Title

Ramp Metering Design Tools and Field Test Implementation of Queue Control

Authors
Sanchez, Rene O.
Gomes, Gabriel
Horowitz, Roberto
et al.

Publication Date
2010-04-01

CALIFORNIA PATH PROGRAM
INSTITUTE OF TRANSPORTATION STUDIES
UNIVERSITY OF CALIFORNIA, BERKELEY

Ramp Metering Design Tools and Field Test Implementation of Queue Control

**Rene O. Sanchez, Gabriel Gomes,
Roberto Horowitz, Pravin Varaiya**

California PATH Research Report
UCB-ITS-PRR-2010-16

This work was performed as part of the California PATH Program of the University of California, in cooperation with the State of California Business, Transportation, and Housing Agency, Department of Transportation, and the United States Department of Transportation, Federal Highway Administration.

Final Report for Task Order 6329

April 2010
ISSN 1055-1425

T.O. 6329: Ramp Metering Design Tools & Field Test Implementation of Queue Control. Final Report.

Rene O. Sanchez, Gabriel Gomes, Roberto Horowitz, and Pravin Varaiya

Abstract

PATH Task Order 6329 has as its two main goals 1) to create a software tool which can be used to design and test ramp metering systems for freeways, and 2) to prepare necessary software to conduct a field test of a queue control algorithm developed under T.O. 5503. Both of these objectives are directed towards improving the effectiveness of ramp metering strategies for freeways by using advanced technologies in traffic sensors (Sensys), data storage (PeMS), and ramp metering algorithms (Alinea, SWARM).

The Ramp Metering Design Tool (RMDT) has been designed to meet the first of these goals. The RMDT is a Matlab based software tool that guides the user through a clearly defined sequence of steps leading from data collection (using PeMS), through calibration, to simulation. The final outcome of the process is a calibrated model of the freeway which can be used to test different operational strategies, such as ramp metering. The Ramp Metering Design Tool was tested by Jan Hueper, an exchange student from the Institute for Transport und Automatisierungstechnik at the Leibniz Universitaet in Hannover who had no prior experience with the software or background in traffic theory, to successfully construct a calibrated model of the I-80 East freeway, in a period of about 4 months. The step by step I-80 East model construction and calibration process is described.

Traffic controllers were evaluated and the 2070 controller was chosen to be used for the field test. The Hegenberger Rd. loop on-ramp to S/B 880, in District 4, was selected as the test site. A 2070 controller was acquired and the Universal Ramp Metering Software (URMS) source code was obtained, studied, and discussed with its developer to identify required modifications for the inclusion of ramp metering and queue control algorithms. In addition, the Sensys wireless vehicle detection system was analyzed and meetings were held with Sensys Networks Inc. personnel to determine the sensor needs for the project. Finally, a new hardware-in-the-loop on-ramp simulation tool was developed to debug and test the Universal Ramp Metering Software.

Keywords: Computer Simulation, Ramp Metering, Traffic Control

Executive summary

The goal of T.O. 6329 project was to develop, along parallel and independent tracks: 1) A Ramp Metering Design Tool (RMDT) and 2) Software tools to conduct a Ramp Metering with Queue Control Field Test.

Track 1: Ramp Metering Design Tool

The Ramp Metering Design Tool has been completed. The following software design tools where integrated:

- **A computationally efficient macroscopic traffic-flow modeling tool that self-calibrates**, using on-ramp and mainline traffic data, and runs at a significantly faster rate than real-time.
- **A tool for evaluating traffic-responsive on-ramp control strategies**, including localized strategies such as ALINEA and the recently developed switching LQI strategy in TO4136/5503, and coordinated strategies such as SWARM, on macroscopic traffic-flow models.

  These tools have been integrated into a larger set of Tools for Operational Planning (TOPL), which is being further developed by our research team under CALTRANS projects TOPL-3 and I-80 Freeway Corridor Monitoring and Control. TOPL is a suite of software tools for specifying freeway corridors operational improvement strategies, such as ramp metering, demand and incident management, and for quickly estimating the benefits of such improvements. Currently, TOPL can model both freeways and arterial traffic flow [9].

Track 2: Software tools to conduct a Ramp Metering with Queue Control Field Test

Traffic controllers were evaluated and the 2070 controller was chosen to be used for the field test. The Hegenberger Rd. loop on-ramp to S/B 880, in District 4, was selected as the test site and a study of its traffic dynamics and queue formation was performed. Furthermore, a 2070 controller was acquired and it was possible to write, compile and load applications using its OS-9 operating system. Moreover, the Universal Ramp Metering Software (URMS) source code was obtained, studied, and discussed with its developer to identify required modifications for the inclusion of ramp metering and queue control algorithms. In addition, the Sensys wireless vehicle detection system was analyzed and meetings were held with Sensys Networks Inc. personnel to determine the sensor needs for the project. For the sensor installation planning, Republic ITS personnel was

consulted and a formal quote was received from them. The report also includes modifications of the methodology of Track 2 of PATH T.O. 6329 RFP. Finally, a new hardware-in-the-loop on-ramp simulation tool was developed to debug and test the Universal Ramp Metering Software.

This work will be continued under RTA-22A0486 Field Test Implementation of Queue Control - Track 2. A new queue control scheme has been developed as an alternative to queue override, which eliminates queue and mainline density oscillations, and can be incorporated into most on-ramp strategies, to enhance their performance when queue length constraints are imposed. A field test will be conducted to implement Queue Control on the Hegenberger Rd. loop on-ramp to S/B 880, in District 4 and study its effect in minimizing queue and mainline density oscillations and enhancing performance. This will be accomplished by using a 2070 traffic controller running a modified Universal Ramp Metering Software (URMS) and utilizing realtime onramp queue counts provided by a Sensys wireless vehicle detection system.

Contents

Introduction

PATH Task Order 6329 has as its two main goals 1) to create a software tool which can be used to design and test ramp metering systems for freeways, and 2) to prepare necessary software in order to conduct a field test of a queue control algorithm developed under T.O. 5503. Both of these objectives are directed towards improving the effectiveness of ramp metering strategies for freeways by using advanced technologies in traffic sensors (Sensys), data storage (PeMS), and ramp metering algorithms (Alinea, SWARM).

The Ramp Metering Design Tool (RMDT) has been designed to meet the first of these goals. The RMDT is a Matlab based software tool that guides the user through a clearly defined sequence of steps leading from data collection (using PeMS), through calibration, to simulation. The final outcome of the process is a calibrated model of a freeway which can be used to test different operational strategies, such as ramp metering. This tool has been integrated into a larger set of Tools for Operational Planning (TOPL), which is being further developed by our research team under CALTRANS projects TOPL-3 and I-80 Freeway Corridor Monitoring and Control. TOPL is a suite of software tools for specifying freeway corridors operational improvement strategies, such as ramp metering, demand and incident management, and for quickly estimating the benefits of such improvements. Currently, TOPL can model both freeways and arterial traffic flow [9].

Toward the second goal we have worked on issues surrounding the deployment of an innovative ramp queue measurement and regulation scheme. Most of the research on the topic of freeway ramp metering has focused on the problem of how to select metering rates such that the throughput on the mainline is kept high. However most of these strategies do not account for the fact that their prescribed rates will be ignored if the queue of vehicles waiting on the ramps spills over into the surface streets. When this happens, the controller enters a state of *queue override*, in which the green rate is increased until the onramp queue length has been reduced to an acceptable size. This queue-override scheme can be viewed as an integral control with a saturated integrating rate and resetting. It has been noted by several researchers [14, 27, 30, 28, 29] that this queue-override scheme leads to oscillatory behavior and underutilization of on-ramp storage capacities. Without an estimate of the current length of the onramp queue, it is not possible to eliminate this oscillatory behavior. In some cases, the inefficiencies of the queue reduction strategy can eliminate any benefits produced by the ramp metering system.

Several researchers have addressed this problem. Gordon attempted to improve the queue-override performance by filtering the occupancy signal and reducing the sampling time interval. Smaragdis and Papageorgiou proposed a proportional controller that relies on the on-ramp vehicle demands. Unfortunately, real-time on-ramp vehicle demand measurements are currently not available in the field, and historical demands, which may greatly deviate from the actual demands, have to be used to realize the controller proposed in [27].

In [28] an innovative queue control scheme is proposed, which is based on estimating the queue length by measuring the vehicle speed using dual queue detectors and devising a Proportional-Integral (PI) controller to regulate the queue length. The additional proportional action in the controller damps the undesirable oscillation in the queue length, while the integral action helps in rejecting the disturbance caused by the unpredicted vehicle arrival rate. To realize the proportional action of the controller, a queue-length estimator was designed based on a simplified model for the driving behavior of a vehicle that is approaching the end of the queue: the vehicle decelerates at a constant rate from its cruising speed, until it stops. The queue controller was tested on a calibrated VISSIM micro-simulation model of a 14-mile (23 km) section of Interstate 210 West (postmiles 25-40) in southern California [13, 12] and the results obtained were extremely promising. However, to further verify the efficacy of this scheme, it is necessary to conduct actual filed tests.

Part I

Ramp metering design tool

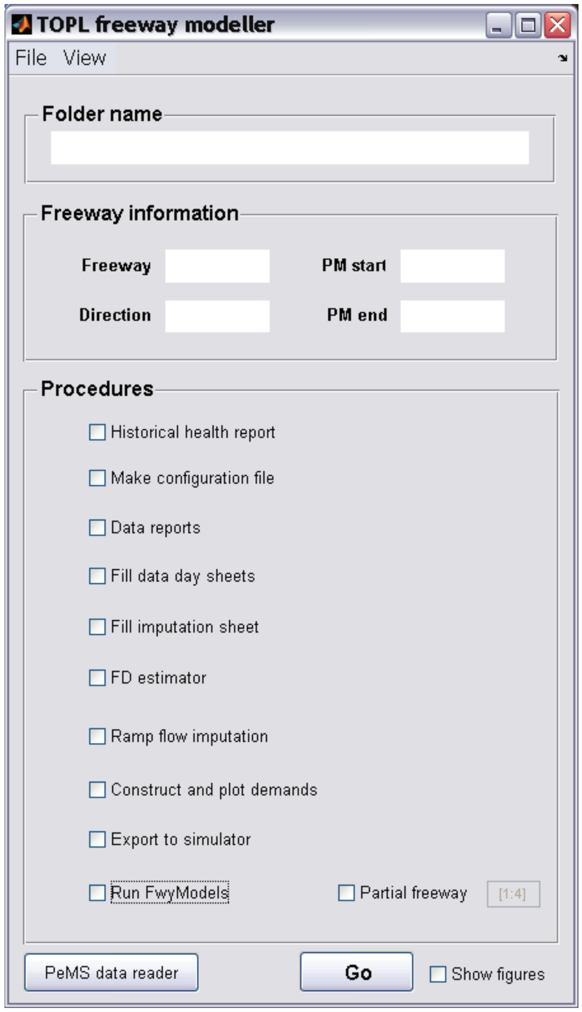

Figure 1: Main interface of the Ramp Metering Design Tool.

The Ramp Metering Design Tool (RMDT) is a computer program based in Matlab which facilitates the construction and calibration of cell-based models of freeways[1]. The main interface of the RMDT, shown in Figure 1, is a control panel from which each of the model construction tasks can be launched. These tasks are:

1. Define the test site. The starting point for the process is to pick a freeway, a direction, and

[1]The RMDT was developed under both T.O. 6329 and the Tools for Operations and Planning (TOPL, T.O. 6611). In the context of TOPL it is referred to as the TOPL Freeway Modeller (TOPL-FM).

limiting absolute postmiles.

2. Assess the state of the loop detector infrastructure. This step involves a historical analysis of PeMS' *detector health index*.

3. Select a set of days on which to base the calibration of the freeway. Ideally the calibration should be based on a single *normal* day with perfect detection, however it is more likely that several days will be needed to construct a complete data set. RMDT helps the user to manage the large amounts of data needed to assemble the demand profiles.

4. Partition the freeway into segments or *cells*, each containing at most one onramp and one offramp. RMDT suggests a partition based on PeMS' configuration data, and allows the user to make adjustments.

5. Calibrate the parameters of the traffic model: capacity, freeflow speed, jam density, and congestion wave speed. RMDT includes a fundamental diagram data fitting routine which computes these parameters for each cell.

6. Impute flows for unmeasured onramps and offramps. Unmeasured ramps are ramps with faulty detectors or no detectors. The flow on these ramps must be estimated in order to create a complete picture of the demand. The program includes a ramp imputation algorithm based on concepts of artificial learning.

7. Simulation and comparison with the measurements. The quality of the calibrated model is determined by comparing the output of the simulation with the measured data. This comparison is made by RMDT in terms of performance measures, such as travel time, delay, and productivity, that are also provided by PeMS.

Section I.1 through I.3 give an overview of the theory underlying the RMDT. The modeling paradigm applied throughout the RMDT is a macroscopic traffic model called the Link-Node Cell Transmission Model (LN-CTM). This model is identical to the original CTM [10] with respect to traffic flow within the links. However the strategy for merging and splitting traffic at nodes is somewhat changed, as explained in Section I.1. The procedures for computing parameters for the LN-CTM based on measured data and for imputing missing ramp flows are described in Sections I.2 andI.3 respectively. More detailed descriptions can be found in [11, 24]. Section I.4 documents a case study of the use of the RMDT. This section was written by Jan Hueper, an exchange student from the Institute for Transport und Automatisierungstechnik at the Leibniz Universitaet in Hannover, who used the RMDT to construct a model of I-80 East without any prior experience with the software or background in traffic theory. Appendix A is a user manual for the RMDT.

I.1 Link-node cell transmission model

The LN-CTM is an extension of the CTM which can be used to simulate traffic in any road network. The traffic network is represented as a directed graph of links. Links represent road segments and nodes are formed at the junctions of links. A time-varying split-ratio matrix is used to specify the portion of traffic moving from a particular input link to an output link. While a normal link connects two Nodes, a "source" link is used to introduce traffic whereas a "sink" is used to accept traffic moving out of the network. A source link implements a queue model. A fundamental diagram (which specifies the flow-speed-density characteristics) is specified for each link, while the source links are also specified with an input demand profile.

Figure 2 shows the directed graph representation of a freeway. The nodes specify the location of a merge between ramps and the mainline (freeway road segment). Each node contains a maximum of one on- and one off-ramp. In California freeways, the onramps are preceded by the offramps, therefore the split ratio matrix is specified to block any flow from the onramp to the offramp. Freeflow is assumed to prevail in both boundaries of the freeway.

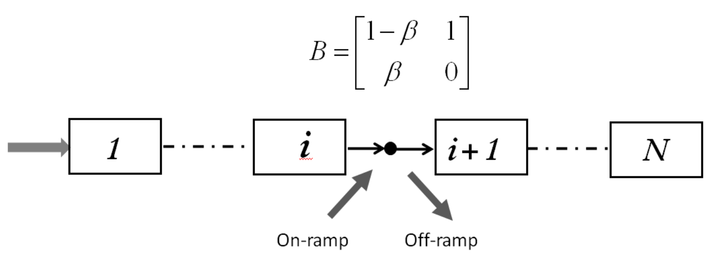

Figure 2: Freeway with N links. Each Node contains a maximum of one on and one off-ramp.

A "source" node attached to the upstream cell is used to introduce traffic flow into the network. The onramps are also represented as source links, while the offramps are represented as sinks. It is also assumed that the off-ramps are in freeflow. Table 1 lists the model variables and parameters.

The LN-CTM is explained in [19]. The algorithm can be represented as a four mode switching model for each link. At each time interval, each link can be in one of four modes - FF, CF, CC, and FC, where F denotes freeflow and C denotes congestion. These modes are decided based on the flow conditions existing at the input and output node of each link. The CF mode is said to be active in Link i , if the input flow into Link i is in congestion and the output flow from Link i (into Link $i+1$ and the offramp) is in freeflow. Other modes can be interpreted similarly. Here, the input into Link i is classified to be in congestion if the flow into it is determined (limited) by its available capacity rather than total input demand, i.e. $c_{i-1}(k) > \bar{w}_i(k)(n_i^J - n_i(k))$, where

Symbol	Name	Unit
F_i	capacity	veh/period
v_i	free flow speed	section/period
w_i	congestion wave speed	section/period
n_i^J	jam density	veh/section
β_i	split ratio	-
k	period number	-
$f_i^{in}(k)$	flow into section i period k	veh/period
$f_i^{out}(k)$	flow out of section i period k	veh/period
$s_i(k)$, $r_i(k)$	off-ramp, on-ramp flow in node i in period k	veh/period
$d_i(k)$	on-ramp demand for Link $i+1$ in period k	veh/period
$c_i(k)$	Total demand for Link $i+1$ in period k	veh/period

Table 1: Model variables and parameters

$c_{i-1}(k) = n_{i-1}(k)v_{i-1}(k)(1 - \beta_{i-1}(k)) + d_{i-1}(k)$ is the input demand into Link i, which consists of flow from Link $i-1$ and the onramp. $\bar{w}_i(k)(n_i^J - n_i(k))$ represents the available output capacity. Thus, in freeflow, the flow into the link is not limited by the available space, thereby allowing for the freeway to cater to the actual demand. However, in the case of congestion, the flow is limited by the capacity in the destination link, and the flow from the onramp and the previous link is scaled accordingly to limit the flow to the capacity. The density update equations for the links of the freeway can be summarized as

1. FF Mode

$$n_i(k+1) = n_i(k) + c_{i-1}(k) - n_i(k)\bar{v}_i(k) \tag{1}$$

2. FC Mode

$$n_i(k+1) = n_i(k) + c_{i-1}(k) - \bar{w}_{i+1}(n_{i+1}^J - n_{i+1}(k))n_i(k)\bar{v}_i(k)/c_i(k) \tag{2}$$

3. CC Mode

$$n_i(k+1) = n_i(k) + \bar{w}_i(n_i^J - n_i(k)) - \bar{w}_{i+1}(n_{i+1}^J - n_{i+1}(k))n_i(k)\bar{v}_i(k)/c_i(k) \tag{3}$$

4. CF Mode

$$n_i(k+1) = n_i(k) + \bar{w}_i(n_i^J - n_i(k)) - n_i(k)\bar{v}_i(k) \tag{4}$$

where $\bar{w}_i(k) = \min(w_i, F_i/(n_i^J - n_i(k)))$ and $\bar{v}_i(k) = \min(v_i, F_i/n_i(k))$. In addition, the mainline flows can be determined by

$$f_i^{out}(k) = \frac{\min(c_i(k), \bar{w}_{i+1}(k)(n_{i+1}^J - n_{i+1}(k)))}{c_i(k)} n_i(k)\bar{v}_i(k) \tag{5}$$

$$f_i^{in}(k) = \min(c_{i-1}(k), \bar{w}_i(k)(n_i^J - n_i(k))) \tag{6}$$

while the ramp flows are determined by

$$s_i(k) = \beta_i(k) f_i^{out}(k) \tag{7}$$

$$r_i(k) = \frac{\min(c_i(k), \bar{w}_{i+1}(k)(n_{i+1}^J - n_{i+1}(k)))}{c_i(k)} d_i(k) \tag{8}$$

$$d_i(k+1) = d_i(k) + fl_i^{in}(k+1) - r_i(k) \tag{9}$$

where fl_i^{in} is the input flow for the onramp i.

The above set of equations correctly represent the LN-CTM for freeways under a few assumptions. It is assumed that the offramp flows are not restricted by the flow capacity/congestion. Similarly onramps are also assumed to have no flow capacity restrictions. These assumptions are not restrictive for modeling purposes, since the offramps are usually specified to be in freeflow, while the capacity of the onramps will not affect the flow calculations if the capacity is defined to be the maximum flow observed in the ramps.

I.2 Automatic model calibration

Like most macroscopic models of vehicular traffic flow, the LN-CTM makes use of the fundamental diagram, an empirical curve relating observed densities to observed flows at a particular point on the road. A calibrated fundamental diagram provides freeflow speed, congestion wave speed, critical density, jam density and capacity (Figure 3). The calibration procedure involves the following.

I.2.1 Freeway Representation

The first step is to define the geometrical characteristics of the site- the locations of onramps and offramps, number of lanes, existence of HOV lanes etc. As the LN-CTM dictates, the freeway should be represented in the form of successive cells. Therefore, the freeway network is divided into cells each with at most one on- and/or off-ramp and one mainline vehicle detector station[2]. The cells must be longer than the freeflow travel distance, i.e. $v_i \leq 1$ so that the algorithm converges. Each

[2]This is done automatically by the RMDT

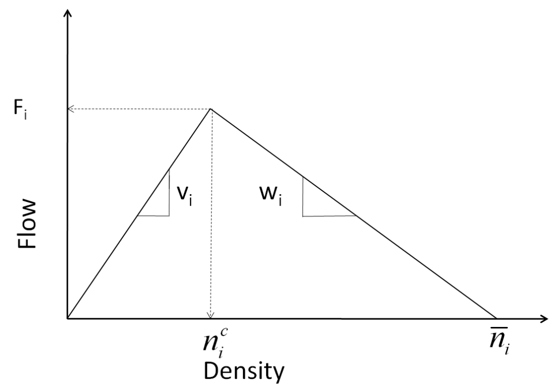

Figure 3: Fundamental diagram for a freeway section.

cell is assumed to be homogeneous in terms of number of lanes, grade and geometrical features so that each cell can be represented by a single fundamental diagram.

I.2.2 Data Acquisition and Selection

Vehicle detector stations(VDS) contain loop detectors that provide flow and occupancy data. PeMS [25] processes and archives these data in form of time series over different days of operation. PeMS also reports detector performance for each day of operation. The RMDT helps the user to collect, process, and visualize the data so that a set of days can be chosen with a relatively high rate of observation. Once the data set has been acquired, it is passed along to the calibration and ramp imputation subroutines.

I.2.3 Calibration of the free-flow speed, v_i

The free-flow speed for each section i, v_i, is estimated by performing a least squares fit on the flow and density data at the time instants where the speed was reported to be above 55 mph. A sample regression line for the free-flow speed can be seen in Figure 3.

I.2.4 Estimation of section capacity, F_i

In the fundamental diagram, the apex of the triangle corresponds to the section capacity and it is the highest observed flow. In fact, the definition and choice of capacity is a rather delicate point. The Highway Capacity Manual [5] defines capacity as the maximum amount of flow that can reasonably be expected to traverse the cross-section of a road segment. This deterministic notion of capacity

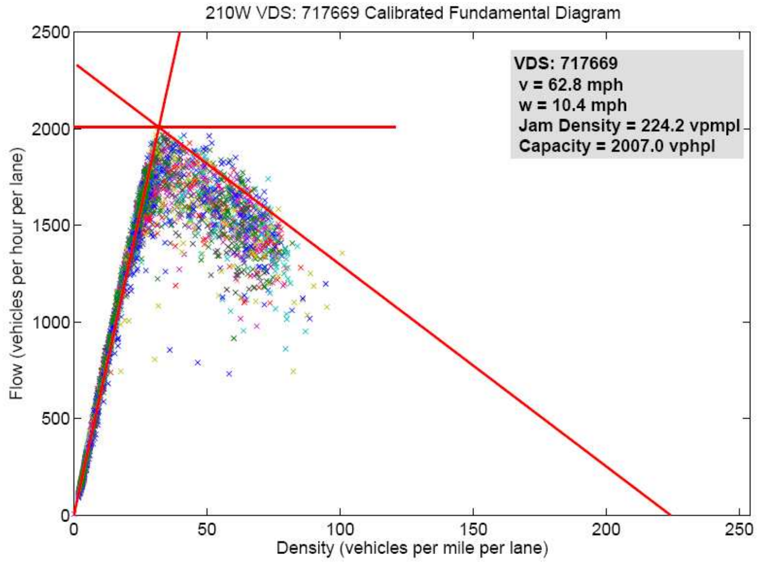

Figure 4: Calibrated Flow vs. Density Scatter Plot of vds 717669 on I-210W over 15 days data.

has been challenged lately by stochastic approaches [32]. A study over the sections of the freeway yields that there is indeed a significant variation in observed maximum flows (Figure 5).

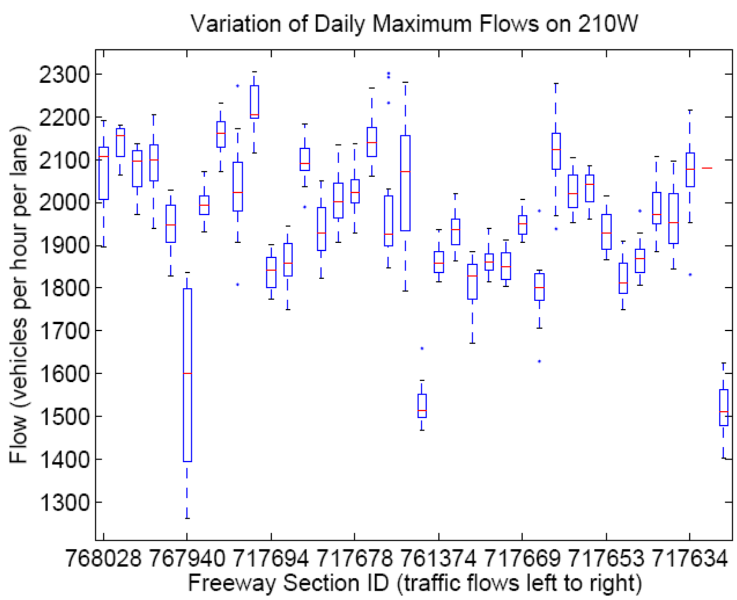

Figure 5: Box plots showing the distribution of observed daily maximum flows for each section of I-210W.

In Figure 5, the horizontal axis is the detector IDs placed on the freeway in upstream to downstream (left to right) succession. The vertical axis reflects the normalized flow values across these sections of the freeway. The horizontal lines inside the boxes correspond to the median of the observed daily maximums among days. The lower and upper box boundaries represent 1st and 3rd quartiles, or 25th and 75th percentiles, respectively. The whiskers span from either end of the box to the smallest and largest data points that are non-outliers, i.e points within 1.5 interquartile range away from box boundaries. The figure reflects significant temporal and spatial variation of section capacities. The stochastic approach to capacity is based on the notion of breakdown, which describes the operation of a freeway near a bottleneck at a time instance where there is a change from free-flow to congestion [1]. Numerous studies on the stochastic nature of capacity [32] suggest that the breakdown occurs randomly, affected by various external factors such as driver behavior, road and weather conditions, incidents, etc. and capacity can be defined as a random variable with a specific probability distribution depending on the probability of breakdown.

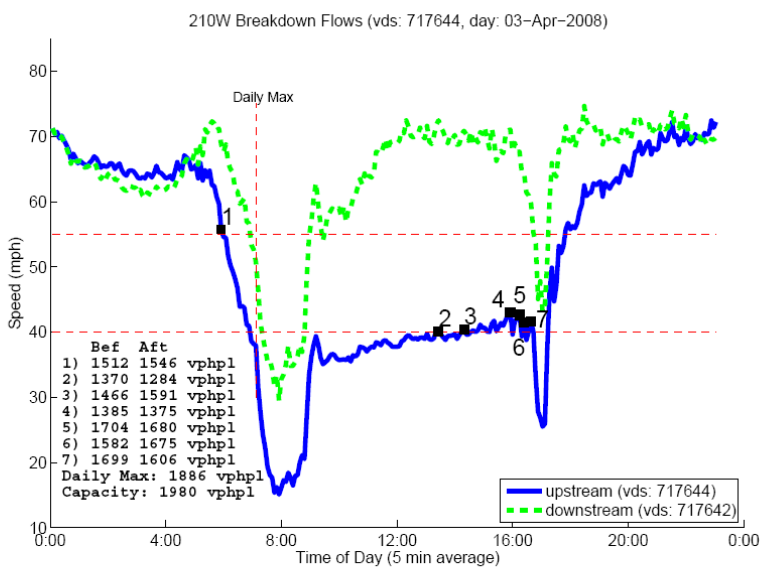

Figure 6: Speed plots of sections 717644 and 717642 on Apr 3rd, 2008

Figure 6 (and many others pertaining to other sections of the freeway not included here) suggests that although related to the capacity of a section, the breakdown analysis is not suitable to the purposes of the capacity estimation in the fundamental diagram framework, since the breakdown flow values differ substantially from the observed maximum flows. A more detailed analysis on the variation of capacity and its comparison to breakdown can be found in [11].

The capacity estimate for model calibration is thus chosen deterministically to be the highest observed flow throughout all investigated days. The choice of this largest observed flow as the estimate of capacity is based on the assumption that external factors such as driver behavior, incidents, weather and road conditions always affect the capacity adversely and the actual capacities of freeway sections are rarely observed, if ever. This capacity estimate enables the model to replicate the ideal operating conditions of the freeway and is also essential to the testing of hypothetical control strategies, such as ramp metering. This maximum value of flow across the section is then projected horizontally to the free-flow line, to establish the tip of the triangular fundamental diagram (Figure 4). The intersection is defined as the critical density for the section, above which the flow is congested.

I.2.5 Calibration of the Congestion Speed Parameter, w_i

The last parameter to be calibrated is the congestion speed parameter, w_i, which also defines the jam density for the section. Similar to capacity, this parameter shows significant variation. Therefore an approximate quantile regression [18] was adopted to estimate this parameter at the higher end of its distribution. After the critical density is determined, the flow-density points with density values higher than the critical density (the data points to the right of the tip) are partitioned along the horizontal axis (density axis) into non-overlapping bins of 10 data points each. Horizontally, each bin is summarized by "BinDensity", the mean of the 10 density values in the bin. Vertically, each bin is summarized by "BinFlow", the largest non-outlier flow values among the the 10 flow values in the bin. Formally, this largest non-outlier is determined as follows:

$$Bin = \{f_1, f_2, \ldots, f_{10}\} \tag{10}$$

$$BinFlow = \max_{f_i}(f_i | f_i \in Bin, f_i < Q_3 + 1.5IQR) \tag{11}$$

where, f_1 through f_{10} are the flow values inside one such bin, Q_3 is the 75th percentile of the data points in the bin and IQR is defined as the difference between the 25th percentile and the 75th percentile of the data.

A constrained least-squares regression is performed on these BinDensity - BinFlow pairs to obtain the congested flow line and complete the fundamental diagram picture (Figure 4). It is required that the regression line passes through the tip of the fundamental diagram, so the regression is constrained accordingly. The point where the regression line crosses zero flow is chosen as the jam density of the section.

I.3 Imputation of missing ramp flows

The LN-CTM model is utilized to impute the missing onramp input flows as well as the off-ramp split ratios for one day (24-hour) traffic flow simulation on a large freeway (eg. 40 miles) segment.

The imputation procedure involves two stages. First, the total demands c_i are determined using an adaptive learning procedure that minimizes the error between the model calculated densities and the observed PeMS density profile. Then the demands and split-ratios are extracted from the total demand, using a linear program that minimizes the error between the model calculated flows and the observed flow profile [24].

The imputation procedure employs the adaptive iterative learning procedure described in [33]. The freeway traffic flow process is assumed to be 24-hour periodic, with respect to the flow and density profiles. This assumption is valid, since the freeway is always found to be in freeflow (with low densities, flows) at midnight. The LN-CTM algorithm is run multiple times, and at each run, the algorithm adapts the unknown demand estimates to minimize the error between the density generated by the model at each link and the data from the corresponding PeMS measurement. The procedure is repeated until the density error reduces to a sufficiently small value or stops decreasing.

As detailed in [33], because of the 24 hour periodicity, the demand vector can be represented as a convolution of a kernel on a constant influence vector, i.e $c_i(k) = K(k)^T C_i$ where $K(k)$ represents a 24 hour periodic time dependent kernel vector, and C_i is the influence vector. Some typical kernel functions ($K(k)$) include a unit-impulse or a Gaussian window centered at time k.

The imputation procedure assumes initial estimates for the influence vectors $\hat{C}_i$. Typical initial estimates incorporate zero onramp and offramp flows. These estimates are then dynamically adapted at each time step, so that the model calculated densities for the whole freeway match with the density profiles obtained from PeMS. At each time step, the mode for each cell is determined, and the corresponding learning update equations are used to adapt the influence vectors. In the following parameter update equations $\hat{n}(k)$ represents density estimates, $\tilde{n}(k) = n(k) - \hat{n}(k)$ represents the density error, and $\tilde{n}^o(k)$ represents the a-priori error estimate.

(a) FF Mode

$$\tilde{n}_i^o(k+1) = \hat{n}_i(k+1) - (\hat{n}_i(k) + \hat{c}_i - 1(k) - \hat{n}_i(k)\hat{v}_i(k) - a\tilde{n}_i(k)) \tag{12}$$

$$\tilde{n}_i(k+1) = \frac{\tilde{n}_i^o(k+1)}{1 + GK^T(k)K(k)} \tag{13}$$

$$\hat{C}_{i-1}(k+1) = \hat{C}_{i-1}(k) + GK(k)\tilde{n}_i(k+1) \tag{14}$$

$$\hat{n}_i(k+1) = \hat{n}_i(k) + \hat{c}_{i-1}(k+1) - \hat{n}_i(k)\hat{v}_i(k) - a\tilde{n}_i(k) \tag{15}$$

(b) FC Mode

$$\tilde{n}_i^o(k+1) = \hat{n}_i(k+1) - \left(\hat{n}_i(k) - a\tilde{n}_i(k) + \hat{c}_{i-1}(k) - \frac{\hat{w}_{i+1}(n_{i+1}^J - \hat{n}_{i+1}(k))}{\hat{c}_i(k)}\hat{n}_i(k)\hat{v}_i(k)\right) \quad (16)$$

$$\tilde{n}_i(k+1) = \frac{\tilde{n}_i^o(k+1)}{1 + G'K^T(k)K(k) + GK^T(k)K(k)} \quad (17)$$

$$\hat{C}_{i-1}(k+1) = \hat{C}_{i-1}(k) + GK(k)\tilde{n}_i(k+1) \quad (18)$$

$$\hat{C}_i(k+1) = \hat{C}_i(k) - \frac{K(k)}{G''}\left(\hat{c}_i(k) - \frac{1}{1/\hat{c}_i(k) - G'K(k)\tilde{n}_i(k+1)}\right) \quad (19)$$

$$\hat{n}_i(k+1) = \hat{n}_i(k) - a\tilde{n}_i(k) + \hat{c}_{i-1}(k+1) - \frac{\hat{w}_{i+1}(n_{i+1}^J - \hat{n}_{i+1}(k))}{\hat{c}_i(k+1)}\hat{n}_i(k)\hat{v}_i(k) \quad (20)$$

(c) CF Mode

$$\hat{n}_i(k+1) = \hat{n}_i(k) + \bar{w}_i(n_i^J - \hat{n}_i(k)) - \hat{n}_i(k)\hat{v}_i(k) \quad (21)$$

(d) CC Mode

$$\tilde{n}_i^o(k+1) = \hat{n}_i(k+1) - \left(\hat{n}_i(k) - a\tilde{n}_i(k) + \hat{w}_i(n_i^J - \hat{n}_i(k)) - \frac{\hat{w}_{i+1}(n_{i+1}^J - \hat{n}_{i+1}(k))}{\hat{c}_i(k)}\hat{n}_i(k)\hat{v}_i(k)\right) \quad (22)$$

$$\tilde{n}_i(k+1) = \frac{\tilde{n}_i^o(k+1)}{1 + G'K^T(k)K(k)} \quad (23)$$

$$\hat{C}_i(k+1) = \hat{C}_i(k) - \frac{K(k)}{G''}\left(\hat{c}_i(k) - \frac{1}{1/\hat{c}_i(k) - G'K(k)\tilde{n}_i(k+1)}\right) \quad (24)$$

$$\hat{n}_i(k+1) = \hat{n}_i(k) - a\tilde{n}_i(k) + \hat{w}_i(n_i^J - \hat{n}_i(k)) - \frac{\hat{w}_{i+1}(n_{i+1}^J - \hat{n}_{i+1}(k))}{\hat{c}_i(k+1)}\hat{n}_i(k)\hat{v}_i(k) \quad (25)$$

where $G'' = K^T(k)K(k)$, $\hat{c}_i(k) = K(k)^T\hat{C}_i(k)$ and G, G' are positive gains. The parameter a is chosen so that the error equation is asymptotically stable. As the adaptation procedure is carried out, the 'error' in the density profile, given by $\sum |n_i(k) - \hat{n}_i(k)|$ decreases. Since the CF mode does not involve adaptation equations, the error may converge to a non-zero value for when this mode is in effect, while other modes shows negligible error. This occurs due to incorrect mode identification at that time instant. In this case, the corresponding estimates are "triggered" automatically so that the correct modes are identified. After the trigger, the adaptation procedure is continued, till the error becomes negligible or stops decreasing.

The above procedure identifies the Total demand vector, from with the on-ramp demand and off-ramp split ratios are decoupled using a linear program. Figure 7 illustrates the position of the mainline detector, from which flow data is available. Depending on the existing flow conditions, the flows preceding the offramp and following the onramp can be described by the equations presented in Figure 7. A linear program that minimizes $|f_{i+1}^{in}(k) - f_{i+1}^{meas}(k) - r_{i+1}(k)| + |f_i^{out}(k) - f_{i+1}^{meas}(k) - s_{i+1}(k)|$ can be used to identify the onramp and offramp flows that best match the observed mainline flow.

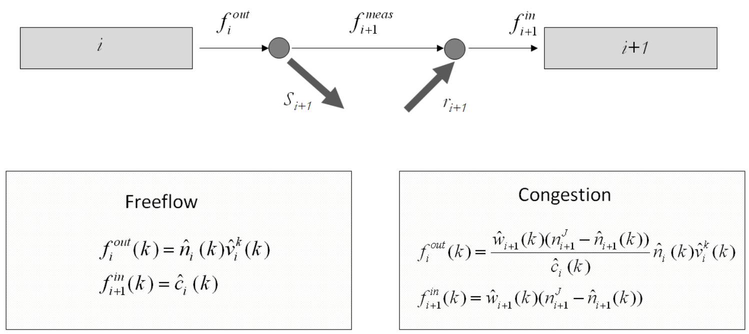

Figure 7: Decouple on-ramp and off-ramp flows.

I.4 I-80 example

This section has been taken from a report written by Jan Hueper [16], an exchange student who worked with our group for a period of 5 months. He was assigned the task of using the RMDT without previous experience with the software, the underlying theory, or PeMS. His report demonstrates that the software greatly simplifies the job of traffic model construction, which involves manipulation of large amounts of geometric and traffic data. It also demonstrates the effectiveness of the automatic model calibration and ramp imputation techniques described in the previous sections. The following is from the report.

This report illustrates the macroscopic modeling and simulation of Interstate 80 Eastbound in the Bay Area. Traffic flow and occupancy data from loop detectors is used for calibrating the model and specifying the inputs to the simulation. The model is calibrated based on the introduced link-node Cell Transmission Model and missing ramp flow data are estimated using an iterative learning-based imputation scheme. The simulation results using the calibrated model exhibit good agreement with loop detector measurements with total density errors of 3.1% and total flow error of 9.1% over the 23 mile stretch of the freeway under investigation and the particular day for which the ramp flows were imputed.

I.4.1 Calibration

The calibration of the model comprises two main steps: 1) The calibration of the fundamental diagram for each link of the freeway, 2) Estimation of ramp flows, which are essential inputs to the simulation but are not monitored by PeMS and thus have to be imputed using the mainline flow data.

The first step of the fundamental diagram estimation is to plot the available PeMS data in a flow-density diagram. This scatter plot readily reveals the typical shape of a fundamental diagram with its rising part at low densities and the falling part at densities over a certain critical density. Figure 3 shows the scatter plot and the fitted fundamental diagram for VDS 400443 on the studied freeway.

The estimation of the free flow parameter follows a simple linear regression whereas the congestion wave speed is estimated using an approximate quantile regression [11]. The capacity of the section is assigned as the maximum observed flow over the freeway segment. This VDS represents a typical cell for I-80 East with maximum flows around 2000 vehicles per hour per lane and a critical density around 35 vehicles per mile per lane. The estimated parameters for this particular section can be read off Figure 8.

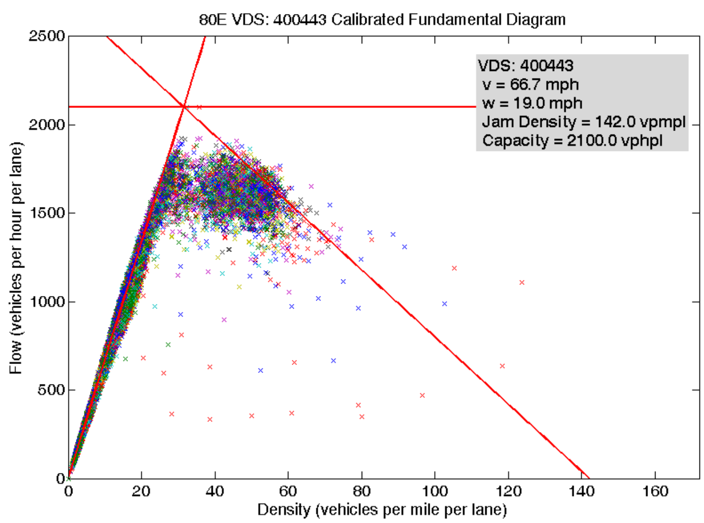

Figure 8: Scatter Plot and fitted Fundamental Diagram.

The second part of the model calibration is the estimation of the missing ramp flows. Generally, data-based macroscopic freeway modeling is constrained by missing data. Despite the widespread collection of induction loop data in California, the simulation of I-80 suffers from the fact that no on- or off-ramp data is provided because there are no detector stations installed. Therefore, an automated imputation procedure is implemented to estimate these values [24]. The imputation of unknown data uses adaptive identification techniques which are adopted from iterative learning control. To determine the on-ramp flows and the off-ramp split ratios, the imputation executes two steps. Firstly, total input demands $c_i(k)$ for every link i are determined and, secondly, on-ramp demands and split-ratios are extracted from the total demands. The total demands $c_i(k)$ can be perceived as the vehicles intending to enter the link $i+1$ from the link i, i.e. the demand from the previous link minus the difference between off-ramp flow and on-ramp demand existing between the two links. The LN-CTM simulation is performed several times and the total demands $c_i(k)$ are adjusted iteratively to minimize the density error of the simulation at each link in comparison with the real data.

This procedure is performed not only for the duration of one simulated day. The iteration for the density profile is done multiple times, always using the parameters from the previous run, so that the overall density error is minimized. The density error is the sum of the differences between the imputed and the measured densities $\sum n_i(k) - \hat{n}_i(k)$, where $\hat{n}_i(k)$ is the imputed value. This error does not converge to zero because one of the flow modes, the CF mode, has no adaptation equations. Thus, an endpoint criteria of the iteration must be defined if the error does not decrease to a negligible value, and this reasonably is when the error does not become smaller any more.

In a next step, the on-ramp demands and the off-ramp split ratios are determined by a linear program. The linear program makes use of the fact that measurements on the freeway are taken right in between off-ramps and on-ramps (as shown in Figure 9). Thus, it is possible to calculate the ramp flows by minimizing the error between the model calculated flows and the flow measurement between the ramps.

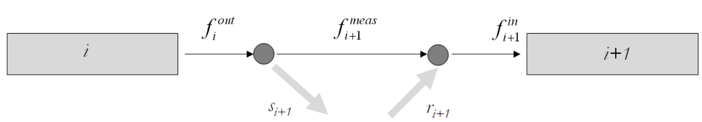

Figure 9: Actual position of ramps and detector at a junction

One point of concern in the imputation process is the low quality of measurement at certain mainline vehicle detector stations and the detection thereof. A set of ad-hoc detection and correction measures were taken to discard irregular data from the imputation procedure. The main approach to identify irregularities was a systematical analysis of the imputation results. Once the vehicle detector stations that report suspicious data are flagged, the imputation is run again, this time omitting the data from flagged detectors, and the results improve in terms of total error in density and flows. Overriding a flagged VDS results in the reduction of the freeway model since the cell it belongs to is now attached to the preceding cell upstream as shown in figure 5 and the freeway model now consists of one less cell whereas the ramps of the adjoined cells are bundled together to represent a single on- or off-ramp each.

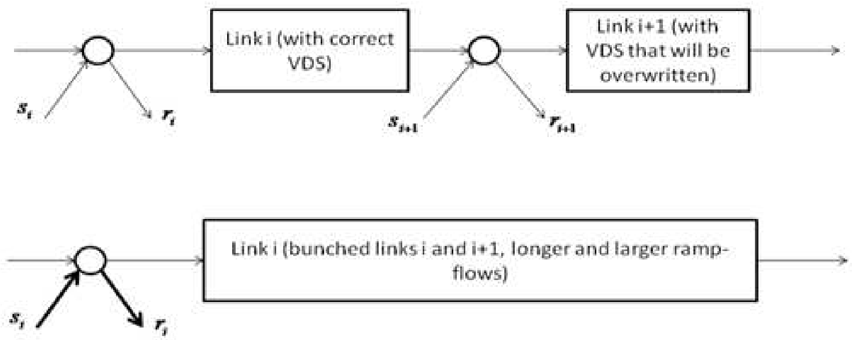

Figure 10: Link structure before and after overriding link i+1.

The approach for I-80 East to find out which VDSs should be overridden was through graphical comparison. For each VDS, the measured data of density and flow are compared to the simulated data, which is based on the imputation. In addition to that, it is useful to review the estimated on- and off-ramp flows as well as the presence or absence of on- or off-ramps. Thus, a graphical overview of the crucial factors is established as seen in Figure 11.

The plots show an example of two VDSs which show almost perfect convergence between the imputed and the PeMS data for both, the density (left) and flow (middle), plots. It also gives the information that the cell of VDS 400976 (top) possesses just an on-ramp ("Onramp Present - 1"; "Offramp Present - 0") and the cell of VDS 400838 (bottom) just an off-ramp. The on-ramp flow is plotted in blue and the off-ramp flow is plotted in red (right plots). By connecting this kind of plots consecutively, it is possible to examine the longitudinal development of the daily flow and density characteristics. This makes it possible to see any disagreement between the simulated and measured densities and flows. A distinction of cases can be made in this analysis. The procedure begins with an analysis of observed density errors, then an analysis of flow errors is performed and, finally, incorrect data is identified inspite of any density or flow mismatch.

A deviation in the simulated density is a sign of a serious irregularity. Density errors in a link can be evoked by the surrounding links, upstream as well as downstream, and depend on the prevalent mode of the LN-CTM (Figure 12).

For instance in the FF mode, the simulated density of link i can only be influenced from upstream, i.e. increased over on-ramp i and decreased over off-ramp i. Vehicles downstream cannot be queued because the traffic is in freeflow and therefore the ramp flows over on-ramp $i+1$ and off-ramp $i+1$ cannot influence the density of link i. So a discrepancy in the measured and simulated

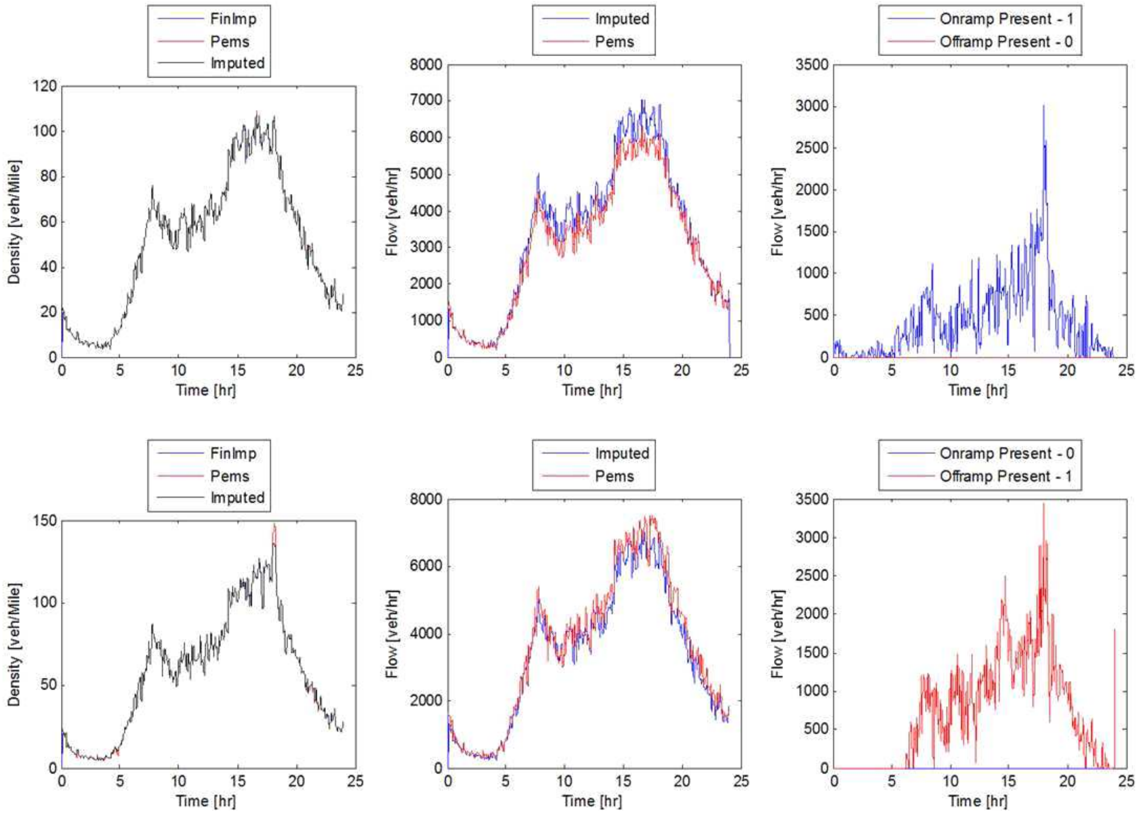

Figure 11: Density, flow and ramp-flow plots for VDS 400976 (top) and 400838 (bottom)

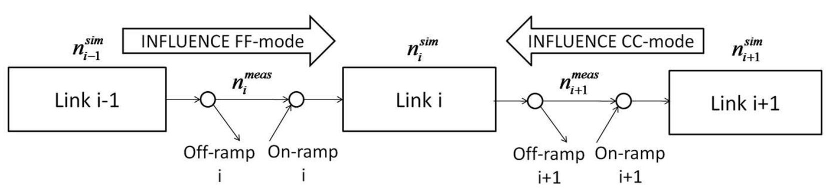

Figure 12: Mode-dependent Influence on simulated density

densities of link i can only be attributed to the ramp-flow estimation of ramps i that precede link i. Regarding the CC-mode, the situation is the opposite. Here, a discrepancy between measured and simulated densities of link i must be attributed to the rampflow estimation of ramps $i+1$ that follow link i. This is due to the fact that upstream traffic and downstream traffic are congested and so, the vehicles downstream can be queued, so that they influence the simulated density in link i. The analysis for the remaining modes follows the same reasoning and conclusions about faulty detectors can be made for all four modes.

The analysis of not converging flow plots is comparatively simple. If there is no ramp between two consecutive cells and the flow plots differ, it is very likely that one VDS is reporting wrong values. In this case, due to the vehicle conservation law, the amount of passing vehicles per time must be the same and only minor variations are tolerable.

It is also possible that simulated and measured values converge smoothly in spite of the fact that the data is corrupt. In this case, one should not willfully stick to the goal of trying to match the simulation to measured data, but should focus on the possibility of false data in spite of converging plots. For example, in the case of present on- and offramps between cells, the occurrence of high imputed onramp flows upstream of the cell and high imputed off-ramp flows downstream is a good indicator of this phenomenon. If, in addition, those ramp-flows show a related shape, it is very likely that the VDS of the reviewed cell measures falsely and the large amounts of imputed ramp-flows only imitate the incorrectness.

The described methods to find out incorrect measurements often narrow the choice of the bad detector to a few rather than pin-pointing the exact malfunctioning detector. To distinguish between the good and the bad detectors, it is useful to regard the plausibility of the plots of the candidates. There are three indicators which help to identify the bad VDS.

Midnight values: For both the density as well as the flow plots, the boundaries, i.e. the hours around midnight, provide good evidence whether the measurements are correct. If the densities and flows reported at night are unreasonably high or low, this VDS is likely to be the bad one.

Maximum values: Another aspect that indicates the bad detector can be found in the maximum values of the plots. If they seem too high or too low compared to the surrounding cells and the whole freeway, the VDS might be faulty.

Exceptional aspects: In addition to the two indicators above, the overall impression of the plots should be examined for exceptional aspects, such as the overall shape, that might denote which VDS reports false data.

If none of the indicators provides a good proof on which detector is bad, giving the different combinations of overridden VDSs a trial is necessary and the combination with the best imputation result is chosen for the model calibration.

I.4.2 Simulation results

For the computer simulation, the whole freeway segment is divided into a certain number of links. This is performed by assuming one link for every VDS except for the case when there is a change in the number of lanes within the link, which results in a partition of the link into several links according to the segments with constant numbers of lanes. All links which belong to one VDS form one cell. To clarify the denotation: Each VDS represents one cell, but one cell can be partitioned into several links. The next step is to determine the locations of the ramps and assign them to their corresponding nodes in the freeway geometry. Once the geometric modeling is done, the calibration and ramp-flow imputations are carried through as described in the previous section and the freeway is ready to be simulated for the given demand and parameters. The following figures summarize the results of the simulation and compare them to the corresponding observations. Figures 13, 14, and 15 reflect the densities, speeds and flows across the freeway, respectively. On the horizontal axis are the successive links of the freeway in the order they appear in the direction of traffic flow and on the vertical axis is the time of the day for which the freeway was calibrated and simulated. In addition to the visually clear agreement of the figures, the numerical values of the total errors for these three measures are quite satisfactory. The total density error was calculated to be 3.1%, whereas the total flow error amounted to 9.1%, which were decreased from 9.0% and 15.4%, respectively, after the fault detection was carried out.

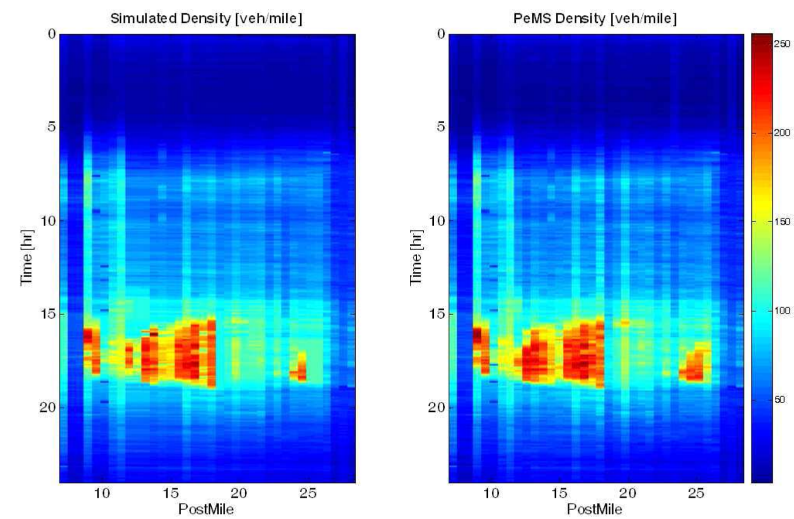

Figure 13: Simulated vs Measured Density contours of I-80 E

I.4.3 Conclusion

The modeling and calibration of I-80E based on the LN-CTM model has been elaborated in this paper. Overall, the macroscopic modeling of I-80E has proven the big service capability of macroscopic traffic models. Two main difficulties had to be overcome in the procedure: 1) Missing ramp data had to be estimated for the whole modeled freeway section, which has been achieved using an automated imputation procedure, 2) Huge extents of false measurements had to be treated, which

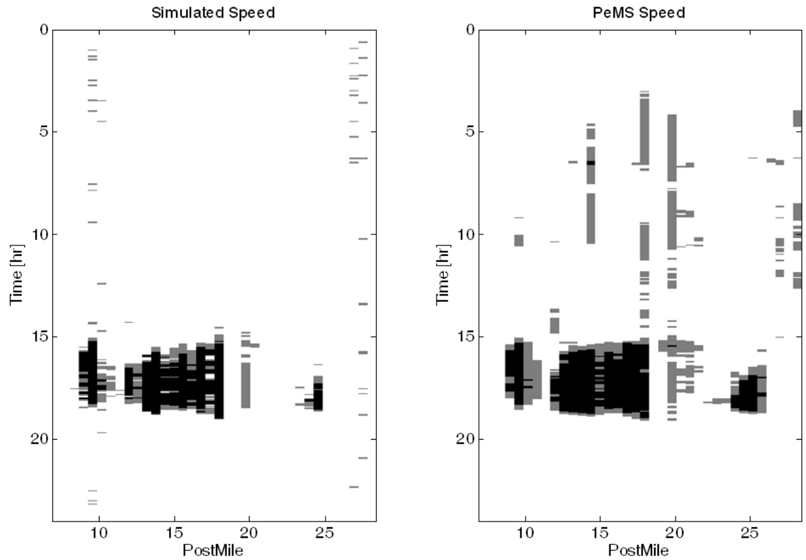

Figure 14: Simulated vs Measured Speed Contours of I-80 E

has been accomplished with ad hoc data manipulation based on in-depth data analysis and the development of a graphical comparative method to detect bad detectors. The results represent a functional calibrated model of I-80 East and can be used for further treatment, such as the implementation of control systems. The simulations, using the calibrated fundamental diagram data as well as the imputed on-ramp flows and off-ramp split ratios, agree closely with the measurements, as shown by the contour plots of the previous section.

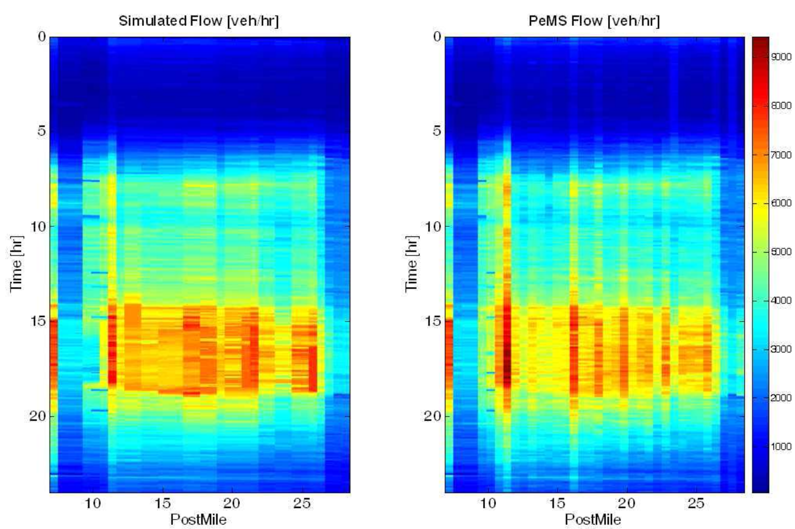

Figure 15: Simulated vs Measured Flow Contours of I-80 E

Part II

Ramp metering with queue control field test

This part of the report describes work that has been directed towards the design and execution of a field test of queue control algorithms. The accomplishments during the past year are described and a modified methodology for the upcoming year is laid out. The main objectives of the proposed research are to test an innovative queue control scheme, developed as part of PATH T.O. 5503 [30, 28, 29], which can be incorporated into most ramp metering strategies as a replacement or enhancement of the queue-override, and has been shown to maintain the on-ramp queue to within its prescribed limits, without inducing any of the oscillations that are observed when the queue-override is used, and significantly improving the performance of the ramp metering scheme. If the queue control field test proves successful and its performance is consistent with the analytical and simulation results so far conducted, its deployment on a large scale could become among the most cost-effective Caltrans measures for attaining significant freeway operational improvements in the near future.

A field test was proposed in order to incorporate Queue Control on one District 4 on-ramp, and study its effect in minimizing queue and mainline density oscillations and enhancing performance. The work related to the field test has been done together with Alan S.Chow and other members of the District 4 Office of Traffic Systems Ramp Metering Group as well as with Caltrans Headquarters Personnel, so that queue control can be incorporated in the functionality of the Universal Ramp Metering Software (URMS) [34]. This modification will permit the use of queue control on any 2070 controller running the URMS, and permit an actual field test of this control scheme on a District 4 on-ramp.

II.1 Accomplishments

During the first year of the project, the focused has been on the planning of the field test. The specific accomplishments to date are:

- Traffic controller selection
- On-ramp test site selection
- Hegenberger on-ramp study
- 2070 controller and OS-9 operating system familiarization
- URMS source code acquisition, familiarization, and analysis with David J. Wells, URMS software developer
- Sensys wireless vehicle detection system analysis
- Sensing equipment discussion and collaboration agreement with Sensys Networks Inc.
- Sensor equipment installation planning with Republic ITS
- Field test methodology modification
- Hardware-in-the-loop on-ramp simulation tool development

II.1.1 Traffic controller selection

In order to make a field test implementation of queue control on an on-ramp in California, it is necessary to interface with the traffic infrastructure managed by Caltrans. Currently there are two traffic controllers commonly used for traffic control in California, the 170 controller and the 2070 controller. Evaluating the features and capabilities of both controllers was necessary in order to chose one for the field test. The information related to the traffic controllers and their operating system and ramp metering software was obtained from Sean Coughlin, Herasmo Iniguez and [8, 35, 6, 7].

The 170 traffic controller is currently used for ramp metering in California. This controller is considered to be outdated by current standards due mainly to its limited processing speed and memory. Additionally, since it only supports assembly language programs, it is complicated to develop or modify software. Moreover, due to the discontinuation of production of the Motorola MC6800 microprocessor, used in the 170 controller, it is difficult to find tools to reprogram the controller. Finally, a major disadvantage of using the 170 controller for the field test is the need to modify one of the 3 different programs used for ramp metering by the different Caltrans districts, where modifications done on one program are useless for the other two.

The 2070 controller, which is based on the 170 controller, was built as a field processor and communication system with the intention to satisfy the high-end needs of the advanced freeway and urban traffic control system. The 2070 controller is easier to work with than the 170 controller. By using this controller for the field test, it is possible to have the resources to implement sophisticated algorithms that require greater performance and flexibility than that available in the 170 controller. Even though Caltrans does not commonly have 2070 controller deployed in freeways for ramp metering, it is possible and very simple to switch a 170 controller by a 2070 controller. All 2070 controllers are compatible with the 170 controller cabinet. At this point, Caltrans does not have a completed ramp metering software for the 2070 controller. However, they have been developing the Universal Ramp Metering Software (URMS), which is in its testing stage.

There were different options considered for the purpose of performing the field test using either the 170 or the 2070 controller, from which two stand out:

1. Modify the current ramp metering software for the 170 controller used in District 4 for ramp metering

2. Collaborate with Caltrans in the completion of a ramp metering software for the 2070 controller and use the 2070 controller instead of the 170 controller

After considering the two possibilities and discussing them with Sean Coughlin, Caltrans District 4 engineer, and Herasmo Iniguez, Caltrans Headquarters engineer, it was decided to use the second approach. Both engineers agreed that using the 2070 controller is better in the long run.

In the T.O. 6329 RFP, it was mentioned that Herasmo Iniguez was going to be involved in the implementation of ALINEA and queue control in the URMS. However, Herasmo Iniguez is not involved in the development of the URMS anymore. David J. Wells, traffic operations/ITS projects & standards engineer is the current developer of the software. For this reason, issues related to the URMS software will be solved with his collaboration.

II.1.2 On-ramp test site selection

An on-ramp test site in the Bay Area District has been chosen in order to perform the field test. Alan S. Chow and Mehran Lajevardi helped in the process of finding a single lane on-ramp with high storage capacity and high demand in order to have favorable conditions for data collection and analysis. The chosen test site is the Hegenberger Rd. loop on-ramp to S/B 880 (see Figure 16). Mehran Lajevardi has provided the "as built" pdf file of the Hegenberger on-ramp. This document was used to discuss with Sensys Networks Inc. and Republic ITS personnel the equipment and installation requirements and costs associated with the field test.

Figure 16: Hegenberger Rd. Loop on-ramp to S/B 880

If for unforeseen reasons it becomes infeasible to perform the field test in the selected on-ramp or any other on-ramp in District 4, the field test could be performed in the North Central District.

II.1.3 Hegenberger on-ramp study

A study of the Hegenberger on-ramp was performed. One of the objective was to understand the specific characteristics of the Hegenberger Rd. loop on-ramp to S/B 880 to determine the sensor and installation requirements for the field test. Furthermore, this study helped to identify the type of vehicles (cars vs trucks) using the on-ramp and their influence on queue formation.

On October 3, 2008, a manual count and characterization of vehicle getting into the on-ramp was conducted. It was possible to determine the capacity of the ramp based on the queue observed (see Figure 17). In addition, it was concluded that traffic lights from the arterial street directly affect the inflow of cars into the on-ramp. Cars usually get into the on-ramp in platoons. Moreover, it was possible to witness the effect that the queue exceeding the on-ramp capacity has on the adjacent streets. Finally, the mainline loop detector station location with respect to the on-ramp was identified. This information is important for the field test because some algorithms like ALINEA need to be modified depending on sensor location (upstream vs downstream).

Figure 17: Hegenberger Onramp with Queue

II.1.4 2070 controller and OS-9 operating system familiarization

In order to learn how to use and write applications for the 2070 controller, a unit was acquired.

To be able to use the 2070 controller for the field test, it was necessary to understand how to interact with the controller using its operating system. Most users of the 2070 controller only need to learn how to use traffic application software like the Traffic Signal Control Program (TSCP). Documentation and training is available for this type of use. However, this project requires writing, compiling and loading applications for the controller using an OS-9 cross-compiler environment. This is a challenging task due to the limited documentation on the 2070 controller device drivers and the need to learn how to use and interact with the OS-9 operating system. Documentation, manuals, and basic code was studied in order to learn how to write, compile and load applications into the controller. It has been possible to compile and load into the 2070 controller modules that perform basic functions like controlling the front panel led or reading/writing to the field I/O port.

Becoming skilled at writing, compiling and loading code for the 2070 controller was needed in order to be able to understand, modify, and compile the URMS source code.

II.1.5 URMS source code acquisition, familiarization, and analysis with David J. Wells, URMS software developer

Getting a source code copy relase of the URMS was a challenging task. Due to the fact that this software is copyrighted by organizations outside of Caltrans, it took time to get access to the complete source code. The URMS was obtained under a confidentiality and non-disclosure agreement with the help of David J. Wells.

Once the URMS source code was available, it was necessary to study and understand the code in order to identify which parts need to be modified. Due to the modular structure of this software, it seems that only one of the ten modules of the URMS needs to be changed. Modifications will be required on the timing module, which handles traffic calculations and meter timing.The timing module was studied in detailed in order to understand its structure and functions, and identify the locations where modifications are required for the implementation of queue control and ramp metering algorithms.

On July 16 2008, a meeting between David J. Wells, Nathan Loebs, Roberto Horowitz, and Rene O. Sanchez took place in Sacramento. In this meeting, the URMS and its timing module were discussed in order to agree on the modifications needed to complete the field test.

II.1.6 Sensys wireless vehicle detection system analysis

Documentation of the Sensys wireless vehicle detection system has been reviewed to determine if this detection system is the most appropriate for the field test.

Based on [17, 26], a Sensys wireless sensor can be installed and connected to a traffic cabinet in such a way that the information it provides to the traffic controller is identical to that provided by an inductive loop detector currently used in California. This feature of the Sensys wireless sensors allows for the use of the URMS without modifying the way sensor data is obtained and processed by the controller. The URMS software will access and process the Sensys wireless sensor data in the same way as it will access loop detector data.

The ease of installation on the field is another advantage of the Sensys wireless sensor in comparison to other sensors. A Sensys wireless sensor can be installed in less than 10 minutes. This ease of installation gives this detector an advantage when compare to inductive loop detectors, which take more time to install and require more wiring.

In conclusion, the Sensys wireless vehicle detection system is the best option to introduce extra detection on the field because it does not require URMS modifications and because is easy to install.

II.1.7 Sensing equipment discussion and collaboration agreement with Sensys Networks Inc.

On July 23, 2008, a meeting with Sensys Networks personnel was held to discuss the field test project. This meeting was useful to identify the ways in which Sensys Networks could contribute towards the completion of the field test. During the meeting, the Hegenberger Rd. Loop on-ramp to S/B 880 was evaluated to determine the Sensys equipment needed for the field test and the proper location for its installation. Finally, the feasibility of vehicle re-identification as a method to measure on-ramp queue was analyzed. After the evaluation it was determined that vehicle re-identification would be used on the field test. Finally, it was accorded that Sensys Networks will help with the development of an application to estimate queue-length based on vehicle re-identification algorithms using arrays of Sensys wireless sensors [22, 21].

On September 24, 2008, another meeting took place with Sensys Networks personnel in order to discuss specific issues of the field test. Of particular interest was to discuss how to interface the Sensys equipment and the 2070 controller in order to communicate queue length estimates between access point and traffic controller. Previous to this meeting, this was discussed with David J. Wells, developer of the URMS, in order to get his advice on the matter. Finally, SNAPS, which is the Sensys application that will allow for the collection of Sensys sensors data directly from the on-ramp into a server, was discussed.

II.1.8 Sensor Equipment Installation planning with Republic ITS

Republic Intelligent Transportation Systems (Republic ITS) is the contractor that will be in charge of the installation of the Sensys equipment on the Hegenberger on-ramp. In order to get a formal quote of the installation cost, a conferences call with Robert Asuncion, operations manager for Republic ITS, was arranged on November 11, 2008. During the conversation, the field test implementation of queue control project was explained and discussed. The following are important things to stand out:

The installation can be completed by Republic ITS in one day. The main challenge is coordinating with Caltrans on the date and on the collaboration for lane closure. Most likely, Caltrans will require Republic ITS to make the installation at night. This means that the installation cost will be higher, since Republic ITS will be required to pay overtime to their workers.

Usually, an encroachment permit for this type of installation will cost around 500 to 1000 dollars. Republic ITS needs to apply for a permit, since Republic ITS workers would be performing the installation. There is the possibility to have the encroachment fee waived, since the field test project is funded by Caltrans. During the conversation it was mentioned that Alan S Chow and

Mehran Lajevardi helped in the selection of the on-ramp and are willing to help in the field test project. Robert Asuncion mentioned that this information will be included in his permit application to see if it is possible to get extra collaboration from District 4, especially for road closure and traffic rerouting.

The Caltrans Encroachment Permit Manual estimates that it should take less than 60 days to get a permit, but since Republic ITS is familiar with the permit process and Caltrans engineers, it usually takes Republic ITS around 10 days. A good time to place the Sensys equipment order will be when Republic ITS starts the encroachment permit process.

After the conference call, an analysis of the installation service needed for the field test was conducted by Robert Asuncion in order to determine the installation cost.

II.1.9 Field test methodology modification

After deciding that a Sensys vehicle re-identification application was going to be used for the field test, the original plan proposed in the T.O. 6329 RFP was no longer applicable. There is a fundamental difference between the new methodology in comparison with the previous one. The new approach includes as one of the queue estimation methods, a vehicle re-identification application using arrays of Sensys wireless sensors instead of placing Sensys detectors along the on-ramp. For this reason, it was necessary to make changes to the original field test plan. The complete modified methodology is described in Section II.2.

II.1.10 Hardware-in-the-Loop On-ramp Simulation Tool to Debug and Test the Universal Ramp Metering Software

A hardware-in-the-loop simulation (HILS) system was developed as a tool to assist in the completion of the ramp metering with queue control field test.

Before deploying a 2070 controller with a modified URMS in the Hegenberger on-ramp, it must be thoroughly debugged. In addition, the modified software must be tested and approved by D4 engineers before it can be used on the field. The unmodified URMS software has already been debugged and tested by Caltrans engineers before its release for preliminary testing in the field using a traditional traffic controller suitcase tester device (see Figure 18). However, one of the main drawbacks of this tester is the need to manually operate mechanical switches to simulate loop detector signals. This debugging and testing approach becomes difficult and sometimes inappropriate when coordination of signal actuation is required, as will be the case for the field test. To debug and test the modified URMS, it will be necessary to recreate the dual detector signals used to measure vehicle speed upstream of the on-ramp, with good accuracy. For this reason, it was decided to design and build a hardware-in-the-loop simulation system to replicate in real-time the Hegenberger on-ramp detector signals. See Section II.3 for a detailed explanation of this tool.

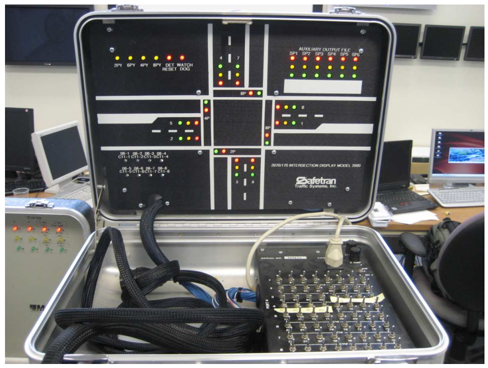

Figure 18: Traffic controller suitcase tester used to evaluate the URMS

II.2 Modified Methodology

To prevent the on-ramp queue from spilling over into surface streets and interfering with the street traffic, the queue length must be regulated. The currently used "queue-override" scheme steadily increases the metering rate whenever the end of the queue reaches the queue detector, until the metering rate saturates to the maximum value. After the queue dissipates, the metering rate is reset to the value determined by the ramp-metering controller. This scheme is equivalent to an integral control with saturated integrating rate and resetting, which can be easily shown not to be asymptotically stable, given that the queue-length dynamics is a simple integrator.

If the queue length could be measured, an asymptotically stable PI regulator could be designed to stabilize the close loop queue dynamics [28]. However, the PI regulator needs the current queue length as its feedback, which unfortunately is not available in the field. Therefore, a suitable queue length estimator has to be designed, as described in [28], using available information, such as the vehicle speed measured by the queue detector.

A field test has been proposed in order to implement Queue Control on a Bay Area District (D4) on-ramp and study its effect in minimizing queue and mainline density oscillations and enhancing performance. This will be accomplished by using a 2070 traffic controller running a modified Universal Ramp Metering Software (URMS).

The field test has been designed to answer the following questions:

1. What is the best technique for estimating queue-length in an on-ramp?
2. Does the simple queue override produce oscillations in the length of the queue?
3. Are these oscillations attenuated by the Queue Controller?
4. Does the Queue Controller decrease the incidence of overspilling on-ramp queues?
5. Does the Queue Controller decrease the frequency of the override, and thus enhance coordination?
6. What data and hardware requirements are necessary to implement the queue regulator?

The criteria used for the test site selection include:

1. the ramp is under local control,
2. it has functioning queue detectors,

3. it is subject to heavy demand and long on-ramp queues, leading to frequent queue overrides.

Two different queue-length estimation methods will be evaluated and one will be chosen to be part of the queue-length estimation and regulation algorithm. The first method is a queue-length estimator based on a simplified model for the driving behavior of a vehicle that is approaching the end of the queue: the vehicle decelerates at a constant rate from its cruising speed, until it stops. By measuring speed upstream of the on-ramp it will be possible to calculate the queue-length. The second method estimates the length of the queue using a vehicle re-identification algorithm [22, 21]. This scheme is based on matching individual vehicle signatures obtained from Sensys wireless sensor arrays placed at the two ends of the on-ramp.

The field test will be conducted in three stages. In the first stage, the two queue-length estimation methods will be evaluated. For this stage it will not be necessary to disrupt the current metering strategy at the chosen on-ramp. The sensor data can be collected and the queue-length estimation algorithms can be validated without the need to modify the URMS or any other ramp metering software currently running on the on-ramp. The data from the Sensys wireless sensors can be collected for analysis using a standard Sensys data collection application. For the second stage, the queue-length estimator and regulator will be tested on the on-ramp. The algorithm will be coded into the URMS and will be used together with the current URMS traffic responsive ramp metering controller. Finally, the last stage and goal of this experiment will be to implement the queue-length estimator and regulator together with ALINEA, a close loop local ramp metering strategy. The algorithms will be coded into the URMS, which will be loaded into a 2070 traffic controller and tested on the on-ramp.

II.3 Hardware-in-the-Loop On-ramp Simulation Tool

II.3.1 Summary

An on-ramp simulation system that can be used to debug and test the Universal Ramp Metering Software (URMS) is presented. The tool includes a simple car following microscopic traffic model for the on-ramp and a Controller Interface Device (CID), which interfaces a standard personal computer with a 2070 traffic controller. The CID consists of the low cost and commonly available National Instruments (NI) USB-6501 24-Channel Digital I/O device and a basic circuit that interfaces the 5-Volt TTL logic from the Digital I/O board to the 2070 controller. The resulting hardware-in-the-loop simulation tool systematically reads the phase states from the controller and changes detector states based on the cars trajectories as displayed on the on-ramp simulator. With this tool it is possible to check the performance of the 2070 controller running the URMS as if the traffic controller was operating on a standard on-ramp managed by Caltrans. Finally, the real-time nature of this tool is discussed based on a quantitative analysis of the simulator performance running on the Windows XP operating system.

II.3.2 Hardware-in-the-Loop Simulation

The hardware-in-the-loop simulation (HILS) concept has been used to create a simulation tool to test the URMS running on a 2070 controller. A particular feature of this type of architecture is that the traffic simulation model does not implement any control logic, instead it controls traffic flow in the simulation based on the phase states produced by the traffic signal control equipment. Simultaneously, the traffic signal control equipment uses the detector signals generated by the simulation to update its control logic (see Figure 19(a)) [4]. HILS has been used in the past to interface with traffic signal control equipment for testing purposes; however, previous systems focused on testing intersection control software. The simulation time step used in these systems is on the order of seconds, and equipment is used primarily to simulate loop detector signals used by traffic controllers to determine car presence and a rough estimate of occupancy [23, 20]. The tool presented in this paper is primarily designed to generate, through simulation, traffic detector signals for an on-ramp/freeway system (see Figure 20(b)) with sufficient resolution to allow the 2070 controller to accurately calculate volume, occupancy, and speed.

The HILS system presented in this paper has three basic components: 1) a controller interface device (CID), 2) a software interface module, and 3) a microscopic simulation engine. A description of each component is presented below.

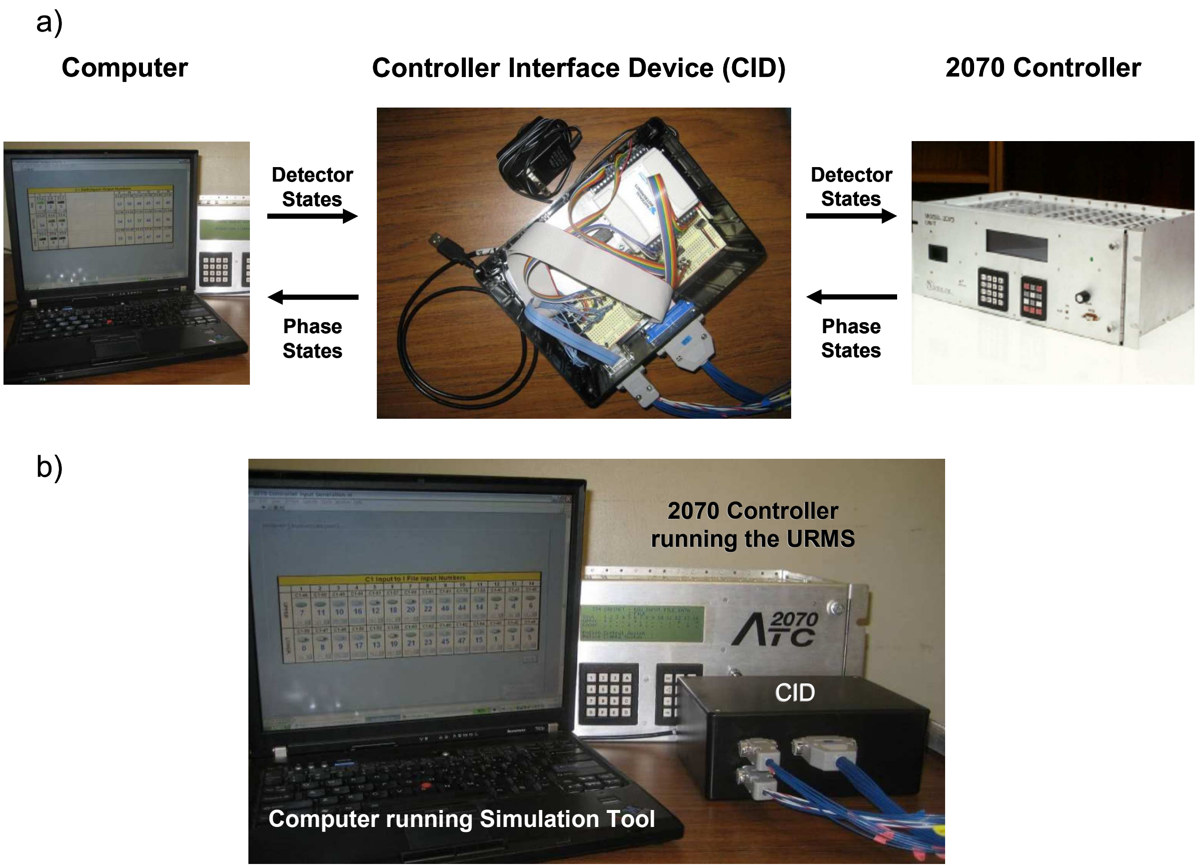

Figure 19: (a) On-ramp simulation tool architecture (b) On-ramp simulation tool setup

II.3.2.1 Computer Interface Device (CID)

This device provides the interface from the 2070 traffic controller to the personal computer running the traffic simulation. The CID has two main elements: 1) the NI USB-6501 device and 2) a custom electronic circuit.

The NI USB-6501 is a portable digital Input/Output device, which provides data acquisition and control capabilities. With plug-and-play USB connectivity, the NI USB-6501 is very versatile and can be used in most personal computers. The NI USB-6501 has 24 single-ended digital lines, which comprise three DIO ports. In this tool, two ports are configured to generate detector signals and one port is used to read the phase states output from the controller. This device was chosen because of its low price, portability, and because when used with LabVIEW, it provides a straightforward procedure to interface with the simulation engine. Signals can be sent and received using LabVIEW standard functions that can easily be accessed from the simulation.

The custom made circuit was designed to interface the 5-Volt TTL logic from the digital I/O board to the 2070 controller. This circuit was built using a modular IC breadboard socket, SN706 TTL hex inverter buffers/drivers with open-collector high-voltage outputs, one 7805A voltage regulator, and a 12-Volt power supply.

Two main goals of the CID design stage were portability and low cost. The portability was ensured with the use of a small USB DIO board that can be used in most personal computers. The low cost was achieved by using one of the cheapest data acquisition boards on the market. The components to build the CID presented in this paper cost less than 200 U.S. dollars.

II.3.2.2 Software Interface Module

The software interface model provides the linkage between the CID and the traffic simulation program. The NI USB-6501 board used to build the CID comes with drivers that can be used to develop customized applications using NI LabVIEW. These drivers serve as the software interface module, and do not require any modification when used in the HILS tool.

II.3.2.3 Microscopic Simulation Engine

The simulation engine was developed using the NI LabVIEW development environment. Before deciding to create a custom traffic microscopic simulator, commercial simulator packages were considered. However, the time steps used in these simulators were not low enough for the resolution desired for this application, e.g. CORSIM uses a 1 second time step while VISSIM can not go lower than 100 milliseconds. This limitation was one reason for developing a microscopic simulation specifically for an on-ramp/freeway system with a time step between 1 and 10 milliseconds. Another reason was to have the flexibility to customize the simulation engine to complement some features of the URMS, e.g. configuration and testing menus.

II.3.3 The Model

In order to simulate the Hegenberger on-ramp/freeway system (see Figure 20(a)), it was necessary to use a simplified layout that would capture the detector location and the ramp characteristics. Figure 20(b) shows a simplified configuration of the Hegenberger onramp/freeway system that follows the NTCIP typical on-ramp layout as close as possible [2], which is also the standard configuration used in the URMS. The ramp layout had to be slightly modified to incorporate dual detection for queue-length estimation.

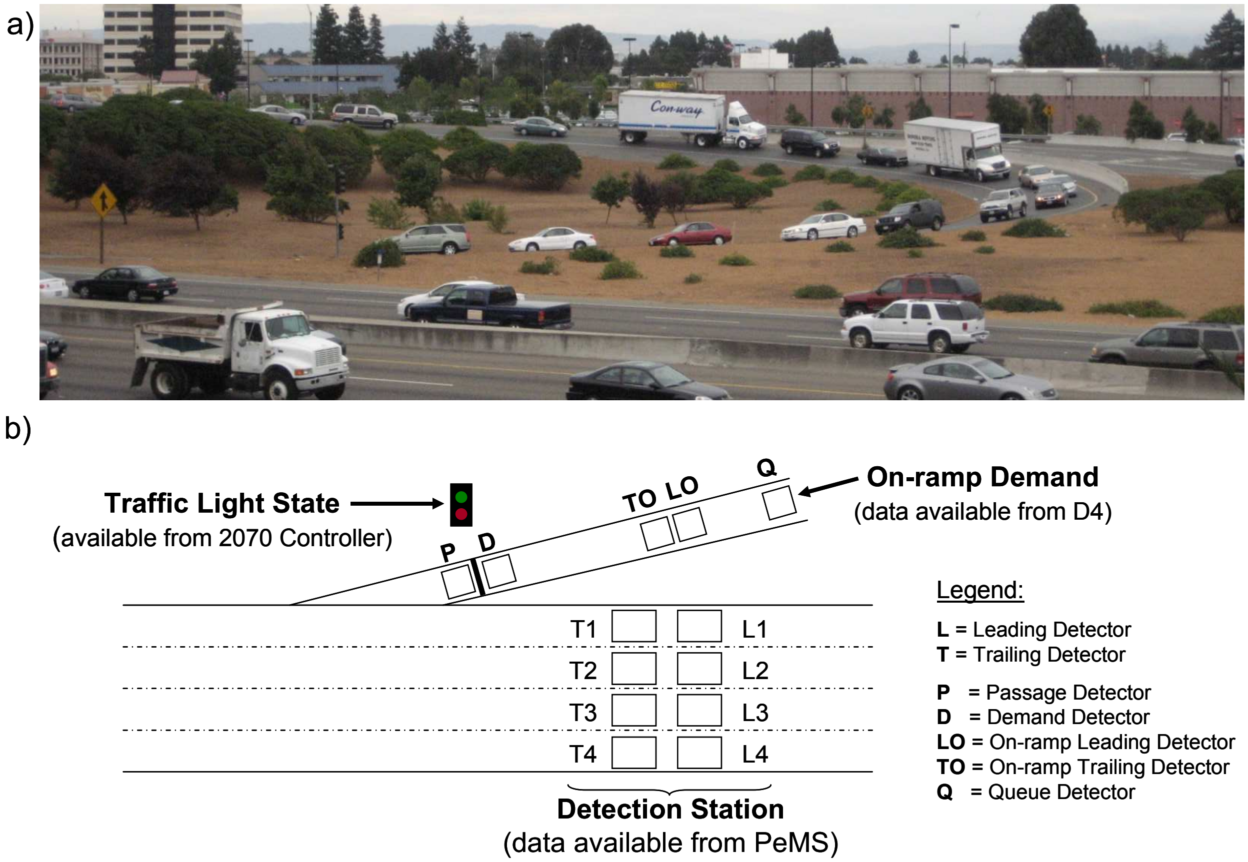

Figure 20: (a) Hegenberger Rd. loop on-ramp to 880 southbound (b) Hegenberger on-ramp/mainline layout used for the simulation tool

A traffic controller operating on an on-ramp in California is usually programmed to collect data from the on-ramp detectors, set the traffic light phase states, and collect mainline detection stations data (sometimes multiple mainline detection stations). In order for the simulation tool to generate the signals that a traffic controller would encounter in the field, it was decided to simulate traffic conditions on the Hegenberger on-ramp/mainline system with two completely different models: 1) a constant velocity microscopic mainline traffic model and 2) a simplified car following on-ramp traffic model. The 2070 controller is able to read the detector states set by both models and can update the on-ramp simulation metering rate (phase states). With this approach, on-ramp traffic conditions do not have any effect on the mainline freeway simulation. However, simulated mainline traffic conditions may have an effect on the on-ramp simulation depending on the ramp metering algorithms implemented in the URMS. The simulation tool was designed in this way in order to test traffic responsive ramp metering algorithms like ALINEA [15], where mainline traffic conditions read by the controller are used to set the metering rate at the on-ramp.

II.3.3.1 Freeway Mainline Model

A constant velocity microscopic mainline traffic model was used to model vehicle trajectories on the mainline. In this model, the cars of a given freeway lane travel at the same speed, and their position is updated every simulation period. The car trajectories start at the beginning of the freeway segment, and end when the car reaches the end of the freeway segment, given by the user-specified freeway length (see Figure 21 (left)). The cars are generated based on the flow specified for every calculation interval. The URMS software calculates aggregates of mainline data every 30 seconds. For this reason, parameters for a given lane can be updated every 30 second calculation interval in the simulation. However, the calculation interval should be set taking the data used to feed the simulator into account. For example, if PeMS [25] data are used, the calculation interval should be set equal to the time granularity used in the data set.

The model for the freeway can be very simple because the main objective of this part of the tool is to generate loop detector signals that the controller can read to calculate aggregate values for each calculation interval, and use these aggregate values as the input for traffic-responsive ramp metering controllers.

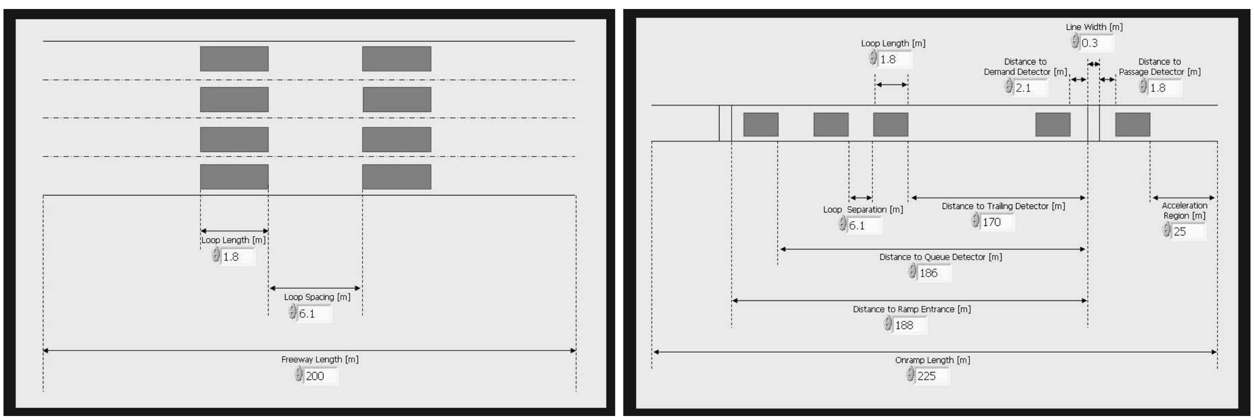

Figure 21: (left) Freeway layout (right) On-ramp layout

II.3.3.2 On-ramp Model

A simplified car following traffic model, based on [3], was used to simulate vehicles on the on-ramp. This is a simple model specifically conceived for a homogeneous highway in which the nth vehicle follows the same trajectory as the (n-1)th vehicle except for a translation in space and time. It was necessary to incorporate the ramp metering traffic signal into the model, which can be considered as an inhomogeneity, by specifying rules of how vehicles react to the signal. The rules that were specified are: 1) a car in front of the traffic light must stop when the light is red, and 2) only a predetermined number of cars can advance per green phase. It was decided to use this model because it is simple but captures dynamics that are important for an accurate generation of detector signals. There is a particular interest in testing algorithms that use vehicle speed close to the on-ramp entrance to estimate queue-length. This model allows for changes in speed based on driver behavior parameters and the presence of vehicles ahead. With this model it is also possible to introduce queue dynamics in the simulation, a feature necessary for the accurate generation and timing of on-ramp detector signals.

For the simulation, it is necessary to know parameters related to the length of the on-ramp, the length of the loop detectors, and the position of the loop detector with respect to the ramp entrance. All these parameters can be specified using the simulator on-ramp layout menu, as shown in Figure 21 (right). For the Hegenberger on-ramp, these parameters were obtained from Goolge Earth^{TM} and [31].

II.3.3.3 Vehicle/Loop Interaction

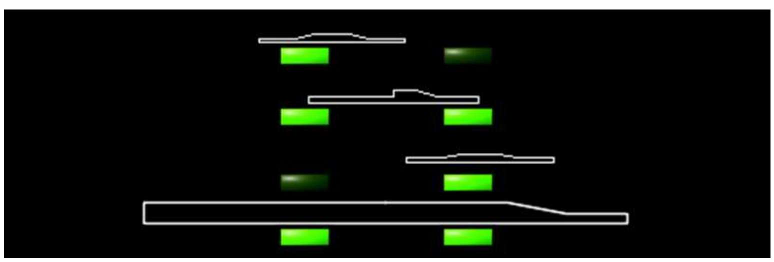

Figure 22: Mainline loop detectors actuation

The loop detector signals generated by the simulation tool are actuated based on vehicle positions. Both simulations have the location of the loop detectors with respect to the beginning of the freeway segment and the beginning of the on-ramp, respectively. When any of the data points representing a vehicle is on the detection zone specified by the location of its leading and trailing edge, the detector signal is triggered. The interaction between loop detectors and cars occurs in real-time. Whenever the display in the simulator shows an active detector, the detector signal read by the controller for that specific detector is active as well (see Figure 22). This is a desired feature for a debugging tool, since it helps to visually identify what is the state of each signal going into the controller.

In order to recreate the signals generated in a real onramp/freeway system more realistically, three types of vehicles can be generated in the simulator: 1) cars, 2) pickups, and 3) trucks. Each vehicle has an independent length, shape, and probability of occurrence. The particular shape of each car can be observed in Figure 22.

II.3.4 Simulation Tool

The simulation software was developed using the NI LabVIEW development environment. This software is composed of four elements: 1) the 2070 controller input configurator, 2) the 2070 controller output configurator, 3) a freeway simulator, and 4) an on-ramp simulator. When the program is run, the user can decide if any configurator will be used. If the configuration process for the inputs or outputs is skipped, the configuration stored in the computer will be used by the program. After the configuration process is completed or bypassed, the on-ramp and freeway simulations start. In the following, each component of the software is described.

II.3.4.1 2070 Controller Input Configurator

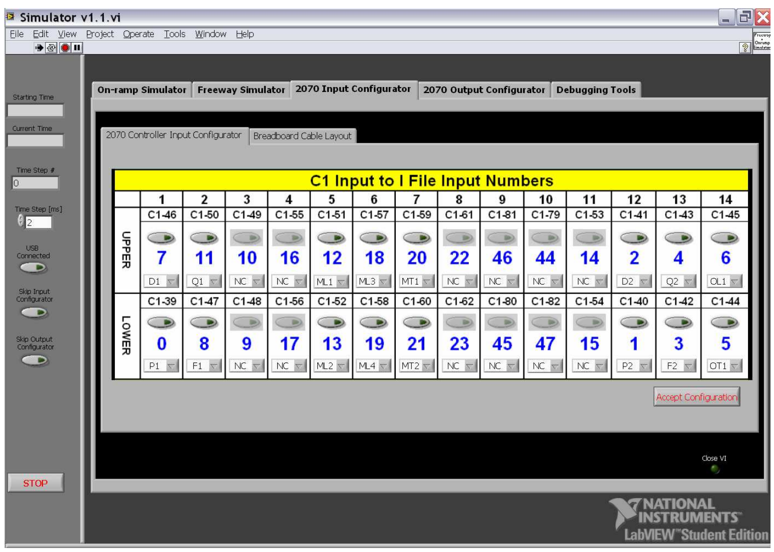

Figure 23: 2070 Controller Input Configurator

The 2070 controller input configuration application was developed to match the URMS configuration convention. The diagram used in the configurator to show the Input File current assignments (see Figure 23) is the same as that used in [34] to describe the physical input number for each input file slot for a Model 334 cabinet. An input configurator was included in this tool, because it is necessary to map every detector used in the simulation to the corresponding detector in the 2070 controller, as recommended in [4]. In this configuration application, it is possible to independently change the state of each active input. When this feature is used in conjunction with the URMS Input File Test utility, it is straightforward to check if the simulation and the URMS signal assignments match and if the detector states are read properly by the 2070 controller.

II.3.4.2 2070 Controller Output Configurator

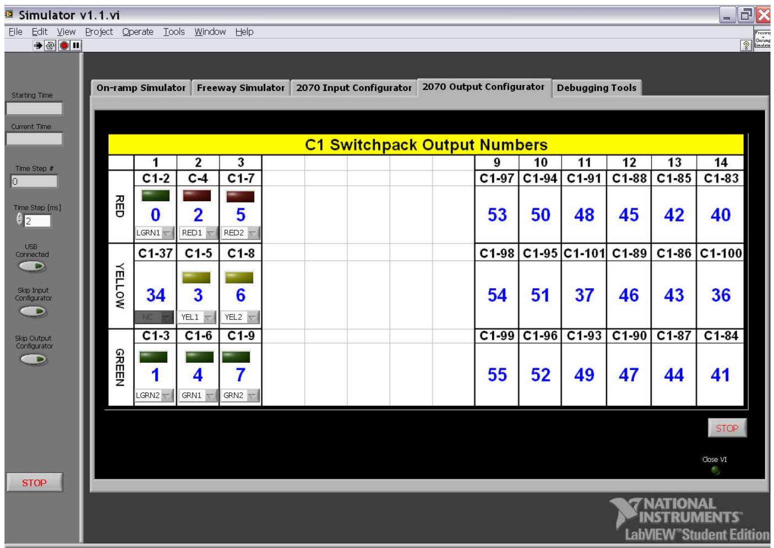

Figure 24: 2070 Controller Output Configurator

The 2070 controller output configuration application was developed to match the URMS configuration convention. The diagram used in the configurator to show the output file current assignments (see Figure 24) is the same as that used in [34] to describe the physical output number for each output file slot for a Model 334 cabinet. An output configurator was implemented in order to map every phase indication used in the simulation to the corresponding phase in the 2070 controller, as recommended in [4]. In this configuration application, it is possible to independently read the phase of each active output. When this feature is used in conjunction with the URMS Output File Test, Output Signal Test, and/or Lights Test utilities, it is easy to check if the simulation and the URMS output signal assignments match and if the phase states are read properly by the simulation.

II.3.4.3 Freeway Simulator

The constant velocity microscopic mainline traffic model described earlier is implemented in the freeway simulator application. There are three components associated with this part of the simulation: 1) the freeway simulation user interface, shown in Figure 25, 2) a freeway layout menu (see

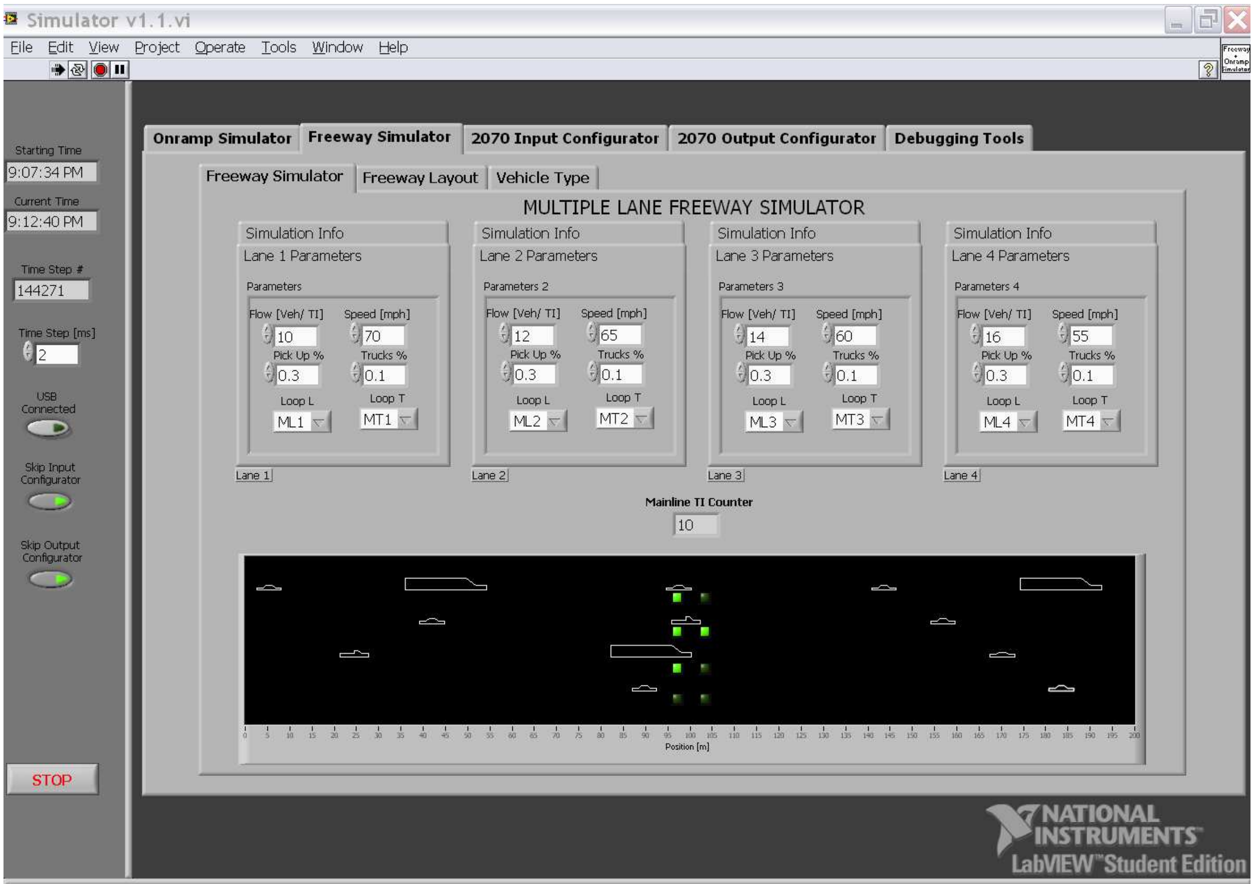

Figure 25: Freeway simulator interface

Figure 21(left)), and 3) a vehicle menu. The simulation interface is used to observe the movement of vehicles through the defined freeway segment. With this interface, information related to the simulation can be accessed and it is possible to set and modify freeway lane parameters independently. In the freeway layout menu, the dimension of the mainline segment, the detector location, and the detector separation can be set. Finally, the vehicle menu is used to determine the properties of the three types of vehicles present in the simulation. The vehicle parameters for the mainline simulation can be different from those used on the on-ramp simulator.

II.3.4.4 On-ramp Simulator

The simplified car following traffic model described earlier is implemented in the on-ramp simulator. There are three components associated with this application: 1) the on-ramp simulation user interface, shown in Figure 26, 2) a freeway layout menu (see Figure 21(right)), and 3) a vehicle menu. The simulation interface is used to observe the movement of vehicles through the on-ramp. This

Figure 26: On-ramp simulator interface

component also displays information related to the simulation, including the phase state output by the 2070 controller. Using this interface, it is also possible to set and modify simulation parameters. In the on-ramp layout menu, the dimension of the on-ramp segment, the detectors location, their length, and their separation can be set. Finally, the vehicle menu is used to determine the properties of the three types of vehicles present in the on-ramp simulator.

II.3.5 Benchmarking

This tool was designed to simulate traffic conditions on an on-ramp/freeway system and update vehicle positions and detector states in real-time. In the context of this project, real-time means that the HILS system should simulate the displacement of vehicles, check if the vehicles are on a detection zone, and update detector states with a time equal or less than the actual time it would take vehicles to travel the same displacement on a real on-ramp/freeway system. Furthermore, it is desired to achieve the smallest possible simulation time step (Δt) in order to increase the resolution of the detector signals sent to the 2070 controller.

The real-time nature of the hardware-in-the-loop simulation (HILS) tool is limited by the performance of the Windows XP operating system, which only permits a 1 ms time resolution. Even though an actual HILS simulation step (Δt_{actual}) may take less than 1 ms, this time is usually larger, since Windows XP does not have sufficient real-time capability to effectively implement such precise timing [4]. To compensate for this limitation, the simulation was designed so that the timing of the simulator would be based on three time stamps: 1) a reference time stamp obtained at the beginning of the simulation run (t_o), 2) a time stamp recorded in the (i-1)th simulation step (t_{i-1}), and 3) a time stamp obtained in the current (ith) simulation step (t_i). To update any quantity that needs the total simulation time, the difference between t_i and t_o is used. For quantities that need the time increment between the (i-1)th and the ith simulation steps, e.g. to calculate position increments, the difference between t_i and t_{i-1} is used. This configuration helps maintain accurate simulation timing even when variations in the actual simulation step execution time (Δt_{actual}) occur. Offsets introduce by having $\Delta t_{actual} \neq \Delta t$ at the ith simulation step can be removed at the (i+1)th simulation step.

In order to show that the HILS system is a reliable tool to debug and test the URMS, it was important to quantify the uncertainty introduced by not developing this tool on a real time operating system environment. As a result, a benchmarking procedure was used to characterize the real-time nature of the software. 8 simulation runs, of 60,000 simulation steps each, were executed using different desired time steps (Δt). The actual time step (Δt_{actual}) was recorded for each simulation step and stored into a file. These data were used to determine the time reliability of this tool as a function of Δt.

Δt	$\Delta t_{average}$	σ	Δt_{max}	$\Delta t_{actual} = \Delta t \pm 1$
1 ms	2.00	0.506	17	97.63%
2 ms	2.09	0.465	17	98.99%
3 ms	3.26	0.553	17	99.20%
4 ms	4.19	0.426	17	99.60%
5 ms	5.20	0.492	18	99.69%
10 ms	10.25	0.476	23	99.76%
25 ms	25.34	0.478	30	99.96%
50 ms	50.93	0.271	56	99.91%

Table 2: Benchmarking results of 60,000 simulation step runs for different Δt.

The results of the analysis are presented in Table 2 and show that Δt_{actual} for at least 97.5% of the steps is within one millisecond of Δt. The worst performance is observed when $\Delta t = 1$ ms. As Δt increases, the percentage of Δt_{actual} that are within one ms of Δt increases, while the standard deviation decreases. Based on Table 2 and Figure 27, choosing $\Delta t = 2$ ms provides the time resolution needed for the HILS tool while introducing an acceptable error on the generation of the detector signals.

	V_{max}	$t_{actuation}$	$t_{controller}$	$V_{controller}$
On-ramp	40 mph	408.32 ms	408.32 ± 16 ms	40 ± 1.5 mph
Freeway	80 mph	204.16 ms	204.16 ± 16 ms	80 ± 6.4 mph

Table 3: Simulated loop detector signal uncertainty for $\Delta t = 2\ ms$, $L_{detector}$=1.8 m *and* L_{car}= 5.5 m .

In order to quantify the effect of time uncertainty on the detector signals, a worst case scenario analysis for $\Delta t = 2$ ms was performed. The shortest signals generated by the simulator, which are also the most affected by the time uncertainty, are those of the smaller cars traveling over a loop detector at the speed limit. There is a speed limit specified for the freeway and one for the on-ramp. Assuming that $\Delta t_{actual} = \Delta t_{max}$, the propagation of the maximum error, of 16 ms, is shown in Table 3. Based on the data collected from the benchmarking procedure, the uncertainty propagated to detector signals generated by the simulators is within ± 16 ms, which when used by the 2070 controller to calculate velocity would yield a ± 6.4 mph uncertainty for the freeway simulator and a ± 1.5 mph uncertainty for the on-ramp simulator. The ± 6.4 mph uncertainty in the mainline speed computations may seem significant. However, it will not considerably affect the aggregate mainline speed since it is calculated as an average over a URMS calculation interval. For any practical purposes, the ± 1.5 mph uncertainty will not affect the velocity estimation on the on-ramp.

Symbol	Name	Unit
Δt	desired simulation time step	ms
$\Delta t_{average}$	average simulation execution time	ms
Δt_{actual}	actual simulation execution time	ms
Δt_{max}	maximum recorded time step	ms
σ	standard deviation	ms
V_{max}	maximum velocity used in simulation	mph
$L_{detector}$	loop detector length	m
L_{car}	average regular car length	m
$t_{actuation}$	theoretical detector actuation time based on V_{max}	ms
$t_{controller}$	detector actuation time recorded by 2070 controller	ms
$V_{controller}$	velocity calculated by 2070 controller	mph

Table 4: List of Symbols

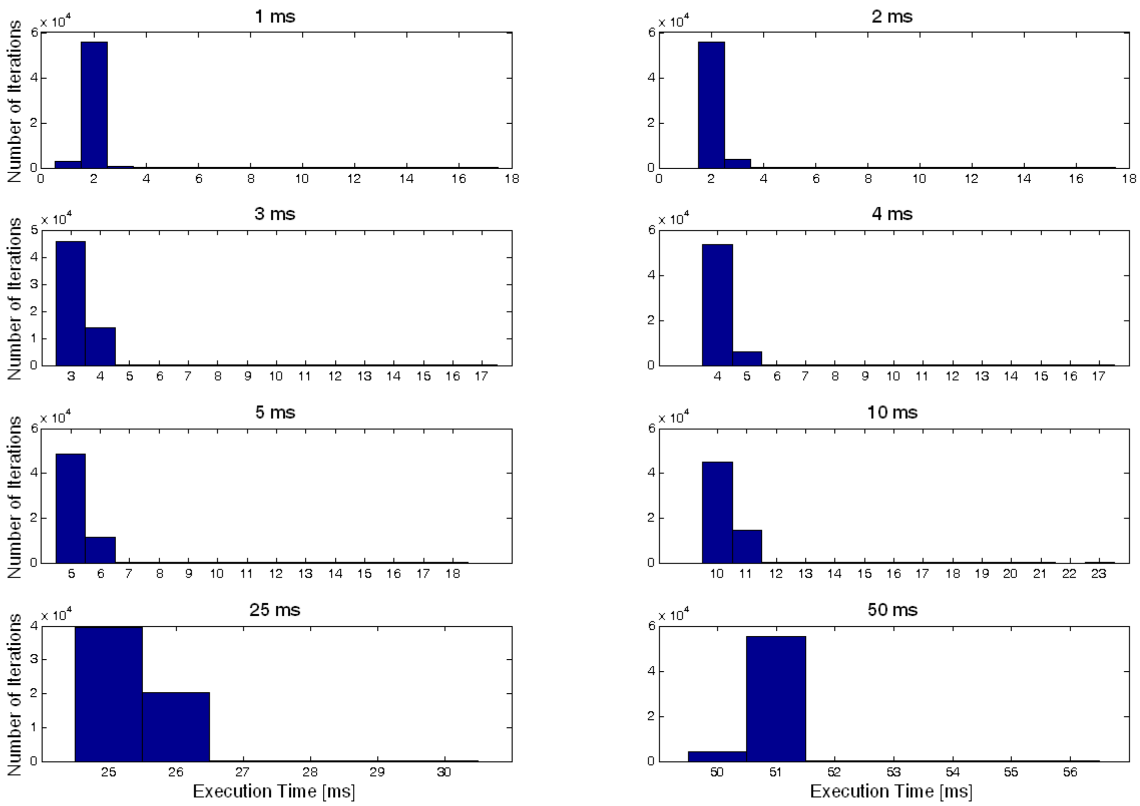

Figure 27: Histograms of Δt_{actual} for multiple simulation runs (60,000 iterations each) using different Δt

II.3.6 Conclusion

This section presented a hardware-in-the-loop on-ramp evaluation system for the URMS, which consists of a personal computer running an on-ramp microscopic traffic model, a CID and a 2070 controller running the URMS. The system was developed to assist in the debugging and testing process involved with the release of a URMS version for deployment in the field. Since this tool was specifically tailored for a 2070 controller running the URMS, it allows for an easy configuration of the system and a user-friendly interface that matches or complements some of the URMS debugging features. The paper also presented an analysis of the real-time nature of the simulator that shows that for all practical purposes, Windows XP limited real-time capabilities do not affect the performance of the HILS tool. Future research involves adding more versatility to the tool and enabling communication with the controller in order to increase synchronization.

Conclusions

This document described the work done over the first year of PATH T.O. 6329. The goal of T.O. 6329 project was to develop, along parallel and independent tracks: 1) A Ramp Metering Design Tool (RMDT) and 2) Software tools to conduct a Ramp Metering with Queue Control Field Test.

The Ramp Metering Design Tool (RMDT) is a Matlab based software tool that guides the user through a clearly defined sequence of steps leading from data collection (using PeMS), through calibration, to simulation. The final outcome of the process is a calibrated model of the freeway which can be used to test different operational strategies, such as ramp metering. This tool has been integrated into a larger set of Tools for Operational Planning (TOPL), which is being further developed by our research team under sponsorship by CALTRANS projects TOPL-3 and I-80 Freeway Corridor Monitoring and Control. TOPL is a suite of software tools for specifying freeway corridors operational improvement strategies, such as ramp metering, demand and incident management, and for quickly estimating the benefits of such improvements. Currently, TOPL can model both freeways and arterial traffic flow [9].

The Ramp Metering Design Tool was tested by Jan Hueper, an exchange student from the Institute for Transport und Automatisierungstechnik at the Leibniz Universitaet in Hannover who had no prior experience with the software or background in traffic theory, to successfully construct a calibrated model of the I-80 East freeway, in a period of about 4 months. The step by step I-80 East model construction and calibration process is described.

In the second track of this project a 2070 controller running under Universal Ramp Metering Software (URMS) was used to test its suitability for the implementation of new onramp queue control algorithms. A new hardware-in-the-loop on-ramp simulation tool was developed to test the 2070 controller and debug the Universal Ramp Metering Software. The Hegenberger Rd. loop on-ramp to S/B 880, in District 4, was selected as the test site and a study of its traffic dynamics and queue formation was performed. The Sensys wireless vehicle detection system was analyzed and meetings were held with Sensys Networks Inc. personnel and a Sensys based onramp queue metering system was designed for the Hegenberger Rd. loop on-ramp. Republic ITS personnel was consulted to conduct the sensor installation plan, and a formal quote was received from them.

This work will be continued under RTA-22A0486 Field Test Implementation of Queue Control - Track 2. A field test is being conducted by implementing Queue Control on the Hegenberger Rd. loop on-ramp to S/B 880, in District 4 and study its effect in minimizing queue and mainline density oscillations and enhancing performance. This will be accomplished by using a 2070 traffic controller running a modified Universal Ramp Metering Software (URMS) and utilizing realtime onramp queue counts provided by a Sensys wireless vehicle detection system.

This work was funded under Task Order 6329 of the California PATH Program.

References

[1] Exploration of the breakdown phenomenon in freeway traffic. *Transportation Research Record*, 1634:64Ű69.

[2] NTCIP 1207:2001, v01.17. National Transportation Communications for ITS Protocol. Object Definition for Ramp Meter Control (RMC) units. Technical report, ASSHTO/ITE/NEMA, 2001.

[3] A simplified car-following theory: a lower order model. *Transportation Research Part B: Methodological*, 36(3):195 – 205, 2002.

[4] Hardware-in-the-loop simulation. *Transportation Research Part C: Emerging Technologies*, 12(1):73 – 89, 2004.

[5] Transportation Research Board. Highway capacity manual 2000, 2000.

[6] Caltrans. The Advanced Transportation Controller Model 2070. Student Guide. Technical report, Caltrans, 2002.

[7] Caltrans. 2070 Controller Operating System User Guide. Technical report, Caltrans, 2004.

[8] Caltrans. 2070 Controller. Introduccion to the Controller. Technical report, Material Engineering and Testing Services, 2007.

[9] A. Chow, V. Dadok, G. Dervisoglu, G. Gomes, R. Horowitz, A. A. Kurzhanskiy, J. Kwon, X.-Y. Lu, A. Muralidharan, S. Norman, R. O. Sanchez, and P. Varaiya. TOPL: Tools for operational planning of transportation networks. In *PAPER: DSCC2008-22432008, 2008 ASME Dynamic Systems and Control Conference*, 2008.

[10] C. Daganzo. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research, Part B*, 28(4):269–287, 1994.

[11] J. Kwon R. Horowitz G. Dervisoglu, G. Gomes and P. Varaiya. Automatic calibration of the fundamental diagram and empirical observations on capacity. *Transportation Research Board, 88th Annual Meeting*, 2009.

[12] G. Gomes. *Optimization and Microsimulation of On-ramp Metering for Congested Freeways.* PhD thesis, University of California, Berkeley, 2004.

[13] G. Gomes, A.D. May, and R. Horowitz. Congested freeway microsimulation model using vissim. *Transportation Research Record*, 1876:71Ũ81, 2004.

[14] R. L. Gordon. Algorithm for controlling spillback from ramp meters. *Transportation Research Record*, 1554:162–171, 1996.

[15] H. Hadj-Salem, J.M. Blosseville, and M. Papageorgiou. Alinea: a local feedback control law for on-ramp metering; a real-life study. pages 194–198, May 1990.

[16] et. al. Jan Hueper. Macroscopic modeling and simulation of freeway traffic flow. 2009.

[17] Chin-Woo Tan Pulkit Grover JD Margulici, Samuel Yang and Andre Markarian. Evaluation of Wireless Traffic Sensors by Sensys Networks, Inc. Technical report, Innovative Transportation Products and Services, 2006.

[18] R. W. Koenker. *Quantile Regression*. Cambridge U. Press, 2005.

[19] A. Kurzhanskiy. *Modeling and Software Tools for Freeway Operational Planning*. PhD thesis, University of California, Berkeley, 2007.

[20] Eil Kwon, Sangho Kim, and Taek Mu Kwon. Pseudo real-time evaluation of adaptive traffic control strategies using hardware-in-loop simulation. In *Industrial Electronics Society, 2001. IECON '01. The 27th Annual Conference of the IEEE*, volume 3, pages 1910–1914 vol.3, 2001.

[21] K. Kwong, R. Kavaler, R. Rajagopal, and P. Varaiya. Arterial travel time estimation based on vehicle re-identification using wireless sensors. *Submited for publication to TRC.*

[22] K. Kwong, R. Kavaler, R. Rajagopal, and P. Varaiya. A practical scheme for arterial travel time estimation based on vehicle re-identification using wireless sensors. In *Transportation Research Board 89th Annual Meeting*, 2009.

[23] Zhen Li, B. Johnson, A. Abdel-Rahim, and M. Kyte. Hardware and software design of an automated testing tool for traffic controllers. In *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, pages 1525–1530, Sept. 2006.

[24] A. Muralidharan and R. Horowitz. Imputation of ramp flow data for freeway traffic simulation. *Transportation Research Board, 88th Annual Meeting*, 2009.

[25] PeMS. PeMS website, 2009. http://pems.eecs.berkeley.edu, accessed 2/3/2009.

[26] Inc. Sensys Networks. The Sensys Wireless Vehicle Detection System. Advanced Technology for 21st Century Traffic management. System Overview. Technical report, 2008.

[27] E. Smaragdis and M. Papageorgiou. Series of new local ramp metering strategies. *Transportation Research Record*, 1856:74–86, 2004.

[28] X. Sun. *Modeling, Estimation, and Control of Freeway Traffic.* PhD thesis, University of California, Berkeley, 2005.

[29] X. Sun and R. Horowitz. A set of new traffic-responsive ramp-metering algorithms and microscopic simulation results. In *Transportation Research Board 85th Annual Meeting.*, 2006.

[30] X. Sun, L. Muñoz, and R. Horowitz. UC researchers develop improved ramp metering algorithms. *PATH Intellimotion*, 2005.

[31] Caltrans Traffic Operations Program. Ramp Meter Design Manual. Technical report, California Department of Transportation, 2000.

[32] J. Geistefeldt W. Brilon and M. Regler. Reliability of freeway traffic flow: A stochastic concept of capacity. 2005.

[33] W.-W. Kao W. Messner, R. Horowitz and M. Boals. A new adaptive learning rule. In *IEEE Transactions on Automatic Control*, volume 36, page 188Ű197, 1991.

[34] D. Wells and E. Torizno. URMS: Universal Ramp Metering Software User Manual. Technical report, Traffic Operations, Caltrans. Version 1.07, 2008.

[35] Yin Yan. Traffic Signal Control Program. Technical report, 2000. http://www.cs.ucsb.edu/ almeroth/classes/193/F00-yan/.

A RMDT User Manual

A.1 Installation

RMDT runs on Windows 2000 or higher, and requires Matlab and Microsoft Office. The program requires no installation other than creating the basic folder structure and copying the source code. Create folders named `\FreewayModeller` and `\Networks` under the TOPL installation directory, which can be anywhere on your hard drive. See Figure 28. The source code should be copied to `\FreewayModeller`.

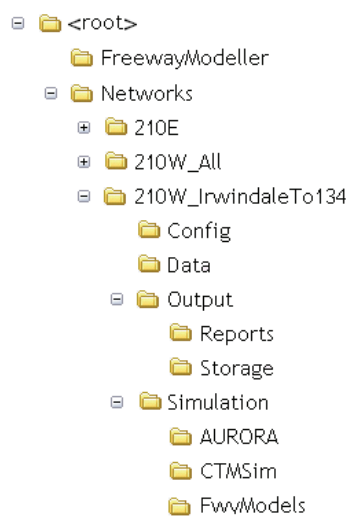

Figure 28: Folders used by RMDT

The `\Networks` folder holds the networks created with RMDT. Figure 28 shows a network folder with three sample networks: 210E, 210W_All, and 210W_IrwindaleTo134. Each network folder contains subfolders `\Config`, `\Data`, `\Output`, and `\Simulation`. These are created automatically by the program when a new network is initialized. Their contents is described below.

- `\Config`. The configuration folder holds the Excel based configuration file. This file is generated automatically by the "Make configuration file" routine, and is the main interface for setting user-defined parameters.
- `\Data`. This is where the data files downloaded from the PeMS Data Clearinghouse should be placed. The output of the PeMS Data Clearinghouse Reader (Section A.3) and Matlab data structures are also stored here.
- `\Output`. This folder contains two subfolders, `\Storage` and `\Reports`. The first holds internal

Field name	Valid entries
Folder name	Character string
Freeway number	Integer
Direction	{N,S,E,W}
Start postmile	Positive real number
End postmil	Positive real number

Table 5: Format for the "New network" window.

data files used by different components of the program. The `\Reports` folder is where Excel and Powerpoint reports are placed.

- `\Simulation`. Input files for the three TOPL traffic simulation programs are stored here in subfolders `\AURORA`, `\CTMSim`, and `\FwyModels`. The input files for FwyModels are used in the "Run FwyModels" routine.

A.2 The user interface

A snapshot of the graphical user interface is shown on the left side of Figure 29. The program is launched from Matlab with the command:

```
>> TOPLfm
```

The "Folder name" and "Freeway information" panels of the interface contain non-editable text boxes that display the basic parameters of the network. The "Procedures" panel comprises a list of checkboxes which activate or deactivate different parts of the program. The "Go" button triggers the execution of checked components, one-by-one, from top to bottom. The "PeMS data reader" button launches a separate application called the PeMS Data Clearinghouse Reader which is used to filter the large text files downloaded from the PeMS Data Clearinghouse. This application is described in Section A.3.

To begin a new network, click on **File≫New**. This opens the "New network" window shown in Figure 29, where the basic details of the freeway are entered. The format for these entries is given in Table 5. Upon clicking "Done", the program creates the required folder structure under the `<root>\Networks\<name>` directory, where `<name>` is the string entered in the "Folder name" field, and `<root>` is the TOPL installation folder. Pressing the "Done" button also saves the information to a file. This file can be used to save and reload the the freeway with **File≫Save** and **File≫Open**.

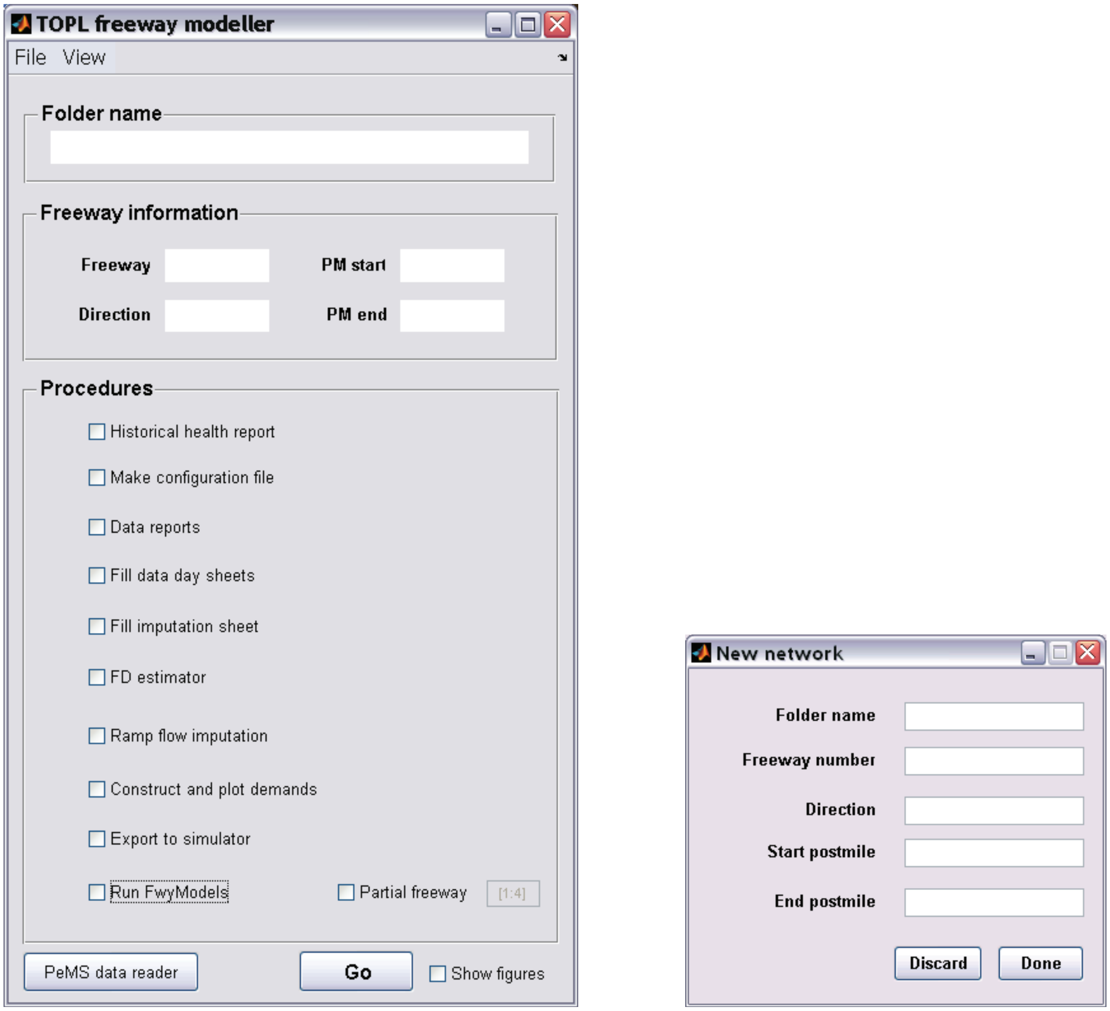

Figure 29: TOPL freeway modeller and New network windows

The **View** pulldown menu is used open some of the most commonly accessed folders. See Table 6. The routines triggered by each of the checkboxes in the "Procedures" panel are described below.

☐ `Historical health report`

Runs the historical health analysis routine, which generates a Powerpoint file with plots tracking the average health index and total daily samples over a period of several months. This report helps the user to select a set of days for further study.

☐ `Make configuration file`

Menu item	Folder
View≫Configuration folder	`<name>\Config`
View≫Reports folder	`<name>\Output \Reports`
View≫Simulation folder	`<name>\Simulation\FwyModels`
View≫Data folder	`<name>\Data`

Table 6: The View menu

This function first checks whether a configuration file already exists, and if so it exits. Otherwise, an initial configuration file is created using PeMS' geometric data. The format of the configuration file is described in Section **??**. The program automatically completes the Nodes, Links, VDSrange, FDnominal, and GoodDays sheets of the configuration file. The user can then review the file and make adjustments. A report on the partitioning of the freeway into cells is produced.

☐ `Data reports`

This routine generates several visualization reports for the daily status and 5-minute data. First, the "Good day health report" is an Excel based, color coded table showing the status for each detector station on each of the days listed in the GoodDays sheet of the configuration file. Second, speeds on the mainline are presented in contour plots for each of the good days. Finally, the raw 5-minute flows for each individual loop detector (i.e. per lane) are plotted in a series of flow reports.

☐ `Fill data day sheets`

The *data day sheets* are a set of 5 sheets in the configuration file that allow the user to assign a 5-minute data set to each detector station. This routine produces a recommended assignment, which the user can either use or override. Several subsequent routines, including the parameter estimator, the ramp imputation, and the demand construction, refer to these sheets when loading data. The format of the data day sheets is described in Section A.5.

☐ `Fill imputation sheet`

The imputation procedure calculates for each unmeasured on- or offramp an estimate of the amount of flow on that ramp, based on nearby mainline measurements. To initialize the imputation algorithm, this routine searches the Links and data day sheets for ramps requiring imputation. This list of ramps is written to the Imputation sheet so that it can be reviewed, but not modified, by the user. To modify the Imputation sheet, the user must make changes to the Link and data day sheets and rerun this routine. An explanation of how this is done is given in Section A.5.

☐ `FD estimator`

The fundamental diagram estimator produces estimates of the capacity, freeflow speed, jam density, and congestion wave speed, for each mainline detector station on the freeway. It distributes these parameters among the cells according to the 'ML det.' column of the Links sheet, as described in Section A.5. For each detector it uses the good day listed in the MLdataDay-E sheet.

☐ `Ramp flow imputation`

Checking this component invokes the ramp flow imputation algorithm. This algorithm computes estimated flows for each of the ramps listed in the Imputation sheet. Results are written to files listed in the Filename column of the Imputation sheet.

☐ `Construct and plot demands`

This function assembles the results of previous steps: the estimated parameters, the imputed and non-imputed flows, and computed split ratios, into a single data structure. It should be run whenever changes are made to the parameters of previous steps so that those changes are reflected in the simulation. It also produces a Powerpoint file with time series plots of the final onramp demands and offramp split ratios.

☐ `Export to simulator`

Using data profiles assembled in the previous step, input files compatible with the three TOPL simulators (AURORA, CTMSim, and FwyModels) are produced. These are exported to respective folders under `<name>\Simulation`.

☐ `Run FwyModels`

The FwyModels simulator is used within RMDT to assess the quality of the calibration. The output of the simulator is recorded in two Powerpoint reports described in Section **??**. Checking the "Partial freeway" box allows the user to restrict the links included in the report to those listed (in Matlab syntax) in the adjacent text box.

A.3 The PeMS Data Clearinghouse Reader

RMDT is designed to be used with the PeMS Data Clearinghouse. This is a section of the PeMS website where configuration, loop status, and detector measurements can be downloaded in text

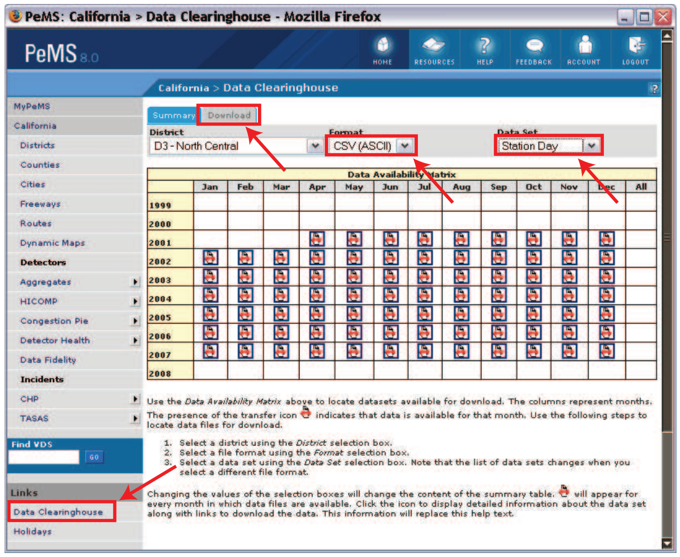

Figure 30: PeMS Data Clearinghouse

files. These text files tend to be too large for Matlab to handle efficiently, so we first filter them using the PeMS Data Clearinghouse Reader.

Three types of files are needed from the PeMS Data Clearinghouse: the station day files, the station 5-minute files, and the metadata files. These can be found by following the "Data Clearinghouse" link on PeMS' front page, clicking the "Download" tab, and then selecting "CSV (ASCII)" in the "Format" menu (see Figure 30).

Each station day file (`dXX_text_station_day_yyyy_mm.txt.gz`) contains loop health information for every detector station in an entire district over a whole month. Download a station file for each month considered in the historical loop health analysis.

Station 5-minute files (`dXX_text_station_5min_yyyy_mm_dd.txt.gz`) contain 5-minute flow, occupancy, and speed measurements for each loop in the district, over a day. Download a station 5-minute file for each good day selected for detailed study.

Metadata files (`dXX_stations_yyyy_mm_dd.txt`) contain current configuration information for each of the detector stations in the district. These files are often updated, and you will need to download all of the metadata files relevant to the selection of good days.

The downloaded files should be placed in the `<name>\Data` folder.

The PeMS Data Clearinghouse Reader is a separate application which can be launched from within RMDT, by pressing the "PeMS data reader" button, or separately from the Windows com-

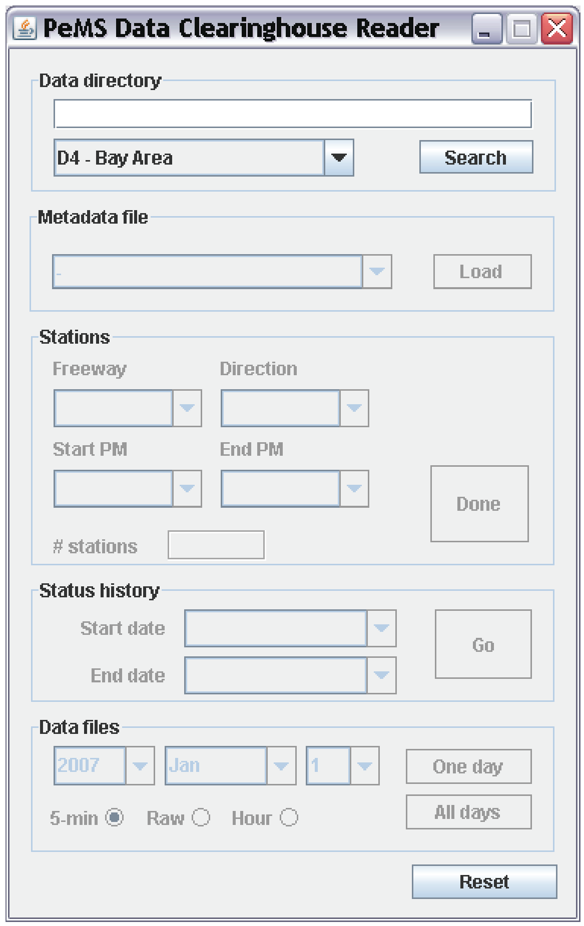

Figure 31: PeMS Data Clearinghouse Reader

mand shell with:

```
>> java -jar <root>\FreewayModeller\PeMSDCReader.jar
```

A screenshot of the program is shown in Figure 31. The following list of steps describes the use of the program.

1. Enter the address of the folder containing the PeMS data files in the "Data directory" field, select a District, and press the "Search" button. The program will search that folder for relevant metadata files and display them in the "Metadata file" pulldown menu.

2. Select a metadata file and press the "Load" button. The information contained in that file will be used to populate the "Stations" panel.

3. In the "Stations" panel, select the desired freeway, direction, starting and ending postmiles. The number of vehicle detector stations covered by your selection will be displayed in the "# stations" window. Press "Done".

4. The "Status history" panel is used to extract data from the station day files for the historical health report. Select the desired start and end dates and press "Go". This generates a reduced detector health file `<fwname>_YYYY_MM_yyyy_mm_histsts.txt`, where `YYYY_MM` and `yyyy_mm` are respectively the start and end dates of the historical health analysis.

5. The "Data files" panel is used to extract information from the daily 5-minute data files. The "One day" button is used to process a single day, the "All days" button processes all of the 5-minute data files for the given district. It is important that each 5-minute data file be processed with the appropriate metadata file. Make sure that the file shown in the "Metadata file" panel is valid for the 5-minute file being processed. The procedure produces three files for each day: `<fwname>_yyyy_mm_dd_5min.txt`, `<fwname>_yyyy_mm_dd_cfg.txt`, and `<fwname>_yyyy_mm_dd_sts.txt`, respectively containing 5-minute measurements, configuration, and status information for the stations defined in the "Stations" panel.

You can restart the process at any time by pressing the "Reset" button. The original Data Clearinghouse files can be deleted once the process is completed.

A.4 Input and output files

A.5 The configuration file

All user-defined parameters in RMDT are read from an Excel spreadsheet called the configuration file. The name of this file is `<fwname>config.xls`, where `fwname` is the name and direction of the freeway. It is stored under `<name>\Config`, which can be accessed with **View≫Configuration folder**. The configuration file contains editable and non-editable areas. The non-editable areas are used by the program to display information which cannot be directly manipulated by the user. In some cases, such as the `Good day` columns, that information can be overridden. The spreadsheet consists of the 11 individual sheets described below.

`Nodes`

This sheet is generated automatically by the program. Its only use is in the generation of input files for the AURORA simulator, which is based on a nodes/link paradigm. The sheet consists of a list of nodes, each with an ID, lat/long coordinates, and an absolute postmile. It is assumed that the ith node connects the ith and $i+1$th links, or equivalently, that the ith link is limited by the ith and $i+1$th nodes. This convention must be kept when a new link is manually added or one is deleted from the list.

`Links`

The Links sheet shows the partition of the freeway into cells, or links, along with the main characteristics of each cell. This sheet is initialized automatically by the program, but should be inspected and possibly manually corrected as its entries have a strong influence on the results of the simulation. Each row in the table represents a single cell. Columns are described below.

- `Link [int]`. ID of the cell.
- `Up Node [int]`. ID of the upstream bounding node.
- `Dn Node [int]`. ID of the downstream bounding node.
- `Length [ft]`. Length of the cell in feet.
- `Lanes [int]`. Number of lanes. This is initialized with the number of lanes reported in PeMS. However, PeMS actually reports the number of loops (or loop pairs) in a vehicle detector station, which may not equal the number of lanes in some cases. This column should therefore

be carefully checked and corrected. The program will issue a warning if the number of lanes listed here does not match the number reported by PeMS. The user-defined number of lanes is always considered correct.

- `Auxlanes [float]`. Number of auxiliary lanes. Auxiliary lanes are additional lanes that, while not qualifying as full lanes, may have some effect on the capacity of the segment. They are used in FwyModels in the computation of capacity and jam density. A segment with 5 lanes and 0.5 auxiliary lanes will be assigned a capacity 5.5 times the estimated per-lane capacity. The only differences between normal lanes and auxiliary lanes are that normal lanes are assigned by PeMS, whereas auxiliary lanes are assigned manually, and the number of normal lanes must be an integer, while auxiliary lanes may take fractional values.

- `ML det. [vds]`. This column lists the mainline VDS from which each cell will receive fundamental diagram parameters. Each entry may itself be a list of VDS numbers, in the case that the cell contains more than one VDS. If so, the parameters will be taken from the first station in the list. However, if there are multiple stations listed for a cell, they must all reside physically within that cell.

- `Has ML [bool]`. Contains a list of 1's and 0's with the same dimension as "ML det.". Each entry is 1 if the corresponding VDS in "ML det." is physically within the cell, and 0 otherwise. According to the rule stated above, a non-scalar entry in this column may not contain any zeros.

- `Onramp data source [vds,file]`. Indicates whether the cell contains an onramp, and the source of data assigned to that onramp. The format is described in Section A.5.1

- `Metered [lanes]`. Number of metered onramp lanes.

- `Storage [veh]`. Maximum number of vehicles that fit on the onramp upstream of the metering light.

- `Onramp Flow Capacity [vph]`. Maximum unmetered onramp flow.

- `Offramp data source [vds,file]`. Indicates whether the cell contains an offramp, and the source of data assigned to that offramp. The format is described in Section **??**

- `Offramp Flow Capacity [vph]`. Maximum offramp flow.

- `Offramp down ML [vds,file]`. Source of data for the mainline flow immediately downstream of the offramp. This data source is used in the calculation of split ratios for the offramp. This column is not filled automatically but must be completed by the user. It is often the case that there is no direct measurement of the mainline flow directly downstream of the offramp.

Linear combinations of mainline measurements, and measured and imputed ramp flows can be used, as described in Section **??**.

`VDSrange`

This read-only sheet displays the starting and ending absolute postmiles for the freeway.

`FDnominal`

This sheet defines the nominal values and ranges for the parameters of the fundamental diagram. These values are used when there is insufficient data to carry out the estimation, or when the loop detector is deemed faulty by PeMS. The status of the detectors can be seen and overwritten in the `MLdataDay-E` sheet.

- `vf [mph]`. Nominal freeflow speed.
- `w [mph]`. Nominal congested wave propagation speed.
- `w min [mph]`. Lower bound on the estimated w.
- `w max [mph]`. Upper bound on the estimated w.
- `Qmax [vphpl]`. Nominal capacity for non-bottleneck sections.
- `Qbottleneck [vphpl]`. Nominal bottleneck capacity.

`GoodDays`

This sheet lists the days for which 5-minute data can be used in the construction of demand profiles. Demand profiles may be constructed from a single day or from several typical days. This sheet lists all of the available days and assigns to each an index. The sheet is initialized by the program with all days found in the `\Data` directory and can be modified by the user.

`MLdataDay-E`

This sheet is used by the fundamental diagram parameter estimator. The first three columns are not editable, they list all of the mainline detector stations along with their best day (according to PeMS's health status and number of samples), and the corresponding status. The status is the average status over all lanes, so that a status of 0.4 means that 40% of the loops in the station are good. The fourth column, `Estimation Good day override`, can be used to override the second column, `Good day`. The program uses the data from the day indicated by either the second or fourth column to estimate traffic parameters. An entry of -1 in the `Good day` column indicates that there are no good days for this detector, and default parameters should be used. A -1 can be used in the `Estimation Good day override` column to force the program to use default parameters.

MLdataDay-I, ORdataDay, FRdataDay, FFdataDay

These sheets are used by the ramp flow imputation algorithm. The `MLdataDay-I` sheet is used to affect the grouping of cells into *mega-cells*, which are the basic topological objects of the imputation algorithm. Cells are grouped into mega-cells such that each mega-cell contains a single good mainline detector station. Thus, adjacent mega-cells can be merged by declaring the mainline station of one of them as bad. This is done by setting the `Good day status` entry of the `MLdataDay-I` sheet to -1. Similarly, mega-cells can be split by setting `Good day status` to 1, although this is probably not useful. If the detector station is good, then the data assigned to it is loaded from the day listed under `Good day [#]`, which can be overridden with `Imputation Good day override`.

The `ORdataDay`, `FRdataDay`, and `FFdataDay` provide information about onramps, offramps, and freeway connectors respectively. This information is used by the "Fill imputation sheet" routine to construct the list of imputed ramps. `Good day` and `Good day status` respectively display the best day and its corresponding status for each ramp. These can be overridden with the `Imputation Good day override` and `Imputation Status override` columns. A ramp with a good day or status of -1 is considered bad, and added to the imputation list. If the ramp is considered good (because its day and status are positive), then the data assigned to it is loaded from the good day.

Imputation

This sheet displays the list of ramps whose flows are to be imputed. The list is not directly editable. It is constructed by the "Fill imputation sheet" routine by scanning the Links sheet for onramp and offramp sources with a `file_` prefix, and by searching the `ORdataDay`, `FRdataDay`, and `FFdataDay` sheets for ramps with no good day or a bad status. The `VDS` column displays the station number of faulty on or offramps, and shows a -1 when there is no associated VDS. The `Filename` column gives the name of the text file where the result of the imputation routine is written. These files are saved in the `<Folder name>\Output\Storage` folder.

A.5.1 Format for data source columns of the Links sheet

The `Onramp data source`, `Offramp data source`, and `Offramp down ML` columns share a common format. A dash ('-') is used to indicate that no ramp is present in the cell. Otherwise, the entry can be any linear combination of detector stations and text file names. Vehicle detector stations are indicated with the prefix `vds_` followed by the VDS number. Text files are indicated with the prefix `file_` followed by the file name (excluding the .txt extension). The format for these files is two columns of numbers separated by a tab character. The first column is a vector of time in minutes, the second is the corresponding value of flow in veh/hour.

Constant multipliers as well as addition, and subtraction symbols are allowed in the data source columns. For example, the following entry indicates that the flow should be computed by taking difference between three time the flow measured by VDS 1234567 and the flow contained in `manualcounts.txt`:

```
3*vds_1234567 - file_manualcounts
```

As explained in the description of the Imputation sheet, the list of ramps requiring flow imputation includes all ramps in the Links sheet whose source contains the `file_` prefix. Thus, entries in the `Onramp data source` and `Offramp data source` columns containing the `file_` prefix are automatically imputed, and it does not make sense to define these flows as linear combinations (since the total is estimated). Linear combinations of sources in the `Onramp data source` and `Offramp data source` columns should therefore contain only `vds_` entries. `file_` entries in these columns are used to force the program to impute the ramp flow.

A.6 Other data sources

Sources of ramp flow data other than the PeMS Data Clearinghouse can be incorporated into RMDT by overwriting the output of the imputation routine. This is done by defining the source of flow in the Links sheet with the `file_` prefix followed by the name of a text file containing the manual counts. The program will consider this as an onramp requiring imputation, and will write the computed flow to the `\Storage` folder. Simply replace this file with the file containing the manual counts before running the "Construct and plot demands" routine.

A.7 Reports

RMDT generates a series of Powerpoint and Excel based reports with results generated by the program. These reports are placed in `<name>\Data\Reports`, which can be accessed by selecting **View≫Reports folder**. The list of reports is given in Table 7.

Name	Generated by	Contents
`*_HealthHistory.ppt`	Historical health report	Slide 1: Top 20 days with respect to status and number of samples. Slide 2: Temporal evolution of samples and status.
`*_FreewayCells.ppt`	Make configuration file	Stages of the cell generation routine.
`*_GoodDayHealthReport.ppt`	Data reports	Daily VDS status for all ML, OR, FR, HV, and FF stations and days.
`*_PeMSSpeeds.ppt`	Data reports	Measured speed contour for each day.
`*_Flows_OR.ppt`	Data reports	Time plots of 5-minute onramp flows.
`*_Flows_FR.ppt`	Data reports	Time plots of 5-minute offramp flows.
`*_Flows_ML.ppt`	Data reports	Time plots of 5-minute mainline flows.
`*_Flows_HV.ppt`	Data reports	Time plots of 5-minute HOV flows.
`*_Flows_FF.ppt`	Data reports	Time plots of 5-minute freeway connector flows.
`*_FundamentalDiagrams.ppt`	FD estimator	Thumbnail flow-density plots with estimated parameters for each mainline station.
`*_ImputationResults.ppt`	Ramp flow imputation	Measured and imputed flows used within the imputation algorithm.
`*_DemandsAndSplitRatios.ppt`	Construct and plot demands	Onramp flows and offramp split ratios for each link.
`*_SimulationResults.ppt`	Run FwyModels	Simulation variables including speed contours.
`*_SimulationVersusPeMS.ppt`	Run FwyModels	Plots of simulated and measured performance measures.

Table 7: Reports. * is the freeway name and direction.