ADVANCED TOLL MODELING

Prepared for

Systems Planning Office

Florida Department of Transportation

Prepared by

Fennessy Associates, Inc.

Under Contract to Cambridge Systematics, Inc

Task Work Order No: 98-WP

Transportation Model Applications for

Systems Transportation Modeling

Agreement No.: C8G23

Task Order No. 980

In Association with

Wilbur Smith Associates

September 2010

TABLE OF CONTENTS

1 BACKGROUND

The MTF Priorities Survey of Florida Modelers, conducted in 2009, identified the application of advanced toll modeling as the third priority – behind only Time-of-Day modeling and Land Use interfaces – to be stressed in the advancement of FSUTMS.

Toll modeling in FSUTMS, under the sponsorship of Florida's Turnpike Enterprise, has migrated over the years from the basic UTPS capability to the current state-of-the-art methods that accommodate the newer toll collection technologies. This document identifies the software migration history from the simple toll modeling origins to the Advanced Toll Modeling features available today in the TRANPLAN version of FSUTMS – advanced features that can be implemented into the CUBE version of FSUTMS.

UTPS Toll Modeling

The UTPS capability permitted 20 types of toll facilities identified in the cost field of the input highway network data. Tolls are input in dollars with a parameter, TOLLS, for up to the 20 types and the tolls are converted to a time equivalent via a CTOLL parameter based upon the average income of the study area. (Note: All the toll features convert the tolls to time via the CTOLL parameter.) The times to service the tolls at the toll facilities are input via the parameter, SERVT, for up to the 20 toll types.

Toll Facilities Model

The Toll Facilities Model (TFM) is the current default toll modeling capability in FSUTMS. This feature is available in both the TRANPLAN version as well as the CUBE version of FSUTMS. This model permits each toll facility to be modeled via an input TOLLLINK file which contains the following information: A-node, B-node, toll in dollars, service time in minutes, facility identification (for reporting), number of lanes, and toll type – ramps and barriers. The TFM requires that toll links be coded with a deceleration link, a toll link (zero distance recommended), and an acceleration link. This capability has been implemented into Voyager scripting, under another contract with the Central Office, Florida Department of Transportation.

In TRANPLAN, the TFM is a fully integrated component of the modeling software. The modeler activates the TFM by supplying a properly formatted TOLLLINK file and selecting the correct option in the highway assignment script. In contrast, the TFM does not exist as an integrated component of the CUBE modeling software package. The TFM can only be

replicated in CUBE by scripting each calculation into a model's existing highway assignment script. Differences in highway assignment procedures between models in Florida make the attempt to develop a uniform TFM approach cumbersome at best. In order to develop a TFM version for FSUTMS in CUBE, a means of efficiently getting toll plaza and acceleration/deceleration link data onto the highway network needs to be implemented. This requires additional scripts that can read a TOLLLINK input file and modify the input highway network appropriately.

It is highly recommended that the TFM be fully integrated as a feature in the CUBE modeling software. This would allow the TFM to be activated by the selection of a specific option (or keyword to use the comparable phrasing in CUBE). This process may require additional standardization in highway network attribute names and the acceptance of a format for the TOLLLINK input file that may be more compatible with CUBE. These potential requirements are consistent with other objectives being advanced by the Florida Model Task Force with regard to establishing, advancing, and maintaining FSUTMS modeling standards.

Advanced Toll Modeling

Several advances since the TFM have been directed by URS Corporation under contract to Florida's Turnpike Enterprise and commissioned by others. The following are the advances:

a) Open Road Tolls

b) Distance-Based Tolls

c) Discrete Tolls

d) HOT-Lane

e) Ramp-to-Ramp Tolls

f) Congestion Pricing Tolls

2 PROJECT SCOPE OF WORK

The scope of work for advanced toll modeling contains the following three basic tasks:

Identification of the Pertinent Advanced Features

This task consisted of discussions to determine which Advanced Features would be included in future versions of CUBE-FSUTMS. It was determined that the Discrete Tolls feature in TRANPLAN would not be implemented in Voyager because it is no longer utilized in Turnpike modeling. The congestion pricing feature in TRANPLAN was being conducted for Turnpike modeling when the economic slowdown in Florida caused the modeling development for Turnpike to be discontinued. Congestion pricing is a feature being developed within CUBE Voyager and is being tested by consultants working on other modeling projects within Florida – no testing was conducted within this project.

Determination of the Features Currently in CUBE

During discussions with Citilabs staff, it was determined that all of these advanced toll features are basically available in CUBE Voyager. Some features are currently in the documented releases of CUBE and others required discussions with Citilabs staff.

Verification of CUBE and TRANPLAN Features

This task encompassed rigorous testing of the selected toll features to ensure that the CUBE version replicates the processing of these features within TRANPLAN. It is recognized that there are inherent differences in the equilibrium assignment algorithms and this task would focus on the handling of the selected toll features. This task also examined reporting capabilities, for the toll features, which should be available in CUBE-FSUTMS.

Voyager scripts were written to replicate the TRANPLAN controls and the following two tests were conducted for each of the tasks. First, a file containing the A-node, B-node, Facility Type, and link impedance – for the inputs to the first iteration of path building – was generated for both TRANPLAN and Voyager. This was done to ensure that the path builders for both systems were working with the same data – this ensured that the Voyager scripts were replicating the impedances for the respective toll scenarios. Secondly, tests were run with several iterations of assignments and the basic differences in the assignments noted.

Toll Scenarios

a) Open Road Tolls

With open road tolling, there is no delay at the toll plaza for deceleration, for service time, or for acceleration. This requires a code change on the TOLLLINK file so that toll facilities with the appropriate code are open road facilities.

b) Distance-Based Tolls

For Florida's Turnpike model, the distance-based tolls were only used with the TOLL DIVERSION option which does a toll diversion based upon time and cost -- yielding a weighted average, based upon the iterations of assignment, for toll and non-toll trips during the assignment process. All current Turnpike models use a PRESET TOLL condition where the mode choice is performed before the assignment, e.g. drive-alone free, drive-alone toll, 2P free, 2P toll, 3+P free, and 3+P toll -- all tolls utilize the ramp-to-ramp process and distances are not directly utilized. To try to replicate the TOLL DIVERSION process in TRANPLAN and the distance-based tolls previously used would not be productive. This toll feature was discontinued by Florida's Turnpike Enterprise and was not analyzed for this project. No work was done in CUBE to implement distance based tolls.

c) Discrete Tolls

Tolls, with this feature, are based upon the number of toll facilities crossed during path building in the assignment process. This required a modification to the TOLLLINK file which was introduced in the TFM. This toll feature was discontinued by Florida's Turnpike Enterprise and was not analyzed for this project. No work was done in CUBE to implement discrete tolls.

d) HOT-Lane

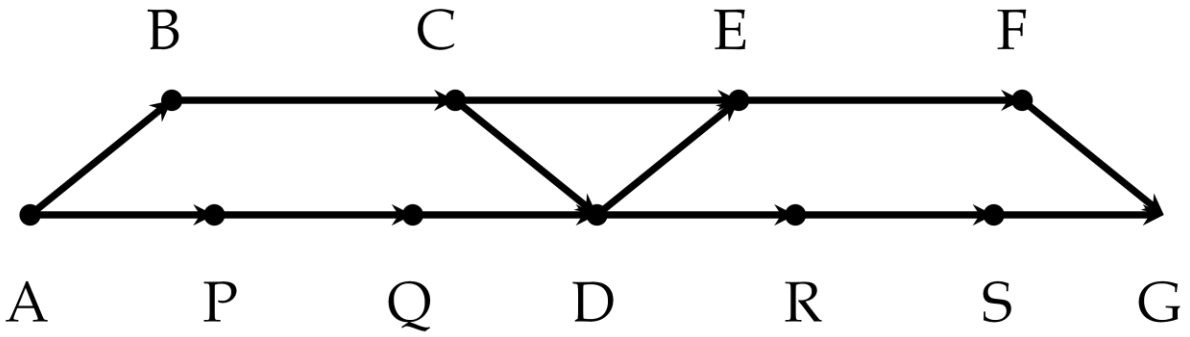
This toll feature utilizes tolls based upon vehicle occupancy for up to three occupancy modes and the tolls are input on a modified TOLLLINK file. The loadings by vehicle occupancy are controlled by the LOAD CLASS specifications in the TRANPLAN equilibrium assignment program.

e) Ramp-to-Ramp Tolls

This toll procedure utilizes toll rates, by mode, for all ramp-to-ramp movements identified as input parameters. For all toll facilities, not identified as ramp-to-ramp tolls, the TFM model is the default. Because of path building, network reconstruction is done prior to the assignment process – pseudo links are constructed, the assignment done and the pseudo links are deconstructed for each iteration of assignment and for output loadings and congested times. Reports were prepared to show the modeled trips and the projected revenue for each ramp-to-ramp movement and for company totals. (See sample reports later in this document.) Ramp-to-ramp toll modeling encompasses all of the various types of toll models: card systems, barrier systems, electronic tolling, open road tolling, discrete tolling, and distance-based tolls. This toll scenario is the current toll modeling structure utilized by Florida's Turnpike Enterprise.

A ramp-to-ramp example is presented below.

Software Problem:

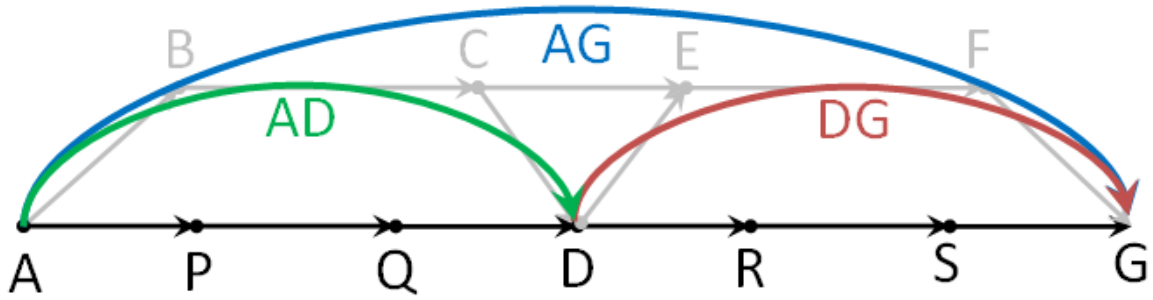


Where: B-C-E-F is the toll road
A-B, C-D, D-E, and F-G are the ramps
A-P-Q-D-R-S-G is the local road

Toll Pathing: ABCD times + 50 cent toll converted to time
DEFG times + 75 cent toll converted to time
ABCEFG time + 125 cent toll converted to time

In this example, a pseudo link is created in the network for each toll pathing link: pseudo link A-D, representing A-B-C-D; pseudo link D-G, representing D-E-F-G; and pseudo link A-G, representing A-B-C-E-F-G. This is done because, for example building paths from A to D, the minimum impedance to D may follow the toll road A-B-C-D; however, adding the toll from on ramp A-B to off ramp C-D may mean that the minimum path could actually be by local streets; A-P-Q-D. The true minimum path from A-D is determined by using the pseudo link A-D or by using the local streets.

Pseudo Links:



f) Congestion Pricing Tolls

With this toll feature, toll pricing is based upon congestion on the toll facility links during the assignment process – at the iteration and lambda loop calculations. An initial (or seed) toll is input and the process attempts to determine an "optimal" toll by facility. This feature has been developed within TRANPLAN but has not been tested because funding for model development/analysis was curtailed by Florida's Turnpike Enterprise at the start of 2009. It is being tested in CUBE Voyager by a consultant with no definitive results at this time. No mature testing could be conducted during this project.

A Congestion Pricing Toll model can be seen as a more general case of the HOT-Lane/Value-Pricing Toll model. The key difference is that congestion pricing can be area-wide or system-wide as opposed to being restricted to a specific facility. Furthermore, whereas HOT-Lanes are usually designed to allow drivers to use HOV lanes by paying a set premium toll, congestion pricing involves a more dynamic relationship between travel demand and supply.

A well-developed Congestion Pricing Toll feature in FSUTMS would require a robust method for dividing vehicle trips into meaningful bins that could react independently to different sets of toll prices. Since the principal utility of Congestion Pricing Toll systems revolves around the ability to manage traffic by changing prices, some means of distinguishing trips by their sensitivity to price fluctuations would be beneficial. This

could be achieved by the development of trip purposes by income group. This would have a dramatic impact on how FSUTMS modeling is done since it would require altering the current trip generation method to account for income. An analysis of driver data and stated preferences may also need to be undertaken to assess how different groups of drivers react to different toll prices. A single CTOLL value as is used in the current TFM to convert toll prices to a value of time for all trips in the model would be insufficient. More work should be done along this vein since there are other benefits to travel demand modeling and policy analysis that can be achieved by tying travel behavior more closely to monetary units.

Another issue to consider is that Congestion Pricing Toll models can be seen as optimization problems. The goal is to optimize the toll price and traffic flow conditions. Congestion Pricing Toll systems fluctuate their prices based on existing or anticipated traffic conditions throughout the day. At a minimum, this suggests that effective Congestion Pricing Toll models need to be time-of-day models. As such, any desire to implement a Congestion Pricing Toll model should be closely coordinated with the MTF's work to develop time-of-day modeling standards for FSUTMS. Additionally, Congestion Pricing Toll models could benefit greatly from the additional analytical detail provided by activity-based modeling. Work currently being conducted in Florida on activity-based modeling may serve as a suitable platform in the future for developing a truly effective Congestion Pricing Toll model. Finally, an integrated toll optimization algorithm may need to be developed that could adjust toll prices iteratively during the highway assignment process. The MTF should continue to take a closer look at enhancing FSUTMS assignment methodologies.

Many of the features needed to develop a successful Congestion Pricing Toll model do not appear to be available as distinct tools in any of the components of CUBE available for testing during this effort. Additional work will need to be done to determine to what extent the existing CUBE features can be applied to Congestion Pricing. Certainly, CUBE Voyager's scripting language is flexible enough to allow for the implementation of a wide range of impedance calculations. These all occur outside of the model's adjustment equations and can only influence model path-building and routing in so far as they affect the assignment cost functions. It is currently unknown as to how effective a Congestion Pricing Toll model can be without embedding toll price adjustments into the software's equilibrium calculations.

Research into developing a congestion pricing toll model should build off of current work being conducted in Florida in the fields of time-of-day modeling, activity-based modeling, and dynamic traffic assignments. Each of the research efforts mentioned deal

with elements that will make the development of a fully functional congestion price model possible.

3 CONCLUSIONS

Open Road Tolls

Again, the only impedance found on an open road tolling link is caused by the value of time related to the price of the toll. Because there is no other delay associated with this type of toll, it is the easiest of the advanced tolls to implement in CUBE. The open road toll process requires a means of identifying specific links in the highway network as open road toll plaza links. The methods for doing this are the same as those used to implement the TFM in CUBE. A set of scripts is developed along with the TFM which takes data stored in the TOLLLINK input file and puts it onto the highway network. The model then automatically updates the highway network with the correct toll plaza information. For the purposes of the tests conducted for this study, open road toll plazas were identified in the TOLLLINK file by adding an "X" in the left most position of the space reserved in the file for toll plaza descriptions. The model scripts were developed to read the toll plaza description for each record in the TOLLLINK file and to interpret any toll plazas which descriptions began with "X" as open road toll plazas.

Since queuing, service time, deceleration, and acceleration do not contribute to the impedance calculated at the open road toll plaza, the impedance remains static from iteration to iteration of the highway assignment. Because of this, differences in the equilibrium models used by either CUBE or TRANPLAN do not have an impact on resulting impedance values of the open road toll links.

The tests conducted for this effort were able to replicate identical initial impedances for both the CUBE and TRANPLAN platforms. The scripts to the open road toll models are in Appendices 1-A and 1-B for the TRANPLAN and Voyager tests, respectively.

A base 10-iteration assignment for both systems showed the following results for sample open road toll links:

North-bound Direction

Link	Voyager	TRANPLAN	Difference	% Diff
Start	8,762	9,469	-707	-8.07%
Middle Link 1	7,489	7,374	115	1.54%
Middle Link 2	18,694	18,182	512	2.74%
Middle Link 3	23,561	23,907	-346	-1.47%
End	14,980	14,271	709	4.73%
Total	**73,486**	**73,203**	**283**	**0.39%**

South-bound Direction

Link	Voyager	TRANPLAN	Difference	% Diff
Start	4,730	4,114	616	13.02%
Middle Link 1	10,759	10,494	265	2.46%
Middle Link 2	17,601	16,678	923	5.24%
Middle Link 3	24,082	26,479	-2,397	-9.95%
End	15,955	15,122	833	5.22%
Total	**73,127**	**72,887**	**240**	**0.33%**

HOT Lane

For the purpose of testing this feature in CUBE, a trip table was provided from TRANPLAN. The table was divided between HOV and non-HOV trips. The HOT-Lane testing was an extremely simplified effort representing an initial attempt to implement this feature. HOT-Lane facilities were coded as HOV lanes and were given a toll price and service time impedance value of zero. There were no impedances assigned to these facilities on the basis of acceleration and deceleration links. The model was able to accurately replicate in CUBE the initial impedances found in TRANPLAN.

A more extensive effort to implement a HOT-Lane feature will need to be undertaken before such a process can be successfully integrated with FSUTMS. The current effort has demonstrated that it is possible to establish proper impedance calculations and path building for volumes belonging to different classes. This makes it possible to measure the impacts of differing impedances among different volume purposes, such as HOV and non-HOV trips. The following table shows comparison of volumes on toll links after convergence:

Links	Volume Group	NB/EB				SB/WB			
		Voyager	Tranplan	Diff	% diff	Voyager	Tranplan	Diff	% diff
Link 1	1	0	0	0	0.00%	0	0	0	0.00%
	2	4,627	4,520	107	2.31%	5,310	5,114	196	3.69%
	3	0	0	0	0.00%	0	0	0	0.00%
	4	1,166	1,150	16	1.37%	1,394	1,365	29	2.08%
	5	0	0	0	0.00%	0	0	0	0.00%
	6	664	674	-10	-1.51%	3,070	3,125	-55	-1.79%
Link 2	1	0	0	0	0.00%	0	0	0	0.00%
	2	6,401	6,575	-174	-2.72%	7,261	6,718	543	7.48%
	3	0	0	0	0.00%	0	0	0	0.00%
	4	1,373	1,419	-46	-3.35%	1,609	1,544	65	4.04%
	5	0	0	0	0.00%	0	0	0	0.00%
	6	707	727	-20	-2.83%	2,541	2,692	-151	-5.94%
Link 3	1	0	0	0	0.00%	0	0	0	0.00%
	2	3,787	3,846	-59	-1.56%	4,125	4,615	-490	-11.88%
	3	0	0	0	0.00%	0	0	0	0.00%
	4	891	907	-16	-1.80%	975	1,108	-133	-13.64%
	5	0	0	0	0.00%	0	0	0	0.00%
	6	502	516	-14	-2.79%	1,458	1,757	-299	-20.51%

Ramp-to-Ramp Tolls

For the ramp-to-ramp task, a program was written to convert the TRANPLAN formats of the input ramp identification file and the ramp-to-ramp tolls into a DBF file which could be input into the Voyager network program. This eliminated any conversion errors.

The Voyager ramp-to-ramp feature was successfully run for several test cases. Turn prohibitors to toll "on" ramps and from toll "off" ramps were promulgated properly and a selected link for one of the toll links had the correct volume. The following problems/deficiencies were noted:

1) Not all the ramp-to-ramp conditions were handled by Voyager. Two off-ramp or two on-ramp links connected at the same node, even to two different toll road directions, cause a script error in Cube Voyager. Citilabs will be looking into these issues.

2) The following toll reports should be available for toll analyses:

Ramp-to-Ramp Detail Report

RAMP-RAMP TOLL DETAIL REPORT -- Mode = 1 -- Company Number: 11 -- Company Name: Veterans Exwy

Ramp-to-Ramp Identification			Toll ($)	Dist. (Miles)	Volume	Rev. ($)	From Ramp	To Ramp
Veterans Exwy SB On at Suncoast	-to-	Veterans Exwy SB Off at SR 60	1.75	11.5	11353	19868	15567-15557	15468-15425
	-to-	Veterans Exwy SB Off at Memorial	1.75	11.1	95	166		15468-15436
	-to-	Veterans Exwy SB Off at Hillsboroug	1.75	10.2	3762	6584		15521-15469
	-to-	Veterans Exwy SB Off at Waters	1.75	8.2	7827	13697		15576-15569
	-to-	Veterans Exwy SB Off at Wilsky	1.25	6.3	441	551		15093-15065
	-to-	Veterans Exwy SB Off at Gunn	1.00	4.5	997	997		15081-15050
	-to-	Veterans Exwy SB Off at Ehrlich	0.75	3.6	2374	1781		15193-15162
	-to-	Veterans Exwy SB Off at Hutchison	0.50	1.6	3777	1889		15700-15653
Veterans Exwy SB On at Dale Mabry	-to-	Veterans Exwy SB Off at SR 60	1.75	14.4	745	1304	16755-16563	15468-15425
	-to-	Veterans Exwy SB Off at Memorial	1.75	13.9	20	35		15468-15436
	-to-	Veterans Exwy SB Off at Hillsboroug	1.75	13.0	400	700		15521-15469
	-to-	Veterans Exwy SB Off at Waters	1.75	11.0	1601	2802		15576-15569
	-to-	Veterans Exwy SB Off at Wilsky	1.25	9.2	112	140		15093-15065
	-to-	Veterans Exwy SB Off at Gunn	1.00	7.4	634	634		15081-15050
	-to-	Veterans Exwy SB Off at Ehrlich	0.75	6.5	2419	1814		15193-15162
	-to-	Veterans Exwy SB Off at Hutchison	0.50	4.5	1243	622		15700-15653
Company Totals					169618*******			

Company Summary Report

RAMP-RAMP TOLL SUMMARY REPORT:

Company Number	Company Name	Volume	Revenue ($)
7	Sunshine Skwy BR	73997	73997.
17	St. Johns Ferry	22963	74630.
12	Suncoast Pky	119374	97214.
42	Suncoast Pky 2	46591	43054.
29	Tpk NorthCoin Sys	318830	409126.
31	Tpk SouthCoin Sys	471758	561620.
28	Tpk SouthConn Ext	50633	28874.
30	Tpk Ticket Sys	214306	699633.
8	Treasure Is BR	8964	4482.
35	Venetian Causeway BR	13442	10082.
11	Veterans Exwy	168637	178632.
Total -- All Companies		6148844	5472082.

Because the networks are restructured with the pseudo link, no direct comparison can be made with input impedances between TRANPLAN and CUBE. The following table shows comparisons with loaded volumes for a simple, one company test of the ramp-to-ramp scenario:

Link	Voyager	TRANPLAN	Difference	% Diff
Northbound	9,320	7,793	1,527	16.38%
Southbound	14,042	13,330	712	5.07%

4 APPENDICES

These appendices contain the script for the advanced toll modeling scenarios selected for this project. The TRANPLAN script is specified, with the specific scripting for the respective toll feature denoted in RED and, then, the CUBE Voyager script is specified.

Appendix 1-A: TRANPLAN – OPEN ROAD TOLLS

The following is the TRANPLAN script for this test:

```
$EQUILIBRIUM HIGHWAY LOAD
$FILES
  INPUT FILE = HWYNET, USER ID = $HNET.F20$
  INPUT FILE = HWYTRIP, USER ID = $ODAM2.TOT$
  INPUT FILE = TRNDATA, USER ID = $TCARDS.20F$
  INPUT FILE = TOLDATA, USER ID = $TOLLLINK.20F$
  OUTPUT FILE = LODHIST, USER ID = $HRLDAM1.2 $
$HEADERS
   TRANPLAN TEST -- OPEN ROAD TOLL ASSIGNMENT
$OPTIONS
   TOLL FACILITIES MODEL
   MULTIPLE SERVER QUEUES
   ZERO NEGATIVE VOLUMES
   TOLL EVASION CONDITIONS
   MINIMUM UNCONNECTED ZONES
   OPEN ROAD TOLLS
   TOLL DEBUG
$PARAMETERS
  EQUILIBRIUM ITERATIONS = 1
  DAMPING FACTOR = 0.25
  CONFAC = 0.3805
  EPS = 0.04
$END TP FUNCTION
```

The following is the TOLLLINK file for this test:

Note the “X” in the records with toll code 92 (columns 1-2) – this signifies that open road tolling will be for that facility.

```
CTOLL = 0.05
 1 2 16003 16004 FT EXT YEEHAW JCT         02 02 04.00 0:12 1 1 00 00 0.00
 1 2 16007 16008 FT ENT YEEHAW JCT         01 01 00.00 0:06 1 1 00 00 0.00
 1 2 12752 12751 FT ML MPOST 236 NB        06 06 04.00 0.12 1 1 00 00 0.00
                                 ▪
                                 ▪
                                 ▪
91 1  9157  9164 WB 438 T/F S              02 02 00.15 0:07 1 1 00 00 0.00
91 1  9160  9163 WB 438 T/F S              02 02 00.15 0:07 1 1 00 00 0.00
92 1 14127 14128XWB CR 535 NB ON           02 02 00.31 0:08 1 1 00 00 0.00
92 1 14125 14126XWB CR 535 SB OFF          02 02 00.31 0:08 1 1 00 00 0.00
92 1  9217  9218XWB North Plaza NB         02 02 00.61 0:08 1 1 00 00 0.00
92 1  9215  9216XWB North Plaza SB         02 02 00.61 0:08 1 1 00 00 0.00
92 1 14146 14147XWB CR 545 NB OFF          02 02 00.31 0:08 1 1 00 00 0.00
92 1 14144 14145XWB CR 545 SB ON           02 02 00.31 0:08 1 1 00 00 0.00
92 1 15202 15203XWB Seidel Rd NB OFF       02 02 00.50 0:08 1 1 00 00 0.00
92 1 15204 15205XWB Seidel Rd SB ON        02 02 00.50 0:08 1 1 00 00 0.00
92 1 14172 14173XWB South Plaza NB         02 02 01.00 0:08 1 1 00 00 0.00
92 1 14168 14169XWB South Plaza SB         02 02 01.00 0:08 1 1 00 00 0.00
92 1  9178  9179XWB 192 SB ON              02 02 00.50 0:08 1 1 00 00 0.00
92 1  9173  9174XWB 192 NB OFF             02 02 00.50 0:08 1 1 00 00 0.00
92 1 18011 18012XSlip Ramp                 02 02 00.25 0:00 1 1 00 00 0.00
```

Appendix 1-B: VOYAGER – OPEN ROAD TOLLS

The following is the Voyager script for this test:

```
;*********************************************************************************************
;         READ IN VFACTORS FILE,TOLL LINK FILE AND SCALE THE TIME & DISTANCE FROM THE TRANPLAN NETWORK
;                        WRITE ACCELERATION AND DECELERATION NODE IDENTIFIERS
;*********************************************************************************************
RUN PGM=NETWORK
FILEI LOOKUPI[1] = "VFACTORS.txt"
FILEI LINKI[2] = "TOLLLINK.DAT",
 var=TOLL,beg=1,len=2,
 var=TOLLTYPE,beg=3,len=2,
 var=A,beg=5,len=6,
 var=B,beg=11,len=6,
 var=PLAZADESC,beg=17,len=25,typ=a,
 var=PLZALNSMIN,beg=44,len=2,
 var=PLZALNSMAX,beg=47,len=2,
 var=CARTOLL,beg=50,len=5,
 var=SVCMINUTES,beg=56,len=1,
 var=SVCSECONDS,beg=58,len=2,
 var=DECELCODE,beg=61,len=1,
 var=ACCELCODE,beg=63,len=1,
 var=EXACTCHGLNS,beg=65,len=2,
 var=AVILANES,beg=68,len=2,
 var=PCTTRUCKS,beg=71,len=4
FILEI LINKI[1] = "HNET.F20"
FILEO NETO = "TEMP1.NET"
FILEO PRINTO[1] = "TEMP1.CSV"
FILEO LINKO = "LNK4CNT.DBF",
 FORMAT=DBF, INCLUDE=A,B

LOOKUP LOOKUPI=1 NAME=VFACTORS,
  LOOKUP[1]=1, RESULT=2,
  LOOKUP[2]=1, RESULT=3,
  LOOKUP[3]=1, RESULT=4,
  LOOKUP[4]=1, RESULT=5,
```

```
  FAIL=0,0,0

PROCESS PHASE=LINKMERGE
      UROAD_FAC= VFACTORS(1,LI.1.LINKGRP2)     ; read in uroadfactor, confac, bpr_coefficient, bpr_exponent
;and adjust time and distance from tranplan network
      CONFAC= VFACTORS(2,LI.1.LINKGRP2)
      BPR_COEFF= VFACTORS(3,LI.1.LINKGRP2)
      BPR_EXP= VFACTORS (4,LI.1.LINKGRP2)
      DISTANCE=DISTANCE/100
      TIME1=TIME1/100
      TIME2=TIME2/100

      IF (TOLLTYPE>0 & TOLLTYPE<>2)      ; check for non-open road toll links and write the acceleration
;and deceleration node identifiers
         PRINT CSV=T LIST=A,1,0 PRINTO=1
         PRINT CSV=T LIST=B,0,1 PRINTO=1
      ENDIF
ENDPROCESS
ENDRUN

;*************************************************************************************************
;        Count Node Usage
;*************************************************************************************************
RUN PGM=MATRIX MSG='Count Times Nodes are Used'
FILEO RECO[1] = "NODECNT.DBF",
 FIELDS=N,TIMESUSED
FILEI RECI = "LNK4CNT.DBF"

ARRAY NODECNT=999999

NODECNT[RI.A]=NODECNT[RI.A]+1
NODECNT[RI.B]=NODECNT[RI.B]+1
IF (RI.A>HINODE) HINODE=RI.A
IF (RI.B>HINODE) HINODE=RI.B

IF (I=0)

  LOOP N=1,HINODE
    IF (NODECNT[N]>0)
     TIMESUSED=NODECNT[N]
     WRITE RECO=1
```

```
    ENDIF
  ENDLOOP

ENDIF

ENDRUN

;*********************************************************************************************************
;        FLAG ACCELERATION AND DECELERATION LINKS USING LINK ATTRIBUTES 'TOLL_ACCEL' AND 'TOLL_DECEL'
;*********************************************************************************************************
RUN PGM=NETWORK MSG='Flag Toll Acceleration and Deceleration Nodes'
FILEO PRINTO[1] = "TEMPNODE.DAT"
FILEI LINKI[1] = "TEMP1.NET"

if (li.1.tolltype>0)
 if (li.1.tolltype<>2)
  a_accel=0
  a_decel=1 ; commented out for ORT
  b_accel=1 ; commented out for ORT
  b_decel=0
  PRINT form=0, LIST=A,' ',a_accel,' ',a_decel,' ',li.1.toll,printo=1
  PRINT form=0, LIST='NODE= ',A,' ACCEL_NODE= ',a_accel,' DECEL_NODE= ',a_decel, 'PLAZA ID= ',li.1.toll
  PRINT form=0, LIST=B,' ',b_accel,' ',b_decel,' ',li.1.toll,printo=1
  PRINT form=0, LIST='NODE= ',B,' ACCEL_NODE= ',b_accel,' DECEL_NODE= ',b_decel, 'PLAZA ID= ',li.1.toll
 else
  a_accel=0
  a_decel=0 ; added for ORT
  b_accel=0 ; added for ORT
  b_decel=0
  PRINT form=0, LIST=A,' ',a_accel,' ',a_decel,' ',li.1.toll,printo=1
  PRINT form=0, LIST='NODE= ',A,' ACCEL_NODE= ',a_accel,' DECEL_NODE= ',a_decel, 'PLAZA ID= ',li.1.toll
  PRINT form=0, LIST=B,' ',b_accel,' ',b_decel,' ',li.1.toll,printo=1
  PRINT form=0, LIST='NODE= ',B,' ACCEL_NODE= ',b_accel,' DECEL_NODE= ',b_decel, 'PLAZA ID= ',li.1.toll
 endif
endif


ENDRUN

;*********************************************************************************************************
******
```

```
;        ASSIGN DECEL AND ACCEL LINKS
;*************************************************************************************************
RUN PGM=NETWORK MSG='Assign Acceleration and Deceleration Links'
FILEO PRINTO[2] = "XY.DAT"
FILEO NETO = "TEMP2.NET"
FILEI LINKI[1] = "TEMP1.NET"
FILEI NODEI[3] = "TEMPNODE.DAT",
 VAR=N,_TOLL_ACC,_TOLL_DEC,_PLAZAID
FILEO PRINTO[1] = "NODES.CSV"
FILEI NODEI[2] = "NODECNT.DBF"


PROCESS  PHASE=INPUT, FILEI=LI.1
  TOLL_ACC=0
  TOLL_DEC=0
ENDPROCESS

PROCESS PHASE=NODEMERGE
   PRINT CSV=T, LIST=N(8.0),X,Y printo=1
   PRINT LIST=N(6.0),X(20.2),Y(20.2),printo=2
   IF (TIMESUSED=0) DELETE
ENDPROCESS

PROCESS PHASE=LINKMERGE

  IF(B._TOLL_DEC=1)
    TOLL_DEC=1
    TOLL=B._PLAZAID
  ENDIF
  IF(A._TOLL_ACC=1)
    TOLL_ACC=1
    TOLL=A._PLAZAID
  ENDIF

ENDPROCESS
ENDRUN

;*************************************************************************************************
;                                                ASSIGNMENT
;*************************************************************************************************
RUN PGM=HIGHWAY
```

```
FILEI TURNPENI = "TCARDS.PEN"
FILEI NETI = "TEMP2.NET"
FILEI MATI[1] = "ODAM2.MAT"
FILEO NETO = "LOADED_TOLL.NET"

PAR MAXITERS= 50 GAP=0.01

LOOKUP, ;Deceleration Curve
 INTERPOLATE=Y, LIST=Y, NAME=DECEL,
        LOOKUP[1]=1,RESULT=2,
R = ' 0 2.35',
    '30 4',
    '70 6.2',
    '99 7.795'

ARRAY ITER=Numlinks ALINK=Numlinks BLINK=Numlinks VOLBC=Numlinks SPDAB=Numlinks SPDCD=Numlinks
ARRAY TT1=Numlinks TT2=Numlinks TT3=Numlinks TT4=Numlinks DD1=Numlinks DD2=Numlinks DD3=Numlinks
ARRAY DISTAB=Numlinks TIMEAB=Numlinks DISTCD=Numlinks TIMECD=Numlinks DELAY=Numlinks
ARRAY LAM=Numlinks MU=Numlinks UTIL=Numlinks SER=Numlinks UTLSER=Numlinks
ARRAY PP0=Numlinks REALN=Numlinks AVEQUE=Numlinks TACC=Numlinks TDEC=Numlinks TPLZ=Numlinks
ARRAY ANODE=Numlinks BNODE=Numlinks TOLTIM=Numlinks TIMT1=Numlinks VTPLT1=Numlinks TIMT3=Numlinks
VTPLT3=Numlinks
ARRAY LISPD1=Numlinks LISPD3=Numlinks TOTDELAY=Numlinks RATIO=Numlinks


PROCESS PHASE=LINKREAD

IF (ITERATION=0 & LI.TOLL>0 & LI.TOLL_ACC=0 & LI.TOLL_DEC=0) NUMTOLLS=NUMTOLLS+1

      CTOLL= 0.05
      CTOLLSCALE=60
      C= (LI.CAPACITY/LI.CONFAC)*LI.UROAD_FAC

      LW.DISTANCE=LI.DISTANCE
      DISTANCE=LW.DISTANCE
      T0=LI.TIME1
      LW.FFTIME=LI.TIME1 ; use for preload purposes (fftime)

; NON TOLL LINKS IS LINKCLASS 1
; OPEN ROAD TOLL LINKS IS LINKCLASS 2
; OTHER TOLL LINKS IS LINKCLASS 3
```

```
; ACCELERATION LINKS IS LINKCLASS 4
; DECELERATION LINKS IS LINKCLASS 5


IF (LI.TOLLTYPE=0 & LI.TOLL_ACC=0 & LI.TOLL_DEC=0) LINKCLASS=1 T0=LI.TIME1
IF (LI.TOLLTYPE=2) LINKCLASS=2 T0=LI.CARTOLL*CTOLL*CTOLLSCALE

IF (LI.TOLLTYPE>0 & LI.TOLLTYPE<>2)
   LINKCLASS=3
       IF (ITERATION=0)
        LW.ARRIVR=(V/LI.UROAD_FAC)*LI.CONFAC/LI.PLZALNSMAX  ; hourly volume per toll lane ie. arrival rate
;in vehicles per hour
        LW.SERVT=LI.SVCMINUTES+(LI.SVCSECONDS/60)           ; plaza lane service time in minutes per
;vehicle
        LW.SERVR=(1/LW.SERVT)*60                            ; plaza lane service rate in vehicle per hour
        IF (LW.ARRIVR>=LW.SERVR) LW.ARRIVR=0.99*LW.SERVR    ; prevent infinite or negative queue

            ITER[LI.TOLL]=ITERATION
            ALINK[LI.TOLL]=A
            BLINK[LI.TOLL]=B
            VOLBC[LI.TOLL]=V
            TT4[LI.TOLL]= CTOLL*LI.CARTOLL*CTOLLSCALE*100

          IF ({QUEUETYPE}=1) ; multiple server queue model
              LAM[LI.TOLL]= VOLBC[LI.TOLL]
              MU[LI.TOLL] = 1/(LW.SERVT/60)
              UTIL[LI.TOLL]=LAM[LI.TOLL]/MU[LI.TOLL]
              UTLSER[LI.TOLL]=UTIL[LI.TOLL]/LI.PLZALNSMAX
              SER[LI.TOLL]=LI.PLZALNSMAX

; Moved from section after "check against maxutl"

              PP0[LI.TOLL]=1
              REALN[LI.TOLL]=1
              NLANE1=LI.PLZALNSMAX-1
              RATIO[LI.TOLL]=1

              IF (UTLSER[LI.TOLL]>0.99) ; check against maxutl
                  LAM[LI.TOLL]= MU[LI.TOLL]*SER[LI.TOLL]*0.99
                  RATIO[LI.TOLL]= VOLBC[LI.TOLL]/LAM[LI.TOLL]
                  UTIL[LI.TOLL]= LAM[LI.TOLL]/MU[LI.TOLL]
```

```
                UTLSER[LI.TOLL]= UTIL[LI.TOLL]/SER[LI.TOLL]
            ENDIF


            LOOP NN=1,NLANE1
                REALI=NN
                REALN[LI.TOLL]= REALN[LI.TOLL]* REALI
                PP0[LI.TOLL]=PP0[LI.TOLL]+(UTIL[LI.TOLL]^REALI)/REALN[LI.TOLL]
            ENDLOOP

            PP0[LI.TOLL]=1/(PP0[LI.TOLL]+(UTIL[LI.TOLL]^SER[LI.TOLL])/(REALN[LI.TOLL]*SER[LI.TOLL]*(1-
UTLSER[LI.TOLL])))
            TT2[LI.TOLL]= PP0[LI.TOLL] * UTIL[LI.TOLL]^SER[LI.TOLL] * UTLSER[LI.TOLL]
            TT2[LI.TOLL]= TT2[LI.TOLL] / (SER[LI.TOLL]* REALN[LI.TOLL] *((1-UTLSER[LI.TOLL])^2))

            IF (LAM[LI.TOLL]=0)
               TT2[LI.TOLL]= 6000/MU[LI.TOLL]
            ELSE
               TT2[LI.TOLL]=(TT2[LI.TOLL]/LAM[LI.TOLL]*RATIO[LI.TOLL]+1/MU[LI.TOLL])*6000
            ENDIF
            DD2[LI.TOLL]=0
        ELSE  ; single server queue model
            LAM[LI.TOLL]=VOLBC[LI.TOLL]/LI.PLZALNSMAX
            MU[LI.TOLL]= 1/(LW.SERVT/60)

            IF (LAM[LI.TOLL]>(0.99*MU[LI.TOLL])) LAM[LI.TOLL]=(0.99*MU[LI.TOLL]); check against maxutl

            UTIL[LI.TOLL]=LAM[LI.TOLL]/MU[LI.TOLL]
            AVEQUE[LI.TOLL]=MU[LI.TOLL]/(MU[LI.TOLL]-LAM[LI.TOLL])
            DD2[LI.TOLL]= AVEQUE[LI.TOLL]*35/5280*100; Car Length = 35 ft
            TT2[LI.TOLL]=1/(MU[LI.TOLL]-LAM[LI.TOLL])*6000
        ENDIF

        IF (TT2[LI.TOLL]>1333.33) TT2[LI.TOLL]=1333.33


    T0=(TT2[LI.TOLL]+TT4[LI.TOLL])/100
    TPLZ[LI.TOLL]=TT2[LI.TOLL]/100
    TOLTIM[LI.TOLL]=TT4[LI.TOLL]/100
    ANODE[LI.TOLL]=LI.A
    BNODE[LI.TOLL]=LI.B
```

```
      ENDIF
ENDIF
IF (LI.TOLL_ACC=1)
   LINKCLASS= 4
        ;Acceleration Lane (CD)
           IF (LI.TOLL>0 & LI.TOLL_ACC=1 & LI.TOLL_DEC=0)
               IF (LI.TIME1=0)
                  TT3[LI.TOLL]=0
                  DD3[LI.TOLL]=0
                  SPDCD[LI.TOLL]=1
               ELSE
                  SPDCD[LI.TOLL]= DISTANCE/LI.TIME1*60
                  DISTCD[LI.TOLL]=DISTANCE
                  TIMECD[LI.TOLL]=LI.TIME1
                  TT3[LI.TOLL]=   SPDCD[LI.TOLL]/2.5/0.6
                  DD3[LI.TOLL]=   SPDCD[LI.TOLL]*0.5*TT3[LI.TOLL]/60
               ENDIF
           ENDIF
       TA=(TT3[LI.TOLL]-((DD3[LI.TOLL]/SPDCD[LI.TOLL])*60))/100
       T0=TA+T0
       TACC[LI.TOLL]=TA
       TIMT3[LI.TOLL]=LI.TIME1
       VTPLT3[LI.TOLL]=V
       LISPD3[LI.TOLL]=SPDCD[LI.TOLL]
ENDIF
IF (LI.TOLL_DEC=1)
   LINKCLASS= 5
        ;Deceleration Lane (AB)
           IF (LI.TOLL>0 & LI.TOLL_ACC=0 & LI.TOLL_DEC=1)
               IF (LI.TIME1=0)
                  TT1[LI.TOLL]=0
                  DD1[LI.TOLL]=0
                  SPDAB[LI.TOLL]=1
               ELSE
                  SPDAB[LI.TOLL]= DISTANCE/LI.TIME1*60
                  DISTAB[LI.TOLL]=DISTANCE
                  TIMEAB[LI.TOLL]=LI.TIME1
                  TT1[LI.TOLL]=   SPDAB[LI.TOLL]/DECEL(1,SPDAB[LI.TOLL])/0.6
                  DD1[LI.TOLL]=   SPDAB[LI.TOLL]*0.5*TT1[LI.TOLL]/60
               ENDIF
```

```
          ENDIF
       TD=(TT1[LI.TOLL]-((DD1[LI.TOLL]/SPDAB[LI.TOLL])*60))/100
       T0=TD+T0
       TDEC[LI.TOLL]=TD
       TIMT1[LI.TOLL]=LI.TIME1
       VTPLT1[LI.TOLL]=V
       LISPD1[LI.TOLL]=SPDAB[LI.TOLL]
ENDIF


ENDPROCESS


PROCESS PHASE=ILOOP
       PATHLOAD  PATH=TIME, PENI= 1, VOL[1]=MI.1.1    ; load trips
ENDPROCESS

PROCESS PHASE=ADJUST

IF (LI.TOLLTYPE>0 & LI.TOLLTYPE<>2)
        LW.ARRIVR=(V/LI.UROAD_FAC)*LI.CONFAC/LI.PLZALNSMAX  ; hourly volume per toll lane ie. arrival rate
in vehicles per hour
        LW.SERVT=LI.SVCMINUTES+(LI.SVCSECONDS/60)           ; plaza lane service time in minutes per
vehicle
        LW.SERVR=(1/LW.SERVT)*60                            ; plaza lane service rate in vehicle per hour
        IF (LW.ARRIVR>=LW.SERVR) LW.ARRIVR=0.99*LW.SERVR    ; prevent infinite or negative queue
ENDIF

IF (ITERATION>0)
;*********************************************************************
;    ROUTINE TO CALCULATE THE DELAY AT A TOLL FACILITY (SETOLL.FOR)
;*********************************************************************
;Deceleration Lane (AB)
   IF (LI.TOLL>0 & LI.TOLL_ACC=0 & LI.TOLL_DEC=1)
       IF (LI.TIME1=0)
          TT1[LI.TOLL]=0
          DD1[LI.TOLL]=0
          SPDAB[LI.TOLL]=1
       ELSE
          SPDAB[LI.TOLL]= DISTANCE/LI.TIME1*60
          DISTAB[LI.TOLL]=DISTANCE
```

```
         TIMEAB[LI.TOLL]=LI.TIME1
         TT1[LI.TOLL]=   SPDAB[LI.TOLL]/DECEL(1,SPDAB[LI.TOLL])/0.6
         DD1[LI.TOLL]=   SPDAB[LI.TOLL]*0.5*TT1[LI.TOLL]/60
      ENDIF
   ENDIF
;Acceleration Lane (CD)
   IF (LI.TOLL>0 & LI.TOLL_ACC=1 & LI.TOLL_DEC=0)
      IF (LI.TIME1=0)
         TT3[LI.TOLL]=0
         DD3[LI.TOLL]=0
         SPDCD[LI.TOLL]=1
      ELSE
         SPDCD[LI.TOLL]= DISTANCE/LI.TIME1*60
         DISTCD[LI.TOLL]=DISTANCE
         TIMECD[LI.TOLL]=LI.TIME1
         TT3[LI.TOLL]=   SPDCD[LI.TOLL]/2.5/0.6
         DD3[LI.TOLL]=   SPDCD[LI.TOLL]*0.5*TT3[LI.TOLL]/60
      ENDIF
   ENDIF

; Toll Plaza (BC)
IF (LI.TOLL>0 & LI.TOLL_ACC=0 & LI.TOLL_DEC=0)
    ITER[LI.TOLL]=ITERATION
    ALINK[LI.TOLL]=A
    BLINK[LI.TOLL]=B
    VOLBC[LI.TOLL]=V
    TT4[LI.TOLL]= CTOLL*LI.CARTOLL*60*100
ENDIF
IF (LI.TOLLTYPE>0 & LI.TOLLTYPE<>2)
   IF ({QUEUETYPE}=1) ; MULTIPLE SERVER QUEUE MODEL
      LAM[LI.TOLL]= VOLBC[LI.TOLL]
      MU[LI.TOLL] = 1/(LW.SERVT/60)
      UTIL[LI.TOLL]=LAM[LI.TOLL]/MU[LI.TOLL]
      UTLSER[LI.TOLL]=UTIL[LI.TOLL]/LI.PLZALNSMAX
      SER[LI.TOLL]=LI.PLZALNSMAX

      PP0[LI.TOLL]=1
      REALN[LI.TOLL]=1
      NLANE1=LI.PLZALNSMAX-1
      RATIO[LI.TOLL]=1
```

```
      IF (UTLSER[LI.TOLL]>0.99) ; CHECK AGAINST MAXUTL
        LAM[LI.TOLL]= MU[LI.TOLL]*SER[LI.TOLL]*0.99
        RATIO[LI.TOLL]= VOLBC[LI.TOLL]/LAM[LI.TOLL]
        UTIL[LI.TOLL]= LAM[LI.TOLL]/MU[LI.TOLL]
        UTLSER[LI.TOLL]= UTIL[LI.TOLL]/SER[LI.TOLL]
      ENDIF


      LOOP NN=1,NLANE1
          REALI=NN
          REALN[LI.TOLL]= REALN[LI.TOLL]* REALI
          PP0[LI.TOLL]=PP0[LI.TOLL]+(UTIL[LI.TOLL]^REALI)/REALN[LI.TOLL]
      ENDLOOP

      PP0[LI.TOLL]=1/(PP0[LI.TOLL]+(UTIL[LI.TOLL]^SER[LI.TOLL])/(REALN[LI.TOLL]*SER[LI.TOLL]*(1-
UTLSER[LI.TOLL])))
      TT2[LI.TOLL]= PP0[LI.TOLL] * UTIL[LI.TOLL]^SER[LI.TOLL] * UTLSER[LI.TOLL]
      TT2[LI.TOLL]= TT2[LI.TOLL] / (SER[LI.TOLL]* REALN[LI.TOLL] *((1-UTLSER[LI.TOLL])^2))

      IF (LAM[LI.TOLL]=0)
         TT2[LI.TOLL]= 6000/MU[LI.TOLL]
      ELSE
         TT2[LI.TOLL]=(TT2[LI.TOLL]/LAM[LI.TOLL]*RATIO[LI.TOLL]+1/MU[LI.TOLL])*6000
      ENDIF
      DD2[LI.TOLL]=0
   ELSE  ; SINGLE SERVER QUEUE MODEL
      LAM[LI.TOLL]=VOLBC[LI.TOLL]/LI.PLZALNSMAX
      MU[LI.TOLL]= 1/(LW.SERVT/60)

      IF (LAM[LI.TOLL]>(0.99*MU[LI.TOLL])) LAM[LI.TOLL]=(0.99*MU[LI.TOLL]); CHECK AGAINST MAXUTL

      UTIL[LI.TOLL]=LAM[LI.TOLL]/MU[LI.TOLL]
      AVEQUE[LI.TOLL]=MU[LI.TOLL]/(MU[LI.TOLL]-LAM[LI.TOLL])
      DD2[LI.TOLL]= AVEQUE[LI.TOLL]*35/5280*100; CARLTH=35 FEET
      TT2[LI.TOLL]=1/(MU[LI.TOLL]-LAM[LI.TOLL])*6000
   ENDIF

   IF (TT2[LI.TOLL]>1333.33) TT2[LI.TOLL]=1333.33
    DELAY[LI.TOLL]=(TT1[LI.TOLL]+TT2[LI.TOLL]+TT3[LI.TOLL]+TT4[LI.TOLL]- (
(DD1[LI.TOLL]/SPDAB[LI.TOLL]+DD3[LI.TOLL]/SPDCD[LI.TOLL]) *60))/100
ENDIF
```

```
ENDIF

FUNCTION {

      TC[1]= (T0*(1.0 + LI.BPR_COEFF*(V/C)^LI.BPR_EXP) - T0)*0.25 + T0
      TC[2]= LI.CARTOLL*CTOLL*CTOLLSCALE
      TC[3]= (TT2[LI.TOLL]+TT4[LI.TOLL])/100; delay for toll links, considering type of Queue Model
      TC[4]= LI.TIME1+((TT3[LI.TOLL]-((DD3[LI.TOLL]/SPDCD[LI.TOLL])*60))/100); congested time toll
;acceleration links
      TC[5]= LI.TIME1+((TT1[LI.TOLL]-((DD1[LI.TOLL]/SPDAB[LI.TOLL])*60))/100); congested time toll
;deceleration links

}

ENDPROCESS

ENDRUN
```

Appendix 2-A: TRANPLAN – HOT LANES

The following is the TRANPLAN script for this test:

```
$EQUILIBRIUM HIGHWAY LOAD
$FILES
  INPUT FILE = HWYNET, USER ID = $HNET.F20$
  INPUT FILE = HWYTRIP, USER ID = $ODAM.2$
  INPUT FILE = TRNDATA, USER ID = $TCARDS.20F$
  INPUT FILE = TOLDATA, USER ID = $TOLLLINK.20F$
  OUTPUT FILE = LODHIST, USER ID = $HRLDAM1.2$
$HEADERS
   TRANPLAN TEST -- HOT LANES ASSIGNMENT
$OPTIONS
   PRESET TOLL DIVERSION
   TOLL FACILITIES MODEL
   MULTIPLE SERVER QUEUES
   ZERO NEGATIVE VOLUMES
   TOLL EVASION CONDITIONS
   MINIMUM UNCONNECTED ZONES
   SUMMARY FILE
   TOLL DEBUG
$PARAMETERS
  EQUILIBRIUM ITERATIONS = 20
  DAMPING FACTOR = 0.25
  CONFAC = 0.3805
  EPS = 0.04
  LOAD CLASS = 1, EXCLUDE, LINK GROUP 2 = 80-99 ~NO TOLL OR HOV
  LOAD CLASS = 2, EXCLUDE, LINK GROUP 2 = 80-88 ~NO HOV (TOLL – OK)
  LOAD CLASS = 3, EXCLUDE, LINK GROUP 2 = 80-99 ~NO TOLL OR HOV
  LOAD CLASS = 4, EXCLUDE, LINK GROUP 2 = 80-88 ~NO HOV (TOLL – OK)
  LOAD CLASS = 5, EXCLUDE, LINK GROUP 2 = 89-99 ~NO TOLL (HOV – OK)
  LOAD CLASS = 6                                    ~HOV AND TOLL – OK
$END TP FUNCTION
```

The following is the TOLLLINK file for this test:

```
CTOLL = 0.05
89 3 14524 14525 SB XL MP 95/MP 92.4       02 02 00.25 0:00 1 1 00 00 0.00
89 3 14526 14527 NB XL MP 92.4/MP 95       02 02 00.25 0:00 1 1 00 00 0.00
89 3 14532 14533 SB XL MP 92.4/MP 90.6     02 02 00.25 0:00 1 1 00 00 0.00
89 3 14534 14535 NB XL MP 90.6/MP 92.4     02 02 00.25 0:00 1 1 00 00 0.00
89 3 14536 14537 SB XL MP 90.6/MP 88.2     02 02 00.25 0:00 1 1 00 00 0.00
89 3 14538 14539 NB XL MP 88.2/MP 90.6     02 02 00.25 0:00 1 1 00 00 0.00
89 3 14540 14541 SB XL MP 88.2/MP 84.3     02 02 00.25 0:00 1 1 00 00 0.00
89 3 14542 14543 NB XL MP 84.3/MP 88.2     02 02 00.25 0:00 1 1 00 00 0.00
89 3 14544 14545 SB XL MP 84.3/MP 82.8     02 02 00.25 0:00 1 1 00 00 0.00
89 3 14546 14547 NB XL MP 82.8/MP 84.3     02 02 00.25 0:00 1 1 00 00 0.00
89 3 14548 14549 SB XL MP 82.8/MP 80.5     02 02 00.25 0:00 1 1 00 00 0.00
89 3 14550 14551 NB XL MP 80.5/MP 82.8     02 02 00.25 0:00 1 1 00 00 0.00
89 3 14552 14553 SB XL MP 80.5/MP 78.5     02 02 00.25 0:00 1 1 00 00 0.00
89 3 14554 14555 NB XL MP 78.5/MP 80.5     02 02 00.25 0:00 1 1 00 00 0.00
```

Appendix 2-B: VOYAGER – HOT LANES

```
;*********************************************************************************************
;         READ IN VFACTORS FILE,TOLL LINK FILE AND SCALE THE TIME & DISTANCE FROM THE TRANPLAN NETWORK
;                        WRITE ACCELERATION AND DECELERATION NODE IDENTIFIERS
;*********************************************************************************************
RUN PGM=NETWORK
FILEI LOOKUPI[1] = "VFACTORS.txt"
FILEI LINKI[2] = "TOLLLINK.DAT",
 var=TOLL,beg=1,len=2,
 var=TOLLTYPE,beg=3,len=2,
 var=A,beg=5,len=6,
 var=B,beg=11,len=6,
 var=PLAZADESC,beg=17,len=25,typ=a,
 var=PLZALNSMIN,beg=44,len=2,
 var=PLZALNSMAX,beg=47,len=2,
 var=CARTOLL,beg=50,len=5,
 var=SVCMINUTES,beg=56,len=1,
 var=SVCSECONDS,beg=58,len=2,
 var=DECELCODE,beg=61,len=1,
 var=ACCELCODE,beg=63,len=1,
 var=EXACTCHGLNS,beg=65,len=2,
 var=AVILANES,beg=68,len=2,
 var=PCTTRUCKS,beg=71,len=4
FILEI LINKI[1] = "HNET.F20"
FILEO NETO = "TEMP1.NET"
FILEO PRINTO[1] = "TEMP1.CSV"
FILEO LINKO = "LNK4CNT.DBF",
 FORMAT=DBF, INCLUDE=A,B

LOOKUP LOOKUPI=1 NAME=VFACTORS,
  LOOKUP[1]=1, RESULT=2,
  LOOKUP[2]=1, RESULT=3,
  LOOKUP[3]=1, RESULT=4,
  LOOKUP[4]=1, RESULT=5,
  FAIL=0,0,0

PROCESS PHASE=LINKMERGE
```

```
      UROAD_FAC= VFACTORS(1,LI.1.LINKGRP2)     ; read in uroadfactor, confac, bpr_coefficient, bpr_exponent
;and adjust time and distance from tranplan network
      CONFAC= VFACTORS(2,LI.1.LINKGRP2)
      BPR_COEFF= VFACTORS(3,LI.1.LINKGRP2)
      BPR_EXP= VFACTORS (4,LI.1.LINKGRP2)
      DISTANCE=DISTANCE/100
      TIME1=TIME1/100
      TIME2=TIME2/100

      IF (TOLLTYPE>0 & TOLLTYPE<>2)      ; check for non-open road toll links and write the acceleration
;and deceleration node identifiers
         PRINT CSV=T LIST=A,1,0 PRINTO=1
         PRINT CSV=T LIST=B,0,1 PRINTO=1
      ENDIF
ENDPROCESS
ENDRUN

;*************************************************************************************************
;        Count Node Usage
;*************************************************************************************************
RUN PGM=MATRIX MSG='Count Times Nodes are Used'
FILEO RECO[1] = "NODECNT.DBF",
 FIELDS=N,TIMESUSED
FILEI RECI = "LNK4CNT.DBF"

ARRAY NODECNT=999999

NODECNT[RI.A]=NODECNT[RI.A]+1
NODECNT[RI.B]=NODECNT[RI.B]+1
IF (RI.A>HINODE) HINODE=RI.A
IF (RI.B>HINODE) HINODE=RI.B

IF (I=0)

  LOOP N=1,HINODE
    IF (NODECNT[N]>0)
     TIMESUSED=NODECNT[N]
     WRITE RECO=1
    ENDIF
  ENDLOOP
```

```
ENDIF

ENDRUN

;*********************************************************************************************************
;        FLAG ACCELERATION AND DECELERATION LINKS USING LINK ATTRIBUTES 'TOLL_ACCEL' AND 'TOLL_DECEL'
;*********************************************************************************************************
RUN PGM=NETWORK MSG='Flag Toll Acceleration and Deceleration Nodes'
FILEO PRINTO[1] = "TEMPNODE.DAT"
FILEI LINKI[1] = "TEMP1.NET"

if (li.1.tolltype> 0)
 if (li.1.tolltype<> 2)
  a_accel=0
  a_decel=1 ; commented out for ORT
  b_accel=1 ; commented out for ORT
  b_decel=0
  PRINT form=0, LIST=A,' ',a_accel,' ',a_decel,' ',li.1.toll,printo=1
  PRINT form=0, LIST='NODE= ',A,' ACCEL_NODE= ',a_accel,' DECEL_NODE= ',a_decel, 'PLAZA ID= ',li.1.toll
  PRINT form=0, LIST=B,' ',b_accel,' ',b_decel,' ',li.1.toll,printo=1
  PRINT form=0, LIST='NODE= ',B,' ACCEL_NODE= ',b_accel,' DECEL_NODE= ',b_decel, 'PLAZA ID= ',li.1.toll
 else
  a_accel=0
  a_decel=0 ; added for ORT
  b_accel=0 ; added for ORT
  b_decel=0
  PRINT form=0, LIST=A,' ',a_accel,' ',a_decel,' ',li.1.toll,printo=1
  PRINT form=0, LIST='NODE= ',A,' ACCEL_NODE= ',a_accel,' DECEL_NODE= ',a_decel, 'PLAZA ID= ',li.1.toll
  PRINT form=0, LIST=B,' ',b_accel,' ',b_decel,' ',li.1.toll,printo=1
  PRINT form=0, LIST='NODE= ',B,' ACCEL_NODE= ',b_accel,' DECEL_NODE= ',b_decel, 'PLAZA ID= ',li.1.toll
 endif

endif


ENDRUN

;*********************************************************************************************************
;        ASSIGN DECEL AND ACCEL LINKS
;*********************************************************************************************************
RUN PGM=NETWORK MSG='Assign Acceleration and Deceleration Links'
```

```
FILEO PRINTO[2] = "XY.DAT"
FILEO NETO = "TEMP2.NET"
FILEI LINKI[1] = "TEMP1.NET"
FILEI NODEI[3] = "TEMPNODE.DAT",
 VAR=N,_TOLL_ACC,_TOLL_DEC,_PLAZAID
FILEO PRINTO[1] = "NODES.CSV"
FILEI NODEI[2] = "NODECNT.DBF"


PROCESS  PHASE=INPUT, FILEI=LI.1
  TOLL_ACC=0
  TOLL_DEC=0
ENDPROCESS

PROCESS PHASE=NODEMERGE
   PRINT CSV=T, LIST=N(8.0),X,Y printo=1
   PRINT LIST=N(6.0),X(20.2),Y(20.2),printo=2
   IF (TIMESUSED=0) DELETE
ENDPROCESS

PROCESS PHASE=LINKMERGE

  IF(B._TOLL_DEC=1)
    TOLL_DEC=1
    TOLL=B._PLAZAID
  ENDIF
  IF(A._TOLL_ACC=1)
    TOLL_ACC=1
    TOLL=A._PLAZAID
  ENDIF

ENDPROCESS
ENDRUN


;*************************************************************************************************
;                                          ASSIGNMENT
;*************************************************************************************************
RUN PGM=HIGHWAY
FILEI TURNPENI = "TCARDS.PEN"
FILEI NETI = "TEMP2.NET"
```

```
FILEI MATI[1] = "ODAM2.MAT"
FILEO NETO = "LOADED_TOLL.NET"

PAR MAXITERS= 50 GAP=0.01

LOOKUP, ;Deceleration Curve
 INTERPOLATE=Y, LIST=Y, NAME=DECEL,
        LOOKUP[1]=1,RESULT=2,
R = ' 0 2.35',
    '30 4',
    '70 6.2',
    '99 7.795'

ARRAY ITER=Numlinks ALINK=Numlinks BLINK=Numlinks VOLBC=Numlinks SPDAB=Numlinks SPDCD=Numlinks
ARRAY TT1=Numlinks TT2=Numlinks TT3=Numlinks TT4=Numlinks DD1=Numlinks DD2=Numlinks DD3=Numlinks
ARRAY DISTAB=Numlinks TIMEAB=Numlinks DISTCD=Numlinks TIMECD=Numlinks DELAY=Numlinks
ARRAY LAM=Numlinks MU=Numlinks UTIL=Numlinks SER=Numlinks UTLSER=Numlinks
ARRAY PP0=Numlinks REALN=Numlinks AVEQUE=Numlinks TACC=Numlinks TDEC=Numlinks TPLZ=Numlinks
ARRAY ANODE=Numlinks BNODE=Numlinks TOLTIM=Numlinks TIMT1=Numlinks VTPLT1=Numlinks TIMT3=Numlinks
VTPLT3=Numlinks
ARRAY LISPD1=Numlinks LISPD3=Numlinks TOTDELAY=Numlinks RATIO=Numlinks


PROCESS PHASE=LINKREAD

IF (ITERATION=0 & LI.TOLL>0 & LI.TOLL_ACC=0 & LI.TOLL_DEC=0) NUMTOLLS=NUMTOLLS+1

      CTOLL= 0.05
      CTOLLSCALE=60
      C= (LI.CAPACITY/LI.CONFAC)*LI.UROAD_FAC

      LW.DISTANCE=LI.DISTANCE
      DISTANCE=LW.DISTANCE
      T0=LI.TIME1
      LW.FFTIME=LI.TIME1 ; use for preload purposes (fftime)

IF (LI.LINKGRP2= 80-99) ADDTOGROUP= 1  ;No TOll or HOV
IF (LI.LINKGRP2= 80-88) ADDTOGROUP= 2  ;No HOV
IF (LI.LINKGRP2= 89-99) ADDTOGROUP= 3  ;No TOll
```

```
; NON TOLL LINKS IS LINKCLASS 1
; OPEN ROAD TOLL LINKS IS LINKCLASS 2
; OTHER TOLL LINKS IS LINKCLASS 3
; ACCELERATION LINKS IS LINKCLASS 4
; DECELERATION LINKS IS LINKCLASS 5

IF (LI.TOLLTYPE=0 & LI.TOLL_ACC=0 & LI.TOLL_DEC=0) LINKCLASS=1 T0=LI.TIME1
IF (LI.TOLLTYPE=2) LINKCLASS=2 T0=LI.CARTOLL*CTOLL*CTOLLSCALE

IF (LI.TOLLTYPE>0 & LI.TOLLTYPE<>2)
   LINKCLASS=3
       IF (ITERATION=0)
        LW.ARRIVR=(V/LI.UROAD_FAC)*LI.CONFAC/LI.PLZALNSMAX  ; hourly volume per toll lane ie. arrival rate
;in vehicles per hour
        LW.SERVT=LI.SVCMINUTES+(LI.SVCSECONDS/60)           ; plaza lane service time in minutes per
;vehicle
        IF (LW.SERVT= 0) LW.SERVT=0.0001
        LW.SERVR=(1/LW.SERVT)*60                            ; plaza lane service rate in vehicle per hour
        IF (LW.ARRIVR>=LW.SERVR) LW.ARRIVR=0.99*LW.SERVR    ; prevent infinite or negative queue

            ITER[LI.TOLL]=ITERATION
            ALINK[LI.TOLL]=A
            BLINK[LI.TOLL]=B
            VOLBC[LI.TOLL]=V
            TT4[LI.TOLL]= CTOLL*LI.CARTOLL*CTOLLSCALE*100

           IF ({QUEUETYPE}=1) ; multiple server queue model
               LAM[LI.TOLL]= VOLBC[LI.TOLL]
               MU[LI.TOLL] = 1/(LW.SERVT/60)
               UTIL[LI.TOLL]=LAM[LI.TOLL]/MU[LI.TOLL]
               UTLSER[LI.TOLL]=UTIL[LI.TOLL]/LI.PLZALNSMAX
               SER[LI.TOLL]=LI.PLZALNSMAX

; Moved from section after "check against maxutl"

               PP0[LI.TOLL]=1
               REALN[LI.TOLL]=1
               NLANE1=LI.PLZALNSMAX-1
               RATIO[LI.TOLL]=1

               IF (UTLSER[LI.TOLL]>0.99) ; check against maxutl
```

```
                    LAM[LI.TOLL]= MU[LI.TOLL]*SER[LI.TOLL]*0.99
                    RATIO[LI.TOLL]= VOLBC[LI.TOLL]/LAM[LI.TOLL]
                    UTIL[LI.TOLL]= LAM[LI.TOLL]/MU[LI.TOLL]
                    UTLSER[LI.TOLL]= UTIL[LI.TOLL]/SER[LI.TOLL]
                ENDIF


                LOOP NN=1,NLANE1
                    REALI=NN
                    REALN[LI.TOLL]= REALN[LI.TOLL]* REALI
                    PP0[LI.TOLL]=PP0[LI.TOLL]+(UTIL[LI.TOLL]^REALI)/REALN[LI.TOLL]
                ENDLOOP

                PP0[LI.TOLL]=1/(PP0[LI.TOLL]+(UTIL[LI.TOLL]^SER[LI.TOLL])/(REALN[LI.TOLL]*SER[LI.TOLL]*(1-
UTLSER[LI.TOLL])))
                TT2[LI.TOLL]= PP0[LI.TOLL] * UTIL[LI.TOLL]^SER[LI.TOLL] * UTLSER[LI.TOLL]
                TT2[LI.TOLL]= TT2[LI.TOLL] / (SER[LI.TOLL]* REALN[LI.TOLL] *((1-UTLSER[LI.TOLL])^2))

                IF (LAM[LI.TOLL]=0)
                   TT2[LI.TOLL]= 6000/MU[LI.TOLL]
                ELSE
                   TT2[LI.TOLL]=(TT2[LI.TOLL]/LAM[LI.TOLL]*RATIO[LI.TOLL]+1/MU[LI.TOLL])*6000
                ENDIF
                DD2[LI.TOLL]=0
            ELSE  ; single server queue model
                LAM[LI.TOLL]=VOLBC[LI.TOLL]/LI.PLZALNSMAX
                MU[LI.TOLL]= 1/(LW.SERVT/60)

                IF (LAM[LI.TOLL]>(0.99*MU[LI.TOLL])) LAM[LI.TOLL]=(0.99*MU[LI.TOLL]); check against maxutl

                UTIL[LI.TOLL]=LAM[LI.TOLL]/MU[LI.TOLL]
                AVEQUE[LI.TOLL]=MU[LI.TOLL]/(MU[LI.TOLL]-LAM[LI.TOLL])
                DD2[LI.TOLL]= AVEQUE[LI.TOLL]*35/5280*100; Car Length = 35 ft
                TT2[LI.TOLL]=1/(MU[LI.TOLL]-LAM[LI.TOLL])*6000
            ENDIF

            IF (TT2[LI.TOLL]>1333.33) TT2[LI.TOLL]=1333.33


        T0=(TT2[LI.TOLL]+TT4[LI.TOLL])/100
        TPLZ[LI.TOLL]=TT2[LI.TOLL]/100
```

```
       TOLTIM[LI.TOLL]=TT4[LI.TOLL]/100
       ANODE[LI.TOLL]=LI.A
       BNODE[LI.TOLL]=LI.B
      ENDIF
ENDIF
IF (LI.TOLL_ACC=1)
   LINKCLASS= 4
        ;Acceleration Lane (CD)
          IF (LI.TOLL>0 & LI.TOLL_ACC=1 & LI.TOLL_DEC=0)
              IF (LI.TIME1=0)
                 TT3[LI.TOLL]=0
                 DD3[LI.TOLL]=0
                 SPDCD[LI.TOLL]=1
              ELSE
                 SPDCD[LI.TOLL]= DISTANCE/LI.TIME1*60
                 DISTCD[LI.TOLL]=DISTANCE
                 TIMECD[LI.TOLL]=LI.TIME1
                 TT3[LI.TOLL]=   SPDCD[LI.TOLL]/2.5/0.6
                 DD3[LI.TOLL]=   SPDCD[LI.TOLL]*0.5*TT3[LI.TOLL]/60
              ENDIF
          ENDIF
       TA=(TT3[LI.TOLL]-((DD3[LI.TOLL]/SPDCD[LI.TOLL])*60))/100
       T0=TA+T0
       TACC[LI.TOLL]=TA
       TIMT3[LI.TOLL]=LI.TIME1
       VTPLT3[LI.TOLL]=V
       LISPD3[LI.TOLL]=SPDCD[LI.TOLL]
ENDIF
IF (LI.TOLL_DEC=1)
   LINKCLASS= 5
        ;Deceleration Lane (AB)
          IF (LI.TOLL>0 & LI.TOLL_ACC=0 & LI.TOLL_DEC=1)
              IF (LI.TIME1=0)
                 TT1[LI.TOLL]=0
                 DD1[LI.TOLL]=0
                 SPDAB[LI.TOLL]=1
              ELSE
                 SPDAB[LI.TOLL]= DISTANCE/LI.TIME1*60
                 DISTAB[LI.TOLL]=DISTANCE
                 TIMEAB[LI.TOLL]=LI.TIME1
                 TT1[LI.TOLL]=   SPDAB[LI.TOLL]/DECEL(1,SPDAB[LI.TOLL])/0.6
```

```
                DD1[LI.TOLL]=   SPDAB[LI.TOLL]*0.5*TT1[LI.TOLL]/60
              ENDIF
           ENDIF
       TD=(TT1[LI.TOLL]-((DD1[LI.TOLL]/SPDAB[LI.TOLL])*60))/100
       T0=TD+T0
       TDEC[LI.TOLL]=TD
       TIMT1[LI.TOLL]=LI.TIME1
       VTPLT1[LI.TOLL]=V
       LISPD1[LI.TOLL]=SPDAB[LI.TOLL]
ENDIF

ENDPROCESS


PROCESS PHASE=ILOOP
       PATHLOAD  PATH=TIME, PENI= 1, VOL[1]= MI.1.1 EXCLUDEGROUP= 1    ; No toll or HOV
       PATHLOAD  PATH=TIME, PENI= 1, VOL[2]= MI.1.2 EXCLUDEGROUP= 2    ; No HOV
       PATHLOAD  PATH=TIME, PENI= 1, VOL[3]= MI.1.3 EXCLUDEGROUP= 1    ; No toll or HOV
       PATHLOAD  PATH=TIME, PENI= 1, VOL[4]= MI.1.4 EXCLUDEGROUP= 2    ; No HOV
       PATHLOAD  PATH=TIME, PENI= 1, VOL[5]= MI.1.5 EXCLUDEGROUP= 3    ; No toll
       PATHLOAD  PATH=TIME, PENI= 1, VOL[6]= MI.1.6
ENDPROCESS

PROCESS PHASE=ADJUST

IF (LI.TOLLTYPE>0 & LI.TOLLTYPE<> 2)
        LW.ARRIVR=(V/LI.UROAD_FAC)*LI.CONFAC/LI.PLZALNSMAX  ; hourly volume per toll lane ie. arrival rate
;in vehicles per hour
        LW.SERVT=LI.SVCMINUTES+(LI.SVCSECONDS/60)           ; plaza lane service time in minutes per
;vehicle
        IF (LW.SERVT=0) LW.SERVT=0.0001
        LW.SERVR=(1/LW.SERVT)*60                            ; plaza lane service rate in vehicle per hour
        IF (LW.ARRIVR>=LW.SERVR) LW.ARRIVR=0.99*LW.SERVR    ; prevent infinite or negative queue
ENDIF

IF (ITERATION>0)
;*********************************************************************
;    ROUTINE TO CALCULATE THE DELAY AT A TOLL FACILITY (SETOLL.FOR)
;*********************************************************************
;Deceleration Lane (AB)
   IF (LI.TOLL>0 & LI.TOLL_ACC=0 & LI.TOLL_DEC=1)
```

```
      IF (LI.TIME1=0)
         TT1[LI.TOLL]=0
         DD1[LI.TOLL]=0
         SPDAB[LI.TOLL]=1
      ELSE
         SPDAB[LI.TOLL]= DISTANCE/LI.TIME1*60
         DISTAB[LI.TOLL]=DISTANCE
         TIMEAB[LI.TOLL]=LI.TIME1
         TT1[LI.TOLL]=   SPDAB[LI.TOLL]/DECEL(1,SPDAB[LI.TOLL])/0.6
         DD1[LI.TOLL]=   SPDAB[LI.TOLL]*0.5*TT1[LI.TOLL]/60
      ENDIF
   ENDIF
;Acceleration Lane (CD)
   IF (LI.TOLL>0 & LI.TOLL_ACC=1 & LI.TOLL_DEC=0)
      IF (LI.TIME1=0)
         TT3[LI.TOLL]=0
         DD3[LI.TOLL]=0
         SPDCD[LI.TOLL]=1
      ELSE
         SPDCD[LI.TOLL]= DISTANCE/LI.TIME1*60
         DISTCD[LI.TOLL]=DISTANCE
         TIMECD[LI.TOLL]=LI.TIME1
         TT3[LI.TOLL]=   SPDCD[LI.TOLL]/2.5/0.6
         DD3[LI.TOLL]=   SPDCD[LI.TOLL]*0.5*TT3[LI.TOLL]/60
      ENDIF
   ENDIF

; Toll Plaza (BC)
IF (LI.TOLL>0 & LI.TOLL_ACC=0 & LI.TOLL_DEC=0)
    ITER[LI.TOLL]=ITERATION
    ALINK[LI.TOLL]=A
    BLINK[LI.TOLL]=B
    VOLBC[LI.TOLL]=V
    TT4[LI.TOLL]= CTOLL*LI.CARTOLL*60*100
ENDIF
IF (LI.TOLLTYPE>0 & LI.TOLLTYPE<> 2)
   IF ({QUEUETYPE}=1) ; MULTIPLE SERVER QUEUE MODEL
      LAM[LI.TOLL]= VOLBC[LI.TOLL]
      MU[LI.TOLL] = 1/(LW.SERVT/60)
      UTIL[LI.TOLL]=LAM[LI.TOLL]/MU[LI.TOLL]
      UTLSER[LI.TOLL]=UTIL[LI.TOLL]/LI.PLZALNSMAX
```

```
      SER[LI.TOLL]=LI.PLZALNSMAX

      PP0[LI.TOLL]=1
      REALN[LI.TOLL]=1
      NLANE1=LI.PLZALNSMAX-1
      RATIO[LI.TOLL]=1

      IF (UTLSER[LI.TOLL]>0.99) ; CHECK AGAINST MAXUTL
        LAM[LI.TOLL]= MU[LI.TOLL]*SER[LI.TOLL]*0.99
        RATIO[LI.TOLL]= VOLBC[LI.TOLL]/LAM[LI.TOLL]
        UTIL[LI.TOLL]= LAM[LI.TOLL]/MU[LI.TOLL]
        UTLSER[LI.TOLL]= UTIL[LI.TOLL]/SER[LI.TOLL]
      ENDIF


      LOOP NN=1,NLANE1
          REALI=NN
          REALN[LI.TOLL]= REALN[LI.TOLL]* REALI
          PP0[LI.TOLL]=PP0[LI.TOLL]+(UTIL[LI.TOLL]^REALI)/REALN[LI.TOLL]
      ENDLOOP

      PP0[LI.TOLL]=1/(PP0[LI.TOLL]+(UTIL[LI.TOLL]^SER[LI.TOLL])/(REALN[LI.TOLL]*SER[LI.TOLL]*(1-
UTLSER[LI.TOLL])))
      TT2[LI.TOLL]= PP0[LI.TOLL] * UTIL[LI.TOLL]^SER[LI.TOLL] * UTLSER[LI.TOLL]
      TT2[LI.TOLL]= TT2[LI.TOLL] / (SER[LI.TOLL]* REALN[LI.TOLL] *((1-UTLSER[LI.TOLL])^2))

      IF (LAM[LI.TOLL]=0)
         TT2[LI.TOLL]= 6000/MU[LI.TOLL]
      ELSE
         TT2[LI.TOLL]=(TT2[LI.TOLL]/LAM[LI.TOLL]*RATIO[LI.TOLL]+1/MU[LI.TOLL])*6000
      ENDIF
      DD2[LI.TOLL]=0
   ELSE  ; SINGLE SERVER QUEUE MODEL
      LAM[LI.TOLL]=VOLBC[LI.TOLL]/LI.PLZALNSMAX
      MU[LI.TOLL]= 1/(LW.SERVT/60)

      IF (LAM[LI.TOLL]>(0.99*MU[LI.TOLL])) LAM[LI.TOLL]=(0.99*MU[LI.TOLL]); CHECK AGAINST MAXUTL

      UTIL[LI.TOLL]=LAM[LI.TOLL]/MU[LI.TOLL]
      AVEQUE[LI.TOLL]=MU[LI.TOLL]/(MU[LI.TOLL]-LAM[LI.TOLL])
      DD2[LI.TOLL]= AVEQUE[LI.TOLL]*35/5280*100; CARLTH=35 FEET
```

```
       TT2[LI.TOLL]=1/(MU[LI.TOLL]-LAM[LI.TOLL])*6000
   ENDIF

   IF (TT2[LI.TOLL]>1333.33) TT2[LI.TOLL]=1333.33
    DELAY[LI.TOLL]=(TT1[LI.TOLL]+TT2[LI.TOLL]+TT3[LI.TOLL]+TT4[LI.TOLL]- (
(DD1[LI.TOLL]/SPDAB[LI.TOLL]+DD3[LI.TOLL]/SPDCD[LI.TOLL]) *60))/100
ENDIF
ENDIF

FUNCTION {
     V=VOL[1] + VOL[2] + VOL[3] + VOL[4] + VOL[5] + VOL[6]

      TC[1]= (T0*(1.0 + LI.BPR_COEFF*(V/C)^LI.BPR_EXP) - T0)*0.25 + T0
      TC[2]= LI.CARTOLL*CTOLL*CTOLLSCALE
      TC[3]= (TT2[LI.TOLL]+TT4[LI.TOLL])/100; delay for toll links, considering type of Queue Model
      TC[4]= LI.TIME1+((TT3[LI.TOLL]-((DD3[LI.TOLL]/SPDCD[LI.TOLL])*60))/100); congested time toll
;acceleration links
      TC[5]= LI.TIME1+((TT1[LI.TOLL]-((DD1[LI.TOLL]/SPDAB[LI.TOLL])*60))/100); congested time toll
;deceleration links

}

ENDPROCESS

ENDRUN
```

Appendix 3-A: TRANPLAN – RAMP-TO-RAMP TOLLS

The following is the TRANPLAN script for this test:

```
$EQUILIBRIUM HIGHWAY LOAD
$FILES
   INPUT FILE = HWYNET, USER ID = $HIGHWAY.NET$
   INPUT FILE = HWYTRIP, USER ID = $2015_CFIG_FinalTrips.TRP$
   INPUT FILE = TRNDATA, USER ID = $TurnFile.trn$
   OUTPUT FILE = LODHIST, USER ID = $DAILY_LOADED_NETWORK.NET$
   OUTPUT FILE = TOLLFILE, USER ID = $DAILY_LOADED_NETWORK.TOL$
$HEADERS
     TEST OF RAMP-TO-RAMP -- SINGLE TOLL FACILITY
$OPTION
   MINIMUM UNCONNECTED ZONES
   TOLL DETAIL
   TURN FILE
   TOLL FILE
   RAMPDBF FILE
$PARAMETERS
   DETAIL RAMPS = 18557-18924, 18913-18545
   CTOLL = 0.04
   CTOLL SCALE FACTOR = 100.0
   UROAD FACTOR = 0.75
   IMPEDANCE = TIME2
   EPS = 0.01
   EQUILIBRIUM ITERATIONS = 10
   COMPANY NAME = 'Goldenrod Extension'
   COMPANY NUMBER = 15
   RAMP LINK = 18557-18579, RAMPID = 'Goldenrod Extension NB On at SR 528'
   RAMP LINK = 18778-18924, RAMPID = 'Goldenrod Extension NB Off at Leevi'
   RAMP LINK = 18913-18757, RAMPID = 'Goldenrod Extension SB On at Leevis'
   RAMP LINK = 18575-18545, RAMPID = 'Goldenrod Extension SB Off at SR 52'
   RAMP TOLL = 0.3800, RAMP FROM = 18557-18579, RAMP TO = 18778-18924
   RAMP TOLL = 0.3800, RAMP FROM = 18913-18757, RAMP TO = 18575-18545
$END TP FUNCTION
```

Appendix 3-A: VOYAGER – RAMP-TO-RAMP TOLLS

The following is the Voyager script for this test:

```
; Do not change filenames or add or remove FILEI/FILEO statements using an ;editor. Use Cube/Application Manager.
RUN PGM=HIGHWAY PRNFILE="PBHWY00E.PRN" MSG='Path Based Tolling Example - Ramp to Ramp Test - Simple Toll'
FILEO PATHO[1] = "ASSIGNPATH.PTH"
FILEI NETI = "GoldenRod.NET"
FILEI MATI[1] = "2015_CFIG_FINALTRIPS.MAT"
FILEI TOLLMATI[1] = "Goldenrod.DBF",
   ENTRYGATE=ON_RAMP, EXITGATE=OFF_RAMP, TOLLS=COST_MODE1,
   NETIENTRY=ONRAMP, NETIEXIT=OFFRAMP, NETITOLLROAD=TOLLROAD, LIST=Y
FILEO NETO = "SIMPLE TOLL.NET"
FILEO MATO[1] = "PBHWY00D.MAT",
MO=1-4, NAME=PATHCOST,PATHTOLL,TOLLDIST,VEHTOLL

PAR GAP=0.01

PROCESS PHASE=LINKREAD
IF  (li.tollroad>=1) LW.TOLLDIST=LI.DISTANCE
C = LI.CAPACITY*0.75
LW.COEF=0.15
LW.EXPO=4.00
ENDPROCESS

PROCESS PHASE=ILOOP
; ---------------------------------------------------------
; trips to be loaded
MW[101]=MI.1.1
; ---------------------------------------------------------
PATHLOAD PATH=COST, VOL[1]=MW[101]*1.00,
                     TOLLMATI=1,    ; TOLL FILE 1
                          TOLLFACTOR=2.4,
                     MW[1]=PATHCOST, NOACCESS=0,
                     MW[2]=PATHTOLL,
                     MW[3]=PATHTRACE(LW.TOLLDIST),NOACCESS=0,
                     PATHO=1, NAME='TOLLA', ALLJ=T, INCLUDECOSTS=T, FULLPATH=T
                     MW[4]=MW[2]*mi.1.1
ENDPROCESS

PROCESS PHASE=ADJUST
function {tc=(t0*(1.0+LW.COEF*((v/c)^LW.EXPO))-t0)*0.25 + t0}

ENDPHASE

ENDRUN
```