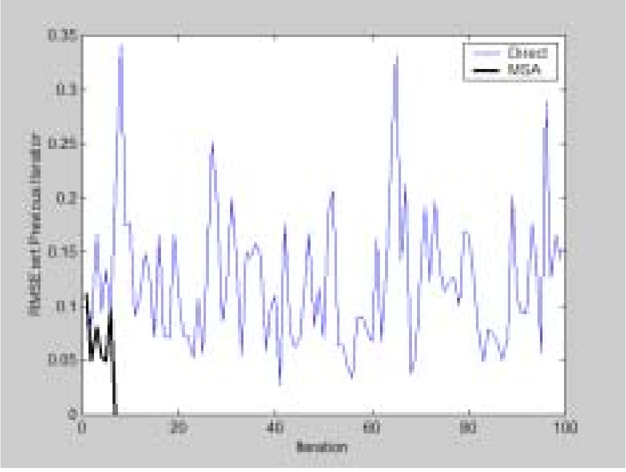


Figure 4.7c. Rate of Convergence for Link Volumes (Medium Scenario).

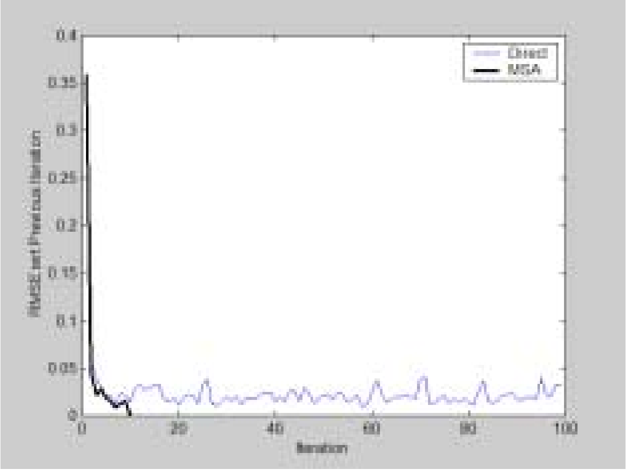


Figure 4.8b. Rate of Convergence for Travel Times (Congested Scenario)

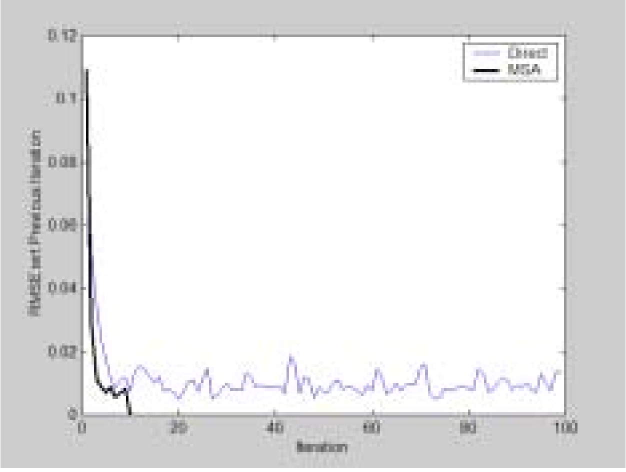


Figure 4.8b. Rate of Convergence for Speeds (Congested Scenario)

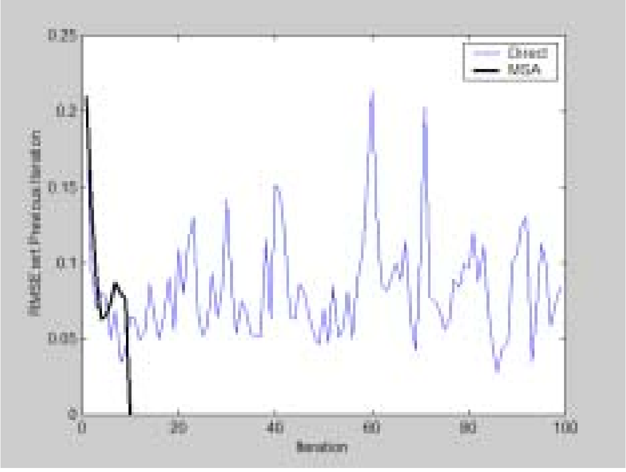


Figure 4.8c. Rate of Convergence for Link Volumes (Congested Scenario)

4.3. Broward (Medium) Urban Area Test Case

4.3.1. Test Scenarios

To further investigate the potential advantages of the feedback implementation, actual planning models with ground count information from Broward and Palm Beach counties were evaluated. The Broward model contains 933 zones and was considered a medium-to-large size model. The Palm Beach model contains 1172 zones and was considered a relatively large-size network. The post-processing batch file, attached in Appendix B, was adapted from the Palm Beach case study file without modifying its existing logic, model parameters or control parameters for running the modules.

4.3.2. Comparison Between Direct and MSA Methods

As shown in Figures 4.9a-c, the results of the convergence behaviors for the Direct and MSA methods were different for the Broward model across various measures. In terms of travel time, the MSA method appears to stabilize compared to the direct method but did not converge within 50 iterations. Both methods failed to converge based on the speed criterion. In terms of link volumes, however, the MSA method stabilized within only a few iterations and stayed at the same level of RMSE ≈ 2×10^{-3}, which was very close to the convergence threshold. The direct method, on the other hand, shows a higher degree of fluctuation and, in all cases, there was no sign of convergence.

Overall, it appears to be difficult to achieve convergence for both methods at the preset convergence criteria (10^{-3}). It is arguable, however, that one could leverage the convergence standard and claim convergence based on a relaxed criterion. From this standpoint, it might make more sense to look at the overall trend of the convergence behavior instead of the preset convergence standard.

4.3.3. Comparison with Ground Counts

One of the important modeling tasks is to validate the model using the ground counts collected in the field. Validation of the planning model is essential for the modeler to project the future demand from the consistent baseline with a degree of confidence. According to practice, several critical measures include count RMSE at screenlines, total VMT, total VHT, congested speed, v/c, etc. These statistics were compiled and listed in Tables 4.1 through 4.4. Two different feedback schemes were performed and compared, namely, the MSA and the pre-assignment procedures that were previously adopted by several models. As shown in Table 4.2, the overall RMSE percentage (RMSEP) from the MSA feedback scheme is slightly lower than the pre-assignment scheme and without feedback. In terms of RMSE by the count level, the MSA feedback scheme consistently performs better than the other two schemes.

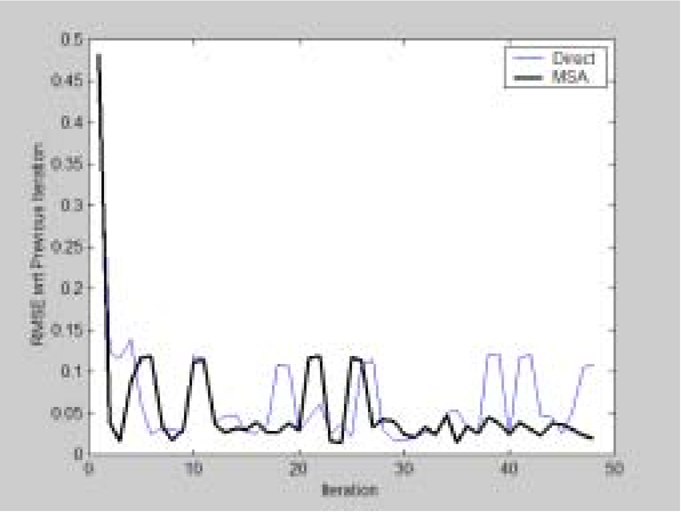


Figure 4.9a. Rate of Convergence for Impedance.

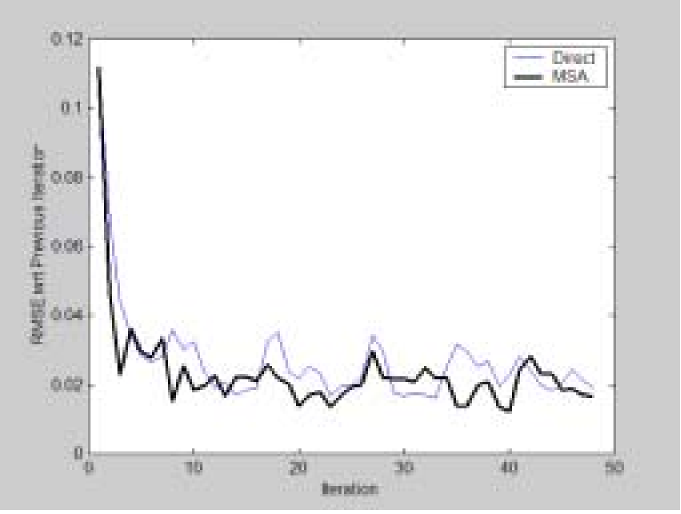


Figure 4.9b. Rate of Convergence for Speed.

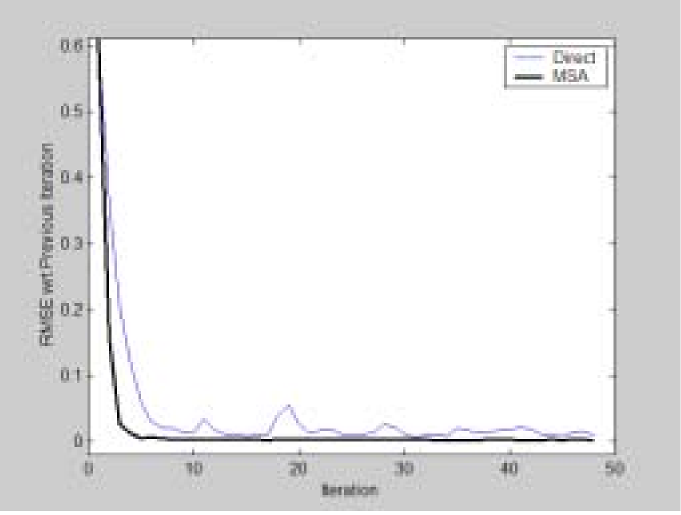


Figure 4.9c. Rate of Convergence for Link Assigned Volume.

Table. 4.1. Volume-to-Count Ratios by Screenlines.

Screenline No.	Without Feedback	Pre-assignment	With Feedback
1	1.02	0.98	0.89
2	1.02	0.96	0.88
3	1.18	1.1	1.02
4	0.95	0.91	0.86
5	1.13	1.09	1.02
6	1	0.98	0.95
7	1	1	1.01
8	1	1	1.01
9	1	1	1.01
12	1.24	1.18	1.08
13	0.86	0.83	0.77
14	1.05	1.01	0.93
59	1	0.99	0.93
75	0.99	0.97	0.94
82	1.09	1.09	1.02
86	0.99	0.82	0.57
91	1.13	1.06	0.91
95	0.99	0.95	0.91
99	1.07	1.04	0.98

Table 4.2. Count RMSEP by Sreenlines.

Screenline No.	Without Feedback	Pre-assignment	With Feedback
1	16.13	15.63	16.79
2	14.37	12.16	12.43
3	23.88	24.01	23.00
4	31.02	27.51	27.20
5	25.56	25.03	26.49
6	19.92	16.42	16.07
7	0.18	0.18	0.45
8	0.26	0.17	0.47
9	0.77	0.94	0.82
12	18.23	17.75	14.11
13	23.42	25.25	25.42
14	19.72	17.25	17.54
59	16.01	16.81	17.14
75	8.98	9.26	10.21
82	23.72	12.72	2.34
86	19.95	17.62	18.21
91	18.56	15.52	16.56
95	14.76	16.14	16.37
99	36.35	35.82	34.96
Overall	**29.78**	**29.53**	**29.2**

Table 4.3. Count RMSEP by Volume Level.

Volume Range ($\times 10^3$)	Without Feedback	Pre-assignment	With Feedback
0 – 5	246.23	202.42	170.96
5 – 10	65.67	59.8	60.84
10 – 20	43.35	42.6	41.55
20 – 30	39.05	38.37	38.15
30 – 40	29.34	28.89	28.04
40 – 50	20.1	19.95	19.39
50 – 60	19.92	19.94	19.1
60 – 70	12.69	13.57	12.07
70 – 80	7.79	8.59	7.53
80 – 90	16.1	15.34	14.97
90 – 100	8.04	7.6	8.01
100 – 400	10.47	10.9	10.52

Table 4.4. Other Measures.

	Without Feedback	Pre-assignment	With Feedback
Congested Speed (mph)	30.64	31.11	31.93
Total VMT (millions)	34.988	33.642	31.572
Total VHT (millions)	1.098	1.03	0.928
v/c	1.05	1.01	0.94

4.4. Palm Beach (Large) Urban Area Test Case

4.4.1. Test Scenarios

The basic modeling structure for the Palm Beach model is quite similar to the Broward model. Therefore, the post-processing batch file for the Broward model can be applied to the Palm Beach model with only minor modifications. All the model and control parameters for executing the various FSUTMS modules remain the same.

4.4.2. Comparison Between Direct and MSA Methods

In this test, the MSA method with and without trip length calibration were performed and compared with the direct method to examine their convergence behaviors. As Figure 4.10 illustrates, the MSA method with trip length calibration, in general, yields the best performance, suggesting that calibrating the trip length distribution somewhat improves the overall convergence process, which is mainly due to the fact that, during the iterations, trip length distribution from the gravity model becomes increasingly consistent with the trip length distribution calculated from the modeled OD trip tables. Convergence based on the direct method, however, behaves poorly.

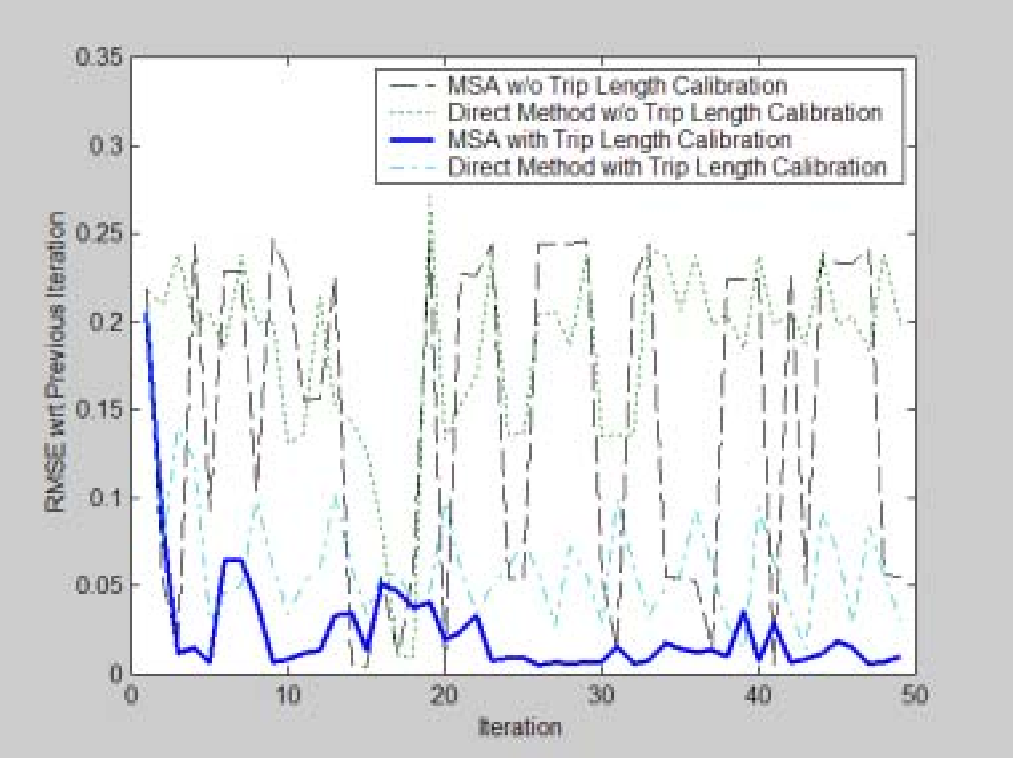


Figure 4.10a. Rate of Convergence for Impedance.

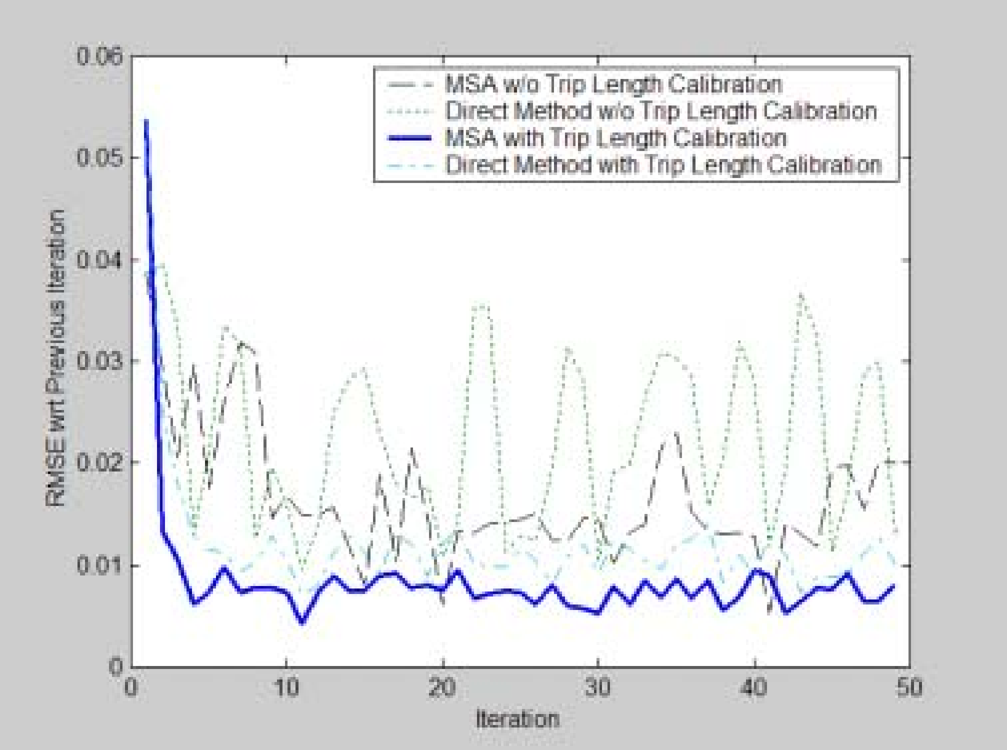


Figure 4.10b. Rate of Convergence for Speeds.

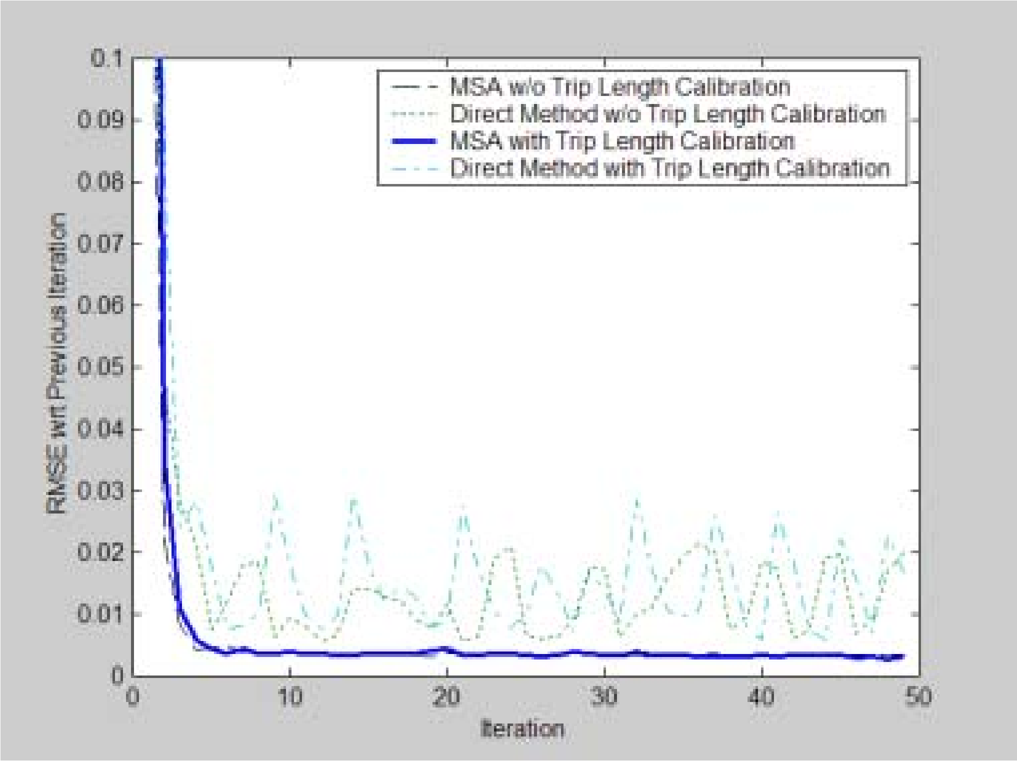


Figure 4.10c. Rate of Convergence for Link Volumes.

4.4.3. Comparison with Ground Counts

For comparison purpose, five schemes, differentiated by with and without feedback, on-line trip length calibration (TLC), and total VHT (Vehicle Hour Traveled) control through PAI (proportional adjustment by iterations), were implemented. The properties of each scheme in terms of friction factor (F-factor) used, total VHT control, and CTOLL (factor used to convert toll fee to equivalent time) are listed below:

	F-factor	VHT Control	CTOLL
1. Without feedback	Predetermined (from Pre-assignment)	No	0.085
2. Pre-assignment	Predetermined	Yes	0.085
3. Feedback w/o on-line TLC	Predetermined (from Pre-assignment)	No	0.085
4. Feedback w/ on-line TLC	Adjusted on-line	No	0.085
5. Feedback w/ on-line TLC and PAI	Adjusted on-line	Yes	0.050

Note that F factors used in scheme (4) and (5) were adjusted on-line in the trip length calibration step using the observed trip length statistics as starting values, while in schemes (1)-(3) they were predetermined using the Automatic Model Calibration Process (AMCP) developed by Carr

Smith Corradino (2002). CTOLL is the only model parameter calibrated in scheme (5) in addition to F factors. The total VHT was controlled in scheme (5) through PAI, while in scheme (2) it was controlled using AMCP.

Tables 4.5 through 4.7 show the volume-to-count ratio and RMSEP information by screenlines and volume levels, as well as the overall performance measures, from different implementation schemes. Tables 4.8 and 4.9 summarize similar information by facility and area types. Table 4.10 shows the first two statistical moments of the trip lengths resulted from pre-assignment scheme, feedback with trip length calibration without PAI and the observed counterpart. In term of the overall RMSEP, the model based on feedback without trip length calibration slightly outperforms the pre-assignment and without-feedback schemes, indicating that the overall model accuracy could be improved through feedback process but only marginally. Feedback with trip length calibration, on the other hand, deviates from the best estimates by 6%. In terms of volume-to-count ratio, the pre-assignment scheme yields an almost perfect ratio (≈ 1.0). Feedback without trip length calibration lowers the overall ratio to 0.97, reflecting the effect of more congested skim times. As a result, total auto trips were reduced due to a mode shift from driving alone to traveling in higher-occupancy vehicles. In addition, the total intra-zonal trips increased (see Table 4.7). On the other hand, feedback with trip length calibration increases the overall volume-to-count ratio by 12% due to an increase in trip length. Under this scheme, trips were more dispersed, increasing both the total VMT and the total VHT by 13%. Table 4.8c shows that the assigned volumes resulting from feedback with trip length calibration are systematically higher than the ground counts, especially for undivided arterials (facility type = 3x) and one-way facilities (facility type = 6x).

Each feedback iteration activates the trip length calibration procedure. Given the initial guesses of the friction factors (here the original friction factors used by the Palm Beach model is assumed), the first iteration of feedback directly applies the gravity model for trip distribution, and the trip tables are immediately stored as a backup. At the second iteration, MSA calculates the skim tables and the backup trip tables are used to generate the reported OD trip length distribution. The reported trip length distribution is then combined with the production and attraction tables to calibrate the friction factors, which are in turn used for trip distribution calculations based on the gravity model. The process is iterated along with feedback until the stopping criterion is satisfied. Note that, with proportional adjustment by iterations (PAI), the friction factors are calibrated using the purpose-specific trip length distribution generated by the Gamma distributional assumption with both mean and standard deviation divided by the overall VHT volume-to-count ratio obtained from the previous feedback iteration. See Section 3.4.2 for details. As shown in Figure 4.11a, except for the home-based school trips, the trip length distributions during the feedback iterations closely match that of the modeled trips by the gravity model. Figure 4.11b shows the close matches of trip length distributions generated from the Gravity model and OD under feedback with on-line trip length calibration.

In addition to trip length, another parameter adjustment made to improve the model performance is CTOLL. The value of CTOLL in the original model is defaulted as 0.085, leading to a low assigned volume to count ratio on the Turnpike facility. To enhance the volume level on Turnpike, probably the most effective way is to reduce the value of CTOLL that converts the toll fee to an equivalency of time for a vehicular trip. According to recent surveys reported in various literatures, the value of time spent on driving per person is ranged from 11 to 13 dollars per hour, and the average vehicle occupancy is between 1.1 to 1.5 persons per vehicle. Therefore, the value of time per vehicle is between 12.1 to 19.5 dollars, and the range of CTOLL can be approximated as 0.05~0.085. After several attempts, CTOLL = 0.05 is selected to enhance the assigned volume level on Turnpike to the greatest extent.

The resultant friction factors from feedback with trip length calibration and PAI were systematically more well-defined in shape (see Figure 4.12) than the pre-assignment counterparts. The original proposed friction factors for pre-assignment were not smooth, in which truncation was found in the beginning of impedance increments. The means and standard deviations of trip lengths from feedback with trip length calibration (without PAI) were mostly shorter than, but closer to, the observed counterparts. In other words, the OD trips under feedback with trip length calibration were distributed onto the network with longer trip distances. Consequently, the assigned volume (flow) pattern may deviate from the originally proposed model.

As a result of implementing feedback with trip length calibration (without PAI), two major effects were observed as compared to the pre-assignment scheme. First, as a result of more congested network, total automobile trips were reduced due to mode shift from driving alone to higher occupancy vehicles. Second, the trip length distribution becomes elongated (dispersed). Both mean and standard deviation increase, indicating that trips were made farer and dispersed spatially. With these two effects compounded, the total VMT and VHT were increased by 12% and 13%, respectively. This leads to a systematically higher level of assigned volumes over ground counts (ratio ≈ 1.12) with the RMSEP (Root Mean Square Error Percentage) degraded by 6%. The PAI scheme was therefore proposed to resolve this issue. Due to PAI, the overall VMT and VHT volume-to-count ratios were controlled at 1.0 and 1.01, respectively, at convergence. The RMSEP was also substantially improved to a comparable level with pre-assignment and feedback without trip length calibration.

Table. 4.5. Volume-to-Count Ratios by Screenlines.

Screenline No.	Without Feedback	Pre-assignment	Feedback Without Trip Length Calibration	Feedback With Trip Length Calibration (w/o PAI)	Feedback With Trip Length Calibration (w/ PAI)
1	1.00	1.00	1.00	0.88	0.86
2	1.06	1.04	1.02	**1.23**	1.04
3	**1.21**	1.16	1.07	**1.55**	1.05
4	1.03	1.00	0.98	1.19	1.01
5	1.07	1.05	1.02	**1.29**	1.06
6	1.04	1.04	1.04	**1.27**	1.11
7	1.00	1.00	1.00	0.97	0.95
8	**1.42**	**1.42**	**1.42**	**1.61**	**1.49**
9	**0.77**	**0.75**	**0.73**	0.85	0.81
10	0.97	0.95	0.94	1.18	1.05
11	0.99	0.98	0.91	1.06	0.98
12	1.08	1.04	1.00	1.12	1.07
13	0.92	0.90	0.89	1.02	1.00
14	1.04	0.99	0.96	**1.23**	1.13
15	**1.21**	1.16	1.13	**1.20**	1.19
16	1.00	1.00	1.00	0.90	**0.78**
17	0.85	0.84	0.83	0.94	0.95
18	0.9	0.88	0.88	0.93	0.98
95	1.01	1.00	0.99	1.07	0.95
96	1.00	1.00	1.00	0.99	0.97
99	1.02	0.99	0.96	1.12	1.02
Overall	1.019	0.996	0.973	1.118	1.014

Table 4.6. Count RMSEP (%) by Volume Level.

Volume Range ($\times 10^3$)	Without Feedback	Pre-assignment	Feedback Without Trip Length Calibration	Feedback With Trip Length Calibration (w/o PAI)	Feedback With Trip Length Calibration (w/ PAI)
0 – 5	68.8	66.2	61.8	80.8	66.2
5 – 10	49.7	48.4	48.3	58.5	50.5
10 – 20	30.6	29.7	29.4	38.2	30.7
20 – 30	23.4	23.2	22.9	22.7	23.5
30 – 40	12.1	13.1	14.4	13.1	16.6
40 – 50	10.9	11.0	11.3	23.0	10.3
50 – 60	0.0	0.0	0.0	0.0	0.0
60 – 70	7.7	7.8	7.3	11.5	8.9
70 – 80	7.0	6.5	6.8	13.1	9.8
80 – 90	12.7	12.6	13.1	16.3	15.9
90 – 100	0.0	0.2	0.5	19.1	23.1
100 – 400	0.0	0.0	0.0	0.0	0.0
Overall	**31.18**	**30.49**	**30.27**	**36.26**	**32.05**

Table 4.7. Other Measures.

	Without Feedback	Pre-assignment	Feedback Without Trip Length Calibration	Feedback With Trip Length Calibration (w/ PAI)
Congested Speed (mph)	40.04	40.35	40.67	40.34
Overall Volume-Count Ratio	1.019	0.996	0.973	1.007
Total Automobile Trips*	-	3,427,221	-	3,422,922
Total Intra-zonal Trip*	-	141,685	-	135,389
Intra-zonal Trip Proportion		4.13%		3.96%
Total VMT*	-	26,536,204	-	26,405,032
Total VHT*	-	751,527	-	748,379

Note: * From HASSIGN.OUT

Table 4.8a. Volume-to-Count Ratio from Pre-assignment.

Area Type Facility Type	1x	2x	3x	4x	5x
1x			0.984	0.963	1.042
2x	1.004	0.968	0.972	1.032	0.961
3x	0.968	0.93	0.959	1.027	1.016
4x	**0.844**	1.041	1.082	0.987	
5x					
6x	1.024	**1.135**	0.976		
7x					
8x			0.967		
9x					0.949

Table 4.8b. Volume-to-Count Ratio from Feedback without Trip Length Calibration.

Area Type Facility Type	1x	2x	3x	4x	5x
1x			0.985	0.942	1.04
2x	0.993	0.950	0.945	1.000	0.962
3x	0.928	0.905	0.933	0.985	0.968
4x	**0.862**	**1.134**	1.066	0.926	
5x					
6x	0.945	1.022	0.961		
7x					
8x			0.973		
9x					0.903

Table 4.8c. Volume-to-Count Ratio from Feedback with Trip Length Calibration (with PAI).

Area Type Facility Type	1x	2x	3x	4x	5x
1x			0.927	0.926	1.002
2x	0.991	1.025	1.015	1.076	0.857
3x	0.984	0.997	0.999	1.052	1.00
4x	0.962	**1.233**	1.126	0.968	
5x					
6x	1.04	**1.158**	0.997		
7x					
8x			0.914		
9x					**0.805**

Table 4.9a. RMSEP (%) from Pre-assignment.

Area Type Facility Type	1x	2x	3x	4x	5x
1x			1.42	0.31	0.52
2x	6.79	4.49	**16.11**	**10.38**	2.47
3x	3.94	1.50	**13.07**	4.21	3.30
4x	4.81	3.52	**21.77**	7.95	
5x					
6x	1.64	0.95	0.06		
7x					
8x			0.95		
9x					1.65

Table 4.9b. RMSEP (%) from Feedback without Trip Length Calibration.

Area Type Facility Type	1x	2x	3x	4x	5x
1x			1.43	0.43	0.56
2x	7.72	4.49	**15.47**	**10.26**	2.44
3x	3.79	1.89	**12.78**	3.44	3.39
4x	5.01	3.61	**21.69**	8.56	
5x					
6x	1.84	0.38	0.1		
7x					
8x			0.89		
9x					1.99

Table 4.9c. RMSEP (%) from Feedback with Trip Length Calibration (with PAI).

Area Type Facility Type	1x	2x	3x	4x	5x
1x			2.11	0.53	0.40
2x	7.70	4.66	**16.01**	**10.48**	2.53
3x	4.27	2.06	**13.63**	4.45	3.14
4x	6.04	4.47	**22.09**	8.10	
5x					
6x	1.63	1.07	0.01		
7x					
8x			1.23		
9x					2.72

Table 4.10. Trip Length Statistics from Pre-assignment Model and Feedback with Trip Length Calibration (Palm Beach County).

Trip Purpose	Pre-assignment		Feedback with Trip Length Calibration (w/o PAI)		Feedback with Trip Length Calibration (w/ PAI)		Observed	
	Mean[1]	Std. [2]	Mean	Std.	Mean	Std.	Mean	Std.
HBW	17.7	10.3	20.6	11.6	17.1	9.3	24.1	16.0
HBSHP	12.4	7.5	16.4	10.1	14.5	7.5	15.5	12.1
HBSR	12.2	8.4	15.2	10.0	13.9	6.7	16.5	12.3
HBSCH	15.8	10.9	17.1	13.5	15.0	8.2	19.6	14.3
HBO	13.8	8.3	17.6	10.7	15.2	8.0	16.8	12.7
NHBW	11.9	7.0	13.3	8.2	14.7	6.7	17.5	14.3
NHBO	12.5	7.6	13.9	8.4	14.6	6.5	16.1	13.7
ARPRT	25.1	12.0	27.8	14.1	28.2	14.2	-	-
TRK-4TIRE	18.7	9.8	20.5	11.6	16.9	9.0	-	-
TRK-SU	18.2	9.3	20.2	11.4	16.8	8.2	-	-
TRK-COMB	30.0	14.8	33.4	18.0	27.4	16.0	-	-

Note: 1. Unit: min. 2. Standard deviation

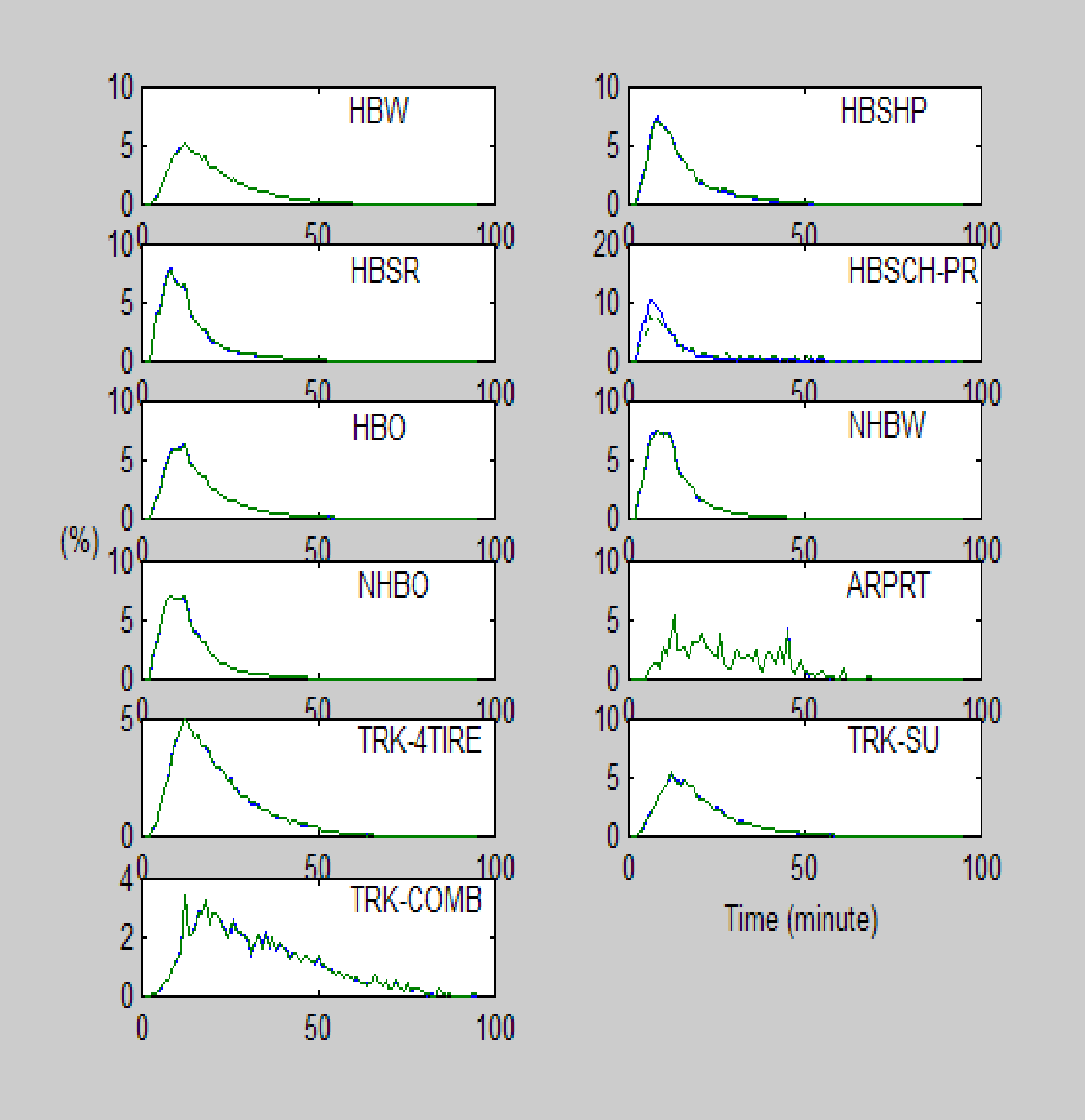


Figure 4.11a. Trip Length Distributions Before (Solid Line) and After (Dash Line) Distribution Step Resulted (Feedback with On-Line Trip Length Calibration).

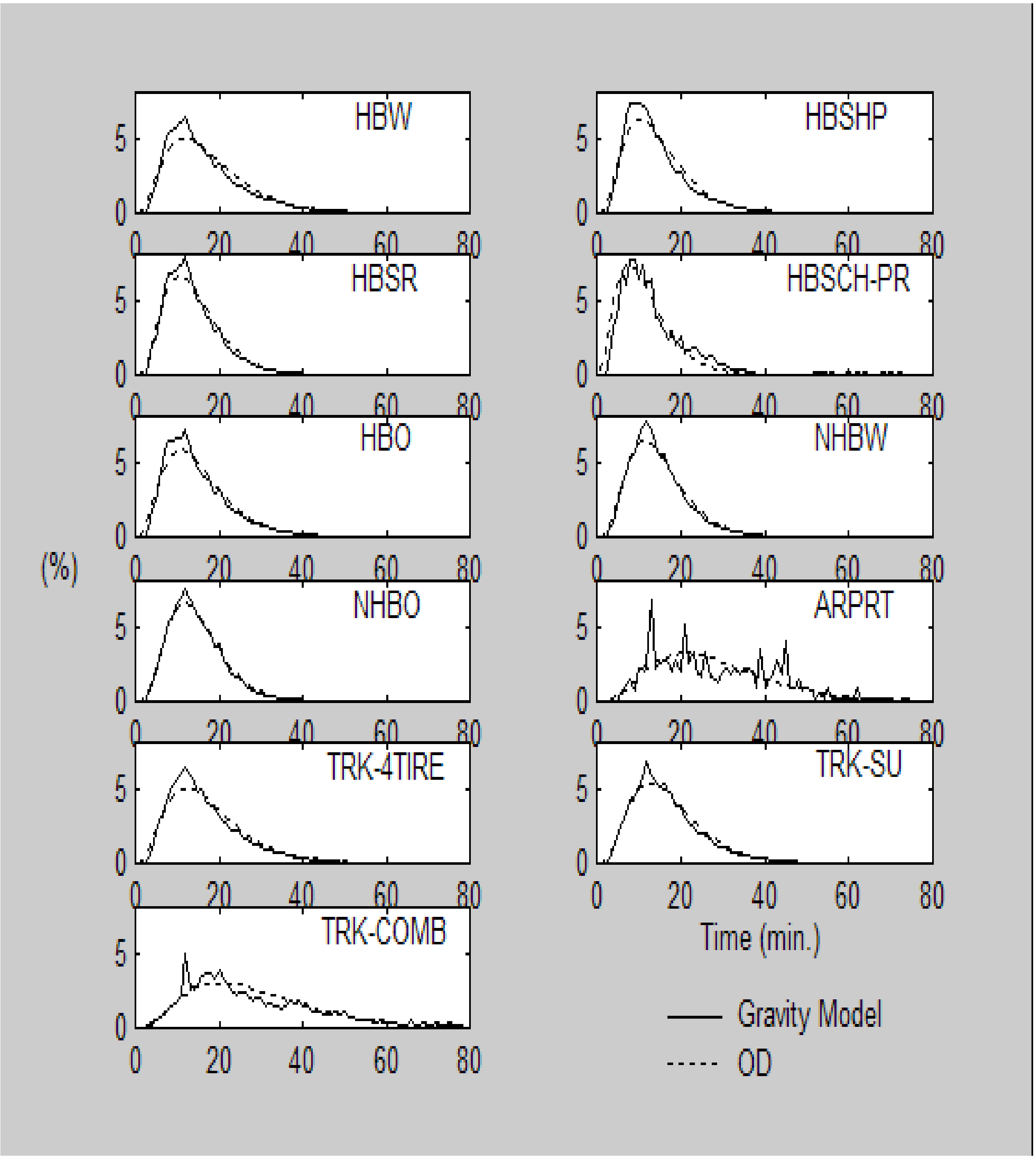


Figure 4.11b. Comparison of Trip Length Distributions Generated from Gravity Model and OD Under Feedback with On-Line Trip Length Calibration.

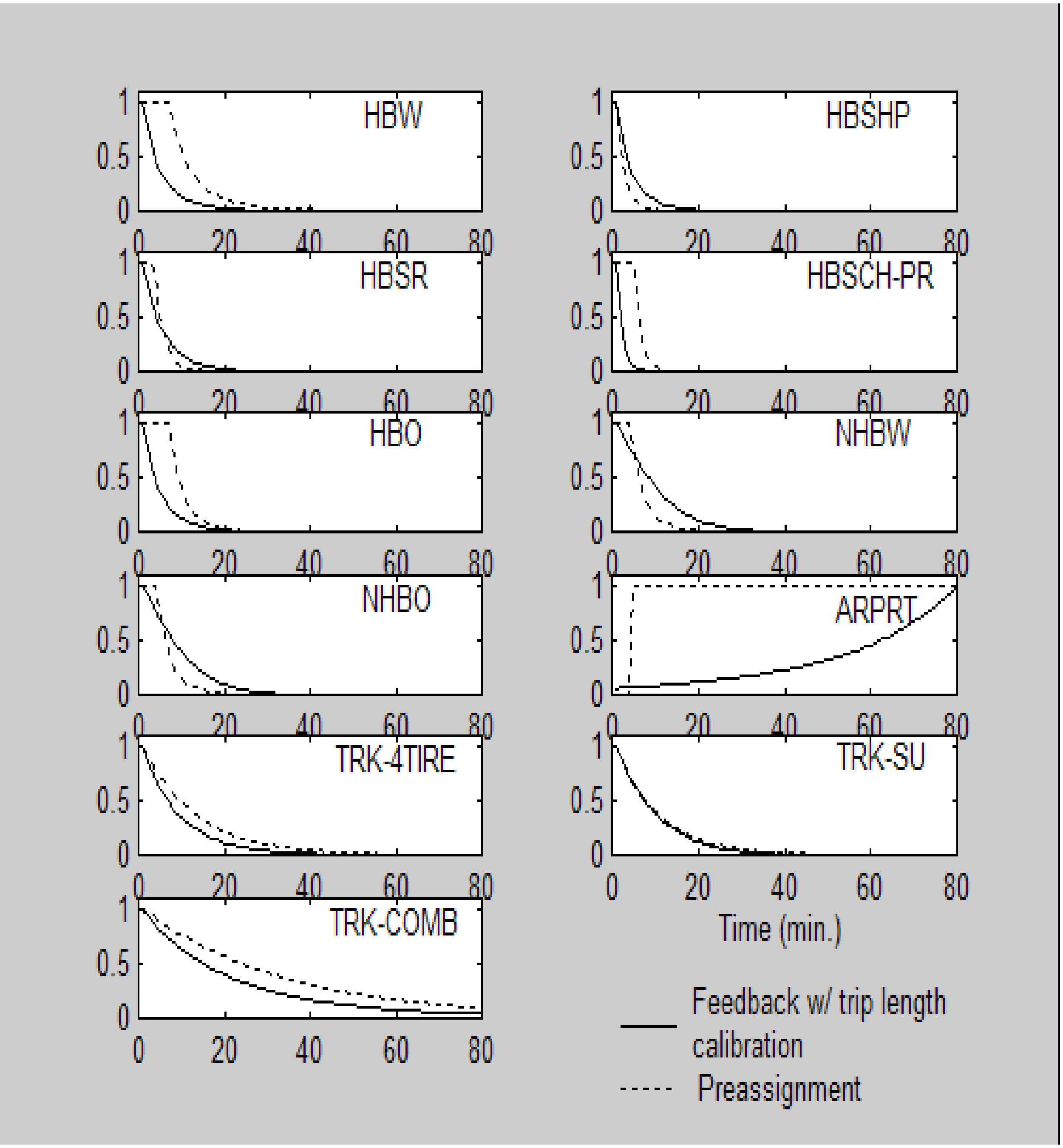


Figure 4.12. Comparisons of Friction Factors from Pre-Assignment and Feedback with Trip Length Calibration.

4.5. Effects of Feedback

The test results from three scenarios of the FTOWN model were used to assess the effects of feedback on the impedance measures and the final trip loading. For evaluation, instead of using the network-wide average measures as in the COMSIS study (1996), averages of changes in link-specific measures were used. As shown in Figures 4.13, 4.14, and 4.15, all three scenarios reveal various degrees of change in link-specific measures, especially for the link loadings. The percentages of change in these graphs were calculated by 1) subtracting the measure with feedback from the measure without feedback and then 2) dividing the difference by the measure without feedback. Therefore, a positive sign in the percentage of change indicates that the process without feedback overestimates the measure. Likewise, a negative sign indicates that the process without feedback underestimates the measure or the process with feedback overestimates the measure.

Under the base scenario, the changes in link-specific travel time and speed are insignificant, as depicted in Figures 4.13a and Figure 4.13b, respectively. However, the change in link loading is significant in a positive direction (Figure 4.13c), indicating that the feedback process redistributes the overall link-loading more evenly over the network. Under the medium scenario, the changes in link-specific travel time and speeds are significant in a positive and a negative direction, respectively (Figures 4.14a-b), indicating that the feedback process results in higher speeds and therefore lower travel time. The change in link-specific volume is highly significant in positive direction, indicating again that the feedback process redistribute the overall demand more evenly over the network. As the congestion level further increases in the “congested” scenario, the changes in all the measures are highly significant, with the direction of change being consistent with those of the previous cases. Table 4.11 also summarizes the average changes in link-specific measures.

Overall, the abovementioned observations were found to be consistent with the findings from the COMSIS study (1996), except that the effects of feedback in this study started to take effect even at the medium level of congestion. Further exercises can be undertaken to investigate the effects of feedback for other networks with larger sizes.

Table 4.11. Effects of Direct Feedback on Model Characteristics (Percent Change from “No Feedback”).

	Test Scenario		
	Base	**Medium**	**Congested**
Number of Iterations	5	10	15
Link Individual Travel Time	-0.0044	-1.0382	-6.5395
Link Individual Speed	0.0043	1.1563	9.5104
Link Individual Volume	-0.3041	-0.6240	-13.0066
System Average V/C Ratio	-0.0764	-5.0777	-14.2087

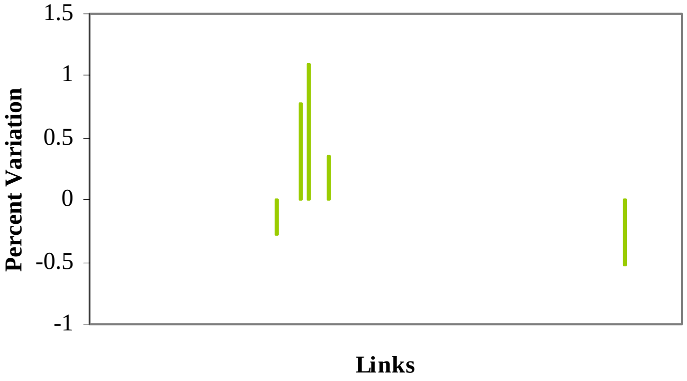


Figure 4.13a. Travel Time Variation with and without Feedback (Base Scenario).

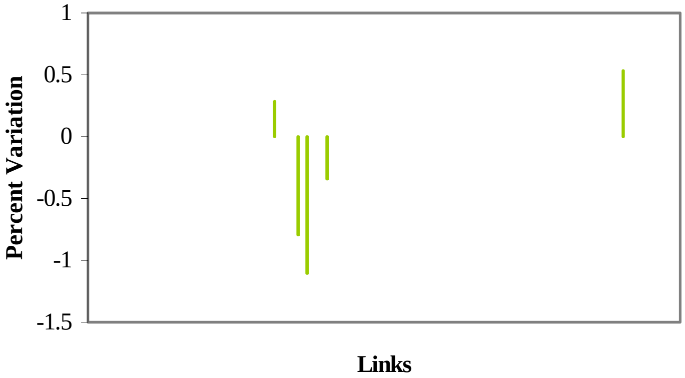


Figure 4.13b. Speed Variation with and without Feedback (Base Scenario).

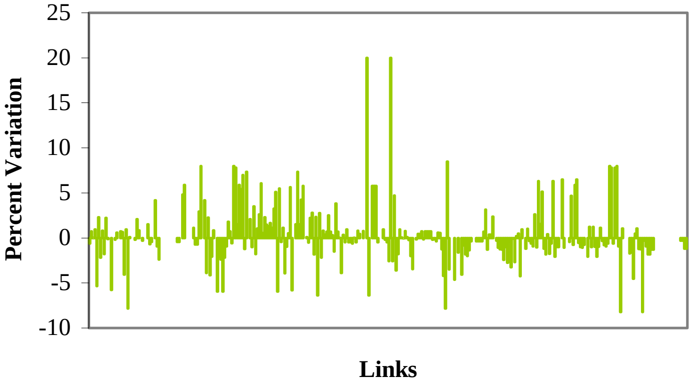


Figure 4.13c. Link Volume Variation with and without Feedback (Base Scenario).

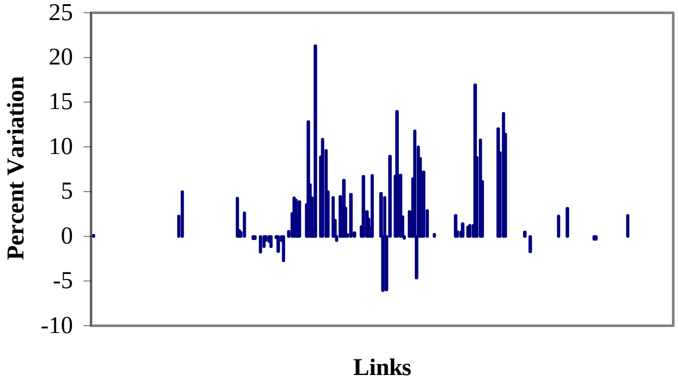


Figure 4.14a. Travel Time Variation with and without Feedback (Medium Scenario).

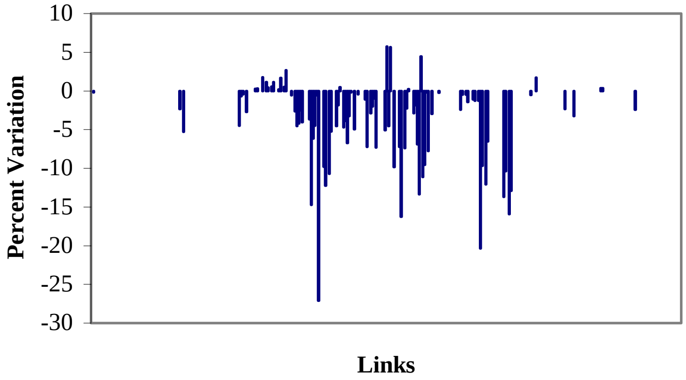


Figure 4.14b. Speed Variation with and without Feedback (Medium Scenario).

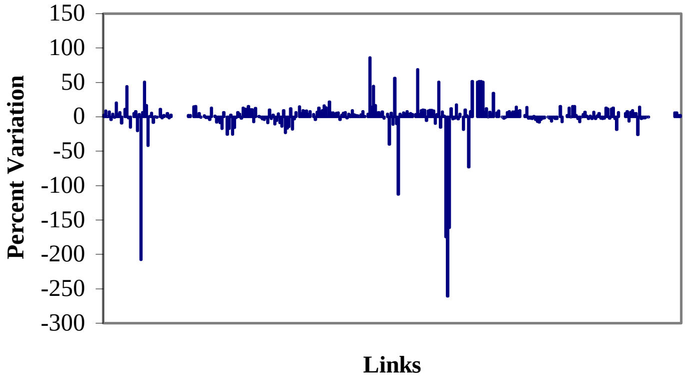


Figure 4.14c. Link Volume Variation with and without Feedback (Medium Scenario).

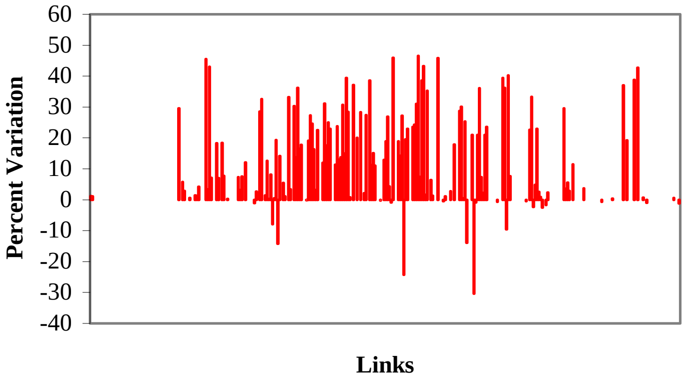


Figure 4.15a. Travel Time Variation with and without Feedback (Congested Scenario).

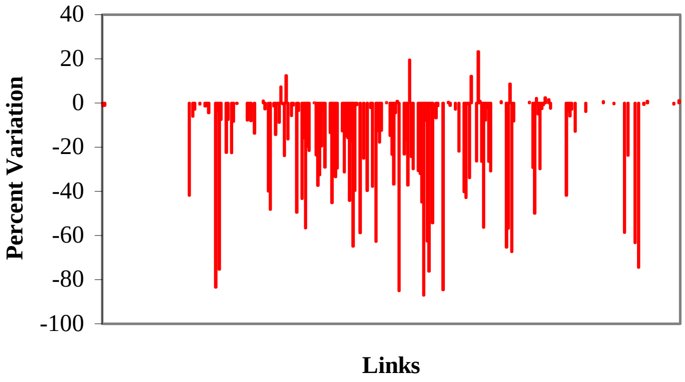


Figure 4.15b. Speed Variation with and without Feedback (Congested Scenario).

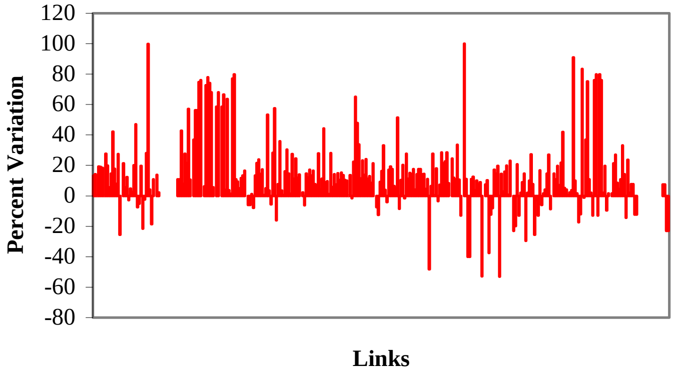


Figure 4.15c. Link Volume Variation with and without Feedback (Congested Scenario).

4.6. Effects of Trip Length Distribution

As suggested by Martin and McGuckin (1998), the *b* and *c* parameters in the Gamma function should both be negative to have a monotonically decreasing shape for friction factors. In some instances, however, the calibrated friction factors behave like an inflected function of time impedance, i.e., increasing when the time impedance is small and decreasing after the point of inflection. To create such a curve, *b* should usually be positive and *c* negative. In the experiments using actual data, a non-positive constraint was imposed on *c* in the optimization routine and *b* was freely determined based on the data. Experiments were carried out on the same two study areas, Broward County and Palm Beach in South Florida. The trip length data were compiled from the trip characteristics survey in South Florida conducted in 2000 (Carr Smith Corradino, 2000). The basic statistics of trip length data by trip purpose are listed in Tables 4.12a-c.

Using the maximum-likelihood procedure, the friction-factor parameters were estimated. The upper bound of zero constraint on the exponential power parameter *c* was active in every case, showing a tendency toward becoming positive. Imposing the non-positive constraint on *c* and allowing the exponent of impedance *b* to be determined freely by the optimization routine, the friction-factor curve preserved a reasonable shape, i.e., decreasing at high time impedances.

The routine 'GMODEL' of TRANPLAN was used to generate calibrated friction factors using the same networks. To implement GMODEL for calibration purpose, several input data are needed, including production and attraction rates, a time impedance matrix, an initial estimate of the friction factors (set at 100 as suggested by TRANPLAN), and a representative trip length frequency distribution over one-minute increments. Given these inputs, GMODEL will generate a friction-factor look-up table per each minute increment. To find a corresponding friction factor, either the modeled skim time can be rounded, or the users can specify an option in GMODEL to interpolate the friction factors based on the modeled skim times. Note that the intra-zonal time impedances were explicitly taken into consideration in model implementation.

Since the calibration process aims to reproduce the observed trip length distribution, reasonable friction factor estimates hinge on whether the observed trip length frequency distribution is representative. Travel surveys usually report trip length in five-minute increments, which is impractical to generate a representative frequency distribution. Tables 4.12a-c illustrate the limitations of using the number of trip length observations by each trip purpose. Figure 4.16 shows a typical trip length frequency distribution over one-minute intervals. Therefore, using the actual frequency distribution as an input data to GMODEL would be impractical and may possibly produce bizarre friction factors. To that end, the empirical distribution needs to be smoothed, or, more rigorously, be fitted by a representative theoretical probability distribution function, such as Gamma distribution, as introduced earlier. Given the first two moments of statistics, the theoretical trip length frequencies were generated using the Gamma distribution for each minute increment, and then used as inputs to GMODEL for calibration. The resulting friction factors were depicted and compared with the proposed ML estimation method, as illustrated in Figure 4.17 (for Broward County) and Figure 4.18 (for Palm Beach County).

Almost all of the friction-factor curves are monotonically decreasing with time impedance, except for the one estimated from TRANPLAN for Home-Based Work (HBW) trips in Palm Beach County, which reveals an increasing trend at high time impedances. Although the resulting friction factors reproduce a closer fit to the "observed" trip length distribution in terms of the first two moments of statistics in TRANPLAN (see Tables 4.12a-c), the shape of the curve is counter-intuitive. Ultimately, the friction factors from the proposed model decline at a more rapid rate than TRANPLAN in all cases.

Both the proposed model and TRANPLAN are capable of producing distributions comparable to the observed distributions at five-minute increments, as shown in Tables 4.12a-c and Figures 4.19 and 4.20. Both models fit the observed distribution reasonably well. However, it is interesting to note that:

1. Both models seem to consistently underestimate the 0-5 minute (including the 5-minute) interval (frequency).
2. The 25-30 minute (excluding the 25-minute but including the 30-minute) interval (frequency) is consistently higher than the neighboring intervals, indicating that the reported trip length from respondents may be rounded to 30-minute for the trip roughly lasting that long.

The first issue is based in the model's code, and can be addressed by replacing the Gamma distribution with a more flexible three-parameter distributional model. Without loss of generality, the two-parameter model was adopted to demonstrate the feasibility of the proposed method. Other possible sources of discrepancy include the imposition of the gravity model in the estimation procedure or inaccuracy in the trip end data. As for the second issue, neither model captures the rounding effect due to the fact that the friction factors calculated from both models are based on smoothed theoretical distributions.

A more robust calibration procedure for the gravity model has been developed, its properties have been discussed and its comparison with traditional calibration method has been made. Based on the Gamma probability density function assumption for observed trip length data, the friction-factor parameters of the gravity model were calibrated using the maximum-likelihood procedure. The results show that, as with the traditional calibration procedure for the gravity model, the proposed method is capable of reproducing trip length distributions by trip purpose comparable to the observed distributions. Explicit constraints can also be imposed on the parameters in the optimization routine to prevent the friction-factor curve from becoming ill behaved, such as increasing time impedances at appropriate intervals. More importantly, the estimation procedure used is more robust in the sense that the traditional calibration procedure may become unstable due to an inherent deficiency associated with the friction function update formula.

One of extensions from this work is to release the imposition of the gravity model from the estimation procedure for calculating trip interchanges for possible reconciliation of the trip end data, assuming that the trip-end data obtained from the trip generation process are erroneous. Especially on the attraction side, the magnitude of errors might be significant, needing reconciliation. To that end, additional data, such as traffic counts, will be required to supplement the possible data insufficiency and enhance the ability of the model to estimate the problem.

Table 4.12a. Trip Length Statistics from Household Travel Characteristics Survey Database.

Trip Purpose	Broward County			Palm Beach County		
	No.	Mean[1]	Std.[2]	No.	Mean	Std.
HBW	1674	23.78	13.77	1678	24.14	16.01
HBSHP	929	14.56	11.69	1157	15.5	12.06
HBSR	633	15.64	11.42	844	16.51	12.29
HBSCH	831	19.6	14.36	682	19.59	14.27
HBO	1880	15.7	12.12	2074	16.81	12.66
NHBW	712	16.88	14.06	739	17.52	14.32
NHBO	1348	16.3	15.49	1619	16.05	13.71
Overall	8007	17.86	13.8	8793	18.14	14.05

Note: 1. Unit: min. 2. Standard deviation

Table 4.12b. Trip Length Statistics from Proposed Model and TRANPLAN (Broward County).

Trip Purpose	Proposed Model		TRANPLAN	
	Mean[1]	Std. [2]	Mean	Std.
HBW	21.47	10.43	20.92	11.11
HBSHP	14.12	8.76	14.48	8.4
HBSR	14.39	9.29	14.68	8.62
HBSCH	16.54	8.93	16.47	9.48
HBO	14.81	9.39	14.83	8.5
NHBW	14.84	8.71	13.82	7.67
NHBO	16.08	9.72	15.45	8.86

Note: 1. Unit: min. 2. Standard deviation

Table 4.12c. Trip Length Statistics from Proposed Model and TRANPLAN (Palm Beach County).

Trip Purpose	Proposed Model		TRANPLAN	
	Mean[1]	Std. [2]	Mean	Std.
HBW	22.66	13.06	24.51	16.54
HBSHP	15.42	10.14	15.84	11.02
HBSR	16.2	10.75	16.64	11.48
HBSCH	19.28	12.38	19.55	14.22
HBO	16.58	11.05	17.02	11.91
NHBW	16.74	10.69	17.52	12.94
NHBO	15.56	10.42	16.95	12.6

Note: 1. Unit: min. 2. Standard deviation

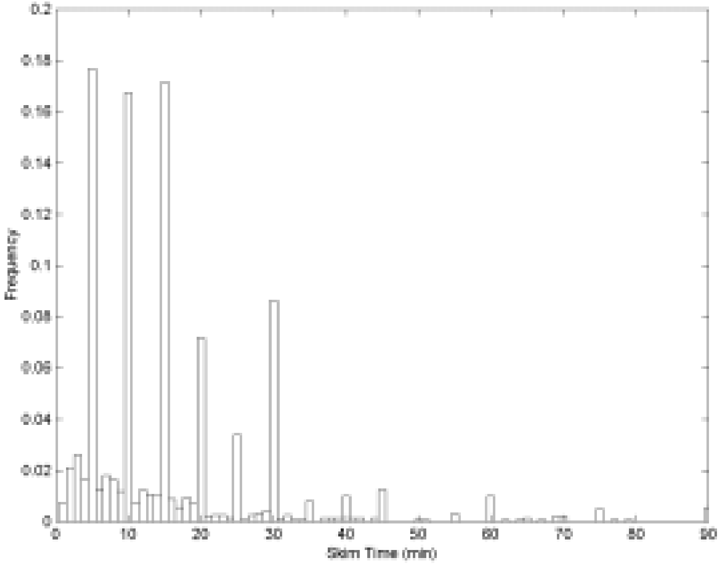


Figure 4.16. A Typical Empirical Trip Length Distribution Over One-Minute Increments.

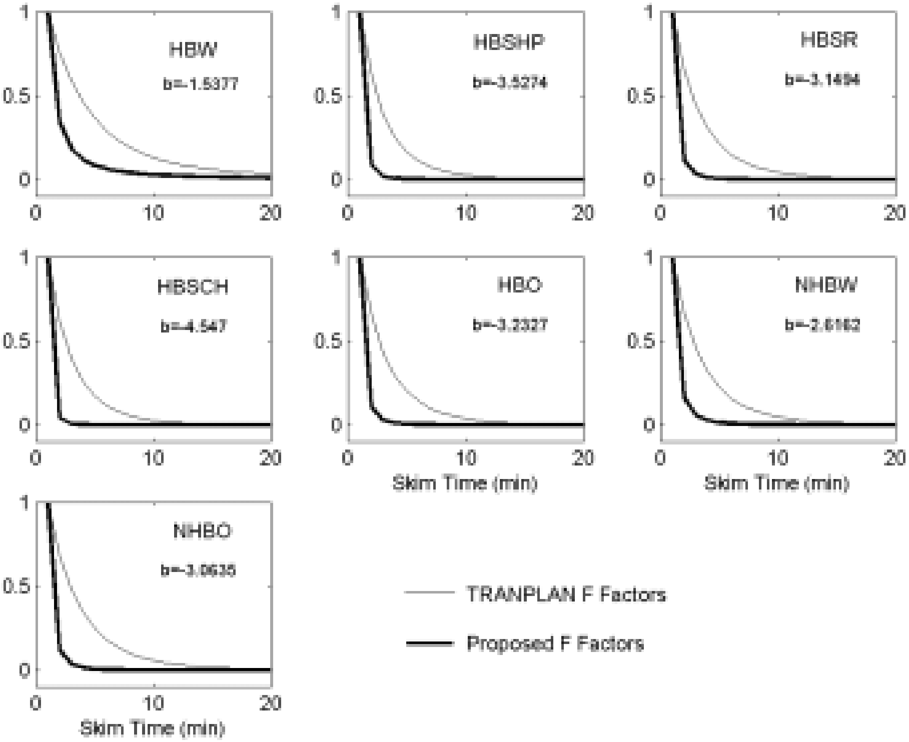


Figure 4.17. Comparison of Proposed Friction Factors (Normalized) with TRANPLAN by Trip Purpose (Broward County).

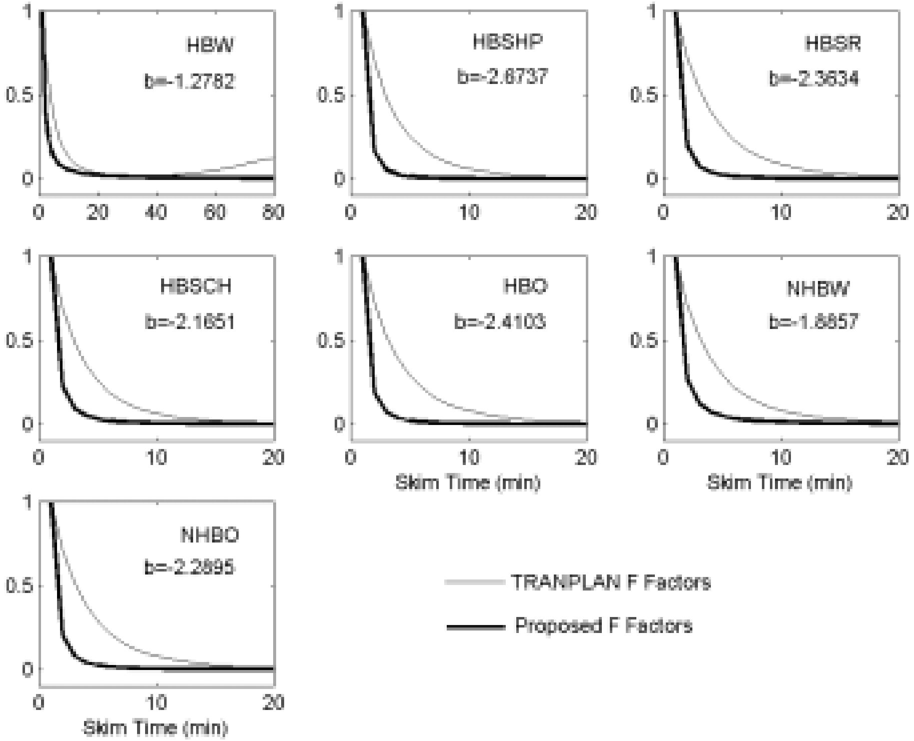


Figure 4.18. Comparison of Proposed Friction Factors (Normalized) with TRANPLAN by Trip Purpose (Palm Beach County).

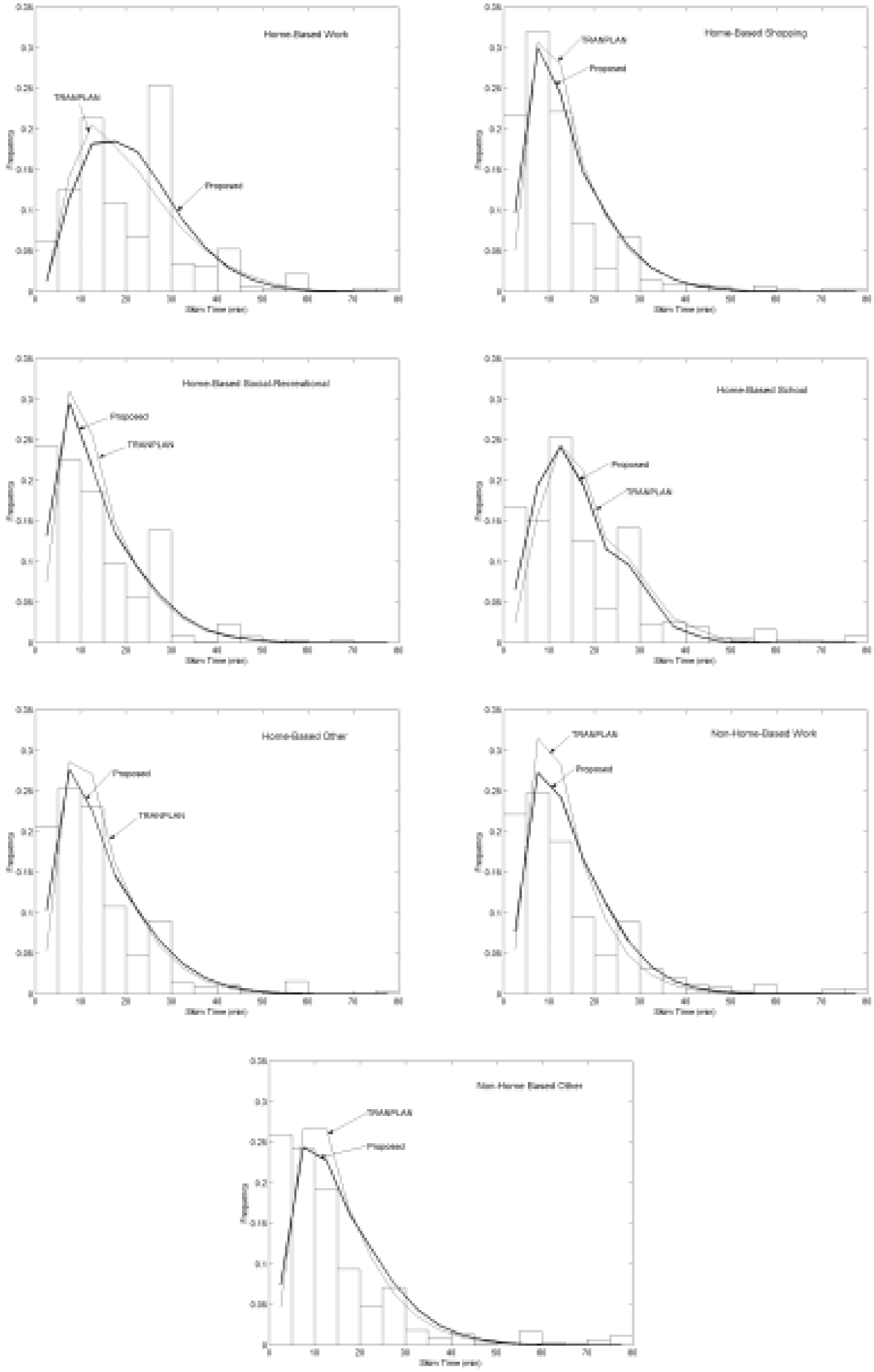


Figure 4.19. Comparison of Derived Trip Length Distributions from Proposed Model and TRANPLAN with the Observed Distribution (in Bars) by Trip Purpose (Broward County).

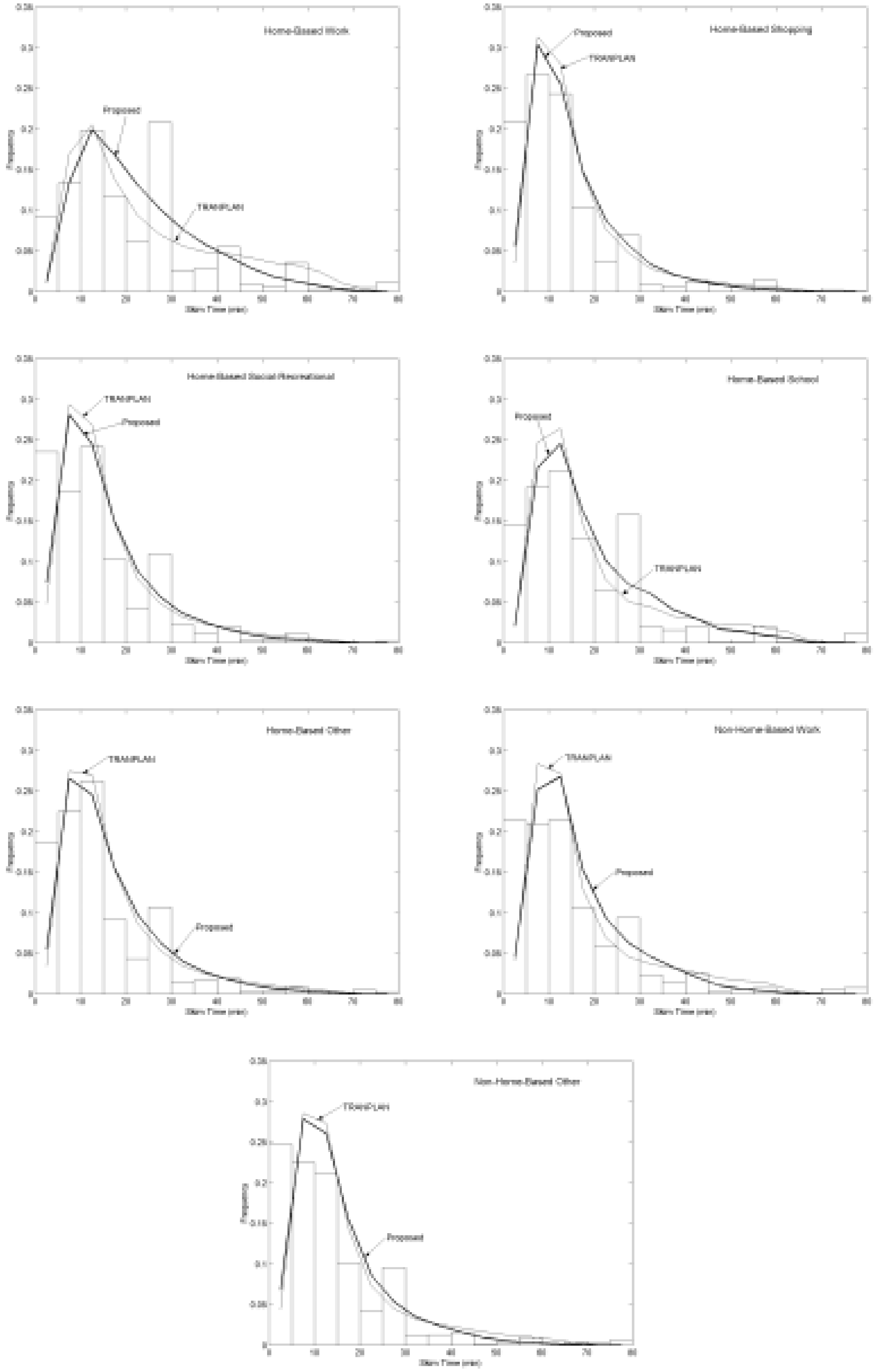


Figure 4.20. Comparison of Derived Trip Length Distributions from Proposed Model and TRANPLAN with the Observed Distribution (in Bars) by Trip Purpose (Palm Beach County).

CHAPTER 5
SUMMARY, FINDINGS, AND RECOMMENDATIONS

Several feedback approaches to possibly achieving consistent travel impedances are attempted, including the direct method, the method of successive average (MSA), and the MSA with trip length calibration. They are performed and compared in this study with the pre-assignment method that is currently used to achieve consistency in FSUTMS. A small artificial network and medium real networks, including Broward and Palm Beach Counties, are used in this report to demonstrate the effects of the proposed feedback methods. The major findings from implementing feedback procedures in FSUTMS are summarized as follows:

1. In general, the direct feedback method in general fails to achieve convergence. In the case of small networks, oscillation may occur between two states of link flow pattern and the network never reaches convergence. This problem becomes less pronounced as the level of congestion increases.
2. The Method of Successive Average (MSA) facilitates the convergence behavior. When the level of congestion increases, the advantage of MSA becomes especially pronounced.
3. Due to feedback, link-specific speed in general increases and consequently the travel time decreases.
4. The feedback process redistributes the Origin-Destination (OD) trips more evenly over alternative routes on network than the four-step process without feedback due to more congested skim time.
5. The trip distance and length are, in general, reduced due to feedback process. This results in lower total VMT and VHT on the network.
6. In general, the feedback process improves the overall volume-to-capacity ratio on the network.
7. The model accuracy in terms of root mean square error (RMSE) improves marginally due to feedback using the friction factor as determined from the pre-assignment scheme.
8. Feedback with on-line trip length calibration using observed trip length as the starting value results in a systematically higher level of assigned volumes and degraded RMSE when compared to ground counts.
9. Feedback with on-line trip length calibration and total VHT (Vehicle Hour Traveled) control through proportional adjustment by iteration (PAI) restores the RMSE and other statistics to a comparable level with the pre-assignment scheme.

As a result of implementing feedback with on-line trip length calibration using observed trip length as starting values, two significant effects were observed as compared to the pre-assignment scheme:

- Total automobile trips were reduced due to mode shift from driving alone to higher occupancy vehicles.
- The trip length distribution becomes elongated (dispersed). Both the mean and the standard deviation increase, indicating that trips were longer and more spatially dispersed.

Compounding these two effects, the total VMT and VHT increased by 12% and 13%, respectively. This leads to a systematically higher level of assigned volumes over ground counts (ratio ≈ 1.12) with the RMSEP (Root Mean Square Error Percentage) degraded by 6%. To resolve this issue, a simple but effective scheme, called *proportional adjustment by iteration* (PAI), was proposed and implemented along with the feedback process. The basic assumption of this approach is that every individual trip length is proportionally adjusted upward or downward according to the overall VHT volume-to-count ratio. This assumption can be validated when there exists difference between the scales of the real the modeled networks. Using this approach, the authors were able to control the overall VMT volume-to-count ratio to 1.0 and VHT volume-to-count ratio to 1.01 at convergence. The RMSEP and other statistics also improved substantially to a comparable level with pre-assignment scheme and feedback without trip length calibration. The resulting trip length distribution was close to the distribution resulting from the pre-assignment scheme. However, curve shapes of the normalized friction factors by trip purpose are more well-defined than the factors resulting from the pre-assignment scheme. Note that all critical model parameters in the pre-assignment model have been fine-tuned to produce the estimated counts consistent with ground counts. The proposed feedback process with on-line trip length calibration and PAI is modified from the pre-assignment model by inserting minimum necessary steps for feedback implementation without further calibrating other model parameters. Further calibration of other model parameters to the feedback process might continue to enhance the accuracy of the model.

The results presented in this report do not provide sufficient evidence to support the significant benefit of feedback process to the model accuracy as claimed by previous research overall. This can be partially attributed to the fact that the proposed feedback model has not been fully calibrated as in the pre-assignment model. Proper calibration of friction factors governing trip length and other important model parameters seems to bear more significance affecting the model accuracy than the feedback process itself.

In terms of future research, effort is definitely needed to determine if additional calibration can be exercised to further enhance the model consistency. The companion report to this project attempts to address the effectiveness of using variable CONFAC for the same purpose. An effort worth pursuing is to integrate these two feedback loops and implement them simultaneously to investigate their overall effects on the consistency issue. In terms of trip length calibration, one of extensions from this work is to release the imposition of the gravity model from the estimation procedure for calculating trip interchanges for possible reconciliation of the trip end data, assuming that trip end data obtained from trip generation process are erroneous. Especially on the attraction side, the magnitude of errors can be significant and needs to be reconciled. To that end, additional data sources, such as traffic counts, will be required to supplement the potentially insufficient data and enhance the estimability of the problem.

REFERENCES

Abdulaal, M. and L. J. LeBlanc, "Methods for Combining Modal Split and Equilibrium Assignment Models," *Transportation Science,* 13, pp. 292-314, 1979

Abrahamson, T. and L. Lundqvist, "Formulation and Estimation of Combined Network Equilibrium Models with Applications to Stockholm," *Transportation Science*, 33, pp. 80-100, 1999.

Blue Ribbon Panel, "Future Direction of Florida's Transportation Models," Report of the Florida Statewide Model Task Force Blue Ribbon Panel, April 2002, (http://www11.myflorida.com/ planning/systems/stm/mtf/02docs/BluRibPn.pdf).

Boyce, D., Y-F Zhang, and M. R. Lupa, "Introducing 'Feedback' into Four-step Travel Forecasting Procedure Versus Equilibrium Solution of Combined Model," *Transportation Research Record 1443*, National Academic Press, Washington, D.C., pp. 65-74, 1994.

Boyce, D. and Y-F Zhang, "Calibrating Combined Model of Trip Distribution, Modal Split, and Traffic Assignment," *Transportation Research Record 1607*, National Academic Press, Washington, D.C., pp. 1-5, 1994.

Carr Smith Corradino. *Southeast Florida Regional Travel Characteristic Study*. Executive Summary Report, Prepared for District 4, Florida Department of Transportation, 2000.

Carr Smith Corradino. *Treasure Coast Regional Planning Model Update – Automatic Model Calibration System*. Version 2, 2nd Technical Report, Prepared for Florida Department of Transportation, 2002.

Chu, Y-L., "A Combined Trip Distribution and Assignment Model with Dogit Destination Functions," *Transportation Research Record 1285*, National Academic Press, Washington, D.C., pp. 70-77, 1990.

COMSIS Corporation, *Incorporating Feedback in Travel Forecasting Methods, Pitfalls and Common Concerns,* Final Report Prepared for the Federal Highway Administration, U.S. Department of Transportation, March 1996.

Evans, A. W. The Calibration of Trip Distribution Models with Exponential or Similar Cost Functions, *Transportation Research*, Vol. 5, pp. 15-38, 1971.

Evans, S. P., *Some Applications of Optimization Theory in Transportation Planning* Ph.D. thesis, Research Group in Traffic Studies, University of London, London, 1973.

Evans, S. P., Derivation and Analysis of Some Models for Combining Trip Distribution and Assignment, *Transportation Research,* Vol. 10, No. 1, pp. 37-57, 1976.

Federal Highway Administration. *Computer Programs for Urban Transportation Planning: PLANPAC/BACK-PAC General Information*, U. S. Department of Transportation, Stock No. 050-001-00125-0, Washington, DC, 1977.

Florian, M., S. Nguyen, and Ferland, On the Combined Distribution-Assignment of Traffic, *Transportation Science*, 9, pp. 43-53, 1975.

Florida Department of Transportation, *Urban Transportation Planning Model Update-Phase II, Task C, Develop Standard Distribution and Assignment Models*, 1981.

Frank, C., *A Study of Alternative Approaches to Combined Trip Distribution-Assignment Modeling,* Ph.D. thesis, Department of Regional Science, University of Pennsylvania, Philadelphia, 1978.

Frank, M. and P. Wolfe, An algorithm for Quadratic Programming, *Naval Res. Logist. Quart.*, 3, pp. 95-110, 1956.

Harvey, G. and E. Deakin, *Manual of Regional Transportation Modeling Practice for Air Quality Analysis*, Prepared for the National Association of Regional Councils, July 1993.

Hasan, M. and K. Safwat, Comparison of Two Transportation Network Equilibrium Modeling Approaches, *Journal of Transportation Engineering*, Vol. 126, No. 1, pp. 35-40, 2000.

Horowitz, A., Tests of an Ad Hoc Algorithm of Elastic-Demand Equilibrium Traffic Assignment, *Transportation Research B*, Vol. 23B, No. 4, pp. 309-313, 1989.

Huang, H-J and W. H. K. Lam, Modified Evans' Algorithms for Solving the Combined Trip Distribution and Assignment Problem, *Transportation Research B*, Vol. 26B, No. 4, pp. 325-337, 1992.

Hyman, G. M., The Calibration of Trip Distribution Models, *Environment and Planning* 1, pp. 105-112, 1969.

Kirby, H. R., Theoretical Requirements for Calibrating Gravity Model, *Transportation Research*, Vol. 8, pp. 97-104, 1974.

Lam, W. H. K. and H-J Huang, A Combined Trip Distribution and Assignment Model for Multiple User Classes, *Transportation Research B*, Vol. 26B, No. 4, pp. 275-287, 1992.

LeBlanc, L. J. and K. Farhangian, Efficient Algorithms for Solving Elastic Demand Taffic Assignment Problems and Mode Split-Assignment Problems, *Transportation Science*, 15, pp. 306-317, 1981.

Levinson, D. and A. Kumar, Integrating Feedback into Transportation Planning Model: Structure and Application, *Transportation Research Record 1413*, National Academic Press, Washington, D.C., pp. 70-77, 1993.

Li, S. C. and A. Lapkowski, Improving the Accuracy of Demand Forecast Models, *Florida Transportation Modeling Newsletter*, pp. 3-4, December 1999.

Martin, W. A. and N. A. McGuckin. *Travel Estimation Techniques for Urban Planning*. NCHRP Report 365, Transportation Research Board, Washington, DC, 1998.

MathWorks Inc. *MATLAB Version 5.3.1*, 1999.

Metaxatos, P., D. Boyce, M. Florian, and I. Constantin, Implementing Combined Model of Origin-Destination and Route Choice in EMME/2 System, *Transportation Research Record 1493*, National Academic Press, Washington, D.C., pp. 57-63, 1995.

Miller, H. J., *Towards Consistent Travel Demand Estimation in Transportation Planning*, Prepared for the Bureau of Transportation Statistics, U.S. Department of Transportation, November 1997.

Papacostas C. S. and P. D. Prevedouros. *Transportation Engineering and Planning*, 3rd Edition, Prentice Hall, 2001.

Rodier, C. and R. Johnston. *Comments on the Final Environmental Impact Statement for the proposed construction of the Legacy Highway*. Department of Environmental Science and Policy. University of California, California, 2000.

Safwat. K. and M. Hasan, Computational Experience with a Simultaneous Transportation Equilibirum Model Under Varying Parameters, *Transportation Research Record 1251*, National Academic Press, Washington, D.C., pp. 17-23, 1989.

Safwat. K. and T. Magnanti, A Combine Trip Generation, Trip Distribution, Modal Split, and Trip Assignment Model, *Transportation Science*, 14, pp. 155-173, 1980.

Sheffi, Y. “Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods,” Prentice-Hall, Inc., 1985.

Sheffi, Y. and C. F. Daganzo, Computation of Equilibrium Over Transportation Network: the Case of Disaggregate Demand Models, *Transportation Science*, 18, pp. 14-30, 1988.

Stopher, P. R., Deficiencies of Travel-Forecasting Methods Relative to Mobile Emissions, *Journal of Transportation Engineering*. Vol 119, No. 5, pp. 723-741, 1993.

Tatineni, M., M. Lupa, D. Englund, and D. Boyce, Transportation Policy Analysis Using a Combined Model of Travel Choice, *Transportation Research Record 1452*, National Academic Press, Washington, D.C., pp. 10-17, 1994.

Van Zijpp, N. J., B. G. and M. G. H. Bell. Heydecker. Improved Algorithms for Calibrating Gravity models. *Transportation Networks: Recent Methodological Advances*, pp. 99-113, 1998.

Walker, W. and H. Peng, Alternative Methods to Iterate a Regional Travel Simulation Model: Computational Practicality and Accuracy, *Transportation Research Record 1493*, National Academic Press, Washington, D.C., pp. 21-28, 1995.

Winslow, A. Bladikas, K. Hausman, and L. Spasovic, Introduction of Information Feedback Loop to Enhance Urban Transportation Modeling System, *Transportation Research Record 1493*, National Academic Press, Washington, D.C., pp. 81-89, 1995.

Appendix A. Codes for Feedback Implementation

A.1. Input File Specification

The program, called fdk.exe, is written in Visual Basic and embedded in the post-processing batch file for implementing feedback. The following input file to fdk.exe specifies the feedback method, convergence criteria, number of iterations, and filenames for immediate files.

```
1                      /1 for direct method; 2 for MSA
0.001                  /criterion of impedance
0.001                  /criterion of speed
0.001                  /criterion of link assigned volumes
5                      /number of iteration
hassign1.out           /hassign.out of previous iteration
hrldxy1.txt            /hrldxy of previous iteration
hassign.out            /hassign.out of current Iteration
hrldxy.txt             /hrldxy of current Iteration
hrldxy1.txt            /output file name
0                      /delay time for ‘pause’
2                      /new volume method   2---for FTOWN; 3---for Broward and Palm Beach
```

A.2. Program Listing

```
Define initial variables

'Define initial variables
Private i As Long                     'Temp Var for counter
Private FileNumber As Integer
Private strFileName As String
Private intSelectMode As Integer      ' 1---for direct   2---for MSA
Private dblCompareResult1 As Double   'Define Compare Result 1
Private dblCompareResult2 As Double   'Define Compare Result 2
Private dblCompareResult3 As Double   'Define Compare Result 3
Private intIterateNumber As Integer
Private strOpenFileName1 As String
Private strOpenFileName2 As String
Private strOpenFileName3 As String
Private strOpenFileName4 As String
Private strOpenFileName5 As String
Private PauseTime As Variant
Private intVolumeNumber As Integer   'xiaofeng add on 06/20/02 to get how many volume
Private strUseOpenFile1 As String    'Current used filename 1
Private strUseOpenFile2 As String    'Current used filename 2

Private strMaxLinkSign As String                    'to see where is Max Link Number
Private strIterationFractionBeginSign As String     'to see where is Iteration Fraction Begin
Private strIterationFractionEndSign As String       'to see where is Iteration Fraction End
Private strNodeBeginSign As String                  'to see where is Node Begin
Private strConfacSign As String                     'to see where is CONFAC normal CONFAC=0.10
'Define initial variables

'Define Variables
Private strAnode() As String                     'define ANode
Private strBnode() As String                     'Define BNode
Private dblFraction() As Double                  'Define Fraction
Private dblAverageImp() As Double                'Define Average Imp
Private dblAverageTravelTime() As Double         'Define average trave time used for data
exchange
Private dblAverageTravelTime1() As Double        'Define average trave time 1
Private dblAverageTravelTime2() As Double        'Define average trave time 2
Private dblAverageSpeed() As Double              'Define average speed
Private dblTotalVolume() As Double               'Define Total Volume
Private intMaxLinks As Long                      'Define Max Link Number
Private intMaxIteration As Integer               'Define Max Iteration Number
Private intIterationLines As Integer             'Define Iteration Line
'Private dblConFAC As Double
'Define Variables

'Define Variables
Private isExit As Boolean                        'Need Exit =1
Private intTotalItemNumber As Integer            'Total Item Number used for dim of Array
Private intTotalItemNumber1 As Long, intTotalItemNumber2 As Long
Private dblImp1() As Double, dblImp2() As Double 'Define imp for two files
Private dblSpd1() As Double, dblSpd2() As Double 'Define speed for two files
Private dblVol1() As Double, dblVol2() As Double 'Define Volume for two files

Private dblSumImp  As Double, dblSumSpd  As Double, dblSumVol  As Double 'Used for caculation
Private dblSumImp1 As Double, dblSumSpd1 As Double, dblSumVol1 As Double 'Used for caculation
Private dblAvgImp1 As Double, dblAvgSpd1 As Double, dblAvgVol1 As Double 'Used for caculation
Private dblSumImp2 As Double, dblSumSpd2 As Double, dblSumVol2 As Double 'Used for caculation
Private dblAvgImp2 As Double, dblAvgSpd2 As Double, dblAvgVol2 As Double 'Used for caculation
Private dblResult1 As Double, dblResult2 As Double, dblResult3 As Double 'Used for caculation
'Define Variables

'Define Variables
Private cntStamp_Old As Variant             'used for check old fdk.cnt file stamp
Private cntFileLength_Old As Long         'used for check old fdk.cnt file length

Private outStamp_Old As Variant             'used for check old fdk.out file stamp
Private outFileLength_Old As Long           'used for check old fdk.out file length

Private hrldxyStamp_Old As Variant          'used for check old hrldxy.txt file stamp
Private hrldxyFileLength_Old As Long        'used for check old hrldxy.txt file length

Private intIterationNumber_Old As Integer
'Define Variables

'Define Variables added by xiaofeng 06/26/02
Private intFT(90) As Integer        'Define Facility Type
Private dblConFAC(90) As Double     'Get CONFAC
'Define Variables added by xiaofeng 06/26/02
```

```
Sub Form_Load

Private Sub Form_Load()
   'Define initial variables
   strMaxLinkSign = "NUMBER OF LINKS   ="
   strIterationFractionBeginSign = "ITERATION  V(LINK)*I(LINK)  V(O-D)*I(O-D)  (S1-S2)/S1  LAMBDA
FRACTION"
   strIterationFractionEndSign = "IC = IMPEDANCE DUE TO CONGESTION ="
   strNodeBeginSign = "ANODE  BNODE    VOLUME  CAPACITY       V/C"
   strConfacSign = "&CONFAC        Factor to convert hourly capacity to daily capacity (E Level)"

   dblSumImp = 0
   dblSumSpd = 0
   dblSumVol = 0
   dblSumImp1 = 0
   dblSumSpd1 = 0
   dblSumVol1 = 0
   dblSumImp2 = 0
   dblSumSpd2 = 0
   dblSumVol2 = 0
   'Define initial variables

   'Following code add in 06/13/02
   On Error Resume Next   ' Defer error trapping.

   Dim intNewValue As Integer
   Dim MyNumber
   FileNumber = 1
   strFileName = App.Path + "\fdk.cnt"

   Dim msg As String

   Open strFileName For Input As #FileNumber   ' Open file for input.

   If Err.Number = 52 Or Err.Number = 53 Or Err.Number = 54 Or Err.Number = 55 Or Err.Number = 58
Or Err.Number = 62 Or Err.Number = 75 Then
      ' Tell user what happened. Then clear the Err object.
      msg = "There was an error attempting to open the file!" & strFileName
      MsgBox msg, , "Deferred Error Test (FDK)"
      Err.Clear   ' Clear Err object fields
   End If


   Input #FileNumber, MyNumber   ' Read data into one variables.
   intNewValue = CInt(MyNumber)
   Close #FileNumber   ' Close file.

   If intNewValue < 1 Then
        intNewValue = 1
   End If
   'Above code add in 06/13/02

   intIterationNumber_Old = intNewValue

   strFileName = App.Path + "\fdk.cnt"
   cntStamp_Old = FileDateTime(strFileName)
   cntFileLength_Old = FileLen(strFileName)

   strFileName = App.Path + "\fdk.out"
   outStamp_Old = FileDateTime(strFileName)
   outFileLength_Old = FileDateTime(strFileName)

   If intIterationNumber_Old = 1 Then
        strFileName = App.Path + "\hrldxy.txt"
   Else
        strFileName = App.Path + "\hrldxy1.txt"
   End If

   hrldxyStamp_Old = FileDateTime(strFileName)
   hrldxyFileLength_Old = FileDateTime(strFileName)
   'Read the initial Values
   ReadInitalValues
   'Read the initial Values

   If intNewValue > 1 Then   ' Add this If...Endif from 06/13/02
        'Write the correct format file in order to the program can read the file line by line
        WriteCorrectFormatFiles strOpenFileName1, strOpenFileName2
        'Write the correct format file in order to the program can read the file line by line

        'Pause for PauseTime Seconds
        'TimeDelay PauseTime
        'Pause for PauseTime Seconds

        'Read data from the file
        ReadDataFromFiles
        'Read data from the file

        'Process1 of the data
        DataProcess1
        'Process1 of the data
   End If

   'Write the correct format file in order to the program can read the file line by line
   WriteCorrectFormatFiles strOpenFileName3, strOpenFileName4
   'Write the correct format file in order to the program can read the file line by line

   'Read data from the file
   ReadDataFromFiles
   'Read data from the file

   'Process1 of the data
   DataProcess2
   'Process1 of the data

   'Rewrite Second Average Travel Time of file HRLDXY1.TXT for new model
   WriteWTTtoHRLDXY1
   'Rewrite Second Average Travel Time of file HRLDXY1.TXT for new model

   If intNewValue > 1 Then   ' Add this If...Endif from 06/13/02
        'Caculate the resultes , compare the results  and write the results and count number
        GetResults
        'Caculate the resultes , compare the results  and write the results and count number
   End If

   'Pause for PauseTime Seconds and check the files
   TimeDelay PauseTime
   'Pause for PauseTime Seconds and check the files


   If isExit Then
      Unload Me
   End If

End Sub
```

Appendix B. Post-Processing Batch File for Palm Beach Model

B.1. File Listing

Instructions: When executing the batch file in the DOS environment, one needs to supply two other parameters. The first parameter is either “f” or “p”, giving the option for running feedback or pre-assignment. The second parameter is either “t” or “o”, giving the option for including or not including the trip length calibration.

```
rem Post-processing batch file for self-calibrating feedback
rem >xpalm f t  :feedback with trip length calibration
rem >xpalm f o  :feedback without trip lng calibration
rem >xpalm p t  :pre-assignment w/ trip lng calibration
rem >xpalm p o  :pre-assignment w/o trp lng calibration
pause
rem goto FSU100

echo off
rem ******************* Batch Initiation *******************
copy *.ink *.ins >NUL
copy trnpln0.con trnplnx.con >NUL
copy titlx title >NUL
if exist endexam.fdk del endexam.fdk
if exist ptrips11.p99 del ptrips11.p99
rem ********************************************************

IF EXIST TRNPLN.ERR DEL TRNPLN.ERR
ECHO OFF
ECHO ***** EXT ***** -- START
datime >> time.log
datime
if exist EXT.OUT del EXT.OUT
if exist *.TEM del *.TEM
if exist EEMAKE.ERR del EEMAKE.ERR
eemake
datime
if exist EEMAKE.ERR goto err
ECHO SKIP1
RTABLE
DEL TRNPL001.INS
IF EXIST EXT.OUT      GOTO FSU001
REN TRNPL001.OUT EXT.OUT
GOTO FSU002
:FSU001
COPY EXT.OUT     +TRNPL001.OUT >NUL
DEL TRNPL001.OUT
:FSU002
IF EXIST TRNPLN.ERR GOTO ERR
datime
echo ***** CREATE INTERNAL-EXTERNAL TRANSIT TRIP TABLE*****
BUILD
DEL TRNPL002.INS
COPY EXT.OUT     +TRNPL002.OUT >NUL
DEL TRNPL002.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
datime >> time.log
ECHO ***** EXT ***** -- END
if exist TRNPLN.ERR goto err
echo off
ECHO OFF
if exist GEN.OUT DEL GEN.OUT
ECHO ***** LSTGEN ***** -- START
datime >> time.log
datime
if exist SRVSCHL.p99      DEL SRVSCHL.p99
if exist PBSC*.TEM DEL PBSC*.TEM
if exist tripgen.out DEL tripgen.out
C:\project\pb99p\pbexecs\lstgenm
```

```
if exist STP283.TEM del STP283.TEM
copy PBSCH.TEM SRVSCHL.p99
copy A1DECK.TEM A1DECK.p99
if exist PBSC*.TEM DEL PBSC*.TEM
if exist HH.TEM del HH.TEM
if exist PRODS.TEM del PRODS.TEM
if exist ATTRS.TEM del ATTRS.TEM
if exist AUTO.TEM del AUTO.TEM
ren TRIPGEN.OUT GEN.OUT
if exist LSTGEN.ERR copy GEN.OUT+LSTGEN.ERR
if exist LSTGEN.ERR DEL LSTGEN.ERR
datime
datime >> time.log
ECHO ***** LSTGEN ***** -- END
if exist TRNPLN.ERR goto err
echo off
ECHO OFF
ECHO ***** HNET ***** -- START
datime >> time.log
datime
if exist HNET.OUT del HNET.OUT
netpro
if exist NETPRO.INF goto HNET1
goto HNET2
:HNET1
ren NETPRO.INF HNET.OUT
:HNET2
if exist NETPRO.ERR goto HNET3
HWYNET
DEL TRNPL003.INS
IF EXIST HNET.OUT     GOTO FSU005
REN TRNPL003.OUT HNET.OUT
GOTO FSU006
:FSU005
COPY HNET.OUT    +TRNPL003.OUT >NUL
DEL TRNPL003.OUT
:FSU006
IF EXIST TRNPLN.ERR GOTO ERR
datime
goto HNET5
:HNET3
copy NETPRO.ERR TRNPLN.ERR
if exist HNET.OUT goto HNET4
ren NETPRO.ERR HNET.OUT
goto HNET5
:HNET4
copy NETPRO.ERR HNET.OUT
del NETPRO.ERR
:HNET5
datime
datime >> time.log
ECHO ***** HNET ***** -- END
echo off
if exist TRNPLN.ERR goto err
ECHO OFF
ECHO ***** HPATH ***** START
datime >> time.log
datime
if exist HPATH.OUT del HPATH.OUT
datime
SLCTSM
rem *------------------UM Insertion------------------------*
rem *  Backup free-flow skim for later use                 *
copy fhskims.p99 fskims.p99 >NUL
rem *------------------------------------------------------*
DEL TRNPL004.INS
IF EXIST HPATH.OUT    GOTO FSU007
REN TRNPL004.OUT HPATH.OUT
GOTO FSU008
:FSU007
COPY HPATH.OUT   +TRNPL004.OUT >NUL
```

```
DEL TRNPL004.OUT
:FSU008
IF EXIST TRNPLN.ERR GOTO ERR
datime
SLCTSM
DEL TRNPL005.INS
COPY HPATH.OUT   +TRNPL005.OUT >NUL
DEL TRNPL005.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
datime >> time.log
ECHO ***** HPATH ***** -- END
if exist TRNPLN.ERR goto err
echo off
ECHO OFF
ECHO ***** DISTRIB ***** -- START
datime >> time.log
datime
if exist DISTRIB.OUT del DISTRIB.OUT
IZTIME
DEL TRNPL006.INS
IF EXIST DISTRIB.OUT  GOTO FSU009
REN TRNPL006.OUT DISTRIB.OUT
GOTO FSU010
:FSU009
COPY DISTRIB.OUT +TRNPL006.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL006.OUT
:FSU010
IF EXIST TRNPLN.ERR GOTO ERR
datime
C:\project\pb99p\pbexecs\ttprepbx
UTABLE
DEL TRNPL007.INS
COPY DISTRIB.OUT +TRNPL007.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL007.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
datime
GMODEL
DEL TRNPL008.INS
COPY DISTRIB.OUT +TRNPL008.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL008.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
BUILD
DEL TRNPL009.INS
COPY DISTRIB.OUT +TRNPL009.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL009.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
MTABLE
DEL TRNPL010.INS
COPY DISTRIB.OUT +TRNPL010.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL010.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
ECHO ***** COMBINE TRIP PURPOSES FOR PRE-MODE CHOICE *****
MTABLE
DEL TRNPL011.INS
COPY DISTRIB.OUT +TRNPL011.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
```

```
del lan.tem>NUL
DEL TRNPL011.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
ECHO ** HWYONLY MODE CHOICE PROGRAM **
ECHO ** GENERATE TRIP TABLE FOR PRE ASSIGNMENT AND CONGESTED SKIM **
ECHO ** (Input Trip Tables generated from Free Flow Skim)**
copy A1DECK.p99     A1DECK.TEM
if exist NLOGIT.OUT del NLOGIT.OUT
if exist MODE.ERR del MODE.ERR
if exist DEFMSAM.p99     del DEFMSAM.p99
if exist HBWORK.p99     del HBWORK.p99
if exist HBNWORK.p99     del HBNWORK.p99
if exist NHB.p99     del NHB.p99
copy FHSKIMS.p99     RHSKIMS.p99
copy HFSKIMS.p99     HRSKIMS.p99
C:\project\pb99p\pbexecs\mrgskim
C:\project\pb99p\pbexecs\nlogitho
if exist MODEIN.TEM del MODEIN.TEM
copy DISTRIB.OUT+NLOGIT.OUT DISTRIB.OUT
copy NLOGIT.OUT MODEPRE.OUT
ECHO ***** CREATE PRE ASSIGNMENT HTTAB FILE *****
ECHO - Airport Vehicle Trip File
datime
MTABLE
DEL TRNPL012.INS
COPY DISTRIB.OUT +TRNPL012.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL012.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
UTABLE
DEL TRNPL013.INS
COPY DISTRIB.OUT +TRNPL013.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL013.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
ECHO - Airport, Trucks and External trips are added to DA
MTABLE
DEL TRNPL014.INS
COPY DISTRIB.OUT +TRNPL014.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL014.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRNSPS
DEL TRNPL015.INS
COPY DISTRIB.OUT +TRNPL015.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL015.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
MTABLE
DEL TRNPL016.INS
COPY DISTRIB.OUT +TRNPL016.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL016.OUT
IF EXIST TRNPLN.ERR GOTO ERR

rem *------------------ UM insertion ----------------------*
rem * backup trip tables for reporting trip length dist.   *
rem if %2==t copy httab.tem hwyttab.p99
rem *------------------------------------------------------*

datime
```

```
EQUILB
DEL TRNPL017.INS
COPY DISTRIB.OUT +TRNPL017.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL017.OUT
IF EXIST TRNPLN.ERR GOTO ERR

rem ****************** feedback entry **********************
:FSU100
rem ********************************************************

datime
rem *------------------ UM insertion ----------------------*
rem * change Impedance=TIME2 in trnpl018.ins               *
rem * recreate DISTRIB.OUT after feedback                  *
    if %1==f copy trnpl018.int trnpl018.ins
rem *------------------------------------------------------*
SLCTSM
DEL TRNPL018.INS
if exist DISTRIB.OUT goto FSU101
ren TRNPL018.OUT DISTRIB.OUT
goto FSU102
:FSU101
COPY DISTRIB.OUT + TRNPL018.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
:FSU102
DEL TRNPL018.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
rem *------------------ UM insertion ----------------------*
rem * change Impedance=TIME2 in trnpl019.ins               *
rem * recreate DISTRIB.OUT after feedback                  *
    if %1==f copy trnpl019.int trnpl019.ins
rem *------------------------------------------------------*
SLCTSM
DEL TRNPL019.INS
COPY DISTRIB.OUT +TRNPL019.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL019.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
rem *------------------ UM insertion 07/02 ----------------*
rem *  mrgskim.exe reads fhskims.b99                       *
copy fskims.p99 fhskims.p99 >NUL
rem *------------------------------------------------------*
C:\project\pb99p\pbexecs\mrgskim
datime
datime
IZTIME
DEL TRNPL020.INS
COPY DISTRIB.OUT +TRNPL020.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL020.OUT
IF EXIST TRNPLN.ERR GOTO ERR
C:\project\pb99p\pbexecs\ttprepbx
UTABLE
DEL TRNPL021.INS
COPY DISTRIB.OUT +TRNPL021.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL021.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
ECHO ***** REPORT TRIP LENGTH BASED ON TIME *****
datime
ECHO ***Apply Weight to Free Flow TIME2 Skim***
datime
```

```
MTABLE
DEL TRNPL022.INS
COPY DISTRIB.OUT +TRNPL022.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL022.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
rem *--------------- UM modification ----------------------*
rem * change weight applied to free-flow skim to zeros     *
rem *------------------------------------------------------*

if %1==f copy trnpl023.int trnpl023.ins

UTABLE
DEL TRNPL023.INS
COPY DISTRIB.OUT +TRNPL023.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL023.OUT
IF EXIST TRNPLN.ERR GOTO ERR
ECHO ***Apply Weight to Constrained TIME2 Skim***
datime
MTABLE
DEL TRNPL024.INS
COPY DISTRIB.OUT +TRNPL024.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL024.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
rem *--------------- UM modification ----------------------*
rem * change weight applied to congested skim to ones      *
rem *------------------------------------------------------*

if %1==f copy trnpl025.int trnpl025.ins

UTABLE
DEL TRNPL025.INS
COPY DISTRIB.OUT +TRNPL025.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL025.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
ECHO ***Compute Weighted skims***
MTABLE
DEL TRNPL026.INS
COPY DISTRIB.OUT +TRNPL026.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL026.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
datime
datime
datime
ECHO del RHSKIMS.p99
ECHO ***** DISTRIBUTION USING WEIGHTED SKIM *****
IZTIME
DEL TRNPL027.INS
COPY DISTRIB.OUT +TRNPL027.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL027.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
if exist TT_T1.TEM del TT_T1.TEM
if exist TT_T2.TEM del TT_T2.TEM
if exist TT_T3.TEM del TT_T3.TEM
if exist TT_T4.TEM del TT_T4.TEM
if exist TT_T5.TEM del TT_T5.TEM
if exist TT_T6.TEM del TT_T6.TEM
if exist TT_T7.TEM del TT_T7.TEM
if exist TT_T8.TEM del TT_T8.TEM
if exist TT_T9.TEM del TT_T9.TEM
if exist TT_T10.TEM del TT_T10.TEM
if exist TT_T11.TEM del TT_T11.TEM
C:\project\pb99p\pbexecs\ttbtx TT_T1.TEM T1
C:\project\pb99p\pbexecs\ttbtx TT_T2.TEM T2
C:\project\pb99p\pbexecs\ttbtx TT_T3.TEM T3
C:\project\pb99p\pbexecs\ttbtx TT_T4.TEM T4
C:\project\pb99p\pbexecs\ttbtx TT_T5.TEM T5
C:\project\pb99p\pbexecs\ttbtx TT_T6.TEM T6
C:\project\pb99p\pbexecs\ttbtx TT_T7.TEM T7
C:\project\pb99p\pbexecs\ttbtx TT_T8.TEM T8
C:\project\pb99p\pbexecs\ttbtx TT_T9.TEM T9
C:\project\pb99p\pbexecs\ttbtx TT_T10.TEM T10
C:\project\pb99p\pbexecs\ttbtx TT_T11.TEM T11
datime
UTABLE
DEL TRNPL028.INS
COPY DISTRIB.OUT +TRNPL028.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL028.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
UTABLE
DEL TRNPL029.INS
COPY DISTRIB.OUT +TRNPL029.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL029.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
UTABLE
DEL TRNPL030.INS
COPY DISTRIB.OUT +TRNPL030.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL030.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
UTABLE
DEL TRNPL031.INS
COPY DISTRIB.OUT +TRNPL031.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL031.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
UTABLE
DEL TRNPL032.INS
COPY DISTRIB.OUT +TRNPL032.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL032.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
UTABLE
DEL TRNPL033.INS
COPY DISTRIB.OUT +TRNPL033.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL033.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
UTABLE
DEL TRNPL034.INS
COPY DISTRIB.OUT +TRNPL034.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
```

```
del lan.tem>NUL
DEL TRNPL034.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
UTABLE
DEL TRNPL035.INS
COPY DISTRIB.OUT +TRNPL035.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL035.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
UTABLE
DEL TRNPL036.INS
COPY DISTRIB.OUT +TRNPL036.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL036.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
UTABLE
DEL TRNPL037.INS
COPY DISTRIB.OUT +TRNPL037.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL037.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
UTABLE
DEL TRNPL038.INS
COPY DISTRIB.OUT +TRNPL038.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL038.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime

if %2==o goto fsu120
if not exist ptrips11.p99 goto fsu120

rem *--------------- UM modification ----------------------*
rem * Report trip length frequency                         *
rem * Run fdkff to retrieve GT inputs and combine with     *
rem *    GP, GA, and initial GF to make gmhist.dat         *
rem *    for generating F factors                          *
rem * Input: trnpl174.out, output: gtff.tem                *
copy trnpl174.int trnpl174.ins >NUL
copy trnpln6.con trnplnx.con >NUL
TLFREQ
del trnpl174.ins
if exist trnpln.err goto ERR
start/w fdkff
copy gmhist.da1 + gtff.tem gmhist.dat >NUL
rem *------------------------------------------------------*

rem *--------------- UM modification ----------------------*
rem * Insert calibration of gravity model (F factors)      *
rem * Create trnpl170.ins for gravity model calibration    *
rem *     Inpupt: gmhist.dat, Output: gmhist.99p           *
rem *------------------------------------------------------*
copy trnpl170.int trnpl170.ins >NUL
copy trnpln2.con trnplnx.con >NUL
GMODEL
del trnpl170.ins
copy distrib.out + trnpl170.out lan.tem >NUL
copy lan.tem distrib.out >NUL
del lan.tem>NUL
del trnpl170.out
if exist trnpln.err goto ERR
rem *--------------- UM modification ----------------------*
rem * Remark for calibration purpose, and                  *
rem * Modify trnpl039.ins to read gmhist.99p as input     *
rem *------------------------------------------------------*
copy trnpl039.int trnpl039.ins >NUL
copy trnpln3.con trnplnx.con
datime

:fsu120

COPY FF2.99p FF.TEM
GMODEL
DEL TRNPL039.INS
COPY DISTRIB.OUT +TRNPL039.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL039.OUT
IF EXIST TRNPLN.ERR GOTO ERR
DEL FF.TEM
datime
MTABLE
DEL TRNPL040.INS
COPY DISTRIB.OUT +TRNPL040.OUT lan.tem>NUL
copy lan.tem distrib.out>NUL
del lan.tem>NUL
DEL TRNPL040.OUT
IF EXIST TRNPLN.ERR GOTO ERR

rem *------------------ UM insertion ----------------------*
rem * Apply MSA on OD tables                               *

if %1==p goto FSU110

copy ptrips11.p99 ptrips1.dat >NUL
del trnpl17?.ins
start/w fdkod
if not exist ptrips0.ins goto FSU110
ren ptrips0.ins trnpl171.ins
ren ptrips1.ins trnpl172.ins
copy trnpln4.con trnplnx.con >NUL
datime
UTABLE
rem del trnpl171.ins
IF EXIST TRNPLN.ERR GOTO ERR
datime
UTABLE
rem del trnpl172.ins
IF EXIST TRNPLN.ERR GOTO ERR
copy trnpl173.int trnpl173.ins >NUL
datime
MTABLE
del trnpl173.ins
IF EXIST TRNPLN.ERR GOTO ERR
copy ptrips.dat ptrips11.p99 >NUL
copy ptrips.dat ptrips0.dat >NUL
copy trnpln5.con trnplnx.con >NUL
rem *------------------------------------------------------*

:FSU110

datime
if exist A1DECK.TEM del A1DECK.TEM
datime >> time.log
ECHO ***** DISTRIB ***** -- END
if exist TRNPLN.ERR goto err
echo off
ECHO OFF
ECHO ***** TNET ***** -- START
datime >> time.log
datime
if exist TNET.OUT del TNET.OUT
copy TPARMAM.99p    TSYSIN.TEM >NUL
copy TSYSIN.TEM+OPTLNKAM.99p    TSYSIN.TEM >NUL
copy TSYSIN.TEM+SDLAYUPD.99p    TSYSIN.TEM >NUL
copy TROUTEAM.99p    TROUTE.TEM >NUL
copy TROUTE.TEM+WALKNETS.99p    TROUTE.TEM >NUL
copy HRLDXYX.p99     HNET.TEM >NUL
rem *------------------ UM insertion 07/02 ----------------*
rem * inet.exe would not execute if inet.err, inet.out     *
rem *   or disk* exist                                     *
rem * Input files: tsysin.tem, hnet.tem                    *
    if exist inet.* del inet.* >NUL
    if exist disk* del disk* >NUL
rem *------------------------------------------------------*
inet
copy HUDNET.TEM HUDNETAM.99p    >NUL
ren INET.OUT TNET.OUT
if exist INET.ERR goto TNET1
goto TNET2
:TNET1
if exist INET.ERR del INET.ERR
goto err
:TNET2
inetrep
copy TNET.OUT+INETREP.OUT TNET.OUT >NUL
if exist INETREP.OUT del INETREP.OUT
if exist PERIOD1 del PERIOD1
echo>PERIOD1
copy HSKIMS.p99     RHSKIMS.p99     >NUL
copy XY.99p    XY.SAV >NUL
copy XY.SAV+NEWXY.99p    XY.TEM >NUL
if exist SIDEAM.p99     del SIDEAM.p99
if exist AUTDELAM.p99     del AUTDELAM.p99
if exist SIDECON.OUT del SIDECON.OUT
C:\project\pb99p\pbexecs\sidecon
copy TNET.OUT+SIDECON.OUT TNET.OUT >NUL
del SIDECON.OUT
if exist SIDECON.ERR goto err
if exist WLKHWYAM.p99     del WLKHWYAM.p99
if exist WLKNEWAM.p99     del WLKNEWAM.p99
if exist CODWAM.p99     del CODWAM.p99
copy SIDEAM.p99     WALKIN.TEM >NUL
copy WALKIN.TEM+HUDNETAM.99p    WALKIN.TEM >NUL
if exist WALKCON.OUT del WALKCON.OUT
if exist WALKCON.ER* del WALKCON.ER*
C:\project\pb99p\pbexecs\walkcon
copy TNET.OUT+WALKCON.OUT TNET.OUT >NUL
del WALKCON.OUT
del WALKIN.TEM
if exist WALKCON.ER1 goto err
if exist AUTOAM.p99     del AUTOAM.p99
if exist STATAM.p99     del STATAM.p99
if exist DUMXYAM.p99     del DUMXYAM.p99
if exist AUTOCON.OUT del AUTOCON.OUT
C:\project\pb99p\pbexecs\autocon
copy TNET.OUT+AUTOCON.OUT TNET.OUT >NUL
copy XY.SAV XY.99p    >NUL
copy STATAM.p99     STATAM.99p
del AUTOCON.OUT
if exist AUTDELAM.p99     del AUTDELAM.p99
if exist RHSKIMS.p99     del RHSKIMS.p99
del PERIOD1
datime
ECHO ***AM Tnet Construction***
copy XY.99p    XY.SAV
rem *---------------- UM revision ------------------------*
rem * not recognize 'NEWXY.&ALTIN'                        *
rem copy XY.SAV+NEWXY.&ALTIN+DUMXYAM.p99     XY.99p    >NUL
    copy XY.SAV+NEWXY.99p+DUMXYAM.p99     XY.99p    >NUL
rem *-----------------------------------------------------*
copy WLKHWYAM.p99     HUDNET.TEM >NUL
rem *----------------- UM revision ------------------------*
rem * copy A + B C to avoid file break during merge        *
    copy HUDNET.TEM+WLKNEWAM.p99     lan.TEM >NUL
    copy lan.TEM+SIDEAM.p99     HUDNET.TEM >NUL
    copy HUDNET.TEM+AUTOAM.p99     lan.TEM >NUL
    copy lan.TEM+HUDNETAM.99p    HUDNET.TEM >NUL
    copy HUDNET.TEM HUDNETAM.SAV
    del lan.tem
rem * HUDNET.exe                                           *
rem * Input files: trnplnx.con, trnplxxx.ins, hudnet.tem   *
rem * Output files: trnplnx.con, trnplxxx.out.             *
rem * Delete hudnet.tem                                    *
rem *------------------------------------------------------*
HUDNET
DEL TRNPL041.INS
IF EXIST TNET.OUT     GOTO FSU011
REN TRNPL041.OUT TNET.OUT
GOTO FSU012
:FSU011
COPY TNET.OUT    +TRNPL041.OUT >NUL
DEL TRNPL041.OUT
:FSU012
IF EXIST TRNPLN.ERR GOTO ERR
copy XY.SAV XY.99p
if exist XY.TEM del XY.TEM
if exist TROUTE.TEM del TROUTE.TEM
if exist TSYSIN.TEM del TSYSIN.TEM
if exist HNET.TEM del HNET.TEM
if exist SIDEAM.p99     del SIDEAM.p99
if exist WLKNEWAM.p99     del WLKNEWAM.p99
if exist WLKHWYAM.p99     del WLKHWYAM.p99
if exist AUTOAM.p99     del AUTOAM.p99
if exist DUMXYAM.p99     del DUMXYAM.p99
if exist HUDNETAM.SAV del HUDNETAM.SAV
if exist HUDNETAM.99p    del HUDNETAM.99p
if exist WALKCON.ER? del WALKCON.ER?
datime
ECHO  ***Midday Construction***
datime
copy TPARMMD.99p    TSYSIN.TEM >NUL
copy TSYSIN.TEM+OPTLNKMD.99p    TSYSIN.TEM >NUL
copy TSYSIN.TEM+SDLAYUPD.99p    TSYSIN.TEM >NUL
copy TROUTEMD.99p    TROUTE.TEM >NUL
copy TROUTE.TEM+WALKNETS.99p    TROUTE.TEM >NUL
copy HRLDXYX.p99     HNET.TEM >NUL
rem *---------------- UM insertion ------------------------*
rem * inet.exe would not execute if inet.err, inet.out     *
rem *   or disk* exist                                     *
rem * Input files: tsysin.tem, hnet.tem                    *
    if exist inet.* del inet.* >NUL
    if exist disk* del disk* >NUL
rem *------------------------------------------------------*
inet
copy HUDNET.TEM HUDNETMD.99p    >NUL
copy TNET.OUT+INET.OUT TNET.OUT >NUL
if exist INET.OUT del INET.OUT
if exist INET.ERR goto TNET3
goto TNET4
:TNET3
if exist INET.ERR del INET.ERR
goto err
:TNET4
inetrep
copy TNET.OUT+INETREP.OUT TNET.OUT
if exist INETREP.OUT del INETREP.OUT
if exist PERIOD2 del PERIOD2
echo>PERIOD2
copy HSKIMS.p99    FHSKIMS.p99     >NUL
copy XY.99p    XY.SAV >NUL
copy XY.SAV+NEWXY.99p    XY.TEM >NUL
if exist SIDEMD.p99     del SIDEMD.p99
if exist AUTDELMD.p99     del AUTDELMD.p99
if exist SIDECON.OUT del SIDECON.OUT
C:\project\pb99p\pbexecs\sidecon
```

```
copy TNET.OUT+SIDECON.OUT TNET.OUT >NUL
del SIDECON.OUT
if exist SIDECON.ERR goto err
if exist WLKHWYMD.p99     del WLKHWYMD.p99
if exist WLKNEWMD.p99     del WLKNEWMD.p99
if exist CODWMD.p99     del CODWMD.p99
copy SIDEMD.p99     WALKIN.TEM >NUL
copy WALKIN.TEM+HUDNETMD.99p    WALKIN.TEM >NUL
if exist WALKCON.OUT del WALKCON.OUT
if exist WALKCON.ER* del WALKCON.ER*
C:\project\pb99p\pbexecs\walkcon
copy TNET.OUT+WALKCON.OUT TNET.OUT >NUL
del WALKCON.OUT
del WALKIN.TEM
if exist WALKCON.ER1 goto err
if exist AUTOMD.p99     del AUTOMD.p99
if exist STATMD.p99     del STATMD.p99
if exist DUMXYMD.p99     del DUMXYMD.p99
if exist AUTOCON.OUT del AUTOCON.OUT
C:\project\pb99p\pbexecs\autocon
copy TNET.OUT+AUTOCON.OUT TNET.OUT >NUL
copy XY.SAV XY.99p    >NUL
copy STATMD.p99     STATMD.99p
del AUTOCON.OUT
if exist AUTDELMD.p99     del AUTDELMD.p99
if exist FHSKIMS.p99     del FHSKIMS.p99
del PERIOD2
datime
ECHO ***MD Tnet Construction***
copy XY.99p    XY.SAV
rem *----------------- UM revision ------------------------*
rem * not recognize 'NEWXY.&ALTIN'                         *
rem copy XY.SAV+NEWXY.&ALTIN+DUMXYMD.p99     XY.99p    >NUL
    copy XY.SAV+NEWXY.99p+DUMXYMD.p99     XY.99p    >NUL
rem *------------------------------------------------------*
copy WLKHWYMD.p99     HUDNET.TEM >NUL
rem *----------------- UM revision ------------------------*
rem * copy A + B C to avoid file break during merge        *
    copy HUDNET.TEM+WLKNEWMD.p99     lan.TEM >NUL
    copy lan.TEM+SIDEMD.p99     HUDNET.TEM >NUL
    copy HUDNET.TEM+AUTOMD.p99     lan.TEM >NUL
    copy lan.TEM+HUDNETMD.99p    HUDNET.TEM >NUL
    copy HUDNET.TEM HUDNETMD.SAV
    del lan.tem
rem * HUDNET.exe                                           *
rem * Input files: trnplnx.con, trnplxxx.ins, hudnet.tem   *
rem * Output files: trnplnx.con, trnplxxx.out.             *
rem * Delete hudnet.tem                                    *
rem *------------------------------------------------------*
HUDNET
DEL TRNPL042.INS
COPY TNET.OUT    +TRNPL042.OUT >NUL
DEL TRNPL042.OUT
IF EXIST TRNPLN.ERR GOTO ERR
copy XY.SAV XY.99p
if exist XY.TEM del XY.TEM
if exist TROUTE.TEM del TROUTE.TEM
if exist TSYSIN.TEM del TSYSIN.TEM
if exist HNET.TEM del HNET.TEM
if exist SIDEMD.p99     del SIDEMD.p99
if exist WLKNEWMD.p99     del WLKNEWMD.p99
if exist WLKHWYMD.p99     del WLKHWYMD.p99
if exist AUTOMD.p99     del AUTOMD.p99
if exist DUMXYMD.p99     del DUMXYMD.p99
if exist HUDNETMD.SAV del HUDNETMD.SAV
if exist HUDNETMD.99p    del HUDNETMD.99p
if exist WALKCON.ER? del WALKCON.ER?
datime
datime >> time.log
ECHO ***** TNET ***** -- END
if exist TRNPLN.ERR goto err
```

```
echo off
ECHO OFF
ECHO ***** TPATH ***** -- START
ECHO **** PEAK PERIOD **** -- START
datime >> time.log
datime
if exist TPATH.OUT del TPATH.OUT
if exist STATAM.p99     del STATAM.p99
if exist STATMD.p99     del STATMD.p99
if exist SFARE.TEM del SFARE.TEM
if exist TFARE?.UPD del TFARE?.UPD
if exist TFARE?.TEM del TFARE?.TEM
if exist TSKIM?.UPD del TSKIM?.UPD
if exist TSKIM?.TEM del TSKIM?.TEM
C:\project\pb99p\pbexecs\statfare
HUDPATH
DEL TRNPL043.INS
IF EXIST TPATH.OUT    GOTO FSU013
REN TRNPL043.OUT TPATH.OUT
GOTO FSU014
:FSU013
COPY TPATH.OUT   +TRNPL043.OUT >NUL
DEL TRNPL043.OUT
:FSU014
IF EXIST TRNPLN.ERR GOTO ERR
datime
HUDPSUM
DEL TRNPL044.INS
COPY TPATH.OUT   +TRNPL044.OUT >NUL
DEL TRNPL044.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRFARE
DEL TRNPL045.INS
COPY TPATH.OUT   +TRNPL045.OUT >NUL
DEL TRNPL045.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
ren TSKIMAM1.p99     TSKIM1.TEM
ren TFAREAM1.p99     TFARE1.TEM
C:\project\pb99p\pbexecs\pathmod
datime
datime
datime
datime
ren TSKIM1.UPD TSKIMAM1.p99
ren TFARE1.UPD TFAREAM1.p99
if exist TSKIM1.TEM del TSKIM1.TEM
if exist TFARE1.TEM del TFARE1.TEM
datime
HUDPATH
DEL TRNPL046.INS
COPY TPATH.OUT   +TRNPL046.OUT >NUL
DEL TRNPL046.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
HUDPSUM
DEL TRNPL047.INS
COPY TPATH.OUT   +TRNPL047.OUT >NUL
DEL TRNPL047.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRFARE
DEL TRNPL048.INS
COPY TPATH.OUT   +TRNPL048.OUT >NUL
DEL TRNPL048.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
ren TSKIMAM2.p99     TSKIM2.TEM
ren TFAREAM2.p99     TFARE2.TEM
C:\project\pb99p\pbexecs\pathmod
```

```
datime
datime
datime
datime
ren TSKIM2.UPD TSKIMAM2.p99
ren TFARE2.UPD TFAREAM2.p99
if exist TSKIM2.TEM del TSKIM2.TEM
if exist TFARE2.TEM del TFARE2.TEM
datime
HUDPATH
DEL TRNPL049.INS
COPY TPATH.OUT   +TRNPL049.OUT >NUL
DEL TRNPL049.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
HUDPSUM
DEL TRNPL050.INS
COPY TPATH.OUT   +TRNPL050.OUT >NUL
DEL TRNPL050.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRFARE
DEL TRNPL051.INS
COPY TPATH.OUT   +TRNPL051.OUT >NUL
DEL TRNPL051.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
ren TSKIMAM3.p99     TSKIM3.TEM
ren TFAREAM3.p99     TFARE3.TEM
C:\project\pb99p\pbexecs\pathmod
datime
datime
datime
datime
ren TSKIM3.UPD TSKIMAM3.p99
ren TFARE3.UPD TFAREAM3.p99
if exist TSKIM3.TEM del TSKIM3.TEM
if exist TFARE3.TEM del TFARE3.TEM
datime
HUDPATH
DEL TRNPL052.INS
COPY TPATH.OUT   +TRNPL052.OUT >NUL
DEL TRNPL052.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
HUDPSUM
DEL TRNPL053.INS
COPY TPATH.OUT   +TRNPL053.OUT >NUL
DEL TRNPL053.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRFARE
DEL TRNPL054.INS
COPY TPATH.OUT   +TRNPL054.OUT >NUL
DEL TRNPL054.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
ren TSKIMAM4.p99     TSKIM4.TEM
ren TFAREAM4.p99     TFARE4.TEM
C:\project\pb99p\pbexecs\pathmod
datime
datime
datime
datime
ren TSKIM4.UPD TSKIMAM4.p99
ren TFARE4.UPD TFAREAM4.p99
if exist TSKIM4.TEM del TSKIM4.TEM
if exist TFARE4.TEM del TFARE4.TEM
datime
HUDPATH
DEL TRNPL055.INS
```

```
COPY TPATH.OUT   +TRNPL055.OUT >NUL
DEL TRNPL055.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
HUDPSUM
DEL TRNPL056.INS
COPY TPATH.OUT   +TRNPL056.OUT >NUL
DEL TRNPL056.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRFARE
DEL TRNPL057.INS
COPY TPATH.OUT   +TRNPL057.OUT >NUL
DEL TRNPL057.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
ren TSKIMAM5.p99     TSKIM5.TEM
ren TFAREAM5.p99     TFARE5.TEM
C:\project\pb99p\pbexecs\pathmod
datime
datime
datime
datime
ren TSKIM5.UPD TSKIMAM5.p99
ren TFARE5.UPD TFAREAM5.p99
if exist TSKIM5.TEM del TSKIM5.TEM
if exist TFARE5.TEM del TFARE5.TEM
datime
HUDPATH
DEL TRNPL058.INS
COPY TPATH.OUT   +TRNPL058.OUT >NUL
DEL TRNPL058.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
HUDPSUM
DEL TRNPL059.INS
COPY TPATH.OUT   +TRNPL059.OUT >NUL
DEL TRNPL059.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRFARE
DEL TRNPL060.INS
COPY TPATH.OUT   +TRNPL060.OUT >NUL
DEL TRNPL060.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
ren TSKIMAM6.p99     TSKIM6.TEM
ren TFAREAM6.p99     TFARE6.TEM
C:\project\pb99p\pbexecs\pathmod
datime
datime
datime
datime
ren TSKIM6.UPD TSKIMAM6.p99
ren TFARE6.UPD TFAREAM6.p99
if exist TSKIM6.TEM del TSKIM6.TEM
if exist TFARE6.TEM del TFARE6.TEM
datime
HUDPATH
DEL TRNPL061.INS
COPY TPATH.OUT   +TRNPL061.OUT >NUL
DEL TRNPL061.OUT
IF EXIST TRNPLN.ERR GOTO ERR
HUDPSUM
DEL TRNPL062.INS
COPY TPATH.OUT   +TRNPL062.OUT >NUL
DEL TRNPL062.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRFARE
DEL TRNPL063.INS
```

```
COPY TPATH.OUT   +TRNPL063.OUT >NUL
DEL TRNPL063.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
ren TSKIMAM7.p99     TSKIM7.TEM
ren TFAREAM7.p99     TFARE7.TEM
C:\project\pb99p\pbexecs\pathmod
datime
datime
datime
datime
ren TSKIM7.UPD TSKIMAM7.p99
ren TFARE7.UPD TFAREAM7.p99
if exist TSKIM7.TEM del TSKIM7.TEM
if exist TFARE7.TEM del TFARE7.TEM
datime
ECHO -- STATION IDENTIFICATION - AM - ALT 99p
TRSTOS
DEL TRNPL064.INS
COPY TPATH.OUT   +TRNPL064.OUT >NUL
DEL TRNPL064.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRSTOS
DEL TRNPL065.INS
COPY TPATH.OUT   +TRNPL065.OUT >NUL
DEL TRNPL065.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRSTOS
DEL TRNPL066.INS
COPY TPATH.OUT   +TRNPL066.OUT >NUL
DEL TRNPL066.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRSTOS
DEL TRNPL067.INS
COPY TPATH.OUT   +TRNPL067.OUT >NUL
DEL TRNPL067.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRSTOS
DEL TRNPL068.INS
COPY TPATH.OUT   +TRNPL068.OUT >NUL
DEL TRNPL068.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRSTOS
DEL TRNPL069.INS
COPY TPATH.OUT   +TRNPL069.OUT >NUL
DEL TRNPL069.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
MTABLE
DEL TRNPL070.INS
COPY TPATH.OUT   +TRNPL070.OUT >NUL
DEL TRNPL070.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
ECHO **** MIDDAY PERIOD *** -- START
datime
HUDPATH
DEL TRNPL071.INS
COPY TPATH.OUT   +TRNPL071.OUT >NUL
DEL TRNPL071.OUT
IF EXIST TRNPLN.ERR GOTO ERR
HUDPSUM
DEL TRNPL072.INS
COPY TPATH.OUT   +TRNPL072.OUT >NUL
DEL TRNPL072.OUT
IF EXIST TRNPLN.ERR GOTO ERR
```

```
datime
TRFARE
DEL TRNPL073.INS
COPY TPATH.OUT   +TRNPL073.OUT >NUL
DEL TRNPL073.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
ren TSKIMMD1.p99     TSKIM1.TEM
ren TFAREMD1.p99     TFARE1.TEM
C:\project\pb99p\pbexecs\pathmod
datime
datime
datime
datime
ren TSKIM1.UPD TSKIMMD1.p99
ren TFARE1.UPD TFAREMD1.p99
if exist TSKIM1.TEM del TSKIM1.TEM
if exist TFARE1.TEM del TFARE1.TEM
datime
HUDPATH
DEL TRNPL074.INS
COPY TPATH.OUT   +TRNPL074.OUT >NUL
DEL TRNPL074.OUT
IF EXIST TRNPLN.ERR GOTO ERR
HUDPSUM
DEL TRNPL075.INS
COPY TPATH.OUT   +TRNPL075.OUT >NUL
DEL TRNPL075.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRFARE
DEL TRNPL076.INS
COPY TPATH.OUT   +TRNPL076.OUT >NUL
DEL TRNPL076.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
ren TSKIMMD2.p99     TSKIM2.TEM
ren TFAREMD2.p99     TFARE2.TEM
C:\project\pb99p\pbexecs\pathmod
datime
datime
datime
datime
ren TSKIM2.UPD TSKIMMD2.p99
ren TFARE2.UPD TFAREMD2.p99
if exist TSKIM2.TEM del TSKIM2.TEM
if exist TFARE2.TEM del TFARE2.TEM
datime
HUDPATH
DEL TRNPL077.INS
COPY TPATH.OUT   +TRNPL077.OUT >NUL
DEL TRNPL077.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
HUDPSUM
DEL TRNPL078.INS
COPY TPATH.OUT   +TRNPL078.OUT >NUL
DEL TRNPL078.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRFARE
DEL TRNPL079.INS
COPY TPATH.OUT   +TRNPL079.OUT >NUL
DEL TRNPL079.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
ren TSKIMMD3.p99     TSKIM3.TEM
ren TFAREMD3.p99     TFARE3.TEM
C:\project\pb99p\pbexecs\pathmod
datime
datime
```

```
datime
datime
ren TSKIM3.UPD TSKIMMD3.p99
ren TFARE3.UPD TFAREMD3.p99
if exist TSKIM3.TEM del TSKIM3.TEM
if exist TFARE3.TEM del TFARE3.TEM
datime
HUDPATH
DEL TRNPL080.INS
COPY TPATH.OUT   +TRNPL080.OUT >NUL
DEL TRNPL080.OUT
IF EXIST TRNPLN.ERR GOTO ERR
HUDPSUM
DEL TRNPL081.INS
COPY TPATH.OUT   +TRNPL081.OUT >NUL
DEL TRNPL081.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRFARE
DEL TRNPL082.INS
COPY TPATH.OUT   +TRNPL082.OUT >NUL
DEL TRNPL082.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
ren TSKIMMD4.p99     TSKIM4.TEM
ren TFAREMD4.p99     TFARE4.TEM
C:\project\pb99p\pbexecs\pathmod
datime
datime
datime
datime
ren TSKIM4.UPD TSKIMMD4.p99
ren TFARE4.UPD TFAREMD4.p99
if exist TSKIM4.TEM del TSKIM4.TEM
if exist TFARE4.TEM del TFARE4.TEM
datime
HUDPATH
DEL TRNPL083.INS
COPY TPATH.OUT   +TRNPL083.OUT >NUL
DEL TRNPL083.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
HUDPSUM
DEL TRNPL084.INS
COPY TPATH.OUT   +TRNPL084.OUT >NUL
DEL TRNPL084.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRFARE
DEL TRNPL085.INS
COPY TPATH.OUT   +TRNPL085.OUT >NUL
DEL TRNPL085.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
ren TSKIMMD5.p99     TSKIM5.TEM
ren TFAREMD5.p99     TFARE5.TEM
C:\project\pb99p\pbexecs\pathmod
datime
datime
datime
datime
ren TSKIM5.UPD TSKIMMD5.p99
ren TFARE5.UPD TFAREMD5.p99
if exist TSKIM5.TEM del TSKIM5.TEM
if exist TFARE5.TEM del TFARE5.TEM
datime
HUDPATH
DEL TRNPL086.INS
COPY TPATH.OUT   +TRNPL086.OUT >NUL
DEL TRNPL086.OUT
IF EXIST TRNPLN.ERR GOTO ERR
```

```
HUDPSUM
DEL TRNPL087.INS
COPY TPATH.OUT   +TRNPL087.OUT >NUL
DEL TRNPL087.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRFARE
DEL TRNPL088.INS
COPY TPATH.OUT   +TRNPL088.OUT >NUL
DEL TRNPL088.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
ren TSKIMMD6.p99     TSKIM6.TEM
ren TFAREMD6.p99     TFARE6.TEM
C:\project\pb99p\pbexecs\pathmod
datime
datime
datime
datime
ren TSKIM6.UPD TSKIMMD6.p99
ren TFARE6.UPD TFAREMD6.p99
if exist TSKIM6.TEM del TSKIM6.TEM
if exist TFARE6.TEM del TFARE6.TEM
datime
HUDPATH
DEL TRNPL089.INS
COPY TPATH.OUT   +TRNPL089.OUT >NUL
DEL TRNPL089.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
HUDPSUM
DEL TRNPL090.INS
COPY TPATH.OUT   +TRNPL090.OUT >NUL
DEL TRNPL090.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRFARE
DEL TRNPL091.INS
COPY TPATH.OUT   +TRNPL091.OUT >NUL
DEL TRNPL091.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
ren TSKIMMD7.p99     TSKIM7.TEM
ren TFAREMD7.p99     TFARE7.TEM
C:\project\pb99p\pbexecs\pathmod
datime
datime
datime
datime
ren TSKIM7.UPD TSKIMMD7.p99
ren TFARE7.UPD TFAREMD7.p99
if exist TSKIM7.TEM del TSKIM7.TEM
if exist TFARE7.TEM del TFARE7.TEM
datime
ECHO -- STATION IDENTIFICATION - MD - ALT 99p
TRSTOS
DEL TRNPL092.INS
COPY TPATH.OUT   +TRNPL092.OUT >NUL
DEL TRNPL092.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRSTOS
DEL TRNPL093.INS
COPY TPATH.OUT   +TRNPL093.OUT >NUL
DEL TRNPL093.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRSTOS
DEL TRNPL094.INS
COPY TPATH.OUT   +TRNPL094.OUT >NUL
DEL TRNPL094.OUT
```

```
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRSTOS
DEL TRNPL095.INS
COPY TPATH.OUT   +TRNPL095.OUT >NUL
DEL TRNPL095.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRSTOS
DEL TRNPL096.INS
COPY TPATH.OUT   +TRNPL096.OUT >NUL
DEL TRNPL096.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRSTOS
DEL TRNPL097.INS
COPY TPATH.OUT   +TRNPL097.OUT >NUL
DEL TRNPL097.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
MTABLE
DEL TRNPL098.INS
COPY TPATH.OUT   +TRNPL098.OUT >NUL
DEL TRNPL098.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
if exist SFARE.TEM del SFARE.TEM
if exist STATAM.99p    del STATAM.99p
if exist STATMD.99p    del STATMD.99p
datime >> time.log
ECHO ***** TPATH ***** -- END
if exist TRNPLN.ERR goto err
echo off
ECHO OFF
ECHO ***** MODE ***** -- START
datime >> time.log
datime
if exist MODE.OUT del MODE.OUT
ECHO ***** COMBINE TRIP PURPOSES FOR MODE CHOICE *****
if exist MODEIN.TEM del MODEIN.TEM
MTABLE
DEL TRNPL099.INS
IF EXIST MODE.OUT     GOTO FSU015
REN TRNPL099.OUT MODE.OUT
GOTO FSU016
:FSU015
COPY MODE.OUT    +TRNPL099.OUT >NUL
DEL TRNPL099.OUT
:FSU016
IF EXIST TRNPLN.ERR GOTO ERR
datime
ECHO ** FULL MODE CHOICE PROGRAM WITH REVISED TRIP TABLE**
copy A1DECK.p99     A1DECK.TEM
if exist NLOGIT.OUT del NLOGIT.OUT
if exist MODE.ERR del MODE.ERR
if exist DEFMSAM.p99     del DEFMSAM.p99
if exist HBWORK.p99     del HBWORK.p99
if exist HBNWORK.p99     del HBNWORK.p99
if exist NHB.p99     del NHB.p99
C:\project\pb99p\pbexecs\nlogitfl
copy MODE.OUT+NLOGIT.OUT MODE.OUT
if exist NLOGIT.OUT del NLOGIT.OUT
copy MODEIN.TEM PTRIPS3P.p99
if exist MODEIN.TEM del MODEIN.TEM
datime
REM: Zero -ve values (if any) of hwy mode tables of
REM: HBWORK, HBNWORK and NHB
UTABLE
DEL TRNPL100.INS
COPY MODE.OUT    +TRNPL100.OUT >NUL
DEL TRNPL100.OUT
```

```
IF EXIST TRNPLN.ERR GOTO ERR
datime
UTABLE
DEL TRNPL101.INS
COPY MODE.OUT    +TRNPL101.OUT >NUL
DEL TRNPL101.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
UTABLE
DEL TRNPL102.INS
COPY MODE.OUT    +TRNPL102.OUT >NUL
DEL TRNPL102.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
datime
ECHO ***** MODE ***** -- END
ECHO ***** MODE TRIP TABLE ***** -- START
ECHO ***** OBTAIN AIRPORT DA and SR2 and SR3 TABLES *****
datime
MTABLE
DEL TRNPL103.INS
COPY MODE.OUT    +TRNPL103.OUT >NUL
DEL TRNPL103.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
UTABLE
DEL TRNPL104.INS
COPY MODE.OUT    +TRNPL104.OUT >NUL
DEL TRNPL104.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
ECHO ***** OBTAIN EE AND EI TRUCK TABLES *****
FRATAR
DEL TRNPL105.INS
COPY MODE.OUT    +TRNPL105.OUT >NUL
DEL TRNPL105.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
MTABLE
DEL TRNPL106.INS
COPY MODE.OUT    +TRNPL106.OUT >NUL
DEL TRNPL106.OUT
IF EXIST TRNPLN.ERR GOTO ERR
UTABLE
DEL TRNPL107.INS
COPY MODE.OUT    +TRNPL107.OUT >NUL
DEL TRNPL107.OUT
IF EXIST TRNPLN.ERR GOTO ERR
MTABLE
DEL TRNPL108.INS
COPY MODE.OUT    +TRNPL108.OUT >NUL
DEL TRNPL108.OUT
IF EXIST TRNPLN.ERR GOTO ERR
UTABLE
DEL TRNPL109.INS
COPY MODE.OUT    +TRNPL109.OUT >NUL
DEL TRNPL109.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
ECHO ***** Extract Truck II and IE Tables *****
MTABLE
DEL TRNPL110.INS
COPY MODE.OUT    +TRNPL110.OUT >NUL
DEL TRNPL110.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
ECHO ***** CREATE HTTAB (HWYTTAB) FILE *****
MTABLE
DEL TRNPL111.INS
COPY MODE.OUT    +TRNPL111.OUT >NUL
DEL TRNPL111.OUT
```

```
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRNSPS
DEL TRNPL112.INS
COPY MODE.OUT    +TRNPL112.OUT >NUL
DEL TRNPL112.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
echo ***** CREATE TOTAL HTTAB FILE FOR EMIS*****
MTABLE
DEL TRNPL113.INS
COPY MODE.OUT    +TRNPL113.OUT >NUL
DEL TRNPL113.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
echo ***** CREATE TRIP TABLES FOR TRANSIT ASSIGNMENT*****
MTABLE
DEL TRNPL114.INS
COPY MODE.OUT    +TRNPL114.OUT >NUL
DEL TRNPL114.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
MTABLE
DEL TRNPL115.INS
COPY MODE.OUT    +TRNPL115.OUT >NUL
DEL TRNPL115.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
if exist A1DECK.TEM del A1DECK.TEM
datime >> time.log
ECHO ***** MODE TRIP TABLE ***** -- END
if exist TRNPLN.ERR goto err
echo off
ECHO OFF
ECHO **** HASSIGN **** -- START
datime >> time.log
datime
if exist HASSIGN.OUT del HASSIGN.OUT
MTABLE
DEL TRNPL116.INS
IF EXIST HASSIGN.OUT  GOTO FSU017
REN TRNPL116.OUT HASSIGN.OUT
GOTO FSU018
:FSU017
rem *----------------- UM insertion -----------------------*
rem * Write to a temporary file to prevent file break      *
rem COPY HASSIGN.OUT +TRNPL116.OUT >NUL
COPY HASSIGN.OUT + TRNPL116.OUT lan.tem >NUL
copy lan.tem hassign.out >NUL
del lan.tem >NUL
rem *------------------------------------------------------*
DEL TRNPL116.OUT
:FSU018
IF EXIST TRNPLN.ERR GOTO ERR
datime
ECHO --Truck Assignment--
EQUILB
DEL TRNPL117.INS
rem *----------------- UM insertion -----------------------*
rem * Write to a temporary file to prevent file break      *
rem * remove VC cap, change Equilibrium Iterations = 50    *
rem * change EPS = 0.01                                    *
rem COPY HASSIGN.OUT +TRNPL117.OUT >NUL
COPY HASSIGN.OUT + TRNPL117.OUT lan.tem >NUL
copy lan.tem hassign.out >NUL
del lan.tem >NUL
rem *------------------------------------------------------*
DEL TRNPL117.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
ECHO --Create Card Image of loaded truck netwok--
```

```
if exist TRKVOL.TEM del TRKVOL.TEM
if exist NXY.TRK del NXY.TRK
netcard<NC.SYN
datime
ECHO --Create Preloaded Truck File--
if exist TRKVOL.PRE del TRKVOL.PRE
C:\project\pb99p\pbexecs\getprel
datime
ECHO --Car and Truck Assignment--
rem *----------------- UM insertion -----------------------*
rem * Write to a temporary file to prevent file break      *
rem * remove VC cap, change Equilibrium Iterations = 50    *
rem * change EPS = 0.01                                    *
rem *------------------------------------------------------*
EQUILB
DEL TRNPL118.INS
COPY HASSIGN.OUT +TRNPL118.OUT lan.tem>NUL
copy lan.tem hassign.out>NUL
del lan.tem
DEL TRNPL118.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
datime
if exist TRKVOL.TEM del TRKVOL.TEM
datime >> time.log
ECHO **** HASSIGN **** -- END
if exist TRNPLN.ERR goto err
echo off
ECHO OFF
ECHO ***** HEVAL ***** -- START
datime >> time.log
datime
if exist HEVAL.OUT del HEVAL.OUT
heval
copy HRLDXY.p99     HRLDXY0.TEM
copy HEVAL.OUT HEVAL.REG
datime
copy HRLDXY.TRK HRLDXY.p99
heval
copy HEVAL.OUT TRKEVL.OUT
copy HRLDXY0.TEM HRLDXY.p99
if exist HRLDXY0.TEM del HRLDXY0.TEM
copy HEVAL.REG HEVAL.OUT
if exist HEVAL.REG del HEVAL.REG
datime
rmse<rmse.syn
datime >> time.log
ECHO ***** HEVAL ***** -- END
if exist TRNPLN.ERR goto err
echo off
ECHO OFF
ECHO ***** TASSIGN ***** -- START
ECHO ***AM TRANSIT ASSIGNMENT***
datime >> time.log
datime
if exist TASSIGN.OUT del TASSIGN.OUT
if exist TLEGSALB.p99     del TLEGSALB.p99
if exist TLEGSMLB.p99     del TLEGSMLB.p99
if exist TLEGAWPM.p99     del TLEGAWPM.p99
if exist TLEGAAPR.p99     del TLEGAAPR.p99
if exist TLEGAAKR.p99     del TLEGAAKR.p99
if exist TLEGSAMW.p99     del TLEGSAMW.p99
if exist TLEGSAMA.p99     del TLEGSAMA.p99
if exist TLEGSAM.p99     del TLEGSAM.p99
if exist TLEGMWPM.p99     del TLEGMWPM.p99
if exist TLEGMAPR.p99     del TLEGMAPR.p99
if exist TLEGMAKR.p99     del TLEGMAKR.p99
if exist TLEGSMDW.p99     del TLEGSMDW.p99
if exist TLEGSMDA.p99     del TLEGSMDA.p99
if exist TLEGSMD.p99     del TLEGSMD.p99
rem
```

```
ECHO -- PEAK TRANSIT ASSIGNMENT
rem
datime
UTABLE
DEL TRNPL119.INS
IF EXIST TASSIGN.OUT  GOTO FSU021
REN TRNPL119.OUT TASSIGN.OUT
GOTO FSU022
:FSU021
COPY TASSIGN.OUT +TRNPL119.OUT >NUL
DEL TRNPL119.OUT
:FSU022
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRLOAD
DEL TRNPL120.INS
COPY TASSIGN.OUT +TRNPL120.OUT >NUL
DEL TRNPL120.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRLOAD
DEL TRNPL121.INS
COPY TASSIGN.OUT +TRNPL121.OUT >NUL
DEL TRNPL121.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRLOAD
DEL TRNPL122.INS
COPY TASSIGN.OUT +TRNPL122.OUT >NUL
DEL TRNPL122.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRLOAD
DEL TRNPL123.INS
COPY TASSIGN.OUT +TRNPL123.OUT >NUL
DEL TRNPL123.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRLOAD
DEL TRNPL124.INS
COPY TASSIGN.OUT +TRNPL124.OUT >NUL
DEL TRNPL124.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRLOAD
DEL TRNPL125.INS
COPY TASSIGN.OUT +TRNPL125.OUT >NUL
DEL TRNPL125.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRLOAD
DEL TRNPL126.INS
COPY TASSIGN.OUT +TRNPL126.OUT >NUL
DEL TRNPL126.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRLOAD
DEL TRNPL127.INS
COPY TASSIGN.OUT +TRNPL127.OUT >NUL
DEL TRNPL127.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRLOAD
DEL TRNPL128.INS
COPY TASSIGN.OUT +TRNPL128.OUT >NUL
DEL TRNPL128.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRLOAD
DEL TRNPL129.INS
COPY TASSIGN.OUT +TRNPL129.OUT >NUL
```

```
DEL TRNPL129.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
rem
ECHO -- COMBINE AM TRANSIT LOADS AND REPORT
rem
ECHO -- COMBINE PREMIUM WALK ACCESS TRANSIT LOADS
rem
copy TLEGSAM2.TEM TLEGS.1
copy TLEGSAM3.TEM TLEGS.2
copy TLEGSAM4.TEM TLEGS.3
tadlodf3
ren TLEGS.123 TLEGAWPM.p99
datime
rem
ECHO -- COMBINE PARK-N-RIDE ACCESS TRANSIT LOADS
rem
copy TLEGSAM5.TEM TLEGS.1
copy TLEGSAM6.TEM TLEGS.2
copy TLEGSAM7.TEM TLEGS.3
tadlodf3
ren TLEGS.123 TLEGAAPR.p99
datime
rem
ECHO -- COMBINE KISS-N-RIDE ACCESS TRANSIT LOADS
rem
copy TLEGSAM8.TEM TLEGS.1
copy TLEGSAM9.TEM TLEGS.2
copy TLEGSAMX.TEM TLEGS.3
tadlodf3
ren TLEGS.123 TLEGAAKR.p99
datime
rem
ECHO -- COMBINE WALK ACCESS TRANSIT LOADS
rem
copy TLEGSAM1.TEM TLEGS.1
copy TLEGAWPM.p99     TLEGS.2
tadlodf2
ren TLEGS.12 TLEGSAMW.p99
ren TLEGSAM1.TEM TLEGSALB.p99
datime
rem
ECHO -- COMBINE AUTO ACCESS TRANSIT LOADS
rem
copy TLEGAAPR.p99     TLEGS.1
copy TLEGAAKR.p99     TLEGS.2
tadlodf2
ren TLEGS.12 TLEGSAMA.p99
datime
rem
ECHO -- COMBINE AM TRANSIT LOADS
rem
copy TLEGSAMW.p99     TLEGS.1
copy TLEGSAMA.p99     TLEGS.2
tadlodf2
ren TLEGS.12 TLEGSAM.p99
datime
TRPRAS
DEL TRNPL130.INS
COPY TASSIGN.OUT +TRNPL130.OUT >NUL
DEL TRNPL130.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
if exist TLEGSAM*.TEM del TLEGSAM*.TEM
rem
ECHO ***MIDDAY TRANSIT ASSIGNMENT***
rem
datime
ECHO -- NO TRANSPOSE OF TRANSIT BECAUSE AUTO EGRESS IS DIFFICULT
datime
MTABLE
```

```
DEL TRNPL131.INS
COPY TASSIGN.OUT +TRNPL131.OUT >NUL
DEL TRNPL131.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRLOAD
DEL TRNPL132.INS
COPY TASSIGN.OUT +TRNPL132.OUT >NUL
DEL TRNPL132.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRLOAD
DEL TRNPL133.INS
COPY TASSIGN.OUT +TRNPL133.OUT >NUL
DEL TRNPL133.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRLOAD
DEL TRNPL134.INS
COPY TASSIGN.OUT +TRNPL134.OUT >NUL
DEL TRNPL134.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRLOAD
DEL TRNPL135.INS
COPY TASSIGN.OUT +TRNPL135.OUT >NUL
DEL TRNPL135.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRLOAD
DEL TRNPL136.INS
COPY TASSIGN.OUT +TRNPL136.OUT >NUL
DEL TRNPL136.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRLOAD
DEL TRNPL137.INS
COPY TASSIGN.OUT +TRNPL137.OUT >NUL
DEL TRNPL137.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRLOAD
DEL TRNPL138.INS
COPY TASSIGN.OUT +TRNPL138.OUT >NUL
DEL TRNPL138.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRLOAD
DEL TRNPL139.INS
COPY TASSIGN.OUT +TRNPL139.OUT >NUL
DEL TRNPL139.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRLOAD
DEL TRNPL140.INS
COPY TASSIGN.OUT +TRNPL140.OUT >NUL
DEL TRNPL140.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
TRLOAD
DEL TRNPL141.INS
COPY TASSIGN.OUT +TRNPL141.OUT >NUL
DEL TRNPL141.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
rem
ECHO -- COMBINE MD TRANSIT LOADS AND REPORT
rem
ECHO -- COMBINE PREMIUM WALK ACCESS TRANSIT LOADS
rem
copy TLEGSMD2.TEM TLEGS.1
copy TLEGSMD3.TEM TLEGS.2
copy TLEGSMD4.TEM TLEGS.3
tadlodf3
ren TLEGS.123 TLEGMWPM.p99
datime
rem
ECHO -- COMBINE PARK-N-RIDE ACCESS TRANSIT LOADS
rem
copy TLEGSMD5.TEM TLEGS.1
copy TLEGSMD6.TEM TLEGS.2
copy TLEGSMD7.TEM TLEGS.3
tadlodf3
ren TLEGS.123 TLEGMAPR.p99
datime
rem
ECHO -- COMBINE KISS-N-RIDE ACCESS TRANSIT LOADS
rem
copy TLEGSMD8.TEM TLEGS.1
copy TLEGSMD9.TEM TLEGS.2
copy TLEGSMDX.TEM TLEGS.3
tadlodf3
ren TLEGS.123 TLEGMAKR.p99
datime
rem
ECHO -- COMBINE WALK ACCESS TRANSIT LOADS
rem
copy TLEGSMD1.TEM TLEGS.1
copy TLEGMWPM.p99     TLEGS.2
tadlodf2
ren TLEGS.12 TLEGSMDW.p99
ren TLEGSMD1.TEM TLEGSMLB.p99
datime
rem
ECHO -- COMBINE AUTO ACCESS TRANSIT LOADS
rem
copy TLEGMAPR.p99     TLEGS.1
copy TLEGMAKR.p99     TLEGS.2
tadlodf2
ren TLEGS.12 TLEGSMDA.p99
datime
rem
ECHO -- COMBINE MD TRANSIT LOADS
rem
copy TLEGSMDW.p99     TLEGS.1
copy TLEGSMDA.p99     TLEGS.2
tadlodf2
ren TLEGS.12 TLEGSMD.p99
datime
TRPRAS
DEL TRNPL142.INS
COPY TASSIGN.OUT +TRNPL142.OUT >NUL
DEL TRNPL142.OUT
IF EXIST TRNPLN.ERR GOTO ERR
datime
if exist TLEGSMD*.TEM del TLEGSMD*.TEM
if exist TLEGS.1 del TLEGS.1
if exist TLEGS.2 del TLEGS.2
if exist TLEGS.3 del TLEGS.3
datime
datime >> time.log
ECHO ***** TASSIGN ***** -- END
if exist TRNPLN.ERR goto err
echo off
echo on
rem **** TEVAL **** -- START
echo off
datime >> time.log
datime
if exist TEVAL.OUT del TEVAL.OUT
datime
tevalmp
```

```
datime
datime >> time.log
echo on
rem **** TEVAL **** -- END
if exist TRNPLN.ERR goto err
echo off

rem ************* feedback convergence checking **************
rem * Utility program: fdk.exe, fdk.inp, fdk.cnt,            *
rem * Input files: fdk.inp, hrldxy(1).txt, hassign(1).out,   *
rem *              fsutms.ctl, vfactors.xxx                  *
rem * Output files: fdk.cnt, fdk.out, endexam.fdk            *
    if %1==p goto END
    IF EXIST HRLDXY.TXT DEL HRLDXY.TXT
    LODUNP <fdk.syn
    start/w FDK
    rem PAUSE 'beginning of iteration #...'
    type fdk.cnt
    if exist endexam.fdk goto END
    IF EXIST HASSIGN1.OUT DEL HASSIGN1.OUT
    copy HASSIGN.OUT HASSIGN1.OUT >NUL
    copy *.ink *.ins >NUL
    copy trnpln1.con trnplnx.con >NUL
    IF EXIST hrldxyx.p99 DEL hrldxyx.p99
    LODPAK <fdk1.syn
    if exist DISTRIB.OUT del DISTRIB.OUT
    copy gtff.tem gtff1.tem >NUL
    GOTO FSU100
rem ********************************************************

:ERR
IF NOT EXIST TRNPLN.ERR GOTO END
DEL *.INS
DEL *.ERR
:END
rem DEL TRNPLNX.CON
:FIN
DEL TITLE
DEL FSUTMS.PTH
ECHO ON
```

B.2. Additional Input Scripts for Palm Beach Model

(TRNPL170.INT)

```
$CALIBRATE GRAVITY MODEL
$FILES
~  INPUT FILE = GMSKIM, USER ID = $FSKIMS.P99$
   INPUT FILE = GMSKIM, USER ID = $WGTSKMTT.TEM$
   INPUT FILE = GRVDATA, USER ID = $GMHIST.DAT$
   OUTPUT FILE = NEWDATA, USER ID = $GMHIST.99p$
$HEADERS
   CALIBRATE GRAVITY MODEL
$OPTIONS
   GRVDATA
   OUTPUT DATA FILE
   PRINT TRIP LENGTH STATISTICS
$PARAMETERS
   F FACTOR CLOSURE =10.0
   F FACTOR ITERATIONS = 5
   MAXIMUM PURPOSE = 11
   MAXIMUM TIME = 95
   SELECTED PURPOSES = 1-11
$END TP FUNCTION
```

(TRNPL171.INS)

```
$MATRIX UPDATE
$FILES
   INPUT FILE = UPDIN, USER ID = $ptrips0.dat$, unload
   OUTPUT FILE = UPDOUT, USER ID = $ptrips0.tem$
$HEADERS
   APPLY MSA WEIGHT (1-1/k) TO OD TALBES OF PREVIOUS ITERATIONS
$OPTIONS
   SUPPRESS DATA REPORT
$DATA
T1, 1-1172     ,1-1172     ,*0.9630
T2, 1-1172     ,1-1172     ,*0.9630
T3, 1-1172     ,1-1172     ,*0.9630
T4, 1-1172     ,1-1172     ,*0.9630
T5, 1-1172     ,1-1172     ,*0.9630
T6, 1-1172     ,1-1172     ,*0.9630
T7, 1-1172     ,1-1172     ,*0.9630
T8, 1-1172     ,1-1172     ,*0.9630
T9, 1-1172     ,1-1172     ,*0.9630
T10, 1-1172     ,1-1172     ,*0.9630
T11, 1-1172     ,1-1172     ,*0.9630
$END TP FUNCTION
```

(TRNPL172.INS)

```
$MATRIX UPDATE
$FILES
   INPUT FILE = UPDIN, USER ID = $ptrips1.dat$, unload
   OUTPUT FILE = UPDOUT, USER ID = $ptrips1.tem$
$HEADERS
   APPLY MSA WEIGHT (1/k) TO OD TALBES OF CURRENT ITERATION
$OPTIONS
   SUPPRESS DATA REPORT
$DATA
T1, 1-1172     ,1-1172     ,*0.0370
T2, 1-1172     ,1-1172     ,*0.0370
T3, 1-1172     ,1-1172     ,*0.0370
T4, 1-1172     ,1-1172     ,*0.0370
T5, 1-1172     ,1-1172     ,*0.0370
T6, 1-1172     ,1-1172     ,*0.0370
T7, 1-1172     ,1-1172     ,*0.0370
T8, 1-1172     ,1-1172     ,*0.0370
T9, 1-1172     ,1-1172     ,*0.0370
T10, 1-1172     ,1-1172     ,*0.0370
T11, 1-1172     ,1-1172     ,*0.0370
$END TP FUNCTION
```

(TRNPL173.INT)

```
$MATRIX MANIPULATE
$FILES
   INPUT FILE = TMAN1, USER ID = $ptrips0.tem$
   INPUT FILE = TMAN2, USER ID = $ptrips1.tem$
   OUTPUT FILE = TMAN3, USER ID = $ptrips.dat$
$HEADERS
  Combined OD tables after MSA
$DATA
TMAN3,T1  = TMAN1,T1  + TMAN2,T1
TMAN3,T2  = TMAN1,T2  + TMAN2,T2
TMAN3,T3  = TMAN1,T3  + TMAN2,T3
TMAN3,T4  = TMAN1,T4  + TMAN2,T4
TMAN3,T5  = TMAN1,T5  + TMAN2,T5
TMAN3,T6  = TMAN1,T6  + TMAN2,T6
TMAN3,T7  = TMAN1,T7  + TMAN2,T7
TMAN3,T8  = TMAN1,T8  + TMAN2,T8
TMAN3,T9  = TMAN1,T9  + TMAN2,T9
TMAN3,T10 = TMAN1,T10 + TMAN2,T10
TMAN3,T11 = TMAN1,T11 + TMAN2,T11
$END TP FUNCTION
```

(TRNPL174.INT)

```
$REPORT TRIP LENGTH FREQUENCY
$FILES
   INPUT FILE = SKIM, USER ID = $wgtskmtt.tem$
   INPUT FILE = VOLUME, USER ID = $ptrips11.p99$
$HEADERS
   REPORT TRIP LENGTH FREQUENCY
$PARAMETERS
   MAXIMUM IMPEDANCE = 95
   SELECTED PURPOSES = 1-11
$END TP FUNCTION
```