STATE OF CALIFORNIA • DEPARTMENT OF TRANSPORTATION
TECHNICAL REPORT DOCUMENTATION PAGE
TR0003 (REV 10/98)

1. REPORT NUMBER	2. GOVERNMENT ASSOCIATION NUMBER	3. RECIPIENT'S CATALOG NUMBER
CA16-2516		

4. TITLE AND SUBTITLE	5. REPORT DATE
DOE Fleet In-Vehicle Data Acquisition System (FIDAS) Technical Support and Testing	September 30, 2015
	6. PERFORMING ORGANIZATION CODE

7. AUTHOR	8. PERFORMING ORGANIZATION REPORT NO.
Kin S. Yen, Bahram Ravani, and Ty A. Lasky	UCD-ARR-15-09-30-02

9. PERFORMING ORGANIZATION NAME AND ADDRESS	10. WORK UNIT NUMBER
AHMCT Research Center UCD Dept. of Mechanical & Aerospace Engineering Davis, California 95616-5294	
	11. CONTRACT OR GRANT NUMBER
	IA 65A0416, Task 2516

12. SPONSORING AGENCY AND ADDRESS	13. TYPE OF REPORT AND PERIOD COVERED
California Department of Transportation P.O. Box 942873, MS #83 Sacramento, CA 94273-0001	Final Report October 1, 2012 – September 30, 2015
	14. SPONSORING AGENCY CODE
	Caltrans

15. SUPPLEMENTARY NOTES

16. ABSTRACT

This report documents the research project "DOE Fleet In-Vehicle Data Acquisition System (FIDAS) Technical Support and Testing." The project goals were to assess the current state-of-the-art for fleet tracking, perform a pilot study for a portion of the Caltrans light vehicle fleet, assess areas for potential improvements and savings, and perform a cost-benefit analysis. The report provides background and literature and product review, including commercial off-the-shelf systems as well as previous research using similar devices. An overview of the FIDAS pilot study with 200 vehicles, along with lesson learned is provided, followed by a cost-benefit analysis based on the data gathered from the pilot study. The pilot study shows that fuel used due to speeding over 80 MPH is about 1%, and due to idling is about 13%. Eliminating 85% of speeding and idling is a realistic goal, and would result in 12% fuel savings. In addition, the study showed that installation of telematics devices doesn't by itself change driver behavior in reducing idling or speeding without coaching, contrary to some FIDAS solution providers' claims. The pilot study also demonstrated that when a speeding policy is implemented and enforced, driver behavior will change, resulting in significant drop in speeding violations and duration. Therefore, policies on reducing idling and speeding must be established and enforced, including coaching and training, in order to gain the benefits of idling and speeding reduction. The Cost-Benefit analysis illustrated that 12% fuel reduction would result in a return of investment in about 5 years for a full FIDAS implementation.

17. KEY WORDS	18. DISTRIBUTION STATEMENT
Telematics, GPS, Fleet management, Automated Vehicle Location (AVL)	No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.

19. SECURITY CLASSIFICATION (of this report)	20. NUMBER OF PAGES	21. COST OF REPORT CHARGED
Unclassified	125	

Reproduction of completed page authorized

DISCLAIMER/DISCLOSURE STATEMENT

The research reported herein was performed as part of the Advanced Highway Maintenance and Construction Technology (AHMCT) Research Center, within the Department of Mechanical and Aerospace Engineering at the University of California – Davis, and the Division of Research, Innovation and System Information at the California Department of Transportation. It is evolutionary and voluntary. It is a cooperative venture of local, State and Federal governments and universities.

This document is disseminated in the interest of information exchange. The contents of this report reflect the views of the authors who are responsible for the facts and accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the State of California, the Federal Highway Administration, or the University of California. This publication does not constitute a standard, specification or regulation. This report does not constitute an endorsement of any product described herein.

For individuals with sensory disabilities, this document is available in Braille, large print, audiocassette, or compact disk. To obtain a copy of this document in one of these alternate formats, please contact: the Division of Research, Innovation and System Information, MS-83, California Department of Transportation, P.O. Box 942873, Sacramento, CA 94273-0001.

Advanced Highway Maintenance and Construction Technology Research Center

Department of Mechanical and Aerospace Engineering
University of California at Davis

DOE Fleet In-Vehicle Data Acquisition System (FIDAS) Technical Support and Testing

Kin S. Yen, Bahram Ravani &
Ty A. Lasky: Principal Investigator

California Department of Transportation

Division of Research, Innovation and System Information

ABSTRACT

This report documents the research project "DOE Fleet In-Vehicle Data Acquisition System (FIDAS) Technical Support and Testing." The project goals were to assess the current state-of-the-art for fleet tracking, perform a pilot study for a portion of the Caltrans light vehicle fleet, assess areas for potential improvements and savings, and perform a cost-benefit analysis. The report provides background and literature and product review, including commercial off-the-shelf systems as well as previous research using similar devices. An overview of the FIDAS pilot study with 200 vehicles, along with lesson learned is provided, followed by a cost-benefit analysis based on the data gathered from the pilot study. The pilot study shows that fuel used due to speeding over 80 MPH is about 1%, and due to idling is about 13%. Eliminating 85% of speeding and idling is a realistic goal, and would result in 12% fuel savings. In addition, the study showed that installation of telematics devices doesn't by itself change driver behavior in reducing idling or speeding without coaching, contrary to some FIDAS solution providers' claims. The pilot study also demonstrated that when a speeding policy is implemented and enforced, driver behavior will change, resulting in significant drop in speeding violations and duration. Therefore, policies on reducing idling and speeding must be established and enforced, including coaching and training, in order to gain the benefits of idling and speeding reduction. The Cost-Benefit analysis illustrated that 12% fuel reduction would result in a return of investment in about 5 years for a full FIDAS implementation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF CODE LISTINGS

LIST OF ACRONYMS AND ABBREVIATIONS

Acronym	Definition
AASHTO	American Association of State Highway and Transportation Officials
AHMCT	Advanced Highway Maintenance and Construction Technology
API	Application Programming Interface
AVL	Automated Vehicle Location
Caltrans	California State Department of Transportation
CAN	Controller Area Network
CDMA	Code-Division Multiple Access
CERT	Center for Environmental Research and Technology
COTS	Commercial-Off-the-Shelf
CSV	Comma Separated Values
DLC	Diagnostic Link Connector
DOE	Division of Equipment
DOT	Department of Transportation
DRISI	Division of Research, Innovation and System Information
DTC	Diagnostic Trouble Code
DVBE	Disabled Veteran Business Enterprise
ECU	Electronics Control Units
EPA	Environmental Protection Agency
FIDAS	Fleet In-vehicle Data Acquisition Systems
GB	Gigabyte
GHG	Green House Gas
GIS	Geographic Information System
GLONASS	Globalnaya Navigatsionnaya Sputnikovaya Sistema or Global Navigation Satellite System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HSGPS	High-Sensitivity GPS
HTML	Hypertext Markup Language
KML	Keyhole Markup Language
MAF	Mass Air Flow
MB	Megabyte
MPG	Miles Per Gallon
MPH	Miles Per Hour
OBD	On-Board Diagnostics
OpenGTS	Open GPS Tracking System
OS	Operating System
PTO	Power Take-Off
RAM	Random Access Memory
RFI	Request for Information
RFID	Radio Frequency Identification
ROI	Return on Investment
RPM	Revolutions Per Minute
SAE	Society of Automotive Engineers
SDK	Software Development Kit
SHP	ESRI shape file format
TAG	Technical Advisory Group
TTFF	Time-to-First-Fix
UCD	University of California-Davis
UCR	University of California, Riverside

Acronym	Definition
USB	Universal Serial Bus
WVDOT	West Virginia Department of Transportation
XML	eXtensible Markup Language

ACKNOWLEDGMENTS

CHAPTER 1:
INTRODUCTION

Background

The California Department of Transportation (Caltrans) has a fleet of about 7,100 light-weight vehicles and 2,000 heavy vehicles. Automated Vehicle Location (AVL) systems using GPS (Global Positioning System) have been used to provide better management of large vehicle fleets in both the private and public sector for decades. Recent technological advancement have lowered the cost of these systems and improved their data collection ability. With increasing availability of the automotive On-board Diagnostic (OBD) II port, the new generation of GPS/AVL systems can collect vehicle data beyond just location. Some of these systems have additional sensor (digital and/or analog) interfaces that enable customers to collect other sensor data on board the vehicle. The collected data are presented to the users through a website, minimizing the need for special software. The Caltrans Division of Equipment (DOE) is assessing the use of cost-effective solutions (fleet in-vehicle data acquisition systems, or FIDAS) to improve Caltrans fleet management.

The initial phase of this assessment is targeting light fleet vehicles (less than 3/4 ton) that typically allow more uniform installations, require less customization, and have reduced per vehicle costs. The objective of this effort was to improve fleet management, reduce fuel use, and produce accurate fuel use reporting. Improvements were accomplished by automating trip-related data collection and report generation, and improving communications to equipment managers. To facilitate FIDAS deployment, AHMCT researchers provided technical support and information gathering, commercial off-the-shelf (COTS) components (hardware and software) testing and evaluation for light fleet vehicles, and subsequent data analysis and reporting.

AHMCT researchers worked with DOE to:

- Improve fleet operation using data collected with FIDAS. This task includes developing algorithms to identify waste and inefficiency.

- Perform cost-benefit analysis and determine the return on investment.

The results have facilitated deployment of Caltrans FIDAS for the light vehicle fleet. The research has led to hands-on experience with a range of systems. This will greatly facilitate procurement and deployment at every stage.

CHAPTER 2:
LITERATURE AND COMMERCIAL PRODUCT REVIEW

Introduction

A preliminary investigation on the use of GPS/AVL in State Departments of Transportation was carried out before the project began. Additional literature review was performed after the start of the project. In addition, a Request for Information (RFI) was sent to thirty-nine solution providers in order to gather the latest state-of-the-art information in December 2011. The RFI document is located in Appendix A. Twenty-one solution providers responded to the RFI in January 2012. Subsequently, fifteen solution providers made presentations to the project Technical Advisory Group (TAG). The RFI and solution providers' presentations gave the TAG up-to-date information on FIDAS solutions.

GPS/AVL Usage in State and Local Government Agencies

The US Department of Energy has published its Guidance for Federal Agencies on Executive Order 13514 Section 12, Federal Fleet Management [2], which focuses on developing a fleet management framework (plan, collect, strategize and implement) to meet the Green House Gas (GHG) reduction goals by Executive Order 13514. These guidelines apply to other agencies that aim to improve their fleet management and reduce GHG.
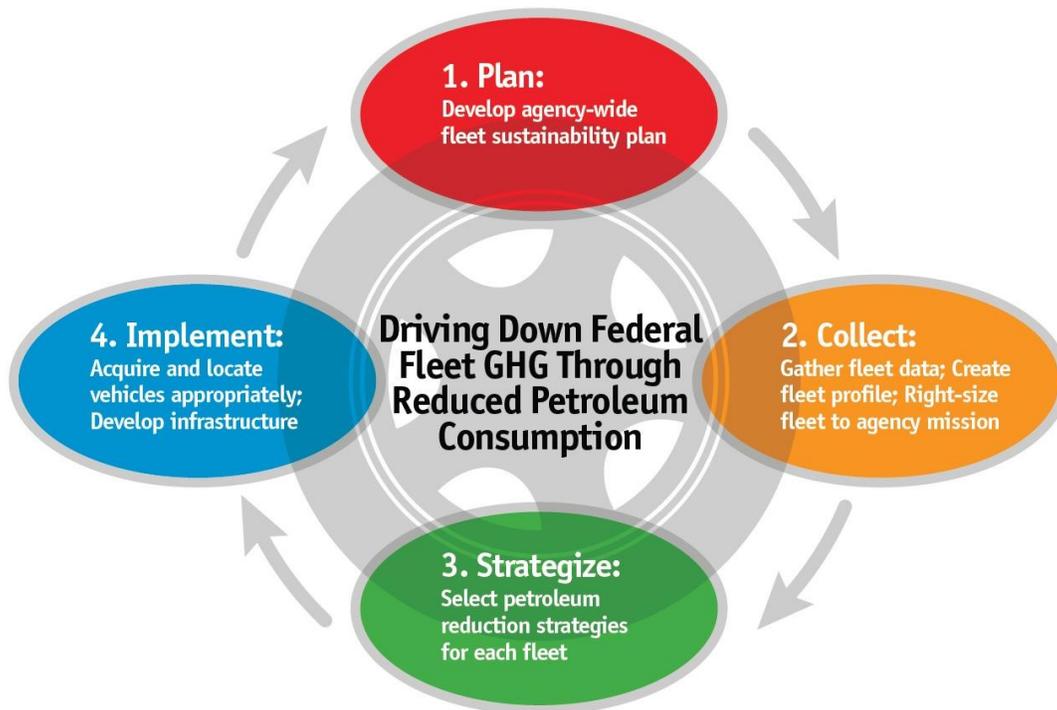
Figure 2.1. Annual fleet strategic planning and implementation process framework for agencies [2]

A Caltrans Division of Research, Innovation and System Information (DRISI) preliminary investigation on the use of GPS/AVL in State Department of Transportation echoed the findings in the presentation "GPS/AVL Technologies in Use at State DOTs" by Marie Venner of Venner Consulting, presented at the AASHTO (American Association of State Highway and Transportation Officials) Subcommittee on Maintenance AVL/GPS/Telematics Expert Panel, July 2011. The majority of the large scale uses of GPS/AVL/Telematics in DOTs are related to snow removal activities. The associated cost-benefit analysis (for example, the Iowa DOT Cost Benefit Analysis Results Geographic positioning systems and automatic vehicle location) included savings specifically applying only to snowplows. The actual cost benefit data of GPS/AVL/Telematics on government agencies are not available.

The California Department of Water Resources started a fleet management project using a GPS/AVL/Telematics solution. However, the project manager, Brian Borlace, indicated that the project completion date has been delayed a few months due to numerous challenges. Their results are not available as of the publication of this report. The West Virginia Department of Administration Office of Fleet Management has developed a Vehicle Telematics Cost-Benefit Analysis - FY2013 Appropriations Request [7]. Their cost benefit analysis is discussed in further detail in Chapter 5.

RFI Results Summary

Requests for Information (RFI) were sent to thirty-nine solution providers. Caltrans received twenty-one responses, and fifteen vendors presented in February 2012. The complete list of the solution providers that responded to the RFI is provided in Appendix B. The RFI provided valuable information on the services and capability of their FIDAS telematics devices. A typical solution was composed of telematics hardware, data services, and web services. Customers interface with a web browser to the web server for visualizing data and creating reports. The solution providers can be classified into three categories:

Table 2.1 – Solution provider categories

Category	Note
Complete solution provider	Provides proprietary hardware, data services, and website solution. Their hardware or web service is not compatible with other solution providers' hardware or web services. Some of these solution providers have multiple resellers which may aid the bidding processes if required by purchasing agency.
COTS FIDAS telematics provider	Provides Commercial Off-the-Shelf telematics hardware only. Their hardware is compatible with several web services solution providers. The communication protocol with the web server is generally known, so that anyone may build a web server to receive their collected data. However, there is no standard in the data communication among different hardware solution providers.

3

Web Services solution provider	Provides web services using multiple COTS telematics hardware. Their web software is compatible with multiple COTS telematics hardware. Their solutions often bundle COTS hardware and data services.

The new generation of FIDAS telematics devices has highly sensitive GPS or GPS combined with Global Navigation Satellite System (GNSS) receivers. GNSS receiver could take advantage of the Russian GLONASS (Globalnaya Navigatsionnaya Sputnikovaya Sistema or Global Navigation Satellite System) and the European Galileo GNSS and provide a better positional solution in GNSS challenged areas such as inside parking structures. The highly sensitive GPS/GNSS receivers eliminate the need for external GPS/GNSS antenna while providing a position solution in most conditions. As a result, the telematics devices may be installed under the dash of the vehicle or in other hidden location inside the vehicle.

In addition, most FIDAS telematics devices have either On-Board Diagnostic (OBD)-II and/or SAE (Society of Automotive Engineers) J-Bus (J1939 and J1708) interfaces. Both OBD-II and J-Bus interfaces allow the telematics devices to collect vehicle operating data from the vehicle on-board electronic control units (ECU). All passenger vehicles (light duty) built after 1996 are required to have OBD-II interface. Most modern heavy trucks have J1708 and/or J1939 J-bus. Telematics systems with both OBD-II and J-Bus interfaces can be used on both light duty automobiles and heavy trucks providing a uniform hardware on all vehicles. In general, OBD-II provides vehicle speed, engine revolutions per minute (RPM), and mass air flow (MAF) rate. The telematics devices calculate the vehicle distance traveled by integrating the vehicle speed data from the OBD-II. This is more accurate than the estimation from the GPS/GNSS positional solution. The solution providers claim that distance traveled accuracy should be within 1 to 2%. Furthermore, the telematics devices could calculate the fuel consumption rate using the MAF and stoichiometric air-fuel ratio of 14.7 for gasoline engine. However, the stoichiometric ratio of 14.7 is not well-maintained when the engine is cold or under high engine load. Nevertheless, by integrating the fuel consumption rate, the total fuel used can be estimated. In our pilot study, the telematics devices were unable to collect fuel consumption data from some vehicles' ECU. These vehicles are mostly made by Chrysler/Dodge/Jeep.

Telematics devices with OBD-II interface also collect vehicle diagnostic fault codes and can deliver alerts to managers and mechanics for vehicle repair. Moreover, some telematics devices have accelerometers to detect hard braking, high acceleration, and hard cornering events that usually indicate unsafe driver behavior. In addition, some offer options for driver identification (Radio Frequency Identification or RFID card, key fob, or user input), digital inputs (for recording strobe warning light activation, power take-off (PTO) usage, or other sensors output such as diesel engine particle trap), and/or analog sensor inputs.

The location data are collected every one to five minutes. A two-minute update frequency is most common. Some telematics systems collect the vehicle position based on vehicle heading change and speed. The collected data are transmitted to the web server via cellular data network, WiFi, satellite communications, or proprietary wireless link. The majority of the systems utilized a cellular data network (GSM, Global System for Mobile Communications, or CDMA, Code-

4

Division Multiple Access). Some systems are capable of using satellite communication when a cellular data network is not available.

Most solution providers generate fleet summary data reports based on collected data. Example reports include trip report (trip start and end: location, address, time, odometer reading, fuel used, trip mileage), speed violation report, idling report, night and weekend driving report, and fleet utilization reports. Most solution providers are willing to develop custom reports for large customers. Some solution providers allow access of raw data through proprietary SDK (Software Development Kit), API (Application Programming Interface), or web services.

The cost of a complete FIDAS solution includes the in-vehicle telematics device cost, monthly services cost (data transmission cost and web services cost), device activation cost, and device installation. Most device installations are fairly simple and could be performed by the customer. However, the customer may choose to pay the solution provider to install the in-vehicle devices. Some solution providers combine all the costs into a single monthly charge resulting in lower initial cost. The service providers often provide a discount to the initial devices cost and the monthly services cost for large quantity order. Based on the RFI responses, the distribution of the initial devices and monthly services costs are shown in Figure 2.2 and 2.3. The specifications of the telematics devices are not the same. The majority of the devices' cost is between $250 and $350, and the majority of the monthly cost is between $20 and $35. Some devices have more features, such as satellite communication, digital and analog inputs, than others. Therefore, direct cost comparison should not be employed. These cost figures should be used only for budgetary estimates.
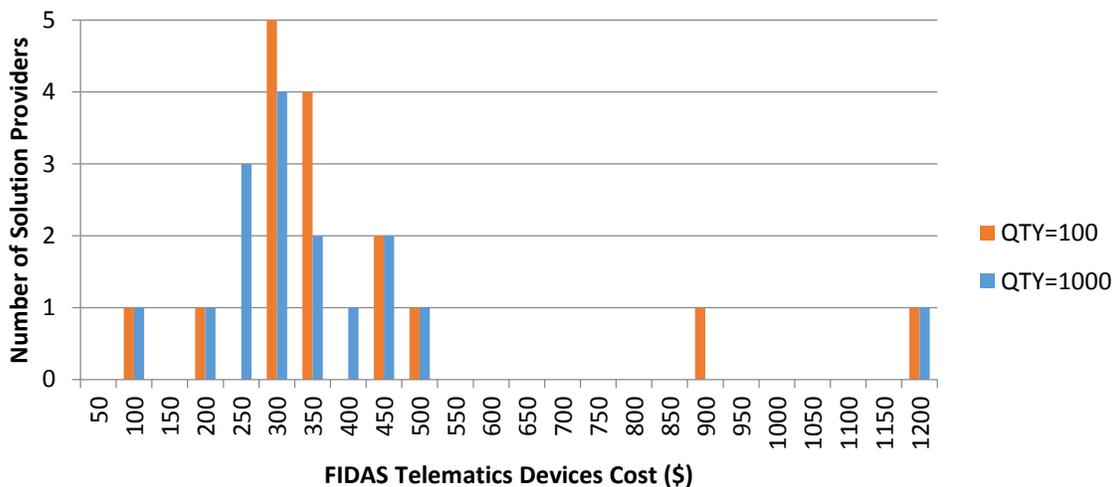
Figure 2.2. Distribution of FIDAS telematics device costs ($)

5
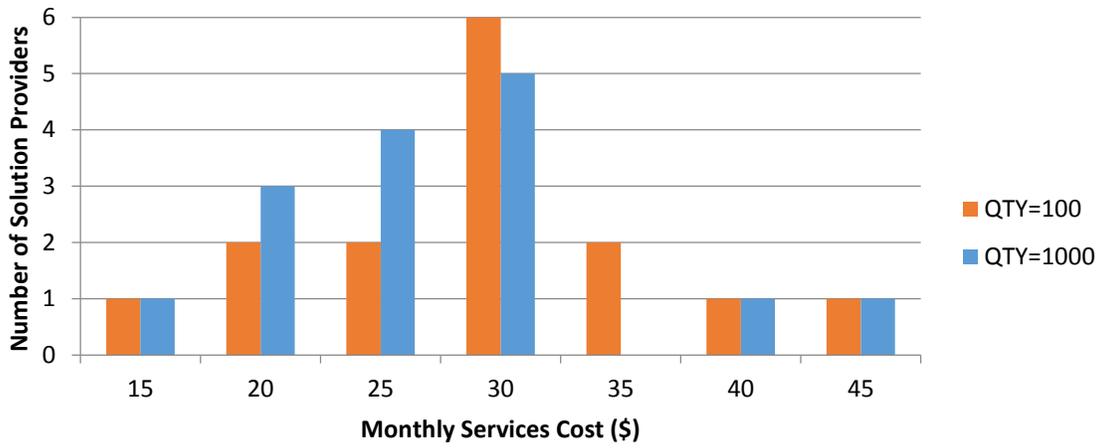
Figure 2.3. Distribution of solution provider monthly service costs

CHAPTER 3:
EVALUATION OF COTS TELEMATICS SOLUTIONS

The researchers and Caltrans DRISI have used various FIDAS telematics devices through their previous and current Caltrans research projects. These projects provided valuable lessons on the use of telematics devices. In addition, other COTS telematics devices, open-source software, and complete solutions were also evaluated under the scope of this project.

Idling Study of Heavy Duty Vehicles in Caltrans

AHMCT researchers have used Preco Systems' telematics devices in a project on assessing Caltrans heavy duty vehicle idling, as well as tracking AHMCT research vehicles for over five years [8]. The Preco system provides API access to the raw data collected and basic reporting and mapping through their website. Their telematics device has two digital inputs, OBD-II and J-Bus interfaces, WiFi and cellular mobile internet connectivity. Preco's willingness to modify the telematics device firmware to fit our research work was a major factor in their selection.

The objective of the Idling Study of Heavy Duty Vehicles in Caltrans project was to determine the amount of idling of vehicle engines and the amount of fuel used during idling periods. AHMCT instrumented thirty Caltrans vehicles with Preco Systems telematics devices which collected the vehicle location, fuel consumption rate, and digital input, and transmitted the data to a central data repository. With the available information, along with AHMCT-developed customized input triggers to identify periods of active work, Caltrans can quantify idling and fuel use, and identify periods of stationary idling when work is being actively performed.

The average vehicle survey period length was 14 months. The Preco system was able to log the vehicle instantaneous fuel rate from 25 vehicles out of the 30 vehicles instrumented. The average idling fuel consumption rate ranges from 0.5 gallons/hr to 2.2 gallons/hr. The idle fuel consumption rate depends on engine size. The fuel used while idling ranges from 0.65 to 9.6 gallons per month. The average for the entire fleet of 25 vehicles is 4.2 gallons/month. The majority of idling events lasted less than 20 minutes, and the average duration of an idling event was 15 minutes. However, the total fuel consumption in idling during the survey period added up to 1584 gallons (1720 hours) and would cost more than six thousand dollars. Given the large size of the Caltrans vehicle fleet, the fuel and cost savings realized by idling reduction could be quite significant. However, without knowing the true function of vehicles and their mode of operation, we may have misidentified idle conditions. Some idling events may be necessary due to repair, maintenance, or vehicle mode of operation. Vehicle operators and managers were not interviewed to determine if some of the idling events are necessary.

The Preco system did not work as well as hoped. It frequently failed to log the state of the ignition. This problem led to uncertainty in determining the idling duration. Idling durations between 5 and 90 minutes are considered in our analysis. Any idling events lasting longer than 90 minutes are considered outlier errors caused by the idling identification algorithm. Any idling events lasting less than 5 minutes are assumed to be caused by waiting in traffic or at a traffic signal.

Alternative Fuel Fleet Monitoring System Development

Working with Caltrans DRISI, researchers at University of California, Riverside (UCR) Center for Environmental Research and Technology (CERT) installed various COTS telematics devices on one hundred Caltrans flex-fuel vehicles in four different Caltrans Districts to monitor the E85 fuel usage. Mike Todd is the Principal Development Engineer at CERT. Twenty Zonar Systems and fourteen Fastracs in-vehicle telematics devices with cellular network communication were used. The Zonar systems provided web services for raw data access by CERT researchers. In addition, three HEM Data OBD-II loggers and eighteen Fastracs with WiFi network communication were also used. Four WiFi base stations were installed in Districts 4, 7, 8, and 11 maintenance yard parking areas to collect data from WiFi logger to the CERT web server. The telematics system interfaces with the vehicles' OBD-II data port, reading several engine operating parameters (e.g. RPM, manifold air pressure, etc.). The system also has an embedded differential GPS receiver and custom software to collect data from the ethanol content sensor on the flex-fuel vehicles.

The engine operating parameters and vehicle location data are packaged and sent wirelessly to a MySQL database server, which then stores, processes, and manipulates the received data. The database structure is designed based on relational database principles in order to easily extract useful information such as current fuel level in the tank, location where the vehicle is refueled, and fuel content (i.e. % ethanol), among others. In addition, the database server is set up to serve internet clients, which organize and display real-time information on web interfaces, allowing fleet managers to more easily monitor their fleet operations. The current website design focused on providing E85 usage data and summary. Additional work will be need for general fleet management applications. This project has shown the feasibility of using WiFi for data transfer link between the telematics devices and the web server. The existing server software may be further developed for fleet management applications.

Networkfleet Software Development Kit

AHMCT researchers evaluated the Networkfleet Software Development Kit (SDK). This SDK provides Java-based web services, with data returned in XML (eXtensible Markup Language) format. The SDK provides access to a subset of the data available on the web interface. Access to vehicle location data is well supported. We used this web service to download vehicle location data, and imported into the OpenGTS tracking system discussed below. There is currently no SDK access to sensor data; however Networkfleet noted that they are working on sensor data web services in the next revision of their SDK.

There is a web service for vehicle data including fuel usage and diagnostic trouble codes. However, this web service did not function during the evaluation. When called, it always returns an exception. Networkfleet did not have the resources to help with this error message. Networkfleet was helpful on several other support requests.

OpenGTS

OpenGTS (Open GPS Tracking System), www.opengts.org, is open-source software for fleet tracking, provided under the Apache Software License version 2. We evaluated OpenGTS for its potential use as a basis for custom software development to provide vendor-independent fleet tracking and management for Caltrans. Based on our evaluation, OpenGTS could serve as a starting point for such development.

The OpenGTS Java source code is modular, well structured, and well-documented. However there does not appear to be a large community of developers. Based on source code comments for the code that was reviewed, all development seems to have been done by the original developer. The program leverages top open-source software packages, such as MySQL and Tomcat. The program offers support for a number of tracking devices, as well as a template to integrate other devices.

OpenGTS provides fleet mapping using Open Street Maps (see Figure 3.1). It appears to offer support for Google Maps as well. The program supports a large number of standard reports, including vehicle location details (Figure 3.2) and overspeed reporting (Figure 3.3). Reports are customizable through XML.

There is one significant structural drawback of OpenGTS. Individual vehicles are assigned to users, and each vehicle's information is accessible only to that user. This would be a significant issue for Caltrans use. It is not clear how ingrained this approach is in OpenGTS, nor how or whether it could be easily modified. One possible workaround would be to have a single "Caltrans" user, shared among the appropriate organization personnel; this is not an ideal solution in terms of security.

There is a commercial version, GTS Enterprise, as well, which is much more capable, with many more reports. Based on a webpage summary of features, this version appears to have capabilities similar to the Networkfleet website. Example reports include OBD alert codes and diagnostic details, fuel usage, idle time, idle fuel used, digital input triggers, etc. A feature chart comparing the capabilities of OpenGTS and the commercial version is available at www.geotelematic.com/gts.html.

9
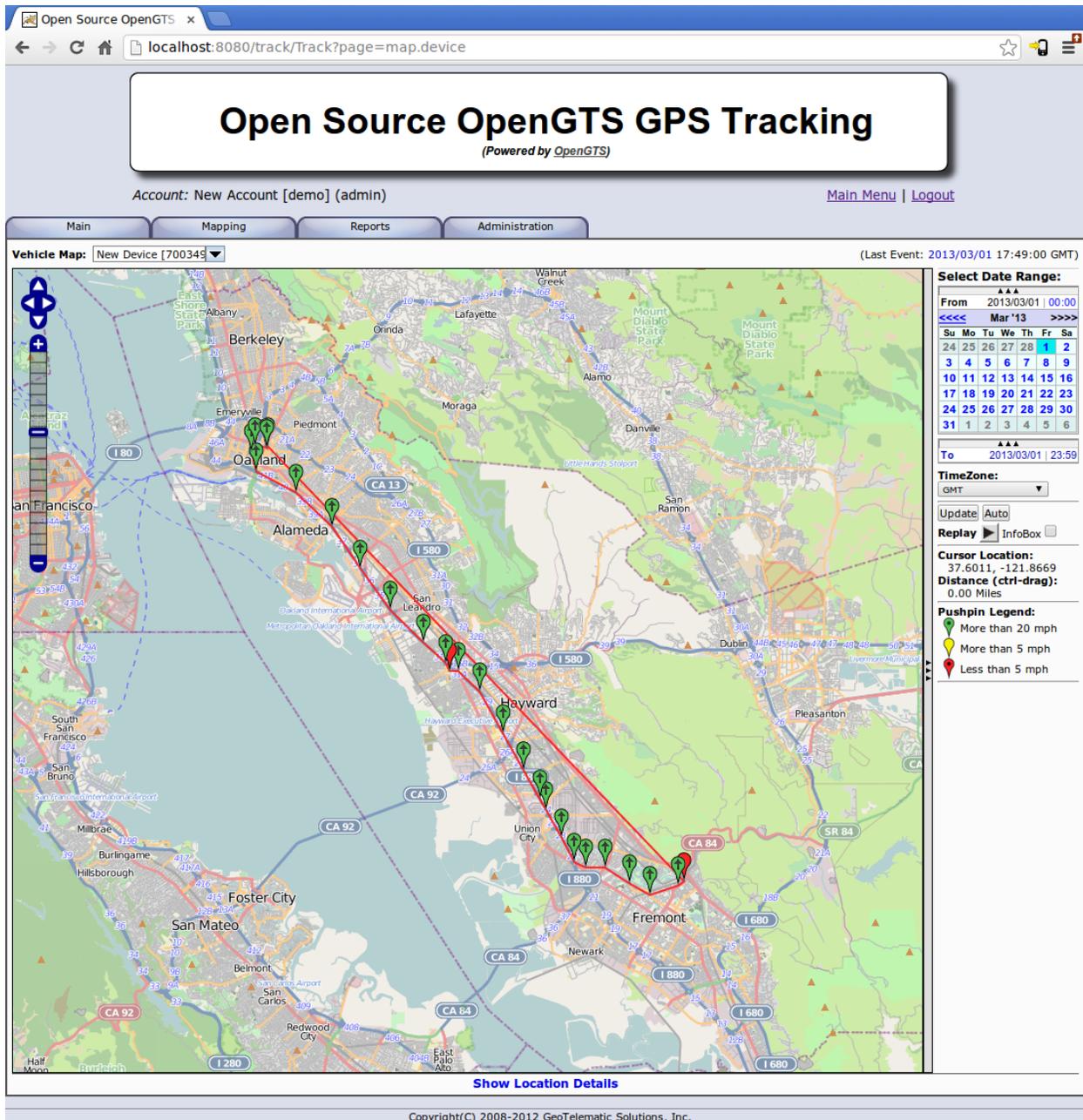
Figure 3.1 – OpenGTS mapping based on Open Street Maps

10
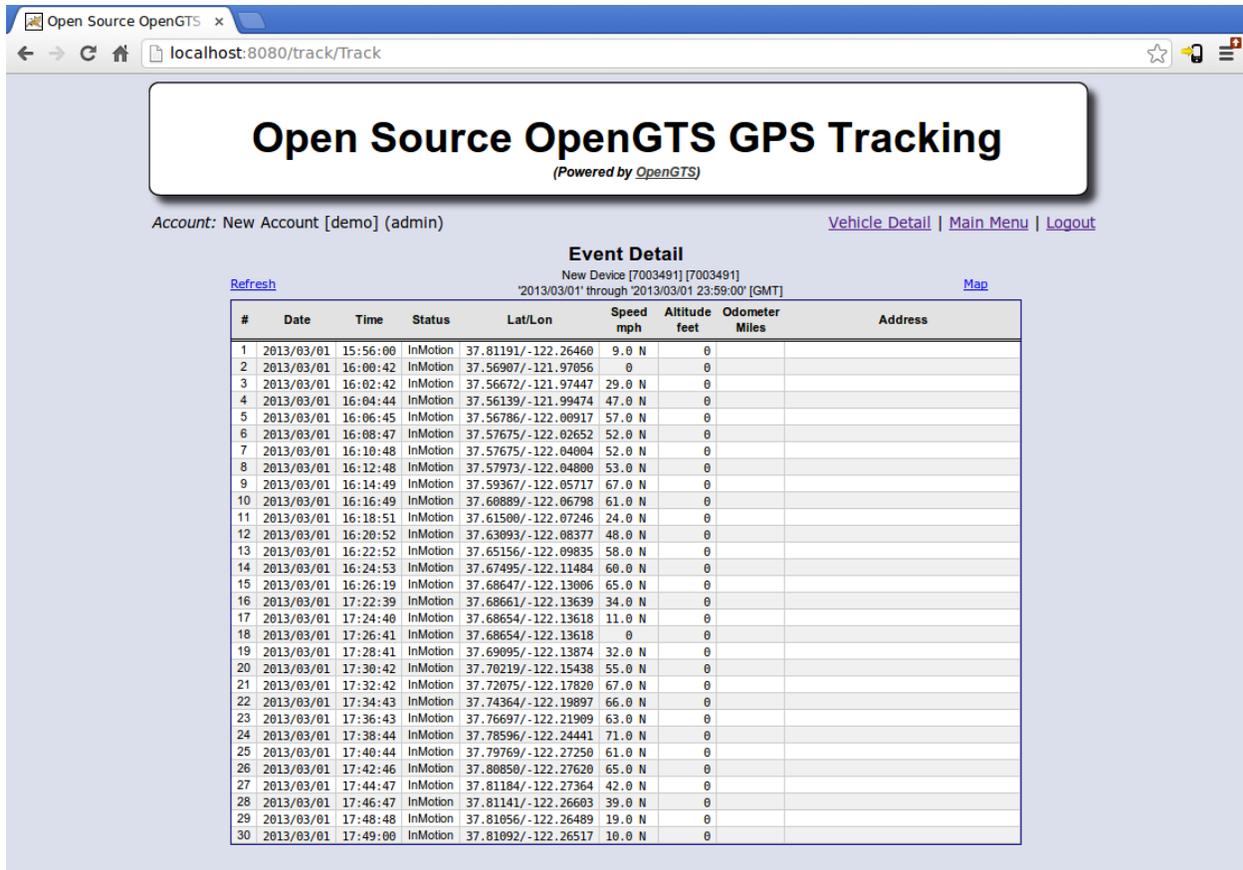
Open Source OpenGTS GPS Tracking

(Powered by OpenGTS)

Account: New Account [demo] (admin)

Vehicle Detail | Main Menu | Logout

Event Detail

New Device [7003491] [7003491]
'2013/03/01' through '2013/03/01 23:59:00' [GMT]

Refresh

Map

#	Date	Time	Status	Lat/Lon	Speed mph	Altitude feet	Odometer Miles	Address
1	2013/03/01	15:56:00	InMotion	37.81191/-122.26460	9.0 N	0		
2	2013/03/01	16:00:42	InMotion	37.56907/-121.97056	0	0		
3	2013/03/01	16:02:42	InMotion	37.56672/-121.97447	29.0 N	0		
4	2013/03/01	16:04:44	InMotion	37.56139/-121.99474	47.0 N	0		
5	2013/03/01	16:06:45	InMotion	37.56786/-122.00917	57.0 N	0		
6	2013/03/01	16:08:47	InMotion	37.57675/-122.02652	52.0 N	0		
7	2013/03/01	16:10:48	InMotion	37.57675/-122.04004	52.0 N	0		
8	2013/03/01	16:12:48	InMotion	37.57973/-122.04800	53.0 N	0		
9	2013/03/01	16:14:49	InMotion	37.59367/-122.05717	67.0 N	0		
10	2013/03/01	16:16:49	InMotion	37.60889/-122.06798	61.0 N	0		
11	2013/03/01	16:18:51	InMotion	37.61500/-122.07246	24.0 N	0		
12	2013/03/01	16:20:52	InMotion	37.63093/-122.08377	48.0 N	0		
13	2013/03/01	16:22:52	InMotion	37.65156/-122.09835	58.0 N	0		
14	2013/03/01	16:24:53	InMotion	37.67495/-122.11484	60.0 N	0		
15	2013/03/01	16:26:19	InMotion	37.68647/-122.13006	65.0 N	0		
16	2013/03/01	17:22:39	InMotion	37.68661/-122.13639	34.0 N	0		
17	2013/03/01	17:24:40	InMotion	37.68654/-122.13618	11.0 N	0		
18	2013/03/01	17:26:41	InMotion	37.68654/-122.13618	0	0		
19	2013/03/01	17:28:41	InMotion	37.69095/-122.13874	32.0 N	0		
20	2013/03/01	17:30:42	InMotion	37.70219/-122.15438	55.0 N	0		
21	2013/03/01	17:32:42	InMotion	37.72075/-122.17820	67.0 N	0		
22	2013/03/01	17:34:43	InMotion	37.74364/-122.19897	66.0 N	0		
23	2013/03/01	17:36:43	InMotion	37.76697/-122.21909	63.0 N	0		
24	2013/03/01	17:38:44	InMotion	37.78596/-122.24441	71.0 N	0		
25	2013/03/01	17:40:44	InMotion	37.79769/-122.27250	61.0 N	0		
26	2013/03/01	17:42:46	InMotion	37.80850/-122.27620	65.0 N	0		
27	2013/03/01	17:44:47	InMotion	37.81184/-122.27364	42.0 N	0		
28	2013/03/01	17:46:47	InMotion	37.81141/-122.26603	39.0 N	0		
29	2013/03/01	17:48:48	InMotion	37.81056/-122.26489	19.0 N	0		
30	2013/03/01	17:49:00	InMotion	37.81092/-122.26517	10.0 N	0		

Figure 3.2 – Vehicle location report in OpenGTS

11

Figure 3.3 – Group overspeed report in OpenGTS

CHAPTER 4:
FIDAS PILOT STUDY AND RESULTS

Introduction

The Caltrans Division of Equipment (DOE) established a contract with a Networkfleet reseller to install Networkfleet in-vehicle telematics devices on two hundred vehicles and provide two years of data and web services. All the telematics devices were installed approximately November 1 2012. The data from December 2012 to May 2014 was analyzed for this report. The pilot fleet has two heavy-duty trucks and four trailers that are not included in the data analysis. In addition, the telematics devices were not able to collect fuel usage data from twenty-nine vehicles in the pilot fleet, and their data are also not included in the analysis. These vehicles are mostly made by Dodge. However, some Dodge vehicles did provide fuel usage data in the pilot fleet. The reasons for not being able to collect fuel usage data on the twenty-nine vehicles are unknown. In addition, Networkfleet reports large erroneous monthly fuel consumption (~ 104,300 or 174,050,517 gallon of fuel) for some vehicles, with unknown reason, in some months of data. These data points are dropped in our analysis. Nevertheless, the 165 vehicle data in the pilot study do provide insight into possible areas of saving and data for cost-benefit study for wider deployment justifications.

Caltrans Fleet Statistics and Pilot Study Fleet Statistics in year 2012

Figure 4.1 shows the vehicle make year distribution of the Caltrans light vehicle fleet. 95.6 % of the Caltrans light fleet vehicles are made after 1996 and have OBD-II interface, and 36% of the vehicles are made after 2005 and have CAN (Controller Area Network) bus interface for faster data transfer. Figure 4.2 shows the vehicle make year distribution of the pilot study vehicle fleet. The pilot study fleet age composition resembles the overall Caltrans fleet distribution even though the pilot study fleet vehicle is not randomly chosen. Figure 4.3 and 4.4 show the manufacturer composition of the entire light vehicle fleet and the pilot study fleet. The 1% Toyota automobiles are Prius hybrids.
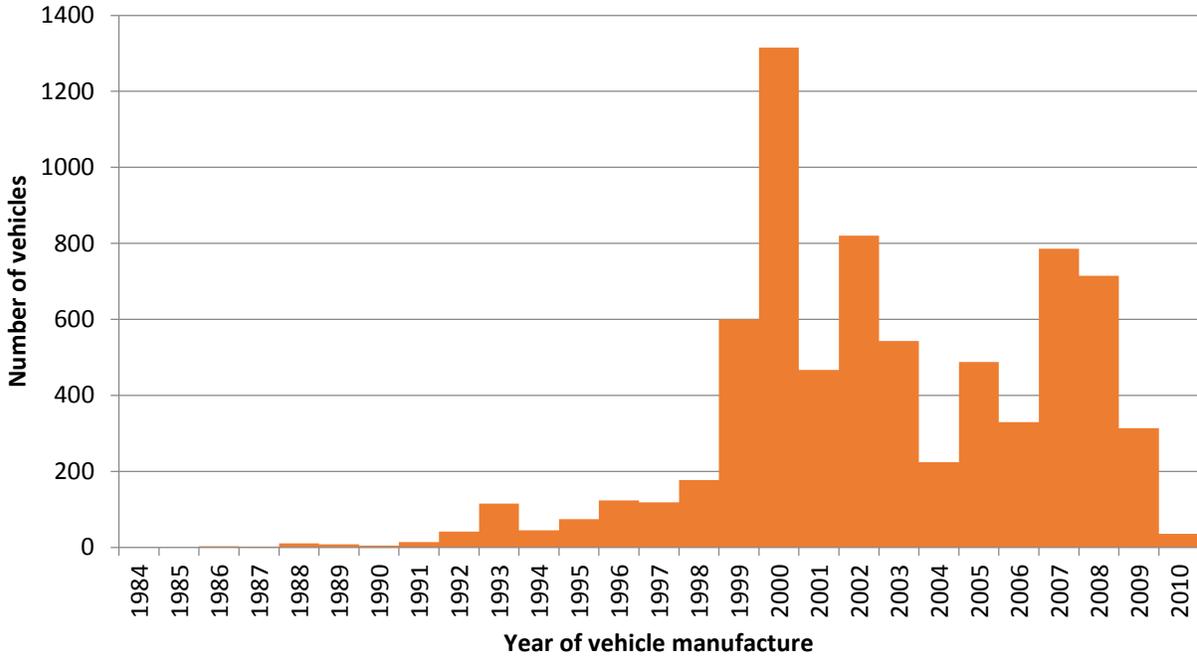
Figure 4.1 – Distribution of Caltrans light vehicle fleet by year of vehicle manufacture in year 2012
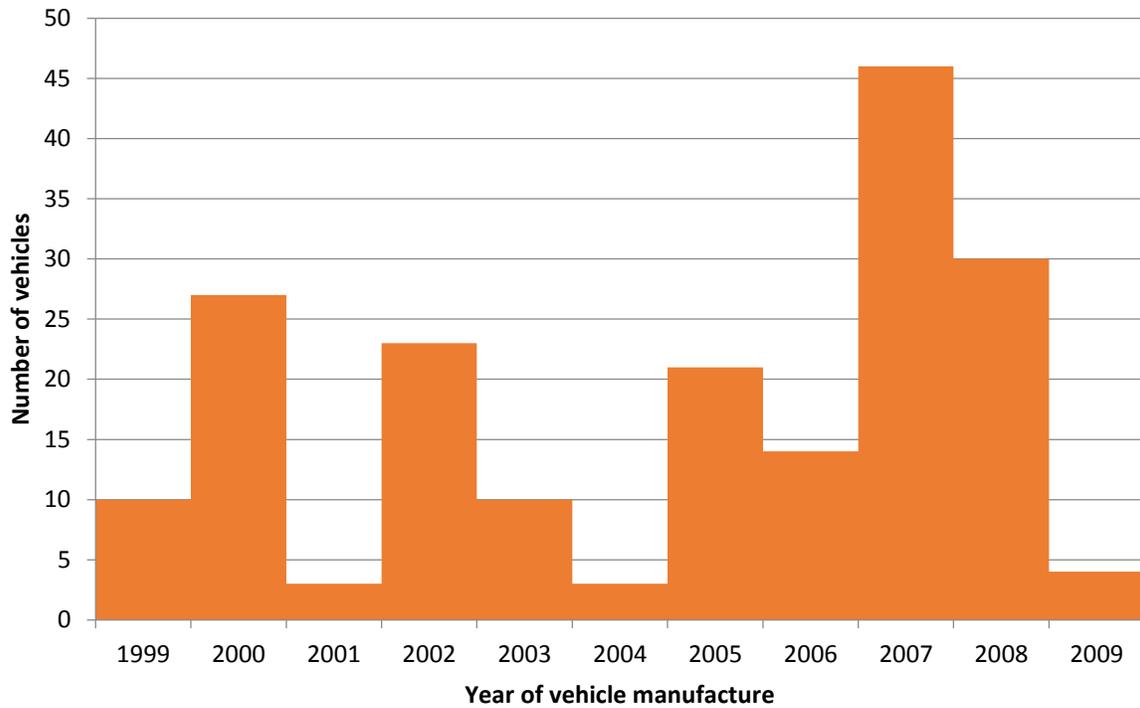
Figure 4.2 – Distribution of pilot study fleet by year of vehicle manufacture in year 2012

14
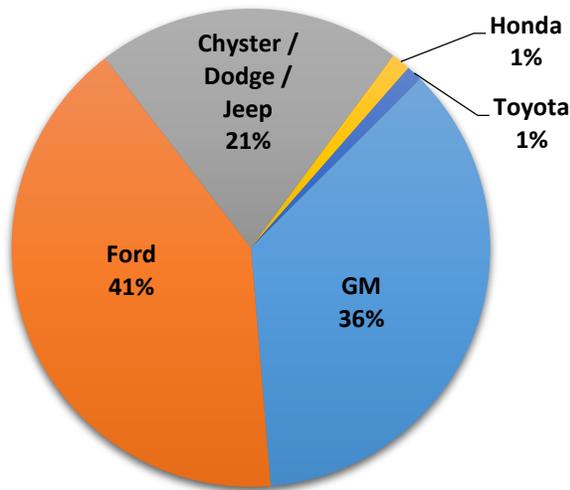
Figure 4.3 – Caltrans light vehicle fleet manufacturer make-up (2012)
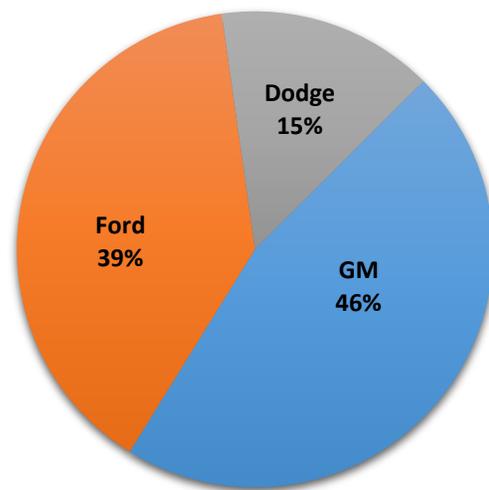
Figure 4.4 – Initial Pilot Study vehicle fleet manufacturer make-up (2012)

Pilot Fleet Utilization

Figure 4.5, 4.6, 4.7 and 4.8 summarize the monthly pilot fleet utilization (pilot fleet total monthly distance traveled, total fuel usage, average monthly pilot fleet fuel mileage (MPG) and average fuel used per active vehicle). Due to clerical error, pilot fleet fuel consumption data was not collected from Networkfleet for the month of June 2013 and July 2013. The blue dotted line in Figure 4.6, 4.7, and 4.8 represent the missing data. There were about 12 vehicles taken out of the pilot fleet and 12 vehicles added to the pilot fleet during the months of January and February 2014. The total fuel consumption does not increase proportional to the total distance traveled because more fuel efficient vehicles are used. The average vehicle travel distance is about 900 miles/month (10,800 miles/year), and the average vehicle fuel usage is about 55 gallon/month (670 gallon/year). Since some of the large erroneous monthly fuel consumption data were rejected in the calculations, the real pilot fleet fuel consumption is likely higher than reported. Not all vehicles in the pilot fleet are used in each month. The average fuel consumed per active vehicles are plotted in Figure 4.8. Figure 4.7 shows increased fuel mileage in the summer months. The average pilot fleet vehicle fuel mileage is about 13.3 MPG. Figure 4.9 shows the pilot fleet vehicles' monthly fuel mileage (MPG) distribution using the first 7 months of pilot fleet data. Figure 4.9 showed that the pilot fleet is composed of two groups of vehicles: large trucks and vans with 10 to 15 MPG, and light passenger vehicles with 25 to 30 MPG.
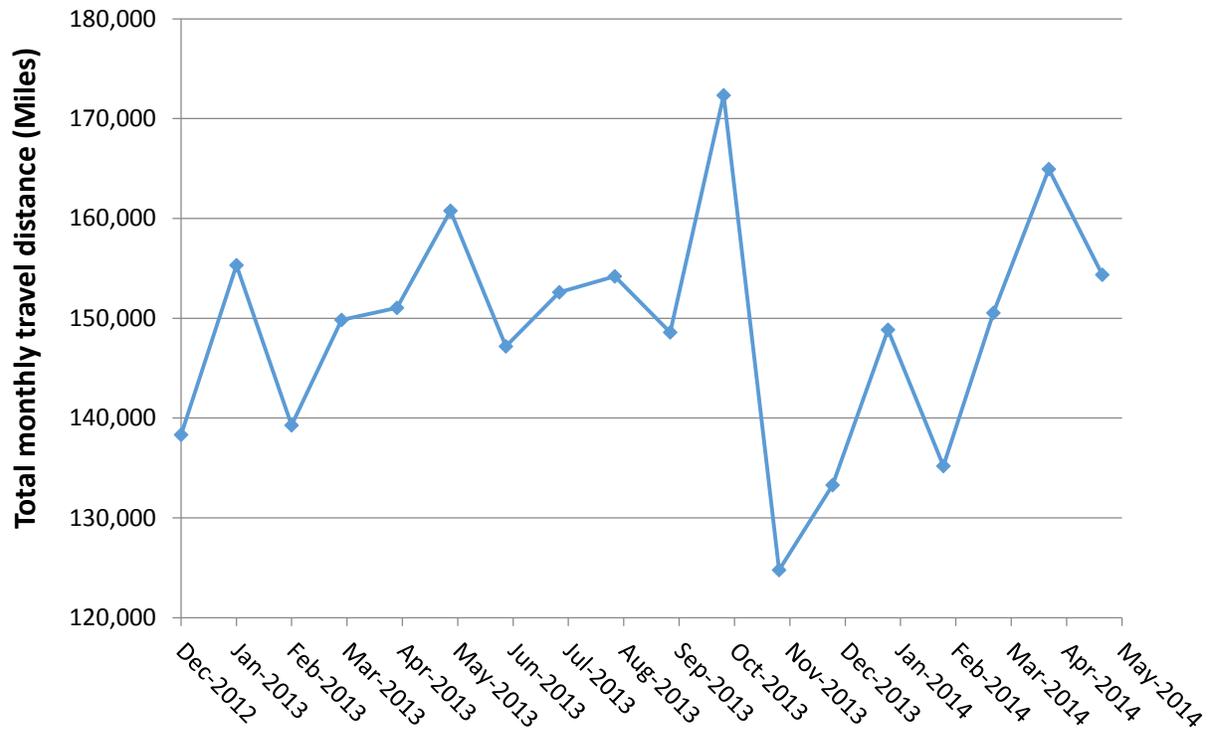
Figure 4.5 – Pilot fleet monthly total travel distance per month

16
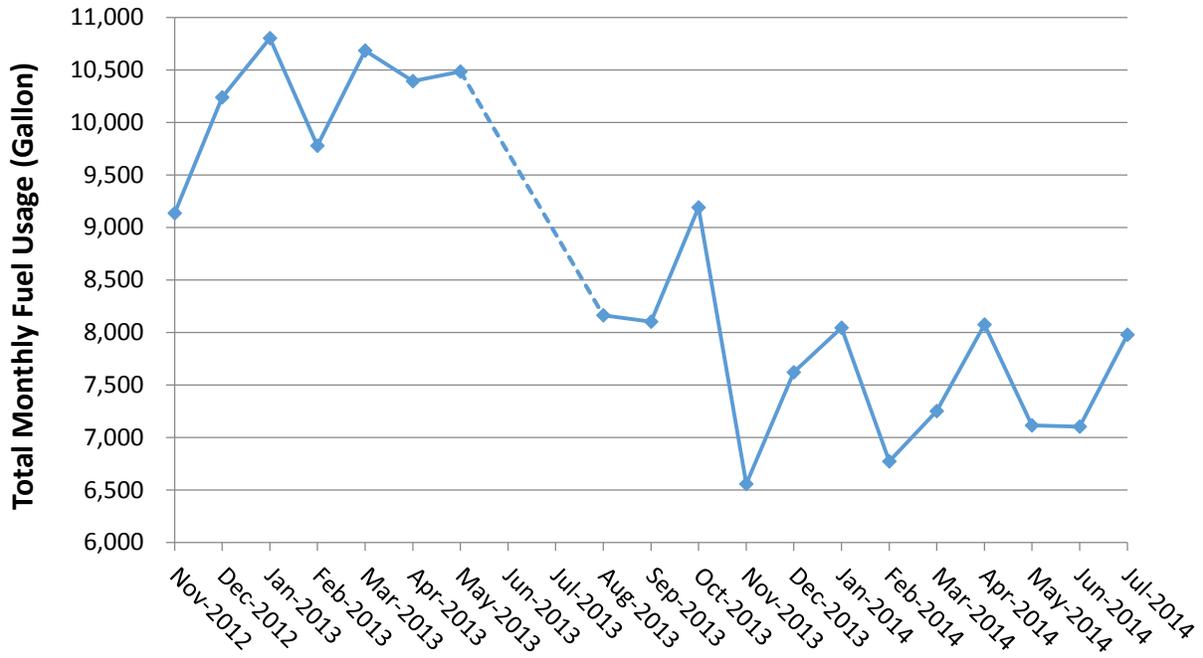
Figure 4.6 – Pilot fleet monthly fuel usage per month
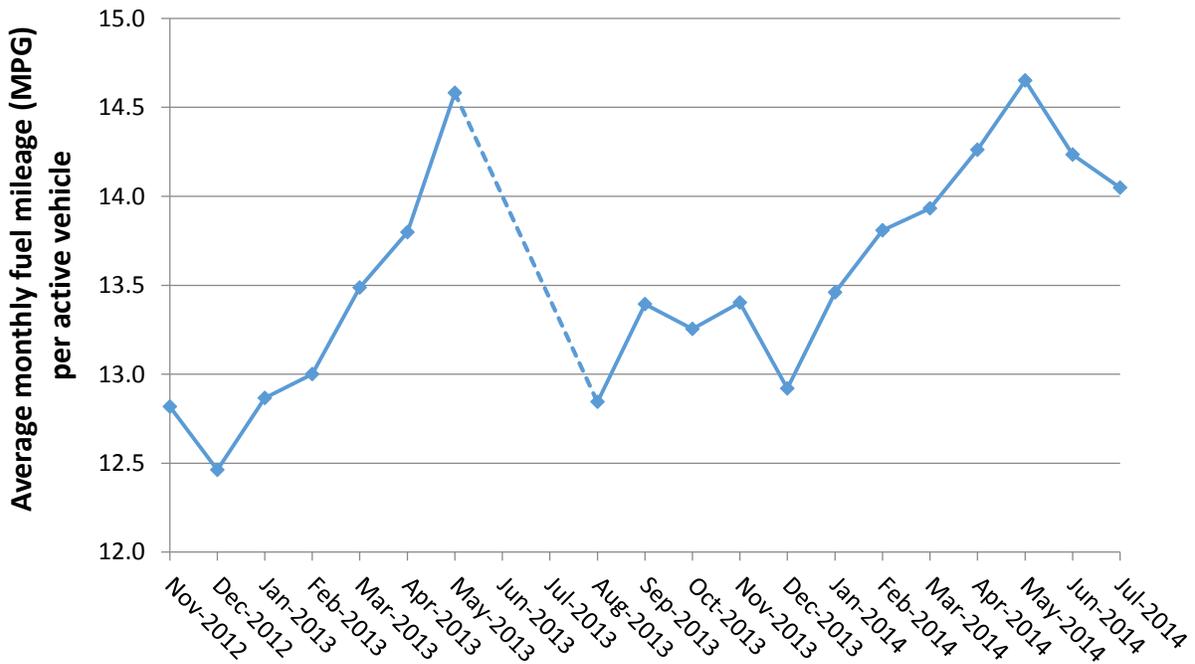
Figure 4.7 – Pilot study fleet average MPG per month

17
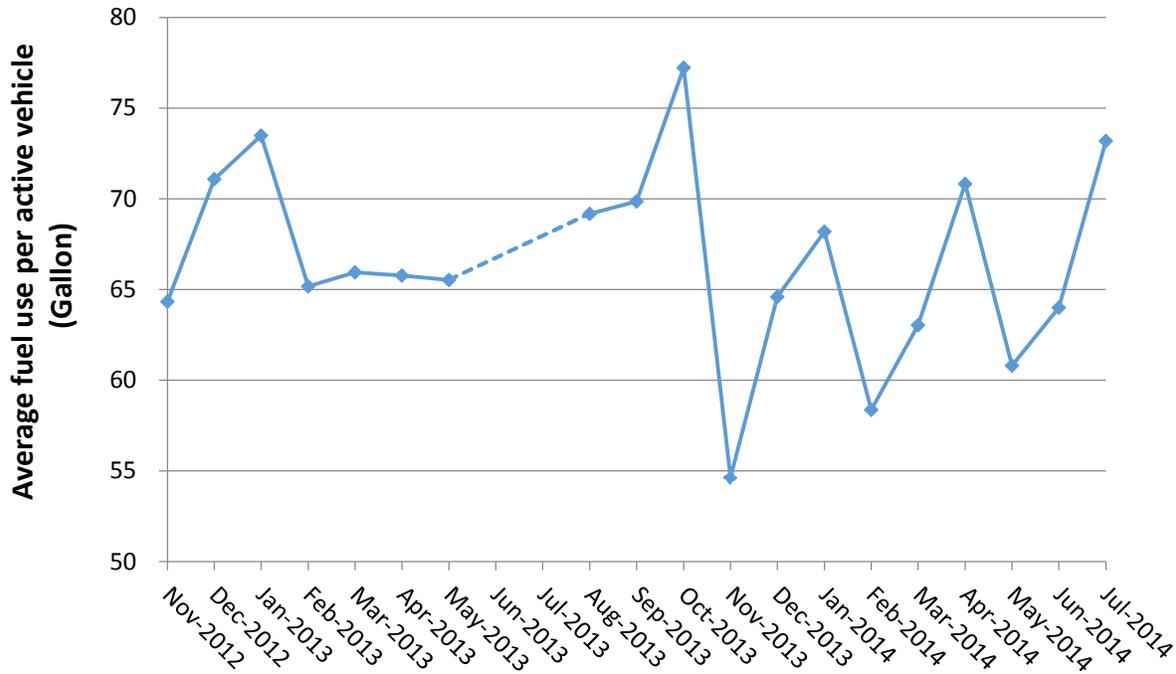
Figure 4.8 – Average fuel usage per month per active vehicle in the pilot study fleet vehicle
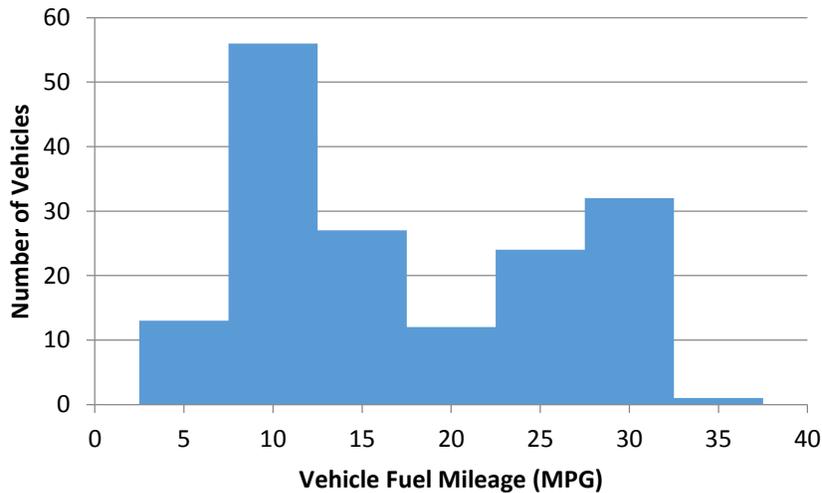
Figure 4.9 – Pilot study fleet vehicle 7 month average MPG distribution

Odometer Accuracy Verification

The Networkfleet telematics device calculates the vehicle distance traveled by integrating the vehicle speed data obtained through the OBD-II. The vehicle speed update frequency is limited by the OBD-II, which could affect the distance traveled accuracy. The vehicle odometer reading was entered into the Networkfleet web server database at the time of the telematics device installation. By adding the distance traveled to the initial odometer reading, the system can keep

18

track of the odometer reading. Thirty-nine pilot fleet vehicles' odometer reading from the maintenance repair shop were compare to the reading from the Networkfleet data. By subtracting the initial odometer reading (of 11/01/2012) from the odometer reading from both data sources, the distance traveled percentage error can be calculated. Since the maintenance shop odometer reading was only marked with date but without the exact time, this current methods does have some error in determining the corresponding Networkfleet odometer data point. This error will be reduced by allowing for a longer duration or travel distance from the initial installation.

The average distance traveled error of the 39 vehicles is 0.82 %. The vehicle odometer is generally higher than the Networkfleet estimation. This result is consistent with the error claimed by the solution providers. For example, Zonar tests show the travel distance accuracy for a Chevrolet Impala is 1.19%. Figure 4.10 shows the distribution of the errors. Most errors are between -0.5% and 2%. However, since the average annual pilot fleet vehicle travel distance is about 10,000 miles/year, 1% error will result in 100 miles/year error.
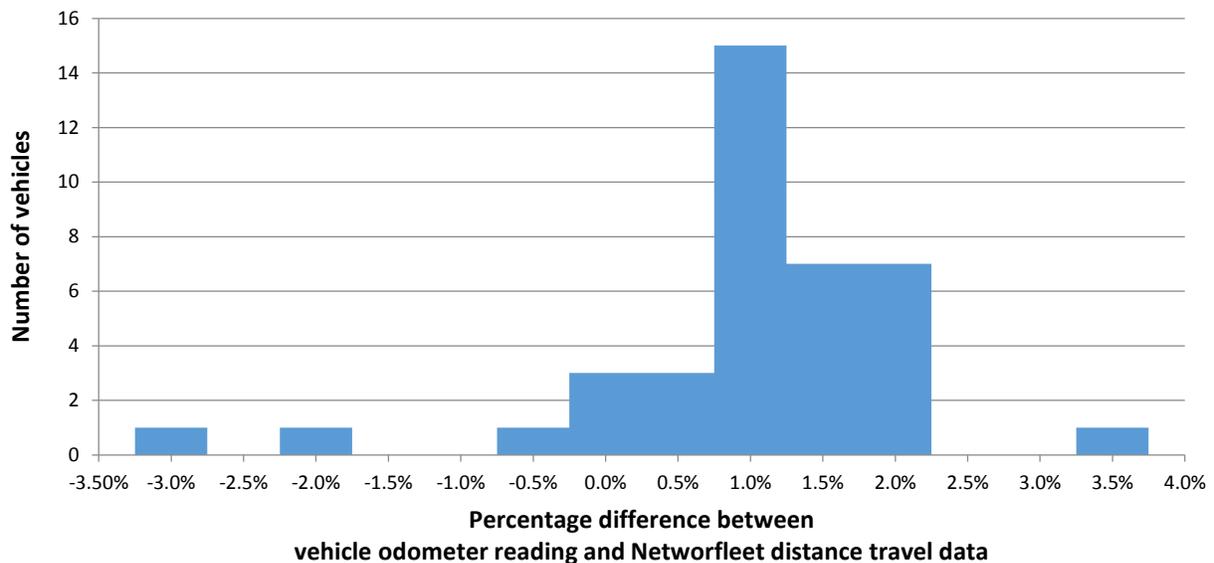
Figure 4.10 – Pilot study fleet odometer error distribution

Fuel Usage Accuracy Verification

Fuel used vs. fuel purchased data was analyzed. Fuel used was obtained from monthly Networkfleet reports. Fuel purchased was obtained from Voyager purchase records and bulk fuel records. The time period for the analysis, January – March 2013, was based on availability of data. There was bad data in the set which had to be culled. For example, some of the Networkfleet data indicates ~104,000 gallons used. In other cases, Networkfleet data indicates zero fuel used for some vehicles known to have mileage in the period. In some cases, the bulk fuel purchased was much higher than that of the Networkfleet data. The likely cause is that the Voyager account was used to fuel up other vehicles as well. Finally, extreme outliers were also removed in the following plots. As a result, only 24 vehicles' data were available for comparison.

19

The majority of the error is between -10 % and 20 %. The fuel bought is generally higher than the Networkfleet estimates. This comparison method is not very accurate because there is always fuel left in the vehicle fuel tank at the end of the month. Thus, the fuel purchase would normally be higher than the fuel consumption reported by Networkfleet. Since there is not a simple way to obtain daily (or hourly) fuel consumption data from the Networkfleet server, this is the only available means assessing the fuel consumption data accuracy. However, the error of this method will be reduced by allowing for a longer duration or fuel consumed from the initial installation. Better data and methods are needed to correctly assess the accuracy of the fuel consumption estimate. The fuel purchased and fuel used data should be compared again at the end of one and two year periods. Zonar Systems reported that their experiments yield a fuel consumption error of about 2.77%.
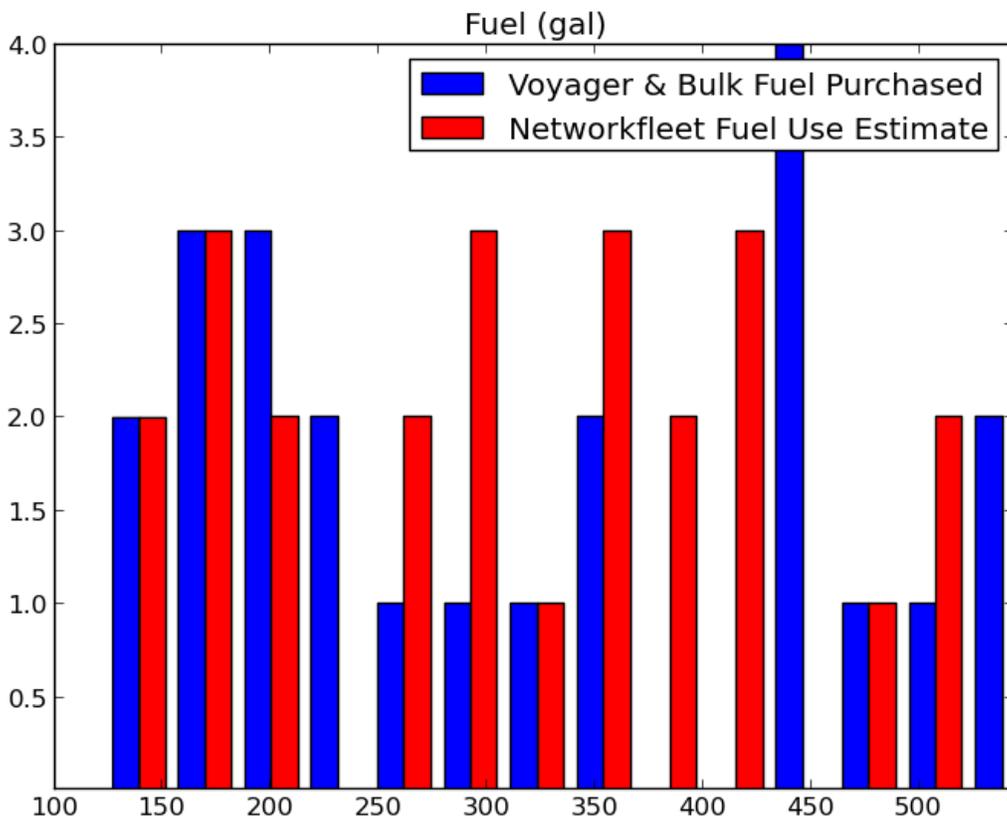
Figure 4.11 – Distribution of fuel bought and used (gallons) for January – March 2013 (24 vehicles)

20
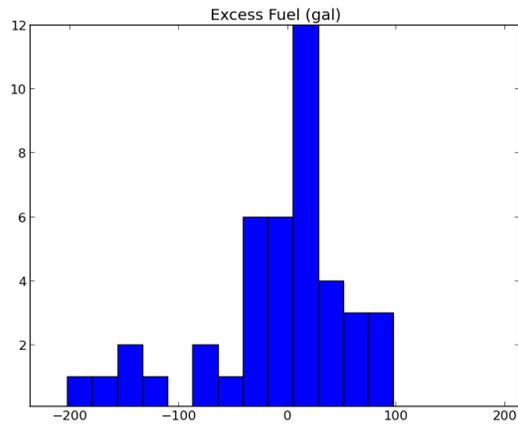
Figure 4.12 – Distribution of [fuel bought – fuel used] (gallons) for January – March 2013 (24 vehicles)
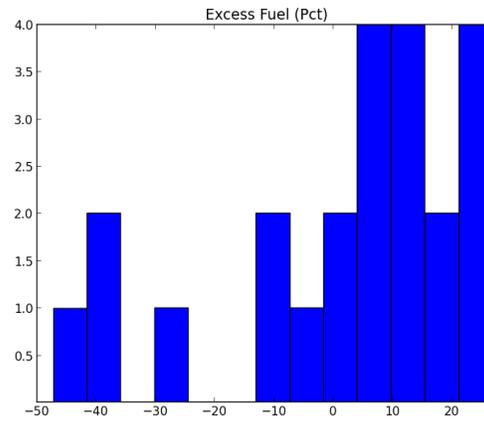
Figure 4.13 – Distribution of [fuel bought – fuel used] (percentage) for January – March 2013 (24 vehicles)

Pilot Study Fleet Idling

Pilot study fleet monthly total idling time data was collected from the Networkfleet reports. There seems to be some erroneous data in trip duration and idling data in the Networkfleet reports. For example, some data points have idling event lasting from 12 hours to 88 hours and single trip events lasting from 12 hours to 140 hours. The data points containing idling or trip duration longer than 12 hours are rejected. If some idling events do indeed last longer than 12 hours, the idling duration percentage and fuel consumption would be higher. The eighteen months (December 2012 to May 2014) total idling duration for the pilot fleet was 20,287 hours (21% of the total vehicle usage duration or engine on time), and the monthly average idling duration per vehicle is 6.8 hours per vehicle per month. The idling duration distribution for the first 7 months is shown in Figure 4.14. The majority of the idling duration is between 5 to 60 minutes. Figure 4.15 indicated that there is an increase in idling time during the winter months. Figure 4.15 shows the distribution of vehicle idling percentages. The majority of vehicle idle time is less than 20%. However, a significant number of vehicles have idle time over 30%. Two vehicles idle over 80% of the time.

21
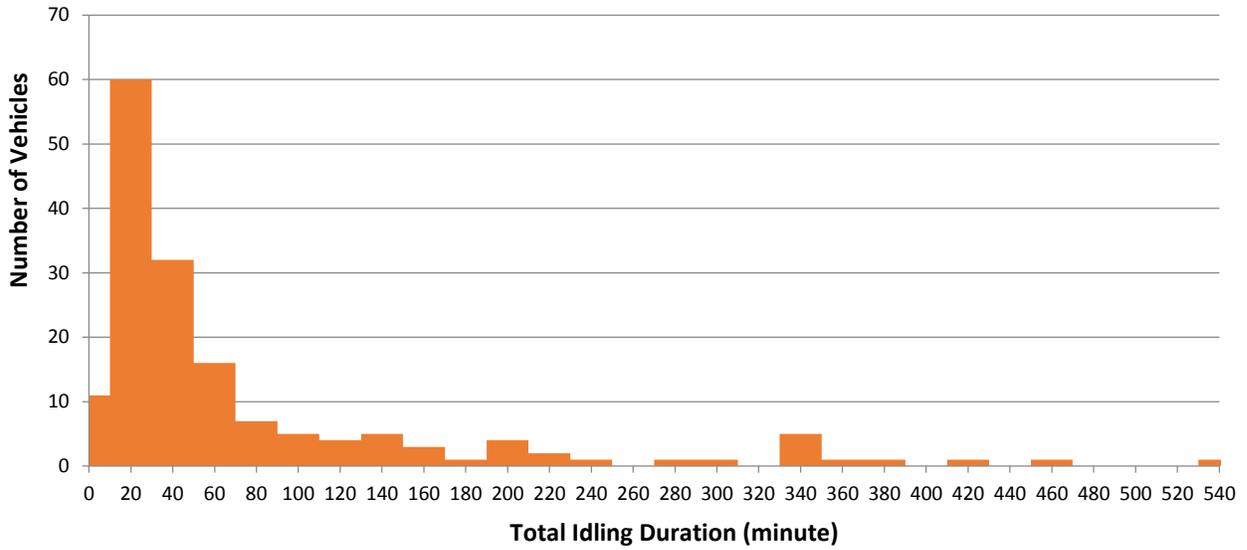
Figure 4.14 – Pilot study idling duration distribution for the first 7 months
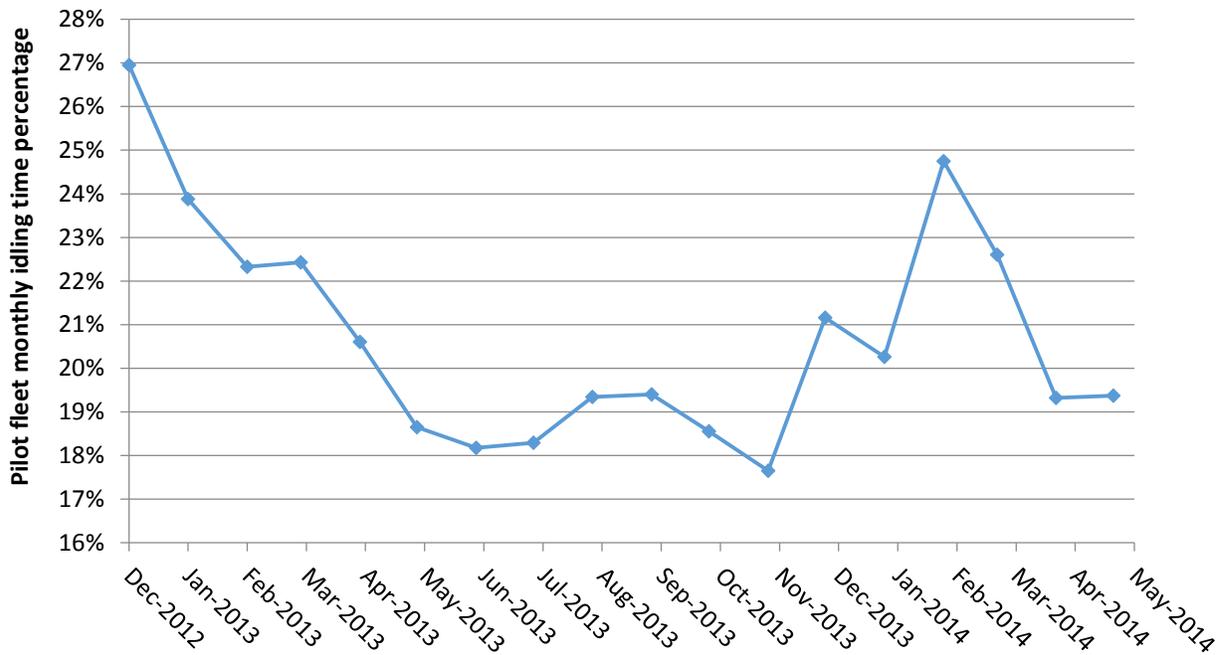
Figure 4.15 – Pilot study idling time percentage (idle time/trip time) per month

22
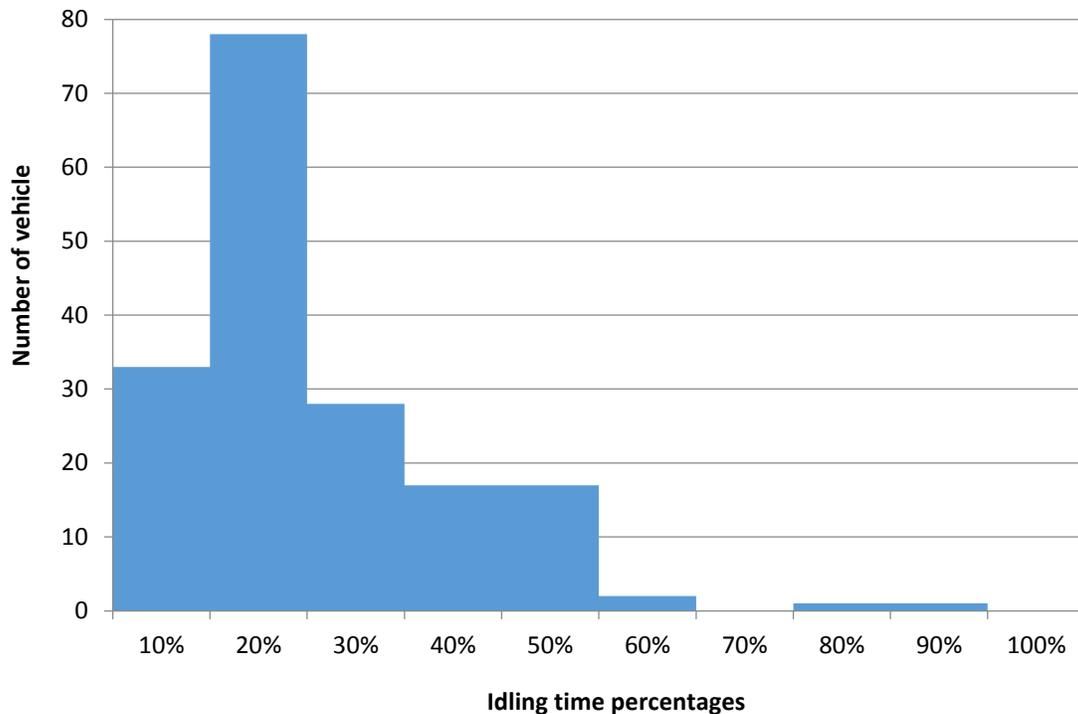
Figure 4.16 – Idling time percentage distribution in the pilot fleet

Unfortunately, each vehicle's idling fuel consumption rate is not available from Networkfleet reports; each vehicle's idling fuel consumption rate was calculated based on the its engine size and idle fuel rate estimate from both Argonne National Laboratory [1] and the Office of Energy Efficiency, Natural Resources Canada [3] as

$$Idle\_fuel\_rate \text{ (Gallon/hour)} = 0.1584*Engine\_size \text{ (Liter)}.$$

Figure 4.18 shows the idling fuel rate estimate from both government research publications. The vehicle's engine size was determined using the VIN number and web database. Each pilot fleet vehicle's idling fuel consumption is then calculated using the idling fuel rate estimate and idling duration. Figure 4.17 illustrates the distributions of the vehicle engine size in the pilot fleet. The statistics for vehicle engine size of the entire Caltrans light vehicle fleet should be collected and compared to the pilot fleet engine size distribution. This would provide a valuable basis to validate idling fuel saving predictions for the entire vehicle fleet. The pilot fleet average idling fuel rate is 0.96 gallon per hour ($0.064 per minute assuming a $4.00 per gallon gasoline cost). The pilot fleet idling cost estimate per minute is slightly higher but consistent with fueleconomy.gov estimate of ($0.02 to $0.04 per minute). The higher idling cost per minute is mostly likely due to higher gasoline price in California ($4.00 per gallon) over the national average ($3.61 per gallon) and the larger than average engine size of the pilot fleet. Over half of the pilot fleet vehicles have a 5 liter or bigger engine as shown in Figure 4.17. The eighteen months total idling fuel consumption estimate for the pilot fleet was 19,812 gallons (13% of the total vehicle fuel usage), and the monthly average idling fuel usage per vehicle is 6.7 gallons. Idling fuel consumption is higher in the winter months, as shown in Figure 4.20.

23
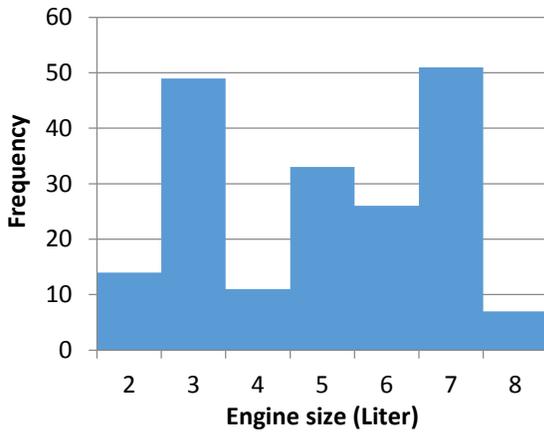
Figure 4.17 – Pilot study fleet vehicle engine size distribution
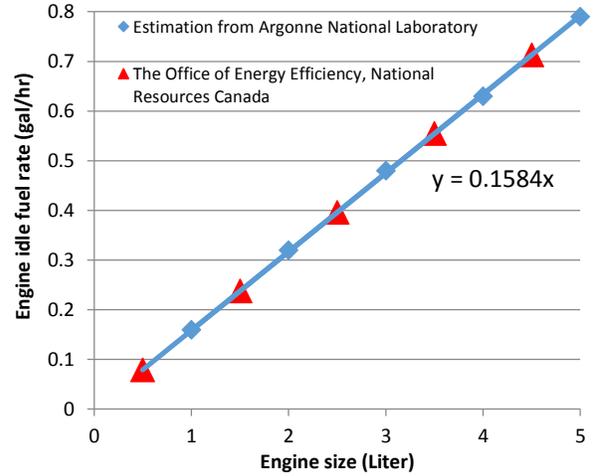
Figure 4.18 – Idle fuel rate vs. engine size
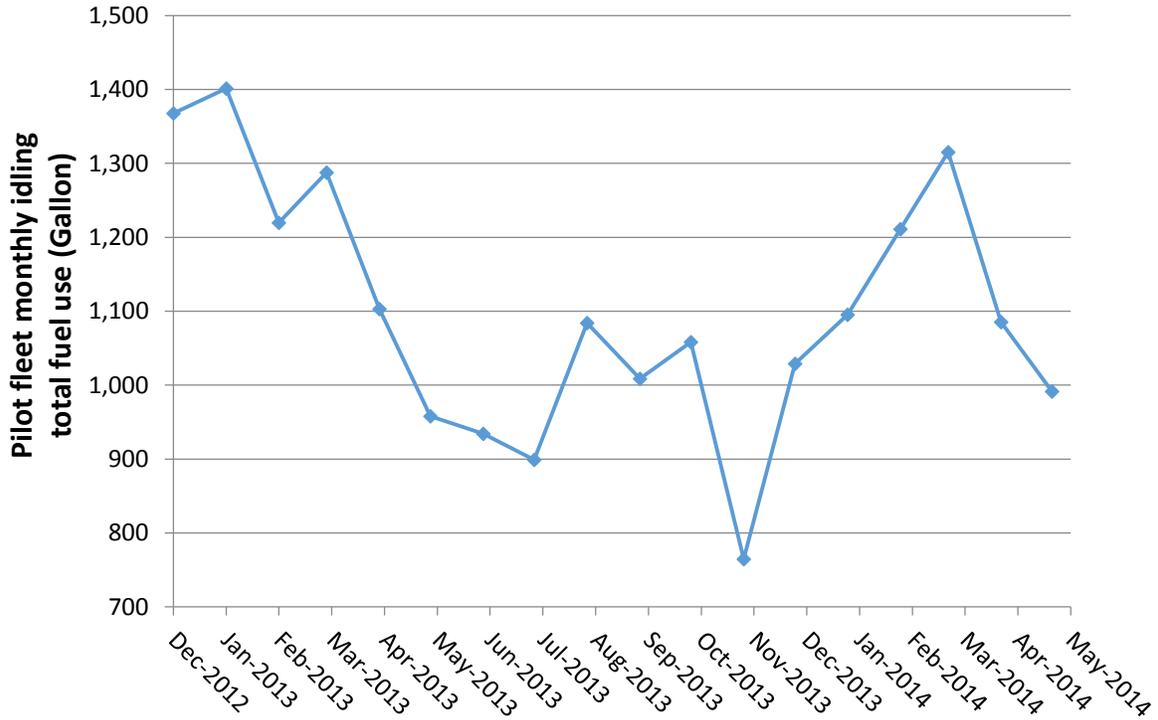
Figure 4.19 – Pilot study fleet idling fuel use per month

24
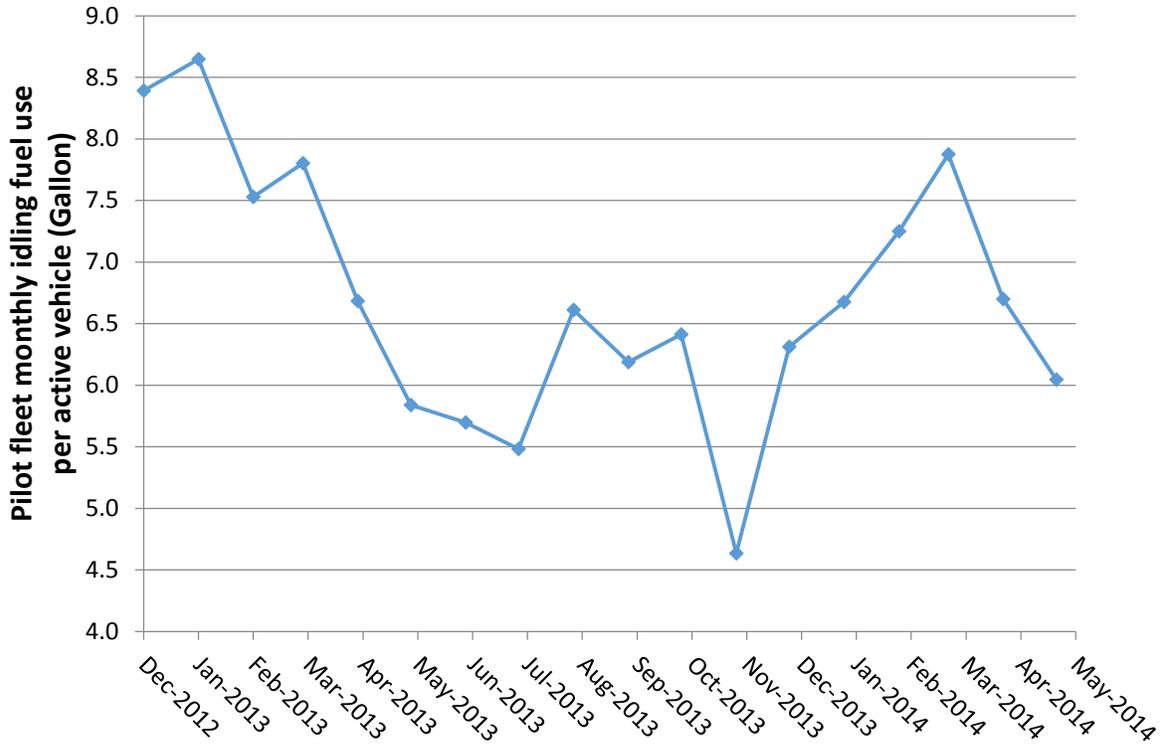
Figure 4.20 – Travel time per month
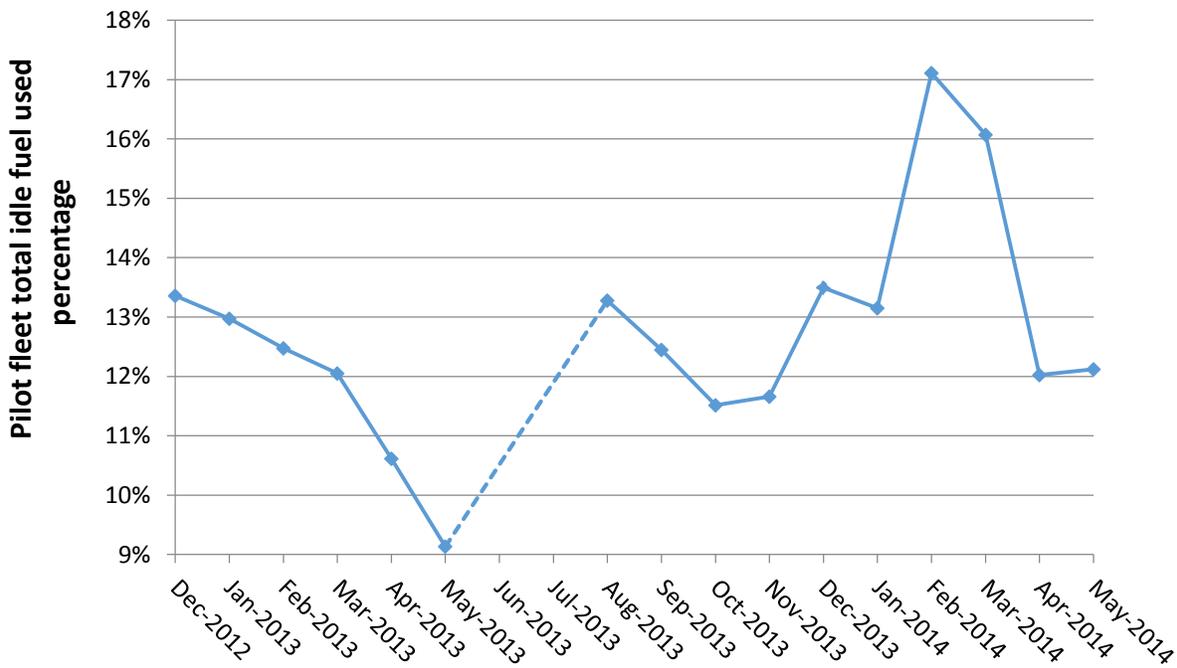
Figure 4.21 – Percentage of idle fuel usage over total fuel usage per month

25

The idling location can provide some insight to the idling driver behavior, and lead to appropriate policy to reduce idling. Idling heat maps based on the idling locations and duration for every month were created. Example idling heat maps for December 2013 are shown in Figure 4.22 and 4.23. Idling events are indicated by blue dots. If there is a cluster for a large number of idling events and/or long idling time, the location will be highlighted in yellow. Excessive idling concentration locations can be quickly identified using idling heat maps. The monthly idling event location and duration data have also been plotted as a cluster map on Google Earth using Keyhole Markup Language (KML), as shown in Figure 2.24 for December 2013 in the Stockton area. The red circle highlighted two Caltrans office/maintenance yard locations where there were high concentrations of idling activities. Similar clusters are seen at many other Caltrans offices and maintenance stations. Idling heat maps for each month of the pilot study are provided in Appendix D, along with idling cluster maps for various regions for January 2014.
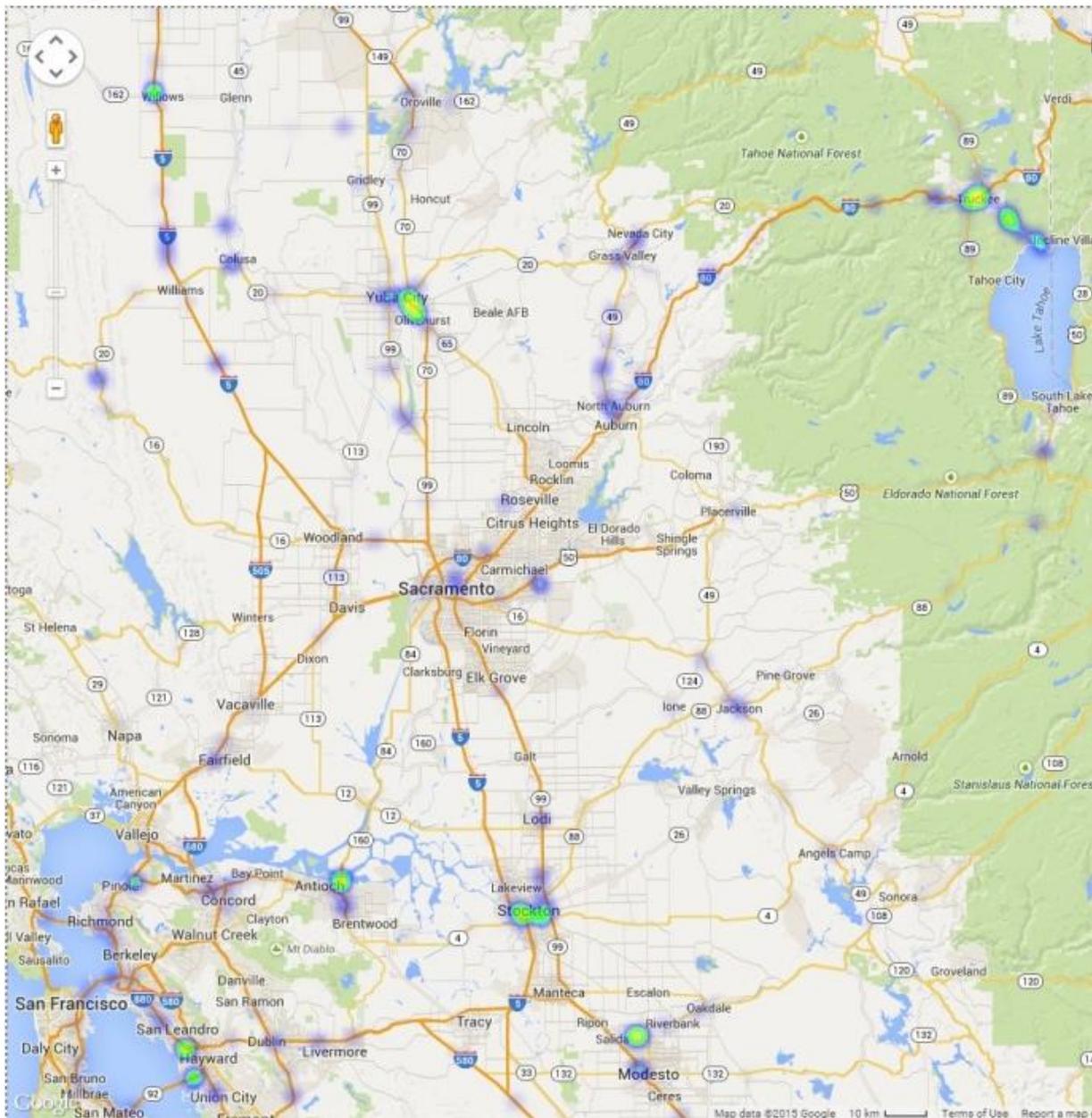
Figure 4.22 – Idling location heat map overlay on Google Maps for December 2013
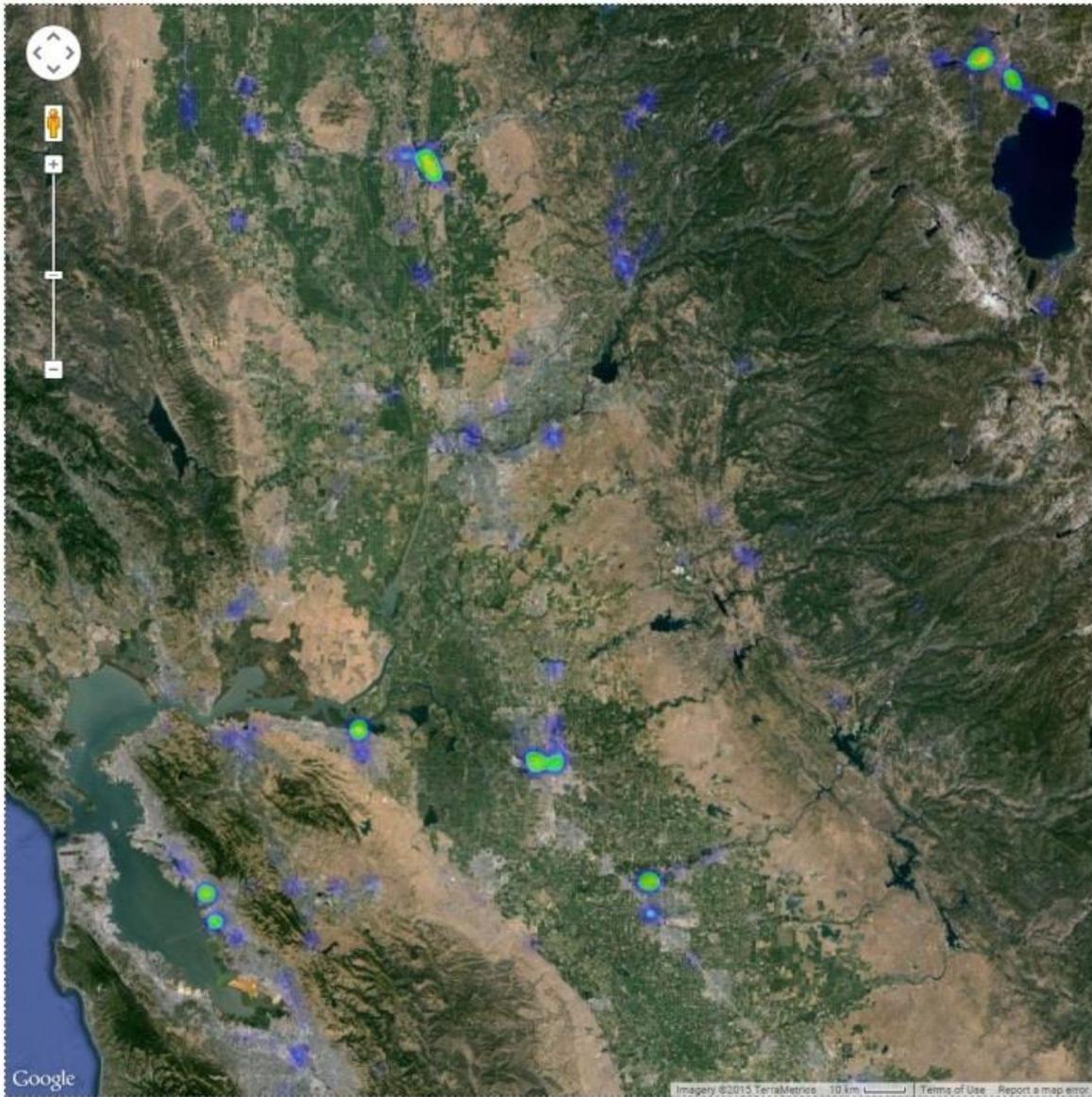
Figure 4.23– Idling location heat map overlay on Google Maps for December 2013 with satellite photo view

28
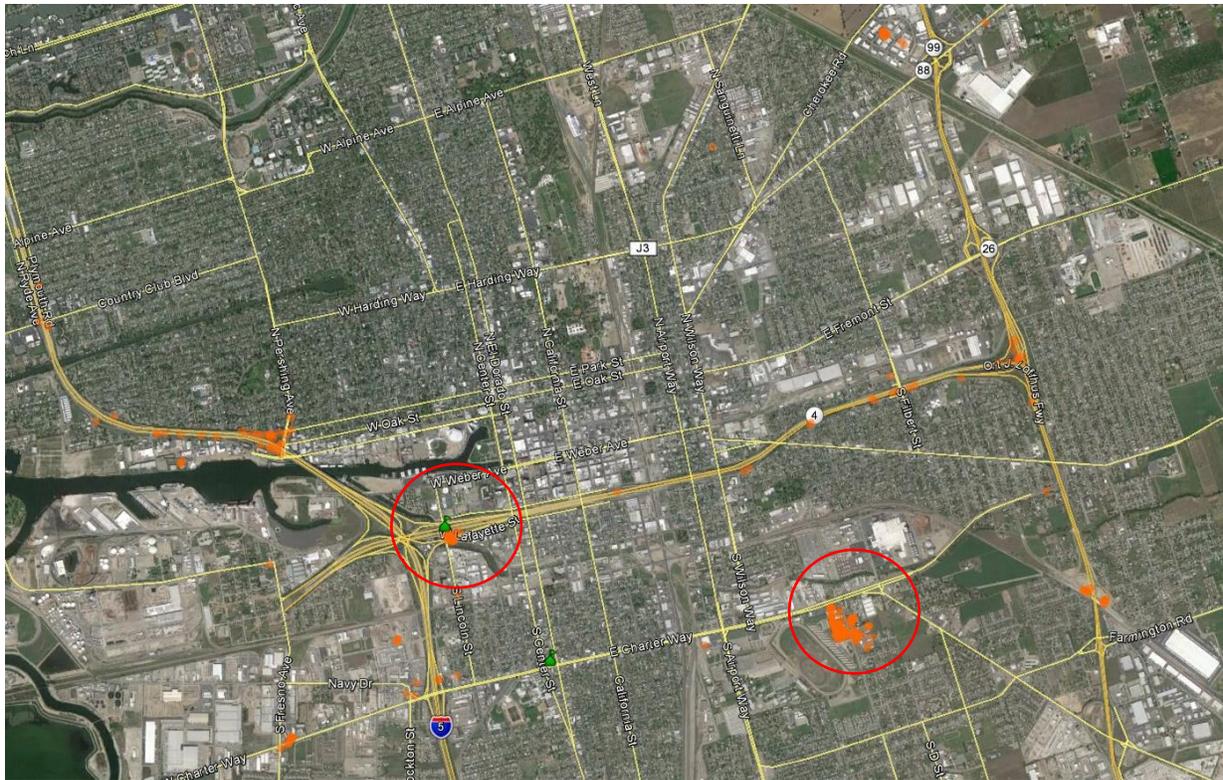
Figure 4.24 – Idling locations plotted on Google Earth using KML
(December 2013 in Stockton area)

Pilot Fleet Speed Violation Statistics

Networkfleet provides speed violation reports based either on the posted speed limit of the location or the maximum speed threshold. The reports included the violation speed, violation duration, and the distance travel during speed violation. The telematics devices occasionally report over 80 MPH speed violation for a short duration during GPS start-up if OBD-II speed data is not available. The speed violation duration is found to be inaccurate. There were a number of data speed violation data points showing over 80 MPH speed with violation duration of over 4 minutes. However, the speed violation travel distances were under 0.5 miles. Therefore, our analysis focused on speed violation data over 80 MPH and speed violation travel distance over 0.5 miles. Figure 4.25 and 4.26 shows the distribution of the speed and distance travel over 80 MPH. The majority of the speed violation is between 80 to 85 MPH.
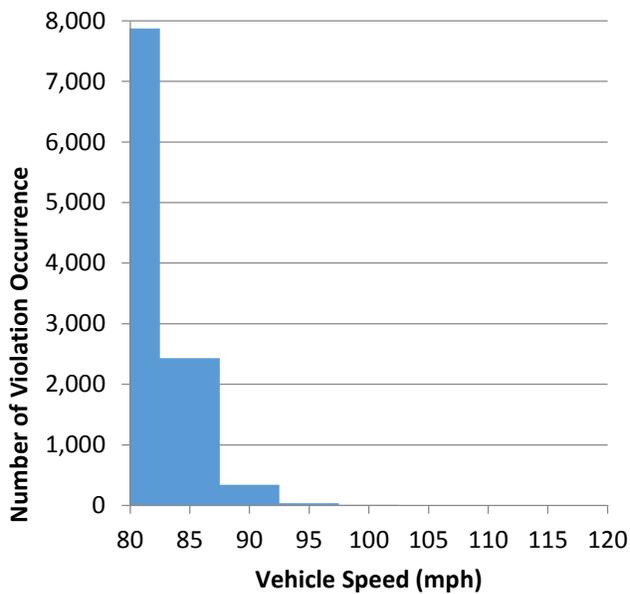
Figure 4.25 – Pilot study fleet over 80 MPH violation speed distribution
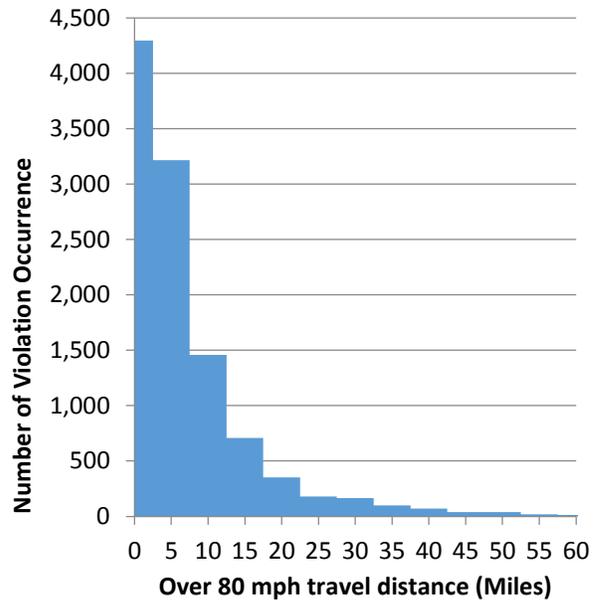
Figure 4.26 – Pilot study fleet over 80 MPH violation distance travel distribution

Figure 4.27 and 4.28 illustrate that the pilot fleet total monthly speeding travel distance and occurrences change with the time of the year and proportionally to the total miles travel. There was a significant drop of speeding violation after October 2013. The Division of Equipment worked with the Division of Maintenance, Human Resource Department, and the labor union to develop and implement a speeding policy. Letters were sent to the vehicle operators in October 2013 regarding the speeding policy. Before the speeding policy implementation (December 2012 to May 2013), the speeding distance traveled was about 6% of the total travel distance (50 miles per month per vehicle). After the speeding policy implementation (December 2013 to May 2014), the speeding distance traveled dropped to 0.4% of the total travel distance (4.3 miles per month per vehicle). Figures 4.27, 4.28, 4.29, and 4.31 show a large drop in both speeding miles traveled and speeding occurrences after November 2013, proving the effectiveness of the speeding policy implementation. Figure 4.31 shows the drop of total speeding over 80 MPH traveled distance six months before and after the speeding policy implementation for each vehicle. The original vehicle ID was masked using an anonymous vehicle ID for privacy reasons. Figure 4.31 shows that there were two vehicles still exhibiting excessive speeding behavior after the speeding policy implementation. Additional coaching and defensive driver training may be required for those two vehicle operators. High risk driver behavior may be identified using a combination of speeding distances and occurrences.

The fuel used over 80 MPH was estimated assuming a 20 % fuel economy reduction for the travel distance at over 80 MPH [4,5]. The fuel saving from eliminating the speed over 80 MPH is estimated based on equations derived in Appendix C. Figure 4.30 shows a large drop in percentage of extra fuel used due to speeding over 80 MPH after November 2013. Thus, a fuel savings from speeding reduction of about 1% of total fuel usage was realized in the successful implementation of the speeding policy.

30
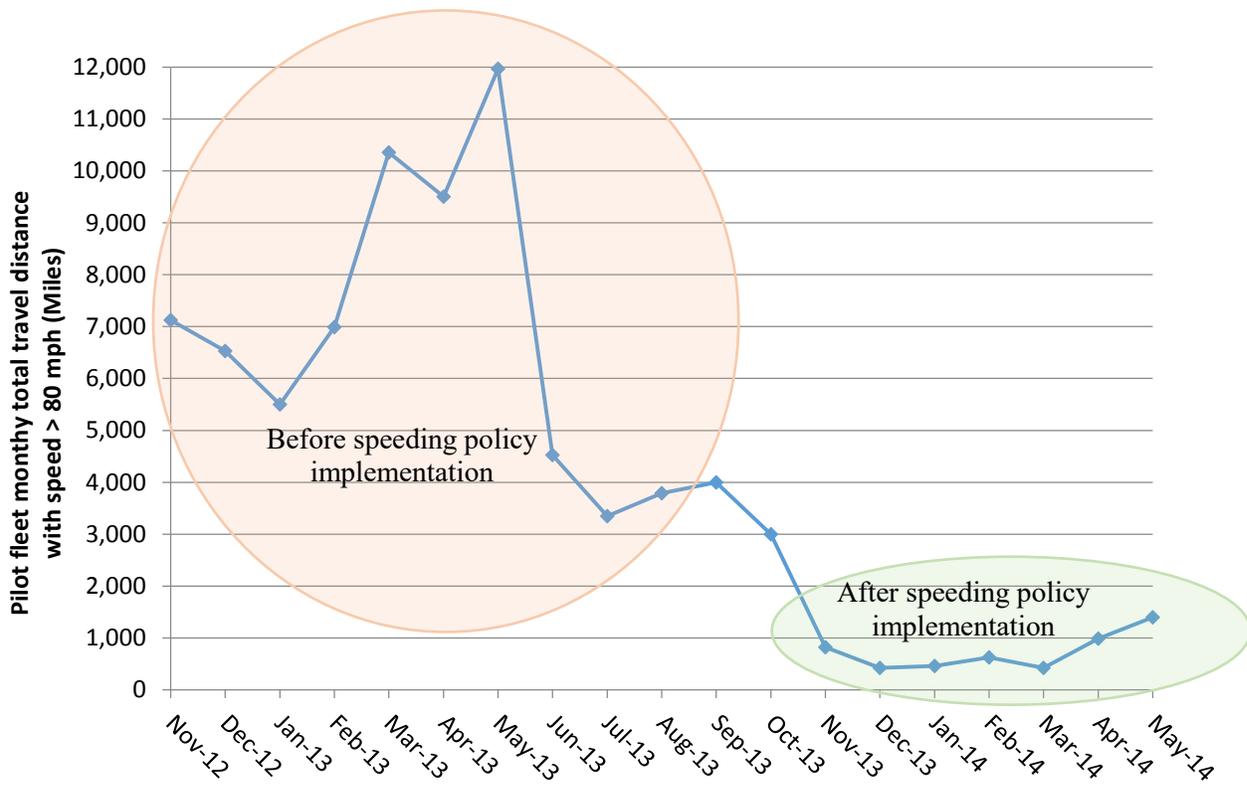
Figure 4.27 – Pilot study fleet over 80 MPH monthly total distance travel per month
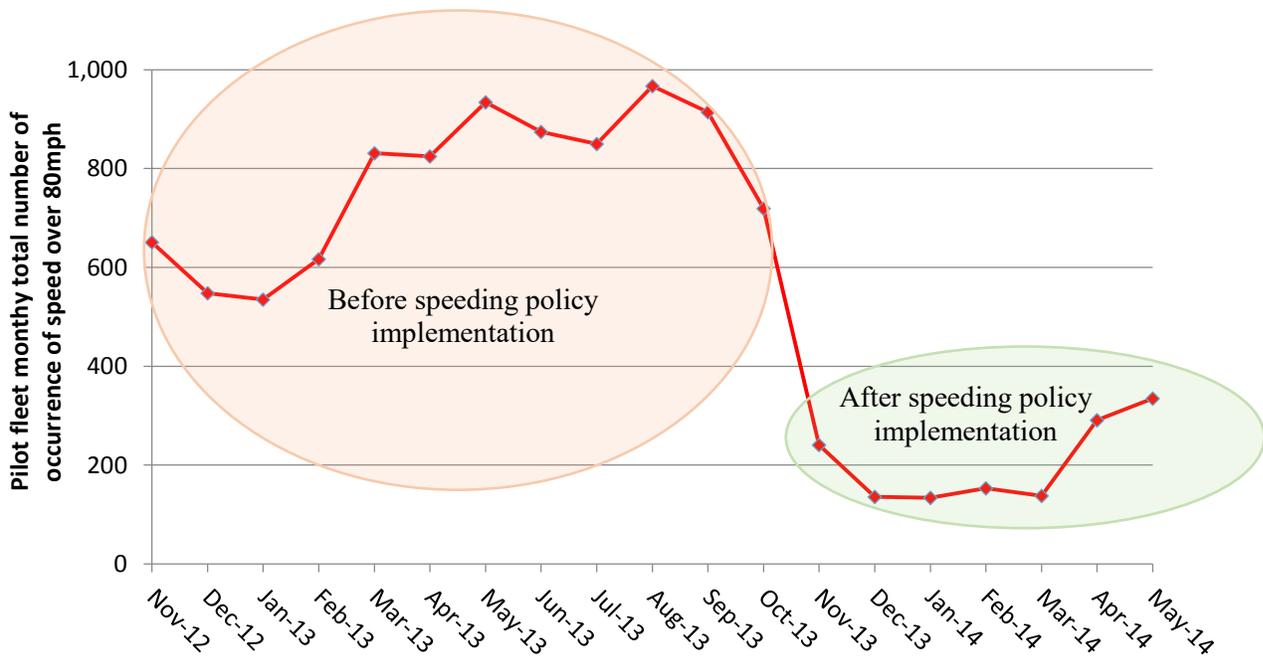
Figure 4.28 – Pilot study fleet over 80 MPH monthly total occurrence per month

31
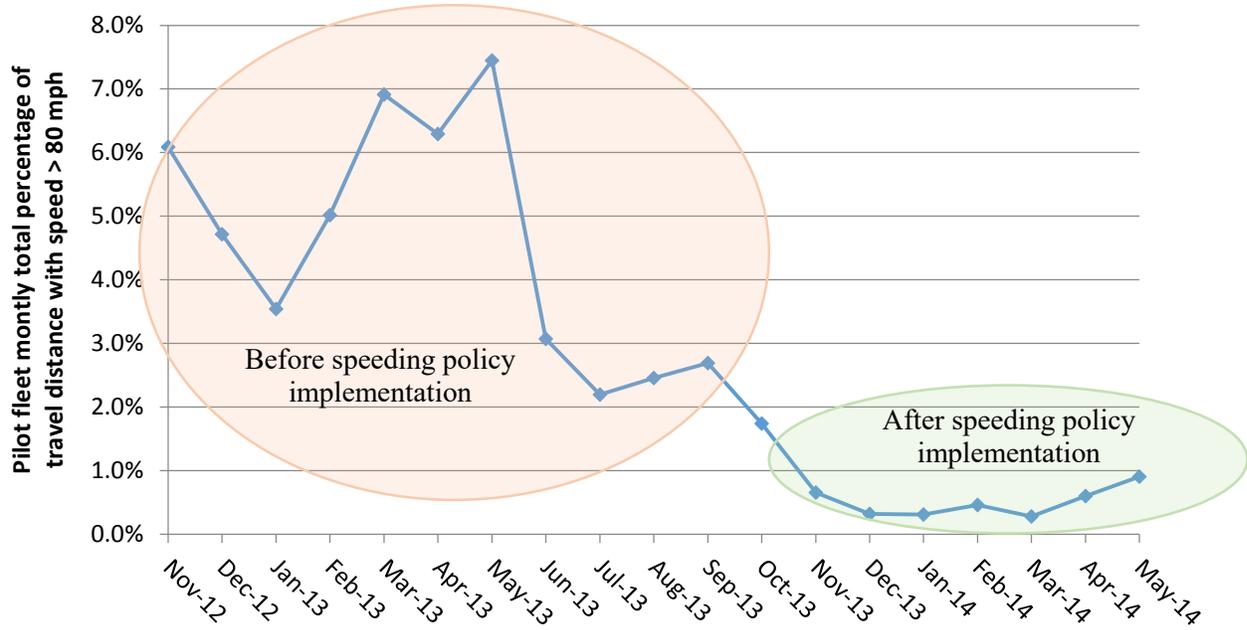
Figure 4.29 – Pilot study fleet monthly percentage total travel distance over 80 MPH
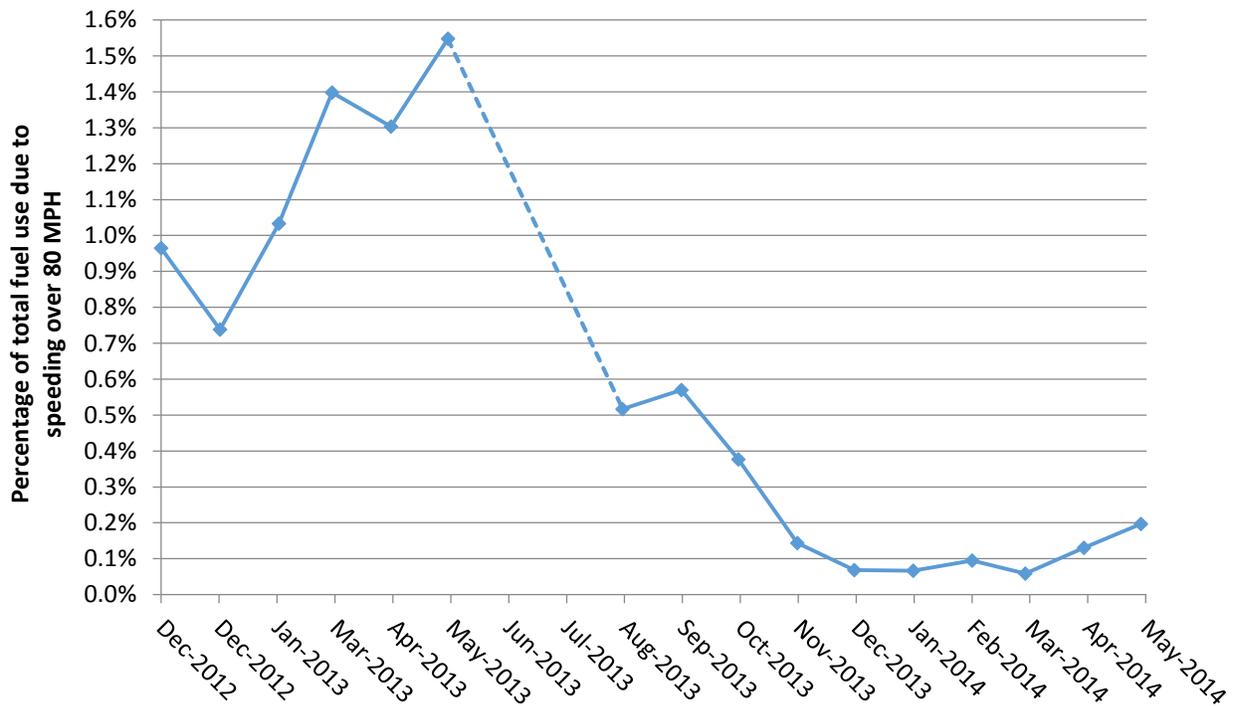
Figure 4.30 – Pilot study fleet monthly percentage total travel distance over 80 MPH
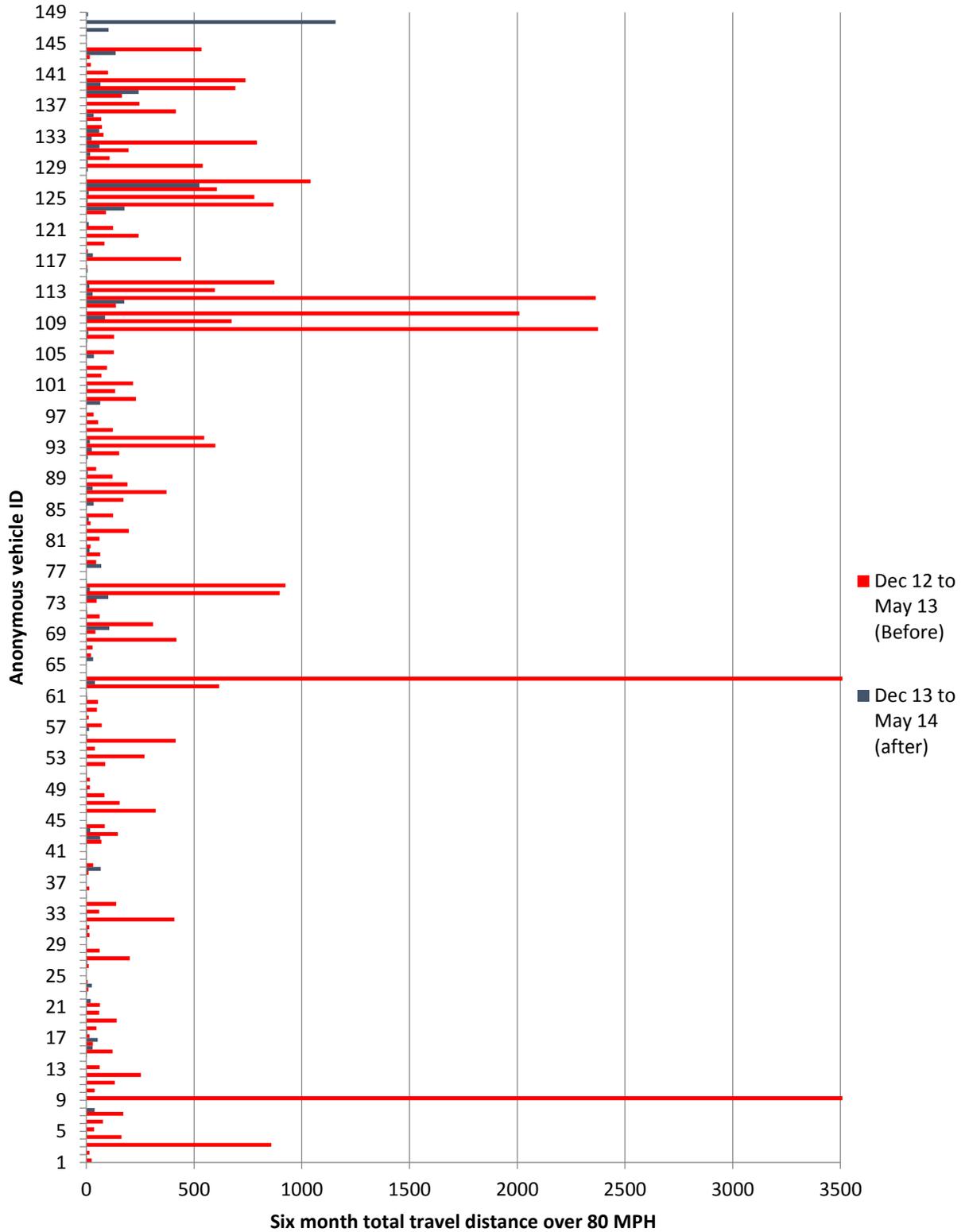
Figure 4.31 – Pilot study fleet six months total travel distance over 80 MPH before and after speeding policy implementation

Conclusions

Figures 4.21 and 4.30 summarize extra fuel usage due to speeding over 80 MPH (1.1%), and idling (13%). 14% fuel and fuel cost could be saved by eliminating all idling and speeding. However, zero percent speeding and idling is unrealistic. Eliminating 85% of speeding and idling would be a realistic goal, and it would result in 12 % fuel saving (6.6 gallon per vehicle per month, or approximately $26.4 per vehicle per month).

In addition, the total monthly idling duration (Figure 4.15) does not exhibit any reduction over time. This suggests that the installation of the telematics devices doesn't change driver behavior in reducing idling and without coaching, contrary to some FIDAS solution providers' claims. On the other hand, total monthly speeding distance vs. time (Figure 4.27) shows that if a speeding policy is implemented and enforced, driver behavior will change, resulting in a significant drop in speeding violations and duration. Therefore, policies on reducing idling and speeding must be established and enforced with coaching and training in order to gain the benefit of idling and speeding reduction. Even though the pilot fleet is not a random sample from the Caltrans fleet, it provided valuable insight on the current fleet operation and justified potential savings opportunity.

Better automation software tools need to be developed to analyze fleet operation performances. The current data analysis process is labor intensive and time consuming, and involves inaccurate data. Access to raw data would enhance the data and analysis accuracy. Data with longer survey period will help improve the analysis accuracy. Risky driver behaviors can be identified using a combination of both speeding travel distances and number of speeding violation occurrences.

CHAPTER 5:
COST-BENEFIT AND RETURN ON INVESTMENT ANALYSIS

The RFI provided valuable cost information, and the pilot study yielded factual tangible benefit evidence for a cost-benefit analysis. The solution providers also supplied benefit information in their presentations to the TAG. The FIDAS systems enable fleet managers to enforce policies to reduce fuel used, vehicle wear, and repair costs. The pilot study showed that the existence of telematics devices per se does not necessarily motivate vehicle operators to reduce idling and speeding—feedback must be provided to the operators in order to change their behaviors and realize the tangible and intangible benefits. The West Virginia Department of Administration Office of Fleet Management has developed a detailed Vehicle Telematics Cost-Benefit Analysis - FY2013 Appropriations Request [6]; this request provided reference data for vehicle telematics cost and benefit.

FIDAS Deployment Cost

The FIDAS cost statistics, gathered through the RFI, correspond to the West Virginia Department of Administration Office of Fleet Management cost estimation for vehicle telematics using cellular data link ($250 per vehicle device, $50 per vehicle installation cost, and $25 per vehicle monthly service cost, as shown in Table 5.1). As stated before, Caltrans has approximately 7,100 vehicles in their light vehicle fleet. The telematics device service life is assumed to be five years in the analysis, even though its service life would mostly likely be longer. Nevertheless, technological advancement may render them obsolete in five years. The devices and monthly service cost may be lower due to large volume discount of 7,100 units.

Another alternative FIDAS deployment scenario, using FIDAS telematics device with WiFi as data link, was explored and cost calculated. In this case, Caltrans will develop a database and web server software and host the web server to receive data from the vehicle telematics devices via WiFi station installed in each Caltrans maintenance yard. In this scenario, the vehicle telematics device will collect and store the automobile operating data, and then transmit data to the server when a Caltrans WiFi station comes into its communication range. Caltrans has a total of 388 station / maintenance yards throughout California. Table 5.1 shows their distributions. The additional cost items applicable with this setup are listed in Table 5.2. We assumed one full-time employee (FTE) for each year ($100,000 per year including benefit and overhead) after the first year deployment to support the web server and its services. The development cost for the web server software was estimated to be $300,000 for one year.

Table 5.2 presents the total cost of the first year and subsequent years as well as the five-year total cost for full deployment to 7,100 vehicles and more limited deployment to 2,000 vehicles. The higher telematics device cost, WiFi station installation, and web server setup cost contributed to the high first-year cost for the FIDAS with WiFi scenario. However, the annual service costs for FIDAS with WiFi are much lower than that of the FIDAS with cellular link. As a result, the five-year total cost for the WiFi option is lower than the cellular link option (~ $4 million, about a 30% reduction over five years for 7,100 vehicles). Moreover, a smaller scale deployment alternative, where only 2,000 vehicles were installed with vehicle telematics, was also considered in the cost analysis. The result is given in Table 5.2 for comparison. In this case, the FIDAS with cellular link option has the cost advantage in both the first year and five-year

total cost. Thus, the WiFi option should only be considered if the eventual monitored fleet size is greater than 2,000.

Table 5.1 – Caltrans maintenance yard distribution throughout districts

Caltrans District	Total Number of Maintenance Facilities
1	19
2	43
3	46
4	74
5	15
6	28
7	60
8	26
9	18
10	29
11	19
12	11

Table 5.2 – FIDAS vehicle telematics cost for entire light fleet (7,100 vehicles) and a subset (2,000 vehicles)

Option	FIDAS with cellular link	FIDAS with WiFi link	FIDAS with cellular link	FIDAS with WiFi link
Number of vehicles	7,100	7,100	2,000	2,000
Telematics devices cost	$250	$450	$250	$450
Installation cost	$50	$50	$50	$50
Month services cost	$25	$0	$25	$0
Devices services life (years)	5	5	5	5
Number of maintenance station	N/A	388	N/A	175
WiFi station cost (hardware + installation Cost)	N/A	$7,000	N/A	$7,000
WiFi station annual maintenance cost	N/A	$1,000	N/A	$1,000
Total WiFi base station cost	N/A	$2,716,000	N/A	$1,225,000
Webserver Cost	N/A	$5,000	N/A	$5,000
Web services annual maintenance cost (1 FTE)	N/A	$100,000	N/A	$100,000
Webserver software development cost	N/A	$300,000	N/A	$300,000
Services cost / vehicle / year	$300	$69	$300	$138
Total cost / vehicle for 5 years	$1,800	$1,255	$1,800	$1,903
Total first year cost (Entire fleet)	**$4,260,000**	**$6,959,000**	**$1,200,000**	**$2,705,000**
Total subsequent yearly cost (Entire fleet)	**$2,130,000**	**$488,000**	**$600,000**	**$275,000**
Total cost for 5 years (Entire fleet)	**$12,780,000**	**$8,911,000**	**$3,600,000**	**$3,805,000**

Table 5.3 – Pros and cons of WiFi and cellular link options

Alternatives	Pros	Cons
FIDAS with cellular link	• Low initial cost • Fast deployment • Low risk of failure • Large pool of solution providers available for the bidding process • Best for small scale deployment • Reduces vehicle wear and repair cost • Reduces labor cost to collect fleet utilization statistics • Improves the speed and accuracy of the fleet utilization statistics • Reduces liability due to speeding • Could poll vehicle current location	• Higher 5-year (long-term) total cost • Limited access to raw data • Limited long term data availability (solution providers typically do not store data over one year) • Requires the development and enforcement of new policies on idling and speeding

FIDAS with WiFi	• Lower 5-year (long-term) total costs
	• Unlimited long-term access to raw data
	• Reduces vehicle wear and repair cost
	• Reduces labor cost to collect fleet utilization statistics
	• Improves the speed and accuracy of the fleet utilization statistics
	• Reduces liability due to speeding
		• High initial cost
		• Limited telematics devices selection available
		• Not suitable for small scale fleet deployment
		• Higher risk
		   ○ Web server software development risk
		   ○ Installation of WiFi Station at maintenance yards
		   ○ Associated IT project risk
		• Slower deployment due to installation and setup of web server and WiFi base station
		• Requires the development and enforcement of new policies on idling and speeding
		• Cannot poll vehicle current location

Status quo (not deploying telematics solution)	• Eliminates risk of FIDAS deployment failure • Eliminates additional personnel time needed for FIDAS deployment	• Continues to waste fuel and not reducing greenhouse gas emission • Continues liability of speeding exposure • Higher vehicle wear and repair cost • Continues the existing labor intensive, higher labor cost method to collect fleet utilization statistics

FIDAS Tangible and Intangible Benefits

The potential benefits of a fleet management telematics solution are listed in Table 5.4. The tangible benefits that may provide direct dollar saving toward funding of the FIDAS solution are the fuel saving from reduced idling and speeding, and decreasing vehicle misuse. However, vehicle misuse is difficult to detect and can require labor intensive work to manually inspect each trip. The solution providers claim fuel saving ranging from 10% to 20%. The pilot study data suggest a maximum possible saving of 21% of total fuel used. Nevertheless, 16% fuel reduction is a more realistic and achievable target. The West Virginia Department of Administration Office of Fleet Management's cost-benefit analysis provided their estimate of fuel saving in three areas [6]:

1. Reduce mileage (2% to 8%) from reduced unauthorized usage and improved routing.

2. 15 minute reduction in idle time per day. The annual fuel savings equates to approximately $500 per vehicle per year (EPA).

3. 5 MPH average speed reduction. Each mile per hour over 60 MPH effectively increases cost per gallon of fuel by $0.04 (www.fueleconomy.gov).

Table 5.4 – Benefits of a fleet management telematics solution

Benefit	Description and saving source
Fuel Savings	• From reducing idling • From reducing speeding • From reducing vehicle mileage created by vehicle misuse
Reduce vehicle wear and repair cost	• From reducing vehicle abuse, speeding, and idling • From reducing vehicle mileage created by vehicle misuse • From timely repair facilitated by the DTC (Diagnostic Trouble Code) alert
Labor savings	• Automated smog check (1.5 hour labor and ~$60 for smog check fee) Caltrans has successfully worked with the California Air Resource Board to implement automated smog check for the pilot fleet. (Estimated $135 per vehicle per year). • Collect fleet utilization data • Reduced work labor in fuel and mileage reports • Non-tangible labor saving in entering driver and trip information into Cartag when driver ID reader is used. • Savings from elimination of Cartag program
Reduce accidents	• From early collection of bad driver behaviors such as speeding, hard braking, and hard acceleration
Theft recovery	• Increase probability of recovering stolen vehicle.

Fuel Cost Assumption

Fuel cost saving is directly proportional to fuel cost. In order words, 10% increase in fuel cost will result in 10% increase in fuel cost saving. Thus, the fuel cost (dollar/gallon) assumption used in the cost-benefit can significantly change the pay-back period. The Energy Almanac (http://energyalmanac.ca.gov/gasoline/retail_gasoline_prices.html), collected by the California Energy Commission, provides historical average fuel cost (shown in Figure 5.1) in California since 1996. The fuel cost fluctuates with season, geopolitical events, and economic activities, and is difficult to forecast. There is also geospatial fuel cost variation in California. For example, the fuel prices in the greater Los Angeles area are generally higher. However, the linear (green solid line and green letter equation in Figure 5.1) and second order curve (red dotted line and red letter equation in Figure 5.1) fit on the historical data yielded some insight into the future fuel cost. The linear fit (y is the fuel cost and x is the number days since the beginning of 1990) suggests an average of $0.18 per gallon per year cost increase. The 2nd-order curve fits suggests even higher annual cost increases than predicted by the linear fit. The average regular gasoline price in California from December 2012 to May 2014 was $3.87. Thus, a $4.00 per gallon fuel prices was used for the cost-benefit analysis. In addition, the analysis also included an $0.18/gallon/year increase in fuel prices to examine the sensitivity of the analysis. Historical gasoline cost data and trends were used to develop bases and justification for fuel cost estimate. However, the authors understand that future gasoline cost may not follow historical gasoline cost data and trends. For example, fuel costs have been abnormally low for late 2014 and early 2015.

The chart is labeled: California average regular gasoline price (y-axis, $1.0 to $4.8) versus dates from Dec-1995 to Dec-2014 (x-axis). Equations shown: $y = 0.0005x - 15.451$ (green) and $y = 1E\text{-}08x^2 - 0.0005x + 3.0227$ (red).
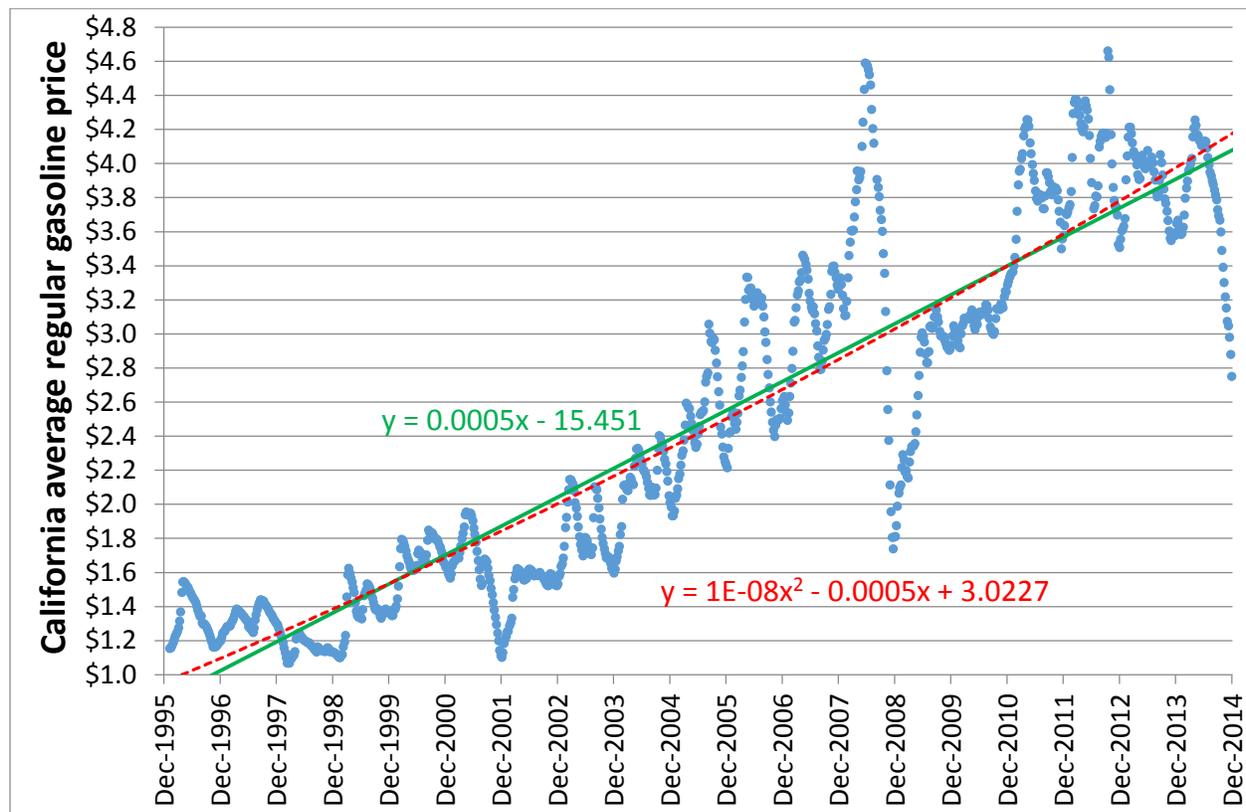
Figure 5.1 – Historical average regular fuel prices in California from 1996 to 2014

With annual gasoline price increase, using the average fuel consumption estimate (700 gallon per vehicle per year) data from the pilot study, fuel cost savings of 16%, 12%, and 8% are calculated and shown in Table 5.5. The West Virginia Department of Administration Office of Fleet Management fuel cost estimates are calculated using their provided Excel file, as tabulated in Table 5.5. The cost and benefit of each scenario are plotted in Figure 5.2. The two solid lines represent the cost of the two deployment options (cellular link and WiFi). The slope of the solid line denotes the annual cost of each option. The intersection of the two solid lines is the time after which the WiFi option will become cheaper.

Table 5.5 – Tangible benefits of a fleet management telematics solution

Yr	FIDAS with cellular link cost	FIDAS with WiFi link cost	Automated smog check saving	16 % fuel saving +$.18/gal/yr fuel cost increase with Automated smog check saving	12 % fuel saving +$.18/gal/yr fuel cost increase with Automated smog check saving	8 % fuel saving +$.18/gal/yr fuel cost increase with Automated smog check saving	WVDOT fuel saving estimate
1	$4,260,000	$6,959,000	$958,500	$3,180,800	$2,385,600	$1,590,400	$7,467,583
2	$6,390,000	$7,447,000	$1,917,000	$6,504,736	$4,878,552	$3,252,368	$14,935,167
3	$8,520,000	$7,935,000	$2,875,500	$9,971,808	$7,478,856	$4,985,904	$22,402,750
4	$10,650,000	$8,423,000	$3,834,000	$13,582,016	$10,186,512	$6,791,008	$29,870,333
5	$12,780,000	$8,911,000	$4,792,500	$17,335,360	$13,001,520	$8,667,680	$37,337,917

The multiple dashed lines are the various benefits estimates. The black dashed line on top is the WVDOT benefit estimate. It shows that the payback period is less than one year. The intersections of the various colored dashed line with the solid line are the corresponding payback period. For example the green dashed line (12 % fuel saving +$.18/gal/yr fuel cost increase with Automated smog check saving) shows a payback period of about 5 years for the cellular link option and about 3 years for the WiFi option. In addition, it illustrates that 12% fuel savings is the minimal saving that must be achieved to arrive at a breakeven point for the cellular option in 5 years. In addition, the slope of the pink dashed line (benefit with 8% fuel savings) is lower than that of the solid red line (cost of the cellular option), and it will never intersect with the solid red line. Thus, 8% fuel savings alone will never be enough to justify the deployment of the cellular option, no matter how long the service life of the telematics device. However, if fuel cost and fuel usage per vehicle increases beyond the assumptions, 8% fuel savings may justify the deployment of a cellular-based fleet management system. There is also some uncertainty in the cost estimation, as the device unit cost and monthly service cost will likely be lower due to the large volume of 7,100 units. Due to sensitive nature of volume discounts, vendors do not disclose large volume discount cost data. In addition, there is additional saving from the elimination of the Caltrans Cartag system. Currently, vehicle operators are required to manually enter the trip information such as date, driver ID, trip location, trip started and end odometer reading into Cartag.
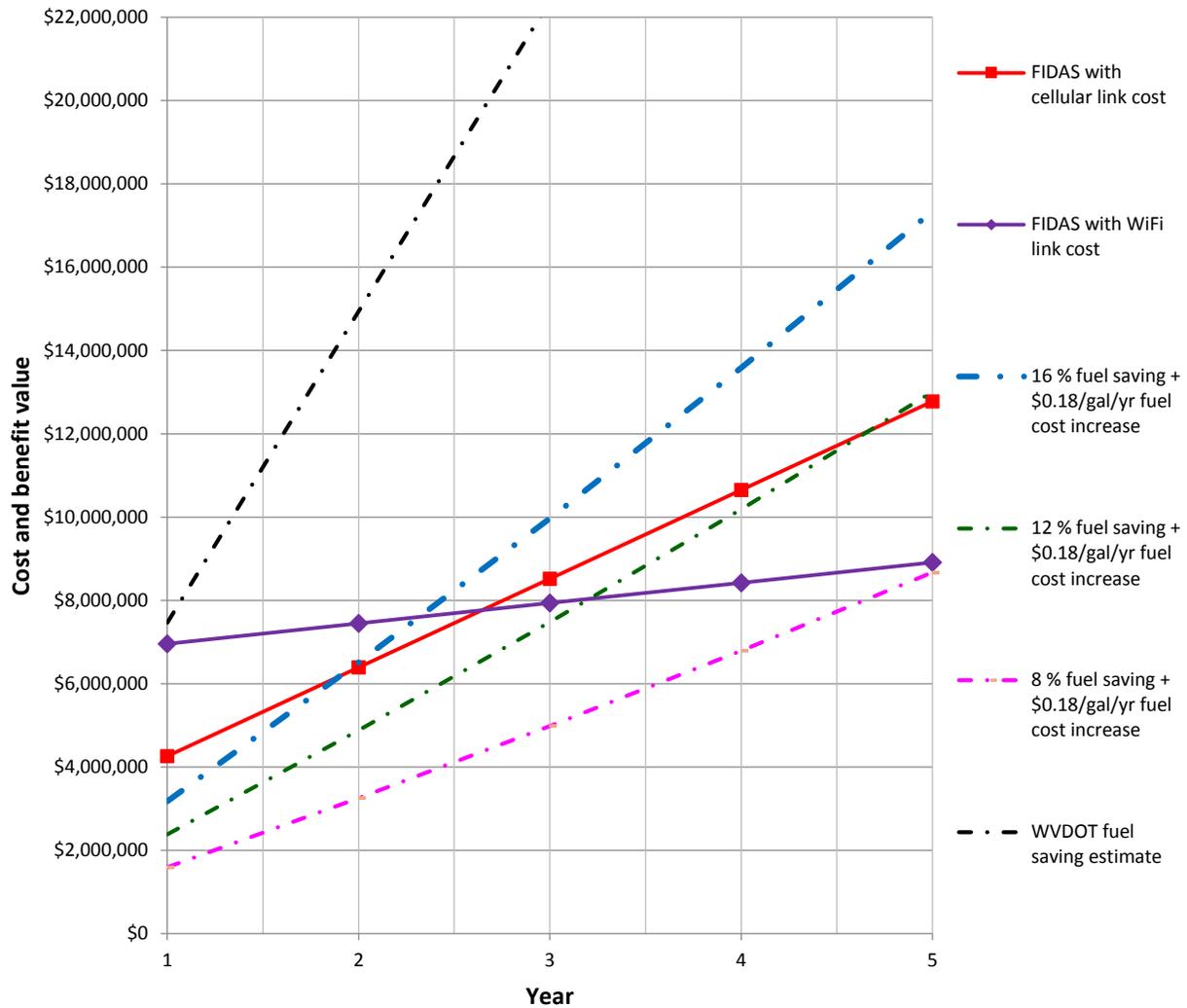
Figure 5.2 – Cost and benefit of various options and scenarios

<u>Cost-Benefit Analysis Summary</u>

Various fuel savings models are presented along with two different deployment cost options (cellular and WiFi). The objective of presenting several benefit models is to give an upper and lower bound of genuine achievable savings as well as providing decision makers with sufficient data to make an informed decision. The analysis depends on many variables that can significantly affect the results. The fuel price alone can change the payback duration. The WVDOT benefit analysis may be overly optimistic. In order to break even within 5 years, a minimum of 12% fuel saving (at $4 per gal.) must be achieved for the cellular option, and a minimum of 8% fuel saving (at $4 per gal.) must be achieved for the WiFi option. Based on the idling (13% of fuel usage) and speeding data (1.2% fuel usage), a DOT should be able to achieve 12% saving by implementing an effective speeding and idling reduction program.

CHAPTER 6:
CONCLUSIONS

The RFI response delivered state-of-the-art information on vehicle telematics and their associated cost. In addition, solution providers suggested several areas of benefit and saving that the solution could achieve. A few vehicle telematics solutions were evaluated in previous Caltrans projects as well as the current project. The OpenGTS open-sourced web server software has potential, but many improvements are required for Caltrans use. The UCR CERT study of alternative fuel usage demonstrated that the use of WiFi for data transfer is a viable option. The current CERT web server software could also possibly be revised for Caltrans use should the WiFi option be chosen.

The pilot study findings are summarized in Figure 4.25. 13% of fuel is squandered on idling, and 2% of fuel is wasted due to speeding. Eliminating 95% of speeding and idling would result in a 12% fuel saving (9.9 gallons per vehicle per month, or approximately $39.6 per vehicle per month). In addition, the study also found that installing the FIDAS telematics devices does not directly change driver behavior in the case of Caltrans. Some form of user feedback is required to change driver behavior. For example, in the early months of data collection, a significant number of over-speed events were noticed. Caltrans developed a notification letter, and sent it to the units where this was occurring. Subsequent to the letters being sent, the frequency and severity of over-speed events dropped dramatically. Vehicle telematics is a management tool. Its fuel saving effectiveness also depends on using the data to affect changes in the vehicle operator and vehicle maintenance. The difficult work is still ahead in creating a sensible policy on idling and speeding, and then training and coaching drivers without adversely affecting employee morale. These new policies must be implemented and enforced in conjunction with the FIDAS deployment in order to attain any fuel savings from reduced idling and speeding. The successful implementation of a speeding policy demonstrated its effectiveness in changing driver speeding behavior and resulted in savings of over 1% of fuel consumption. Additional fuel saving of up to 13% can be realized by implementing an idling reduction policy.

The cost-benefit analysis presented two deployment options: cellular data link and WiFi data link. Their pros and cons are listed in Table 5.3. The WiFi option has high initial cost but lower 5-year total cost. However, if the fleet size or deployment fleet size is less than 2,000, the cellular option is more advantageous. In addition, a minimum of 12% fuel saving is required to achieve breakeven for the cellular option in five years based on the cost assumptions noted in Chapter 5.

Recommendations

1. In order to accomplish fuel saving along with the deployment of a FIDAS solution, new policy on idling and speeding must be implemented, including training, coaching, and positive and negative enforcement to the drivers. Many states and municipalities have laws or ordinances to prevent extended idling. These generally prohibit idling beyond 3 or 5 minutes, except in cases of congested traffic, extreme temperatures, or for other extenuating circumstances. The summary of various state regulations provided by the California Air Resources Board in "Light Duty Anti-Idling Laws, Appendix B,"

(http://www.arb.ca.gov/toxics/sbidling/appb.doc), as well as the US DOE Idle Reduction Incentives and Laws may be used as reference.

2. Since the cost saving primarily comes from idling and speeding, the vehicle telematics solution will be most effective on vehicles with larger engines, lower MPG, and high annual mileage. Thus, if a limited deployment is pursued, we recommend targeting such vehicles. The maximum achievable savings on a hybrid vehicle, which automatically prevent engine idling, may not cover the cost of the solution.

3. The pilot study idling data provided information which can be used to justify the use of idling fuel consumption reduction technologies such as hybrid vehicles, cylinder deactivation systems, and automatic engine start-stop to avoid idling. We recommend that Caltrans consider the use of these and similar technologies.

4. There also seems to be some compatibility issue with obtaining fuel consumption estimate data from Chrysler and light vehicles with diesel engines. This should be a consideration factor in the vehicle telematics deployment. The telematics devices are quite reliable: only one device failed in the 18 months evaluation. The failed device was replaced at no charge to Caltrans.

5. A large portion of the potential benefit can be gained by analyzing the reasons for idling at maintenance stations and offices, and developing effective policies for reducing such idling. This can be seen by looking at the idling maps of Figures 4.22 – 4.24, and in Appendix D.

Future Work

Based on the successful results of the current study, a variety of future work is recommended:

1. Define fleet reporting policies and requirements.

2. Define long-term data storage policies and requirements.

3. Establish and implement a policy on idling and its reduction.

4. Research additional automated or semi-automated means to identify inappropriate trips, speeding driver, excessive idling.

5. Assess need for integration to Assetworks FA, and perform integration if warranted.

6. Consideration of collecting additional data through FIDAS for heavy fleet applications, such as pavement temperature, plowing location, and sanding location and rates.

REFERENCES

1. Argonne National Laboratory, "How Much Could You Save by Idling Your Light-Duty Vehicle Less?," http://www.transportation.anl.gov/pdfs/idling_worksheet_light-duty.pdf.
2. Federal Energy Management Program, "Federal Leadership in Environmental, Energy, and Economic Performance: Guidance for Federal Agencies on E.O. 13514 Section 12, Federal Fleet Management," *Executive Order 13514*, **http://www.epa.gov/oaintrnt/documents/fleetguidance_13514.pdf**, 2010.
3. G.W.R. Taylor, "Review of the Incidence, Energy Use and Costs of Passenger Vehicle Idling," Office of Energy Efficiency, Natural Resources Canada, 2003.
4. J.F. Thomas, H.-L. Hwang, B.H. West, and S.P. Huff, "Predicting Light-Duty Vehicle Fuel Economy as a Function of Highway Speed," in *SAE 2013 World Congress & Exhibition*, Detroit, MI, April, 2013.
5. U.S. Department of Energy, www.fueleconomy.gov.
6. West Virginia Department of Administration Office of Fleet Management, "Vehicle Telematics Cost-Benefit Analysis - Fy2013 Appropriations Request," http://www.fleet.wv.gov/SiteCollectionDocuments/About/West%20Virginia%20FY2013%20Budget%20AR%20CBAs.zip.
7. West Virginia Department of Administration Office of Fleet Management, "Vehicle Telematics Cost-Benefit Analysis," *FY2013 Appropriations Request*, 2013.
8. K. Yen, T. Swanston, V. Reveles, B. Ravani, and T. Lasky, "Identifying Excessive Vehicle Idling and Opportunities for Off-Road Fuel Tax Credits for Stationary Operations in the Caltrans Fleet - Phase 1," AHMCT Research Center Rept. # UCD-ARR-09-09-30-04, 2009.

APPENDIX A:
RFI QUESTIONAIRE

Caltrans Fleet In-vehicle Data Acquisition System: Request for Information

OVERVIEW

The California Department of Transportation (Caltrans) is assessing the availability of cost effective in-vehicle data acquisition systems for fleet management. The initial phase of this assessment is targeting light fleet vehicles that typically allow more uniform installations, require less customization and reduced per vehicle costs. The first phase of the assessment will consider system installations in 100 to 1,000 vehicles with two to five year duration. The objective of this effort is to improve fleet management and fuel use. Improvements will be accomplished by automating trip related data collection, report generation and improved communication to equipment managers.

The results of this Request for Information (RFI) may be used to finalize functional requirements into a biddable specification for advertisement. Advertisement will be contingent on several factors such as available budget, funding authority, obtaining purchasing authorization from control agencies and the evaluation alternative options.

RFI REQUIREMENTS PROCESS

RFI PARTICIPATION

This RFI is open to organizations / companies that have systems meeting the functional requirements described in this document and that can meet the timetable described in Section 2.2.

RFI SCHEDULE

December 22, 2011	RFI made available to the responders
January 4, 2012	Deadline for submitting questions
January 6, 2012	Response to all questions
January 9, 2012	Deadline for receiving response (all material)
January 10 – February 28, 2012	Responses to be evaluated. Responders may be invited to present solution to Department (presentation meetings, not negotiation meetings). Live demonstrations may be launched at this presentation.

47

RFI RELATED QUESTIONS / CLARIFICATIONS / SUBMISSION

Please email your questions, clarifications and completed RFI to both:

Kin Yen: ksyen@ucdavis.edu

Bob Meline: bob_meline@dot.ca.gov

Please provide the RFI response in electronic format: MS Word, Adobe Acrobat or Excel.

LIABILITIES OF AGENCY

CONFIDENTIALITY & RFI OWNERSHIP

Once received, the RFI response will become the property of the State of California with the exception of RFIs or portions of RFIs marked exempt as noted below. All non-exempt RFI information may be summarized, published and available for general public access.

MARKING RECORDS EXEMPT FROM DISCLOSURE (PROTECTED, CONFIDENTIAL, OR PROPRIETARY)

Any RFI submissions marked exempt from disclosure will be treated as confidential, used only internally by the State of California and the University of California and not included in external publications.

HIGH LEVEL BUSINESS REQUIRMENTS

OVERVIEW OF PRELIMINARY FUNCTIONAL REQUIREMENTS

- Web interface
- Passive idle
- Speed (from OBD-II and/or GPS)
- Odometer (from OBD-II)
- Fuel used (from OBD-II)
- Location (coordinates and addresses)
- OBD-II compatible
- Data point collection frequencies (min, max)
- Download raw and processed data in Excel, CSV format, or direct data access through API (Application Programming Interface)
- Hard braking, hard acceleration
- Data security and privacy protection
- Options for driver identification (RFID, manual input, fingerprint, others)
- Summary data reports
  - o Trip start and end: location (lat., long.), address, time, odometer reading, fuel used, trip mileage
  - o Over speed
  - o Excessive idling
  - o Night and weekend driving
  - o Usage reports (individual vehicle, fleet, subset of the fleet, etc.).
- Cellular, WiFi, and/or other wireless communications
- OBD-II fault code alert

48

RESPONSE FORMAT

No	Question	Response
	Vendor Information	
1	Name of person responsible for the information contained in this RFI	
2	Telephone number Fax number Email address Web page Product Specifications, Brochures, etc. Identify current California certification(s): small business (SB), microbusiness (MB) or disabled veteran business enterprise (DVBE) programs	Attach as required.
3	Have you implemented your system to any state or federal agencies? If so, please provide reference information.	
4	How many active units does your company have in vehicles? Provide date and evidence for support of claim, client list and references.	
5	Describe any third party alliances, relationships, or dependencies.	
6	Please provide information on your implementation methodology.	Attach as required.
7	What documentation is provided with the system and in what format?	

No	Question	Response
	System Administration	
8	What type and complexity are the system administration roles? Such as multiple administrators and different access levels, grouping, maximum number of users, etc.?	
9	What security or privacy protection have you implemented on your system?	
10	What type of technical support does your company provide?	
	Training / Anticipated Learning Curve	
11	Do you offer formal user training?	
12	What types of training courses do you run and what are their durations?	
13	What level of training would you recommend?	
14	Describe any training materials offered?	
15	What is your anticipated learning curve for users to generate reports?	
16	What is your anticipated learning curve for Technical Administrators?	
	Requirements	
17	What are the installation requirements (equipment, cables, other hardware, time, personnel)?	Attach as required.
18	What are the infrastructure requirements (Wi-Fi stations, GSM, CDMA, others)?	Attach as required.

50

No	Question	Response
19	Requirements for future scaling [expandable/upgradable]	Attach as required
	Capabilities	
20	Does your system integrate with other systems? [include other cell phone provider, if system depends on cellular communication]	
21	What other third party hardware and software are needed for the integration of your system?	
22	Does your system facilitate the setting of thresholds and provide alerts to users? [geo-fencing, speed, hard breaking, acceleration, vehicle fault code alert, system health, tampering, others]	
23	How are these alerts communicated to the end user? [email, text, phone …etc.]	
24	How does your system monitor and report engine fuel use [total and / or instantaneous]? What is the accuracy?	
25	How does your system determine trip mileage and vehicle odometer (GPS based and / or OBD-II)? What is the accuracy?	
26	Does your system monitor and report engine performance, including idle, and diagnostics? [idle, usage, tire pressure, & diagnostics]	
27	Does your system monitor and report driver behavior such as over-speed, hard breaking, hard acceleration or	

No	Question	Response
	impact data?	
28	Does your system monitor and report external auxiliary systems / devices [PTO, rotator light, hydraulics, work idle, digital or analog input, etc.]	
29	Does your system provide options for driver identification (RFID, manual input, fingerprint, others)	
30	Name all compatible web browsers that work with your system.	
31	Does your web host support mobile devices? [Blackberry, iPhone, iPad, Android...etc.]	Attach as required.
32	What are the data collection frequencies by the in-vehicle device? What are the in-vehicle data storage capabilities (number of days, storage size)?	
33	What are the data upload options (near real time or at end of day/trip)?	
	Third Party Integration	
34	Name all third party products required to meet the criteria described in Section 4.1.	
35	Does your system provide direct database query?	
	Data Exporting	
36	What are data export options supported by your system (CSV, Excel, Word, html, xml, kml, shape file shp, etc.)? List data fields available for	

No	Question	Response
	export.	
	Detailed Cost Model	
37	Is there anything that would require an additional or third party purchase to meet the requirements?	Attach as required.
38	Do you offer different support levels? (Hourly, methods of contact)	
39	Do you provide a monthly or yearly fee option for service-only that includes the hardware, data service and web service? If yes, provide estimated costs? (Quantity 10, 100, 1000)	
40	What is your annual maintenance cost estimate breakdown? (Quantity 10, 100, 1000) • Web access • Data hosting • Long term data storage (1, 2, 3, 5 years) • Data transmission costs • Device maintenance costs	
41	What is your system initial cost estimate? (Quantity 10, 100, 1000) • Total • Device cost (device, cabling, etc.) • Installation cost • Web hosting setup cost • Data transmission setup cost • Leasing cost (option) • Training costs • Other costs	
42	How is cost of data collection affected by frequency of vehicle data update?	

No	Question	Response
	Availability	
43	What is the initial lead time for hardware delivery? (in quantity of 10, 100, 1000)	
44	What is the lead time for replacement of hardware?	
45	What is your expected product life and warranty?	
46	What is the setup time for web site and report customization?	
47	What is your web access uptime percentage?	

APPENDIX B:
LIST OF FLEET MANAGEMENT SOLUTION VENDORS

Solution Provider	Website
Advanced Tracking Solutions - Parnered w/ Xirgo Technologies	http://www.AdvancedTrackingSolutions.com
Atrack	http://www.atrack.com.tw/main.php?p=product
Azure Dynamics	http://www.azuredynamics.com/
Broadband And Networking Consultants	http://www.bn-c.com/
CAN Telematics	http://www.cantelematics.com/
Car Chip (Car Chip connect)	http://www.davisnet.com/
Drone	http://www.dronemobile.com/
EarthRise Technology	www.earthrisetech.com
Fastrax	http://www.fastraxgps.com
Field Logix	http://www.fieldtechnologies.com/gps-fleet-tracking-system/
Fleet Mgmt. Solutions	http://www.fmsgps.com
FLEET TEK, Inc.	www.FleetTek.com
Fleet Tracking	http://www.fleettracking.com/
Fleetilla	http://www.fleetilla.com/
Gentrifi	www.gentrifi.com
Geotab	http://www.geotab.com/gps-tracking-solutions.aspx
GPS Fleet Solutions	http://www.gpsfleetsolutions.com
GPS Heroes	http://www.gpsheroes.org/
GPS Insight	www.gpsinsight.com
GPS North America	http://www.gpsnorthamerica.com/
HEM Data	http://hemdata.com/
Industrack	www.industrack.com
Inretti Technologies	
InterFleet and Webtech Wireless	http://www.webtechwireless.com/
Inthinc - Tiwi Pro	http://www.inthinc.com
Keytroller LLC	
Mix-Telematics	http://www.mixtelematics.com/
Enfora	http://www.enfora.com
Networkcar	http://www.networkfleet.com/
Networkfleet, Inc.	http://www.networkfleet.com/
Precise MRM	http://www.precisemrm.com/service_fleet/specs/index.html
TeleTrac	http://www.teletrac.net/
TrackNet	http://www.tracknetonline.com/
TransCore (ROVR)	http://www.transcore.com
Trimble	http://www.trimble.com/
WebtechWireless Vancouver, BC	http://www.webtechwireless.com/
Xata	http://www.xata.com/
Xirgo Technologies (XT 2000) - Partnered w/ Advanced Tracking	http://www.xirgotech.com
Zonar	http://www.zonarsystems.com/

55

APPENDIX C:
FORMULA FOR FUEL SAVINGS FROM ELIMINATION OF SPEEDING OVER 80 MPH

$$\text{MPG}_{speeding} = \text{K} * \text{MPG}_{normal} \quad where\ K = 0.80 \tag{1}$$

$$\text{Mileage}_{Total} = \text{Mileage}_{normal} + \text{Mileage}_{speeding} \tag{2}$$

$$\text{Fuel}_{with\_speeding} = \frac{Mileage_{normal}}{\text{MPG}_{normal}} + \frac{Mileage_{speeding}}{\text{MPG}_{speeding}} = \frac{Mileage_{normal}}{\text{MPG}_{normal}} + \frac{Mileage_{speeding}}{\text{K}*\text{MPG}_{normal}} \tag{3}$$

From equation (3):
$$\text{Fuel}_{with\_speeding} = \frac{K*Mileage_{normal} + Mileage_{speeding}}{K*\text{MPG}_{normal}} \tag{4}$$

From equation (4):
$$\text{MPG}_{normal} = \frac{K*Mileage_{normal} + Mileage_{speeding}}{K*\text{Fuel}_{with\_speeding}} \tag{5}$$

$$\text{Fuel}_{saving\_without\_speeding} = \text{Fuel}_{with\_speeding} - \text{Fuel}_{without\_speeding} \tag{6}$$

Thus, from equation (3) and (6):
$$\text{Fuel}_{saving\_without\_speeding} = \frac{K*Mileage_{normal} + Mileage_{speeding}}{K*\text{MPG}_{normal}} - \frac{Mileage_{normal} + Mileage_{speeding}}{\text{MPG}_{normal}} \tag{7}$$

From equation (7):
$$\text{Fuel}_{saving\_without\_speeding} = \frac{K*Mileage_{normal} + Mileage_{speeding} - K*Mileage_{normal} - K*Mileage_{speeding}}{K*\text{MPG}_{normal}} \tag{8}$$

From equation (8):
$$\text{Fuel}_{saving\_without\_speeding} = \frac{(1-K)*Mileage_{speeding}}{K*\text{MPG}_{normal}} \tag{9}$$

From equation (4) and (8):
$$\text{Fuel}_{saving\_without\_speeding} = \frac{(1-K)*Mileage_{speeding}}{K*(\frac{K*Mileage_{normal} + Mileage_{speeding}}{K*\text{Fuel}_{with\_speeding}})} \tag{10}$$

After simplification of equation (10):
$$\text{Fuel}_{saving\_without\_speeding} = \frac{\text{Fuel}_{with\_speeding}*(1-K)*Mileage_{speeding}}{K*Mileage_{normal} + Mileage_{speeding}} \tag{11}$$

APPENDIX D:
IDLING LOCATION MAPS

The idling location can provide some insight to the idling driver behavior, and lead to an appropriate policy to reduce idling. Idling heat maps based on the idling locations and duration for every month were created for each month of the pilot study. These heat maps are shown in Figures D.1 – D.18. Idling events are indicated by blue dots. If there is a cluster for a large number of idling events and/or long idling time, the location will be highlighted in yellow. Excessive idling concentration locations can be quickly identified using idling heat maps.
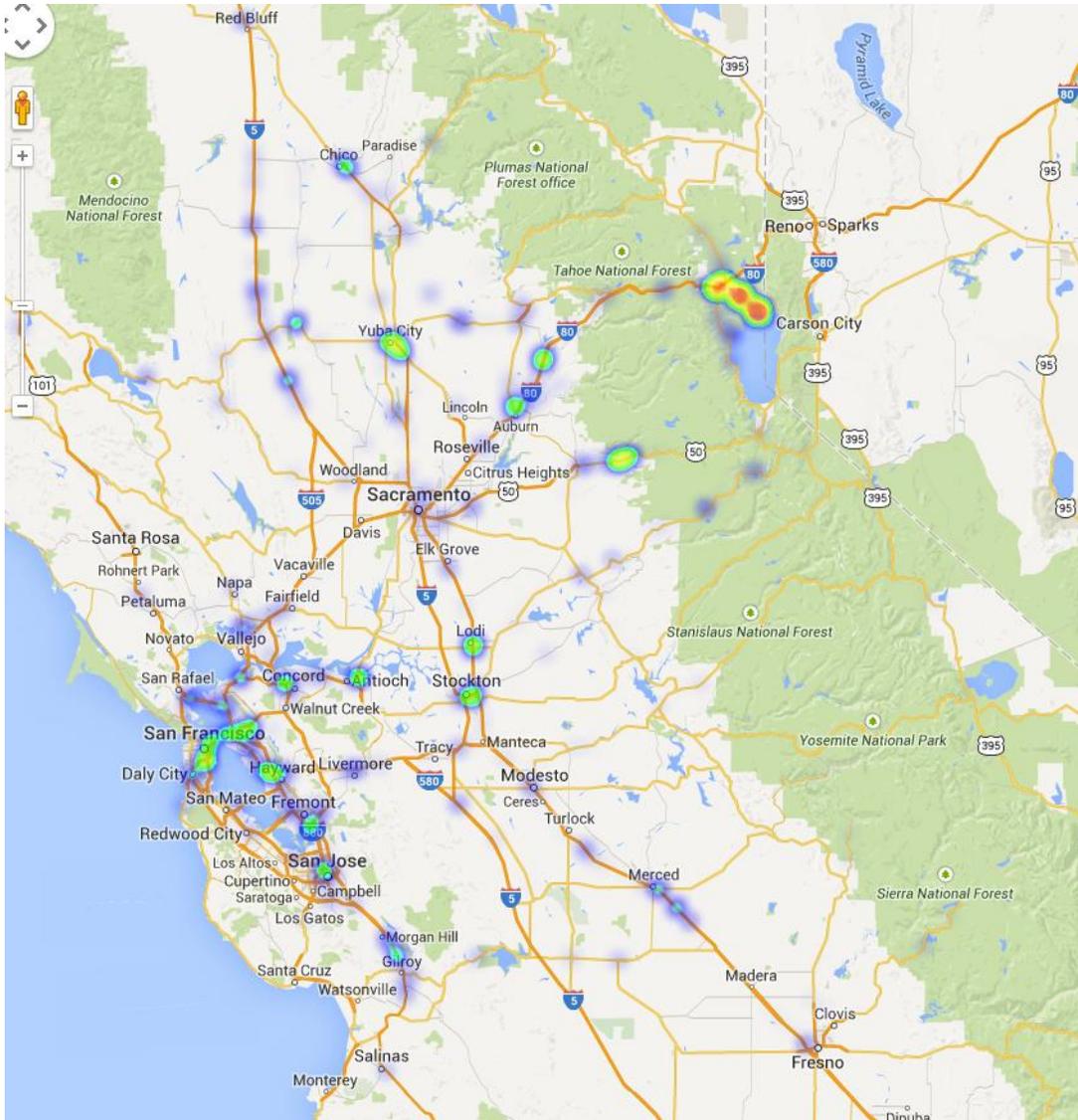
Figure D.1 – Idling location heat map for December 2012

58
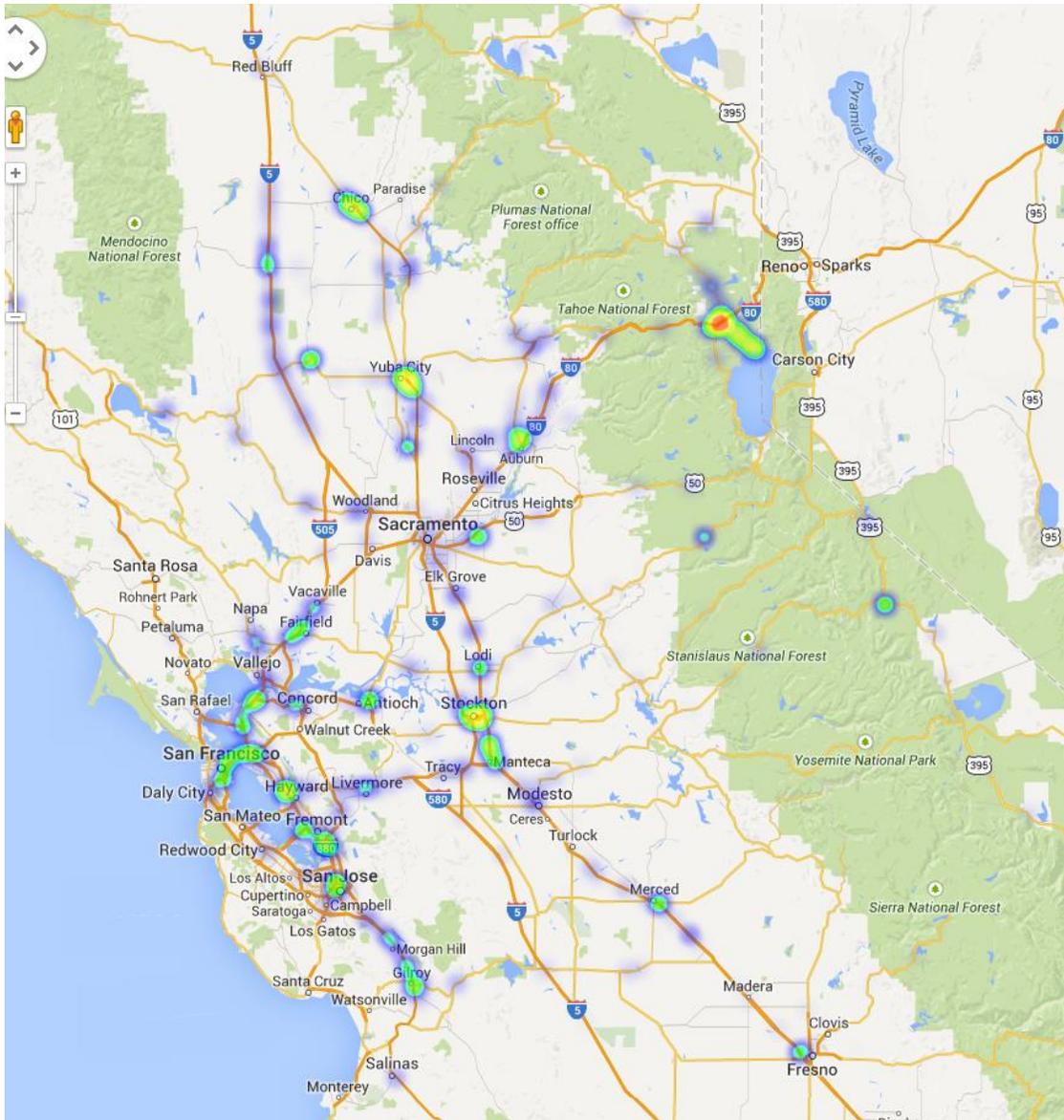
Figure D.2 – Idling location heat map for January 2013
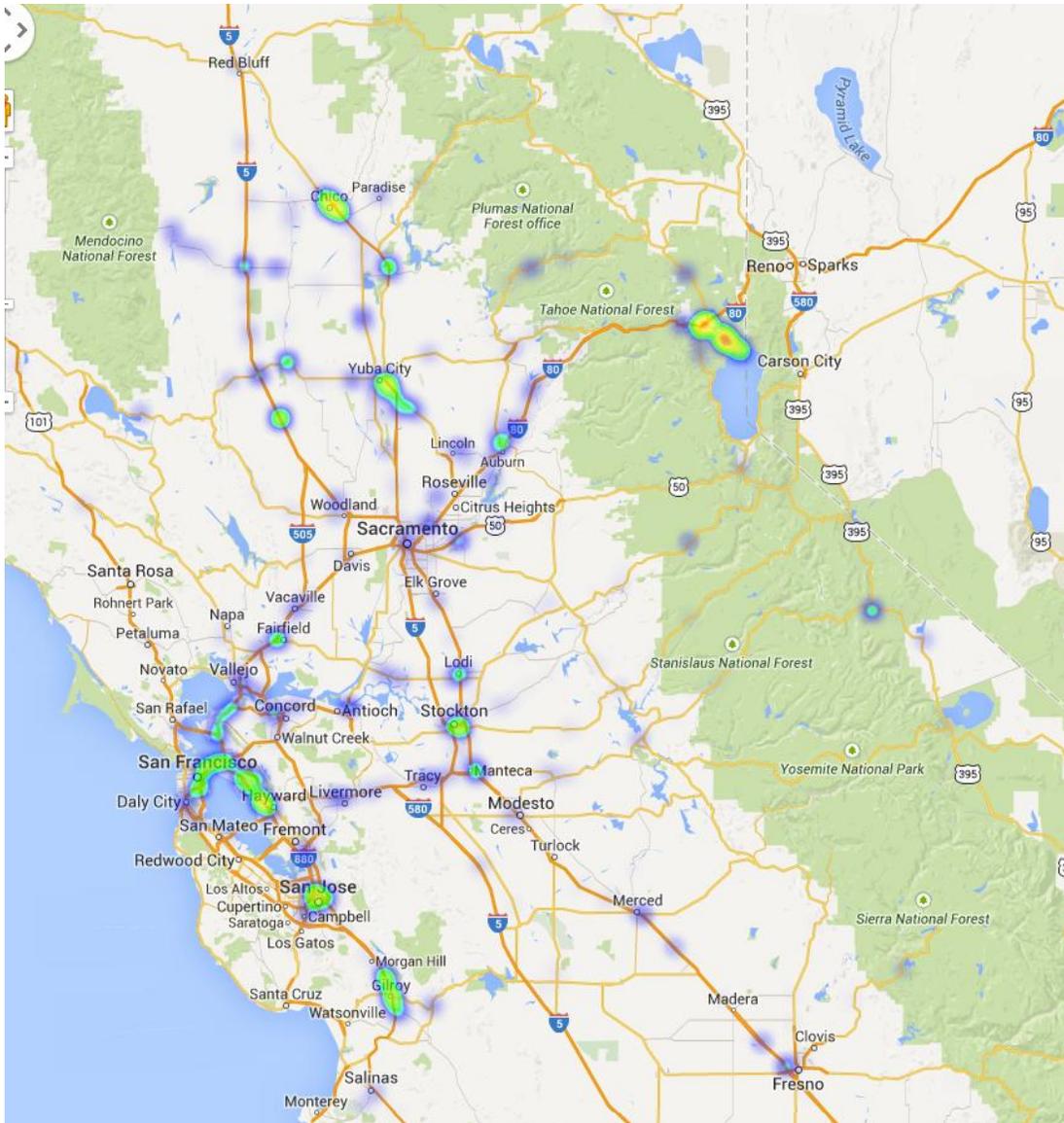
Figure D.3 – Idling location heat map for February 2013
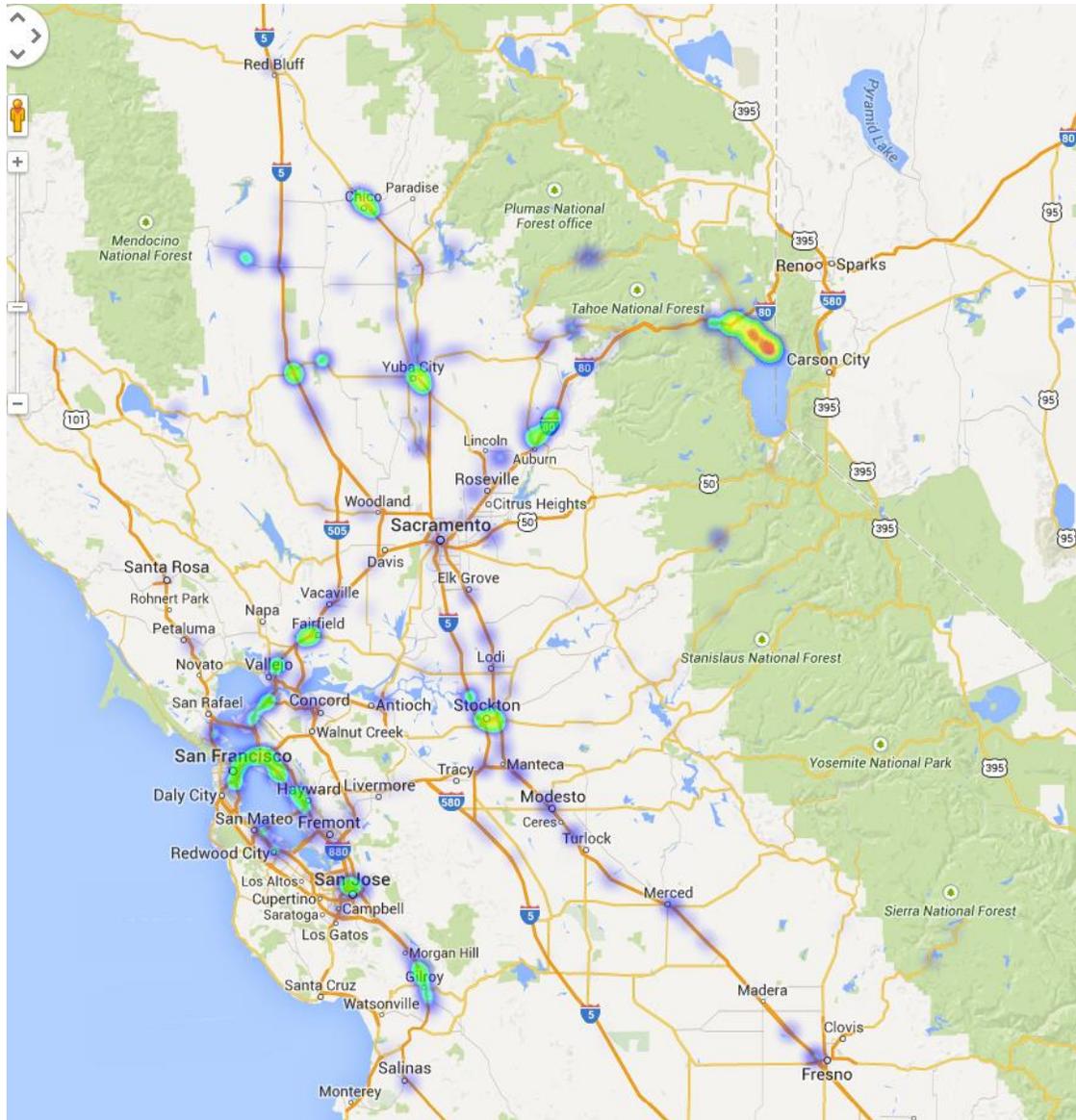
Figure D.4 – Idling location heat map for March 2013

61
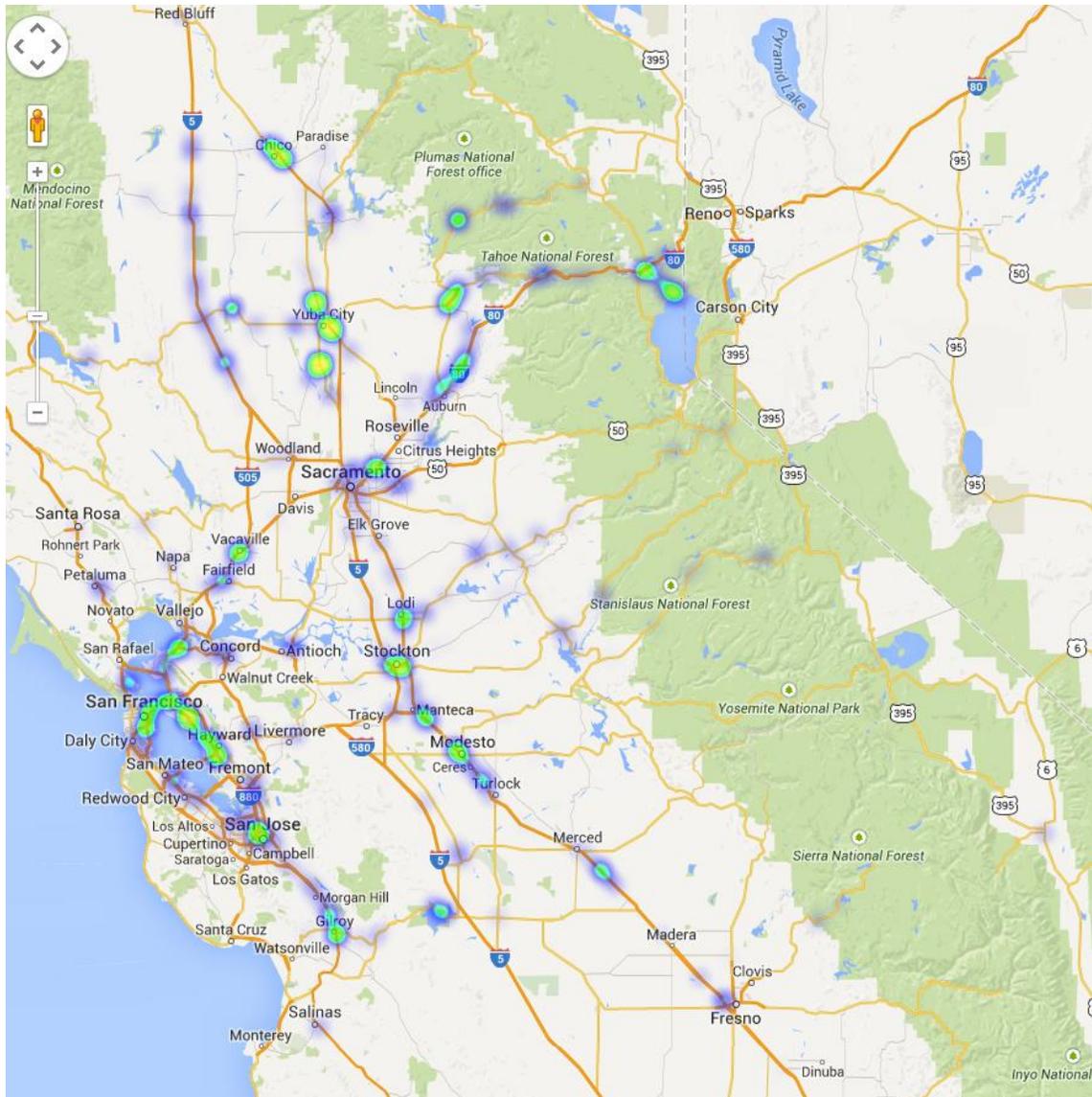
Figure D.5 – Idling location heat map for April 2013
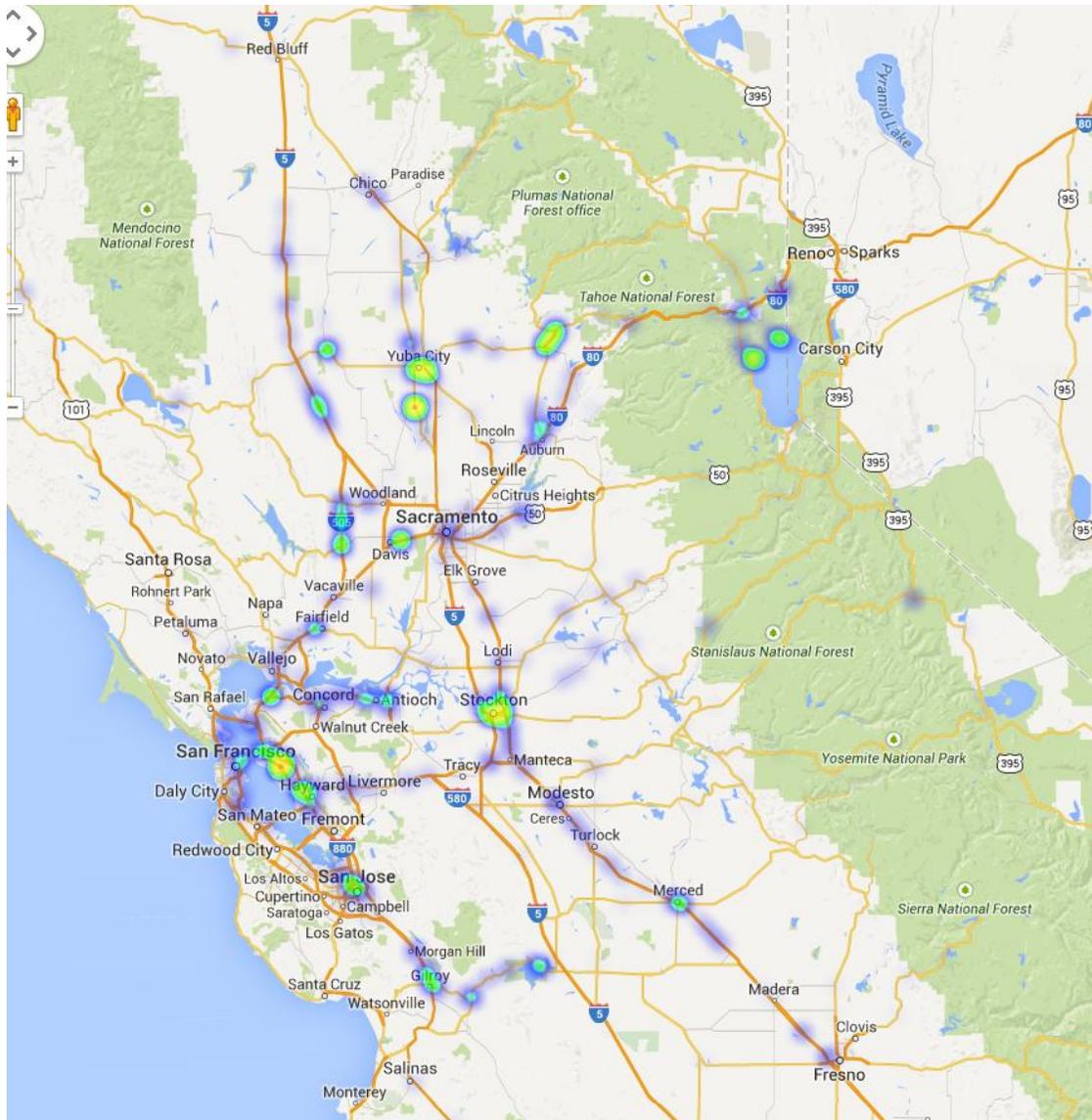
Figure D.6 – Idling location heat map for May 2013

Figure D.7 – Idling location heat map for June 2013
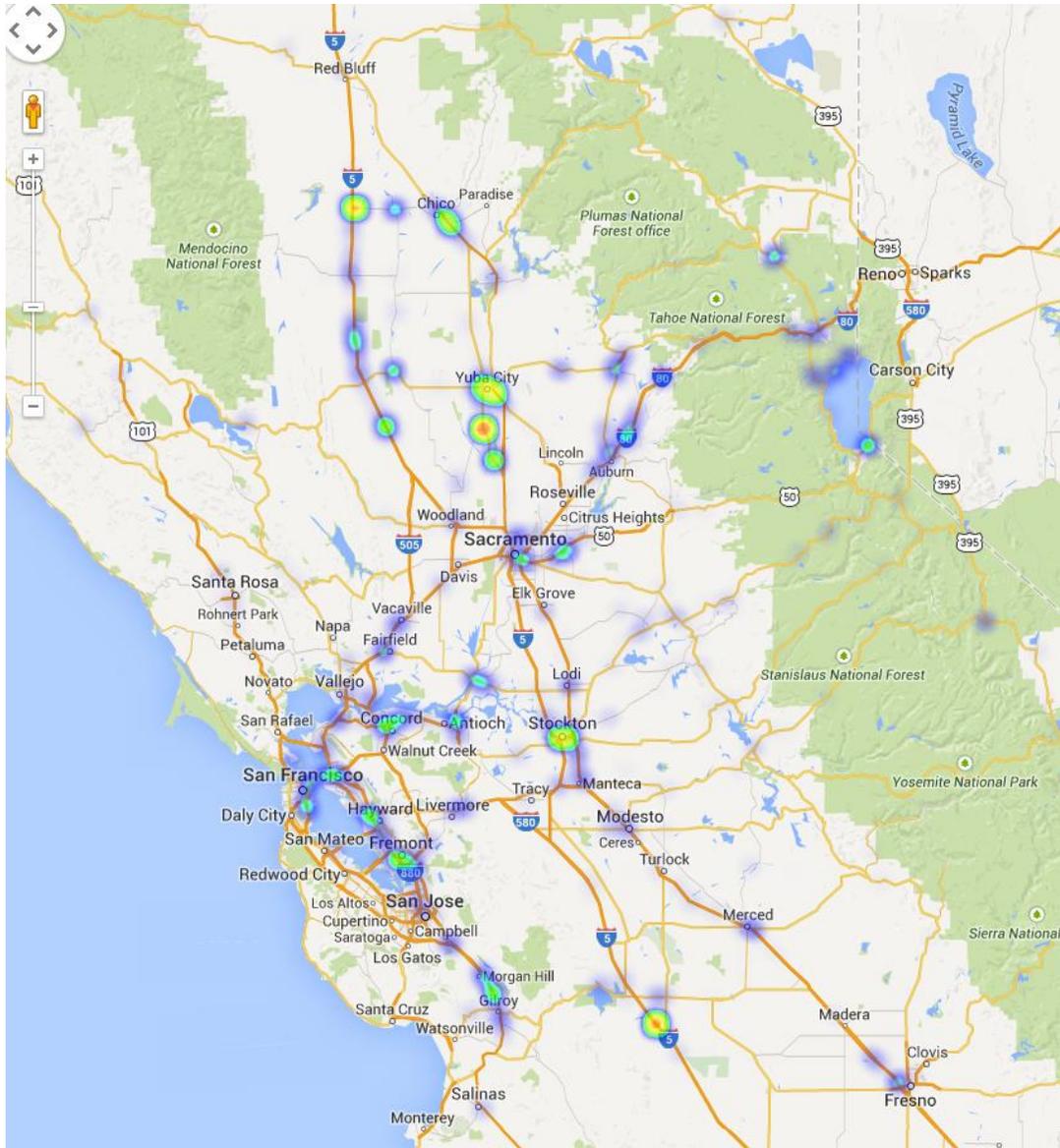
Figure D.8 – Idling location heat map for July 2013

65
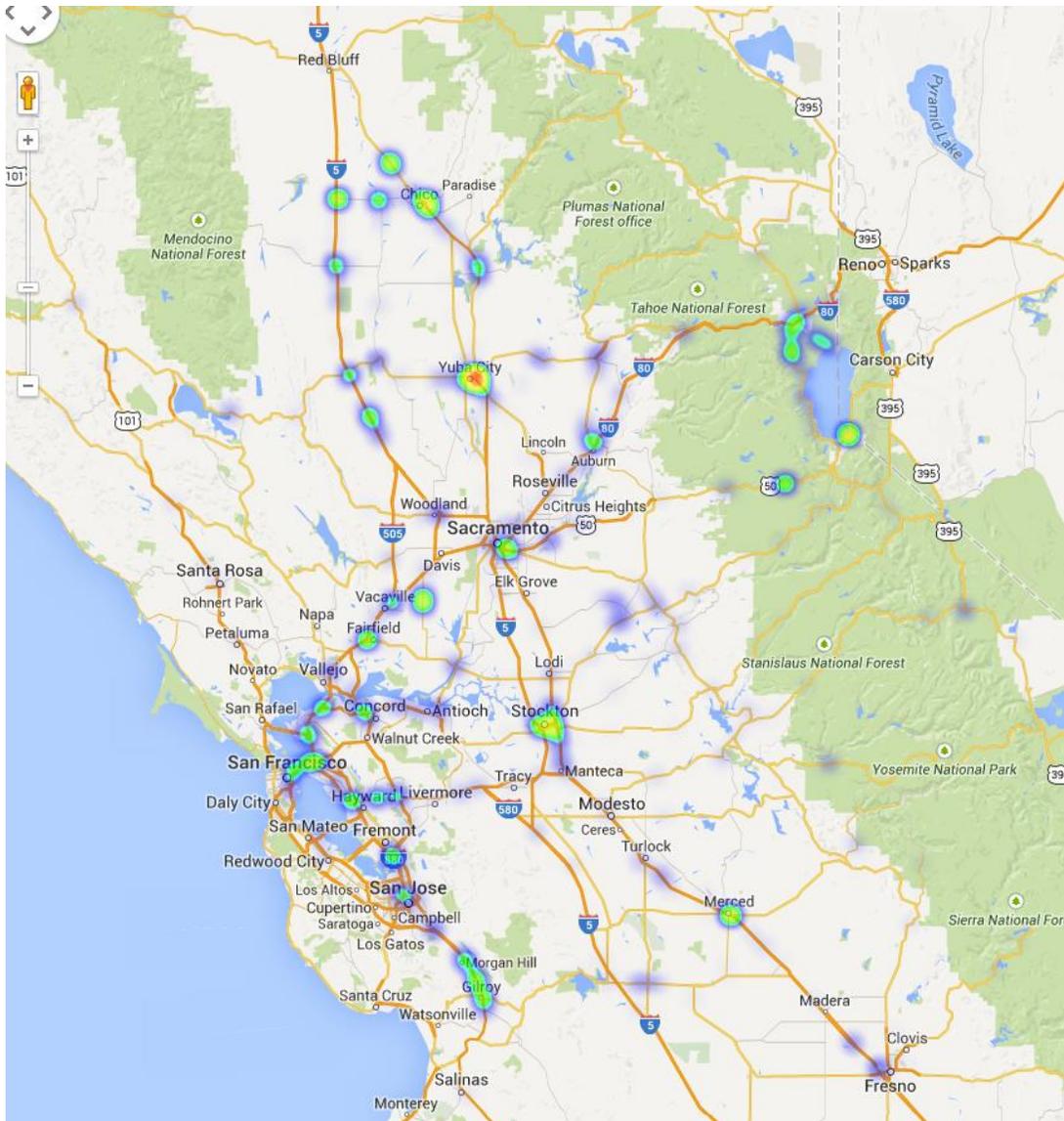
Figure D.9 – Idling location heat map for August 2013
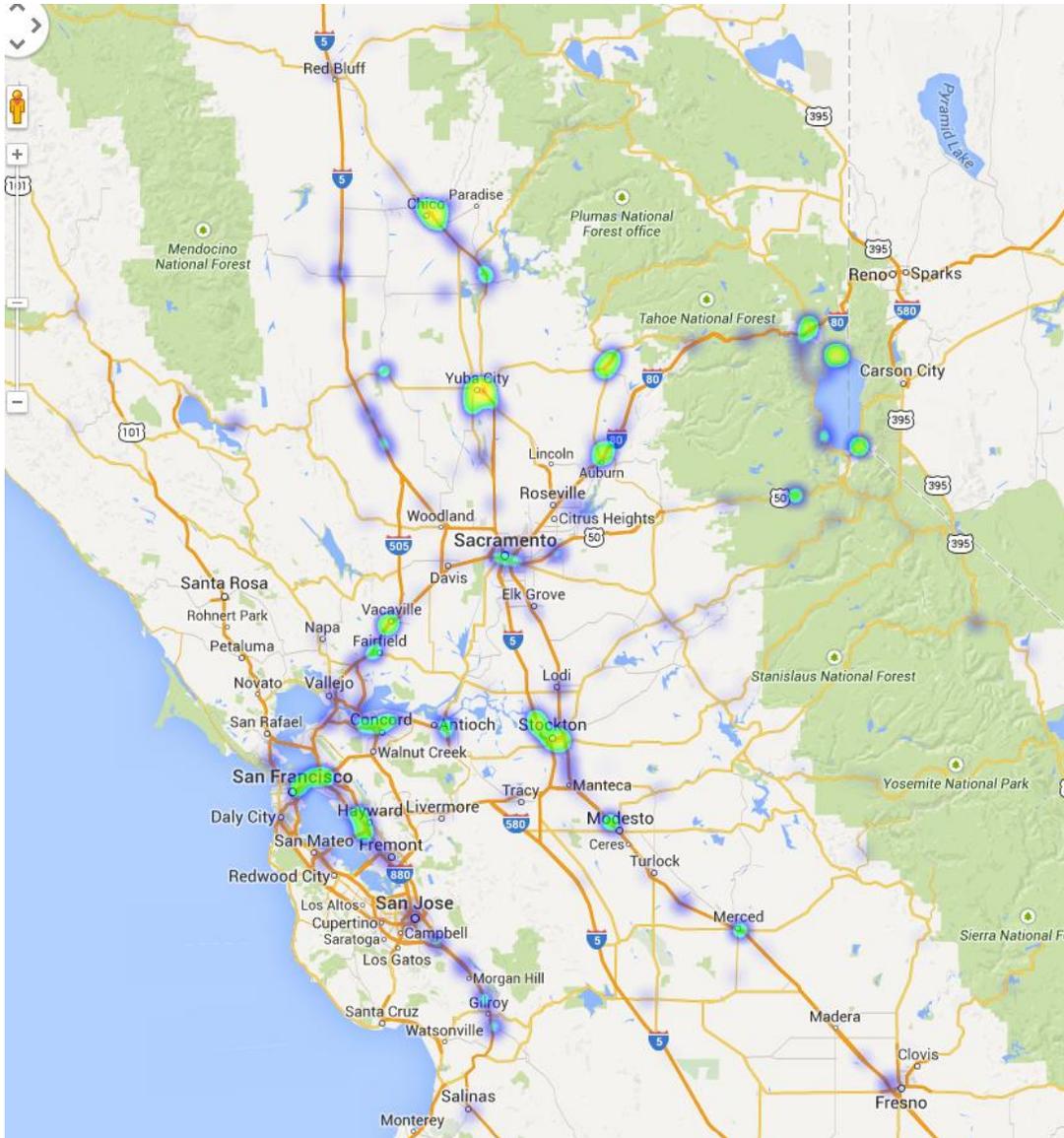
Figure D.10 – Idling location heat map for September 2013

67
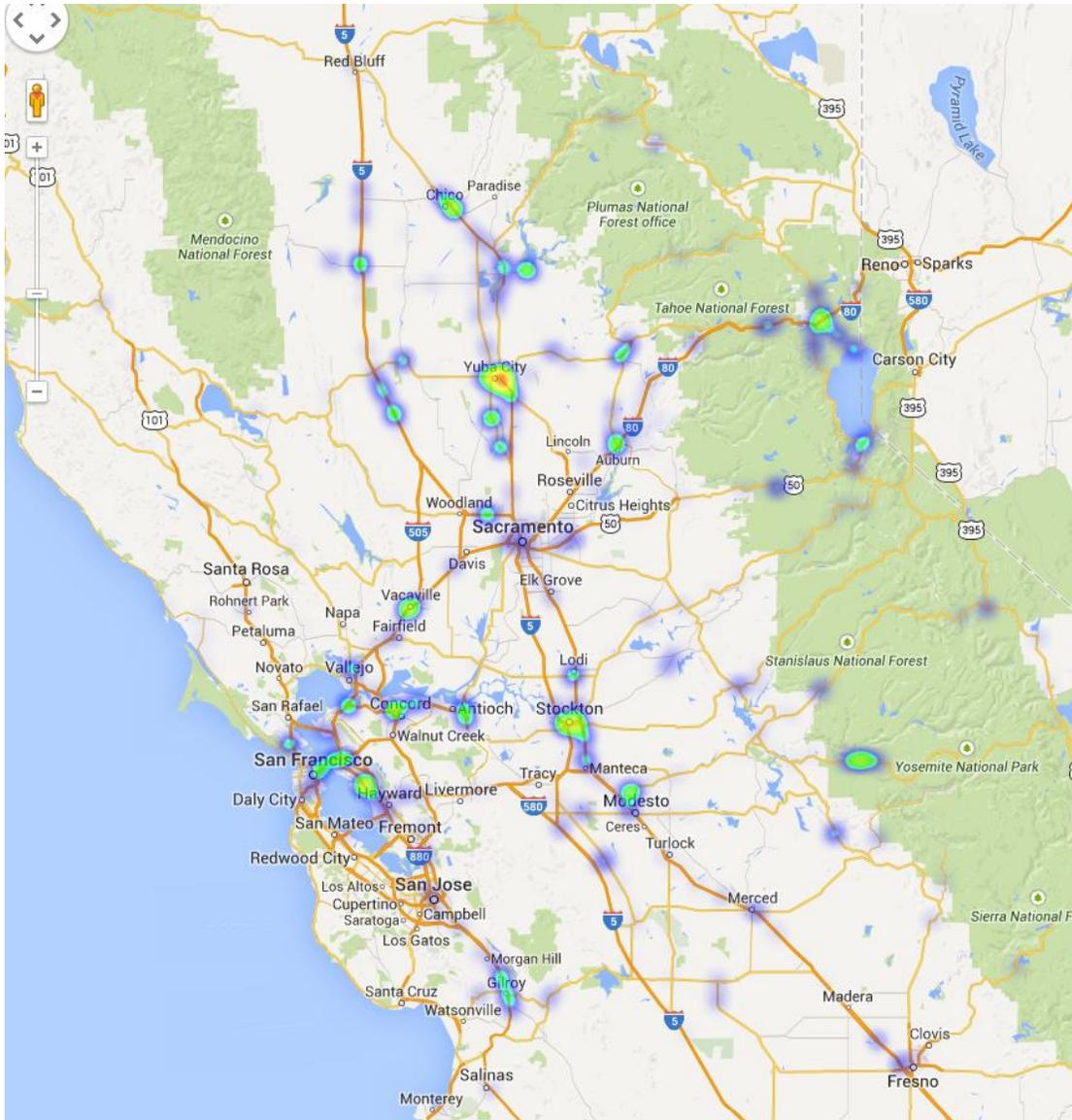
Figure D.11 – Idling location heat map for October 2013
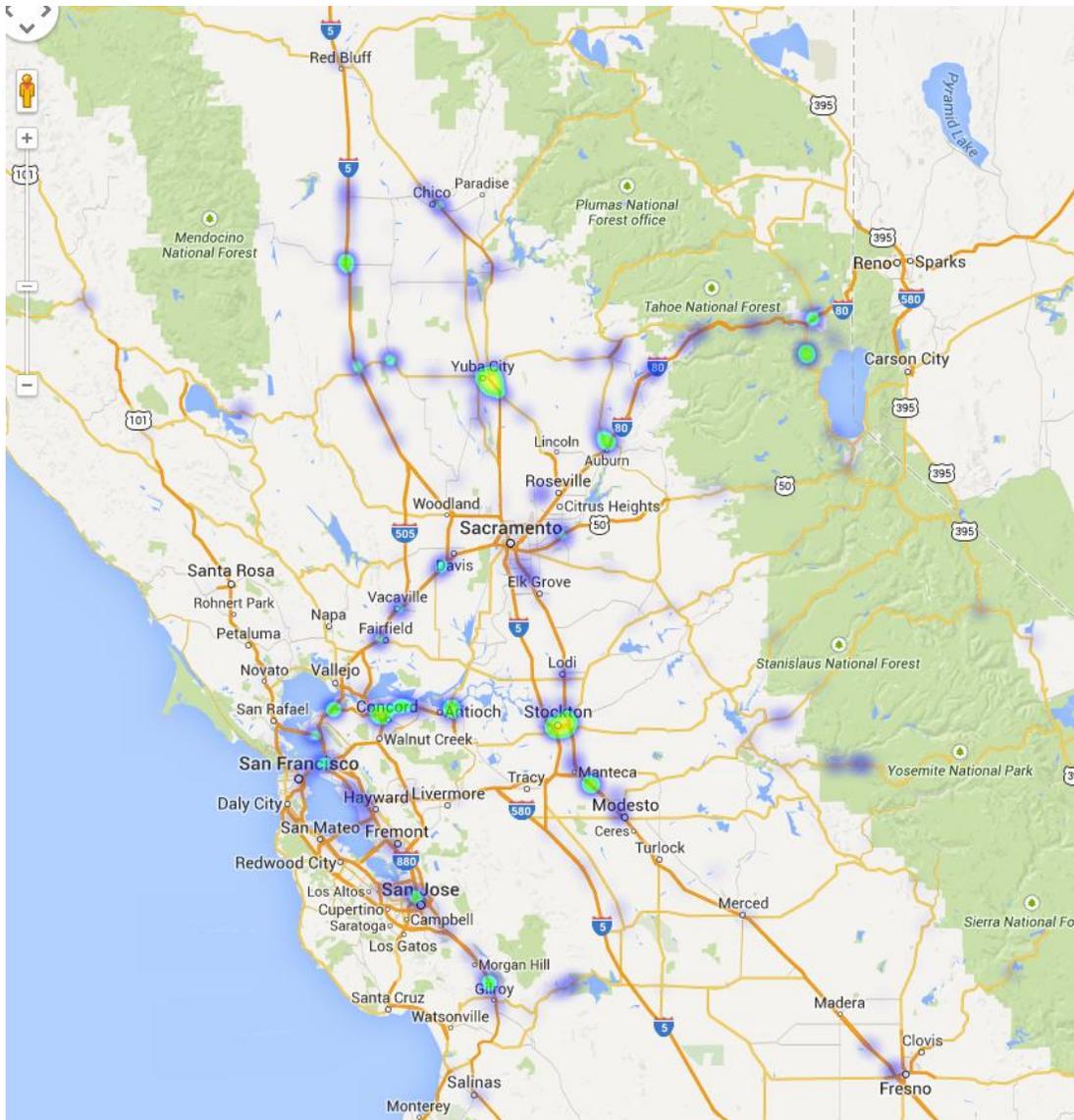
Figure D.12 – Idling location heat map for November 2013
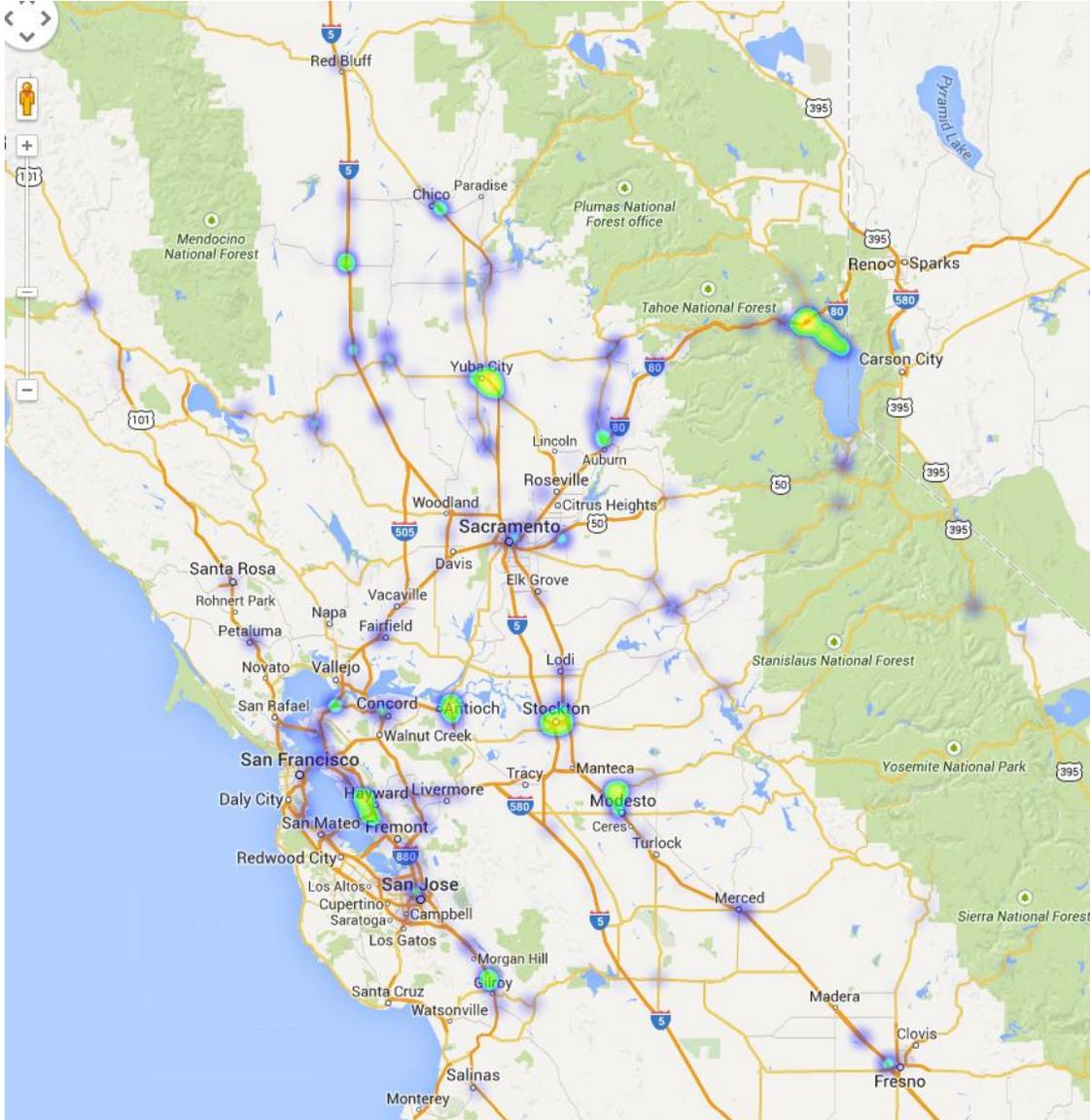
Figure D.13 – Idling location heat map for December 2013
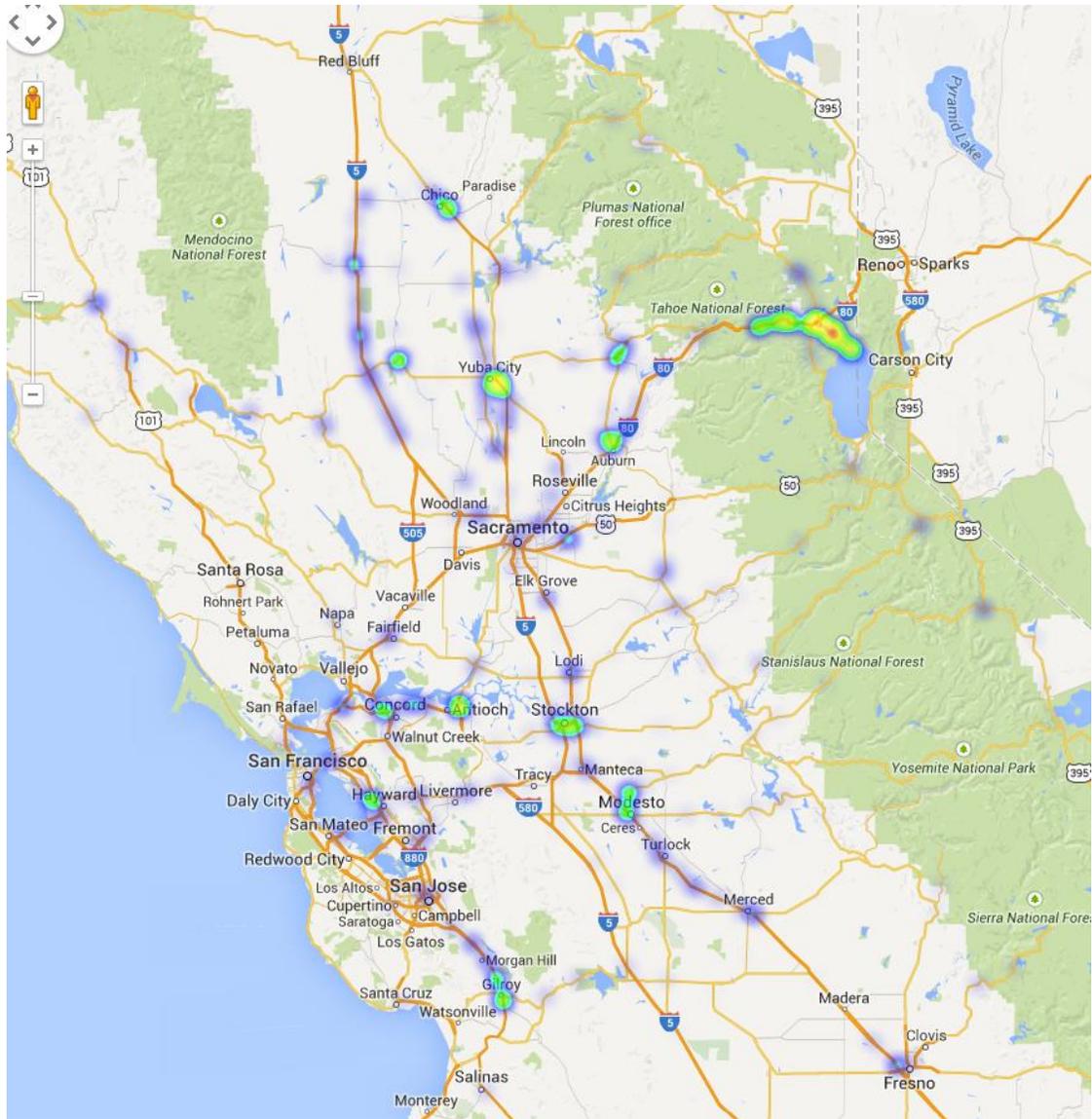
Figure D.14 – Idling location heat map for January 2014
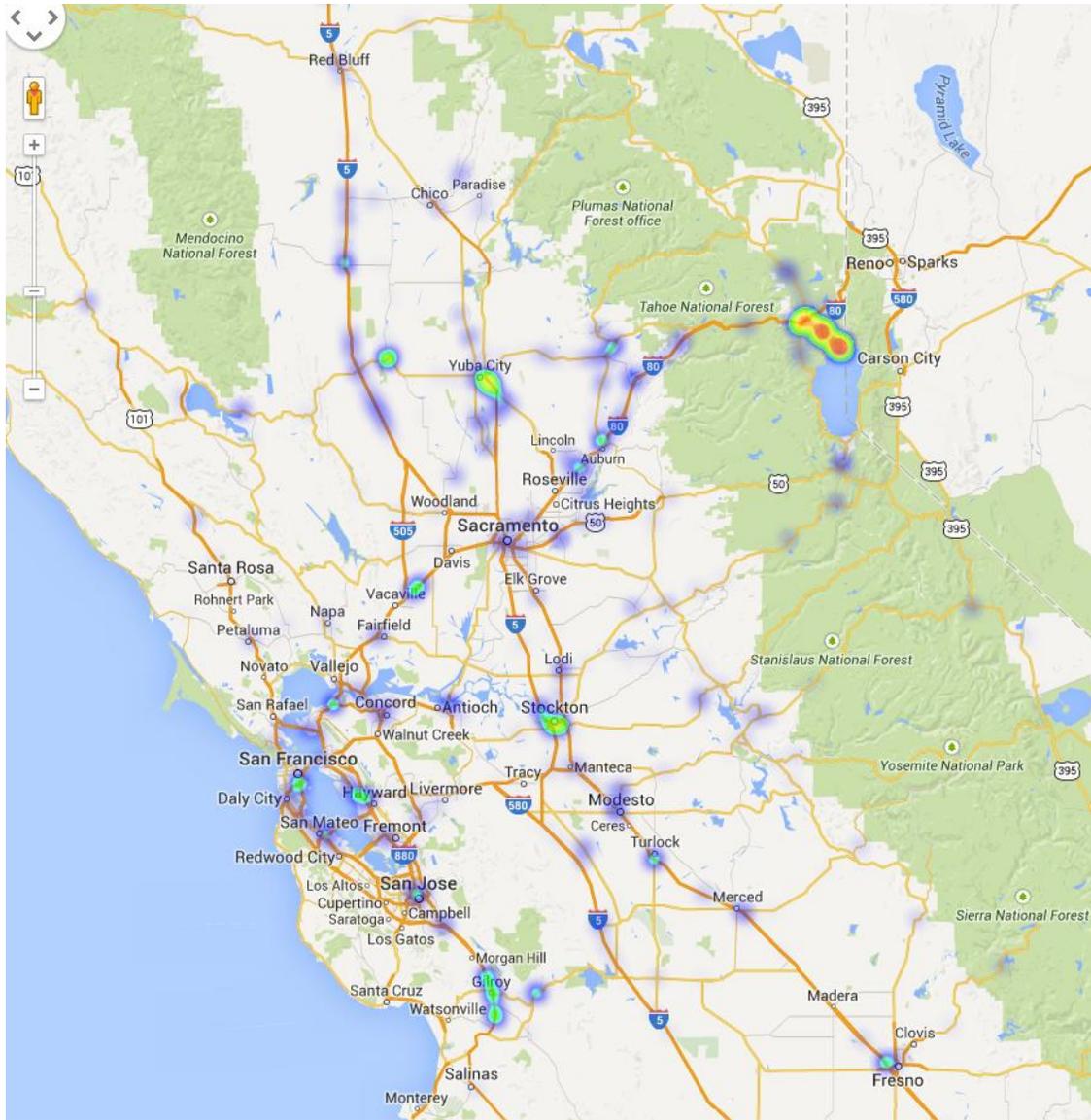
Figure D.15 – Idling location heat map for February 2014
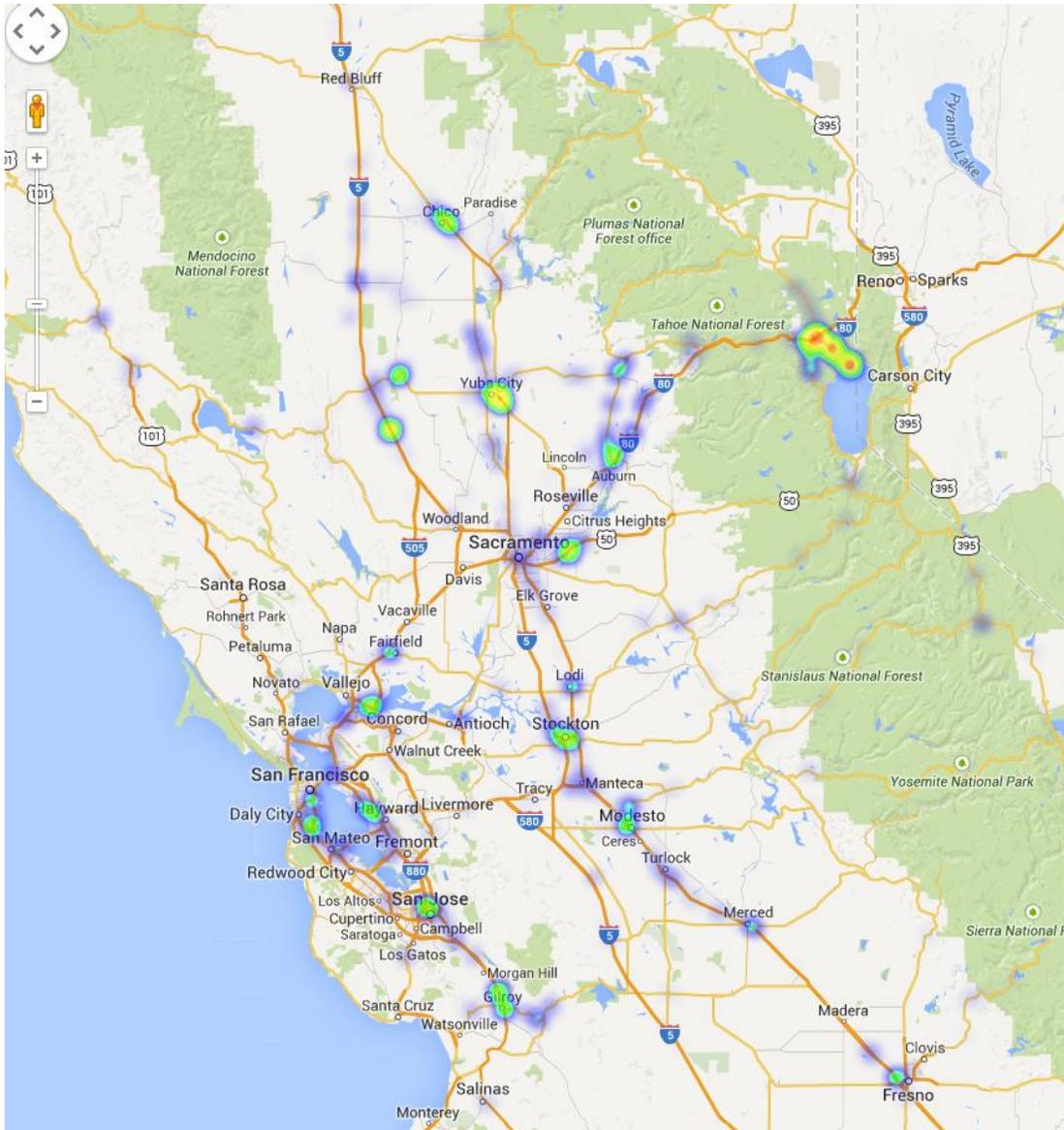
Figure D.16 – Idling location heat map for March 2014
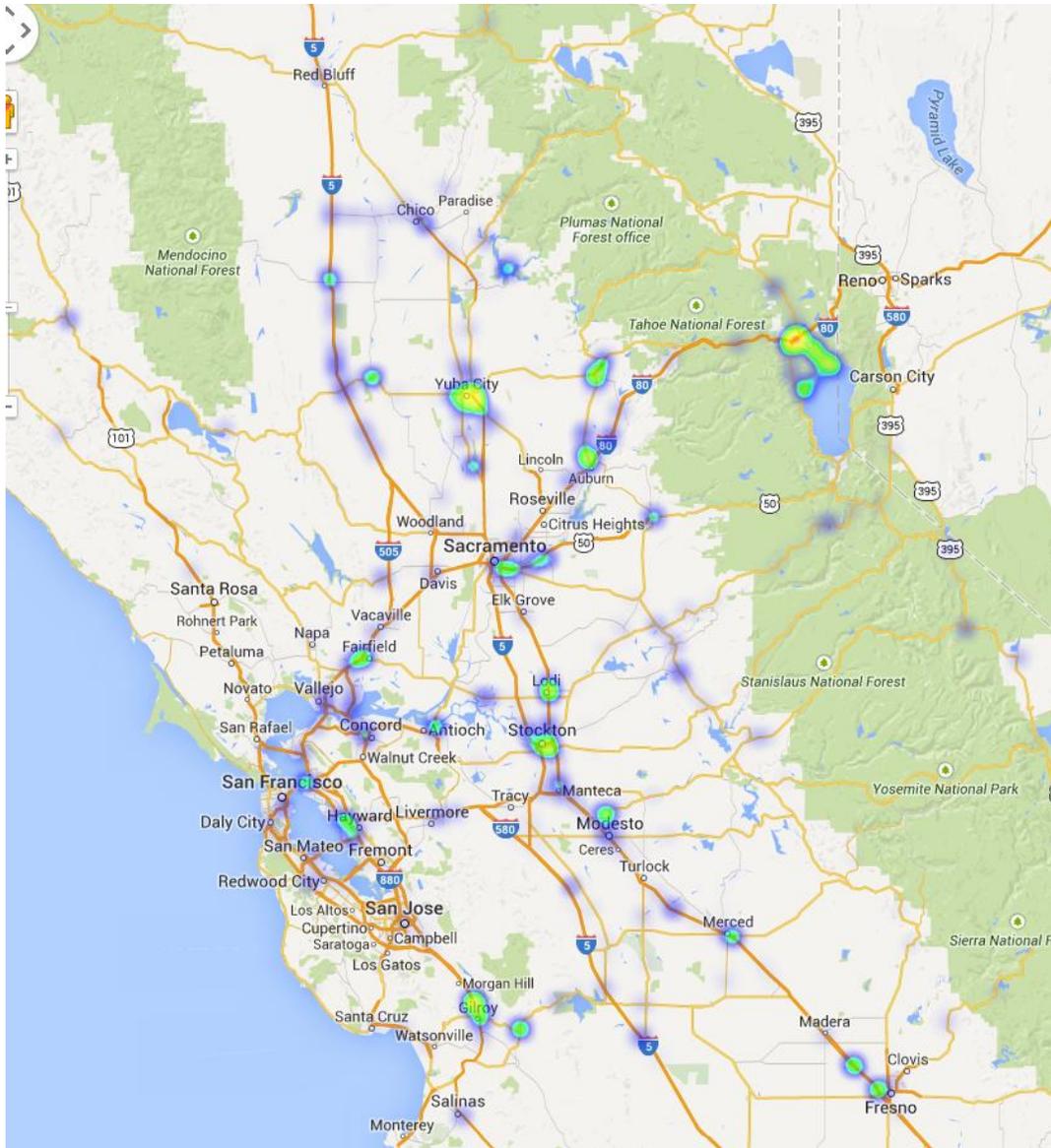
Figure D.17 – Idling location heat map for April 2014

74
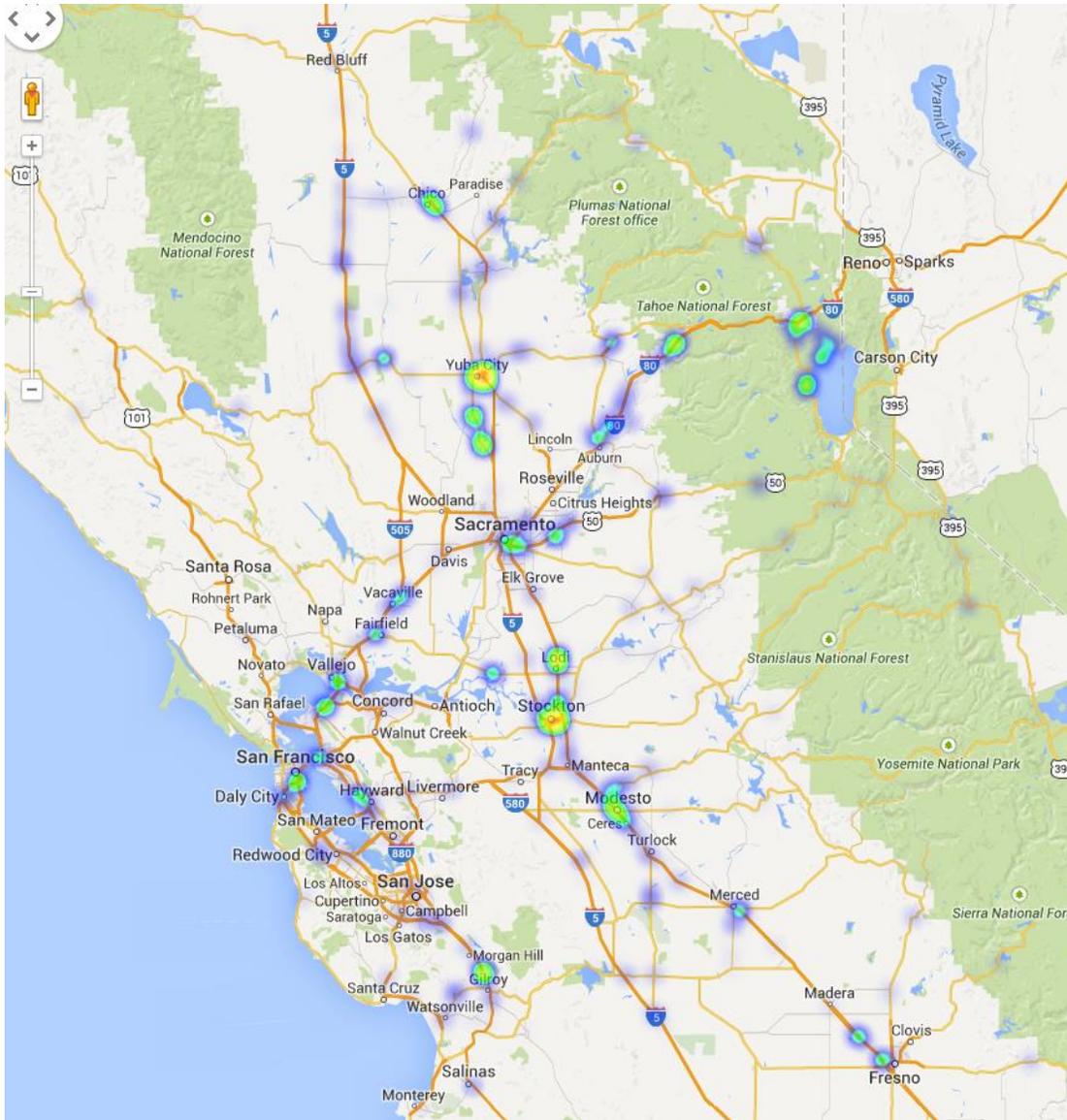
Figure D.18 – Idling location heat map for May 2014

Idling cluster maps also help to identify locations where a significant number of idling events have occurred. Idling cluster maps for one month, January 2014, are provided in Figures D.19 – D.33. Larger dot size corresponds to longer idling period. The green, yellow, and red "house" icons correspond to some of the Caltrans maintenance stations or office locations. All images are obtained from custom KML files viewed in Google Earth.
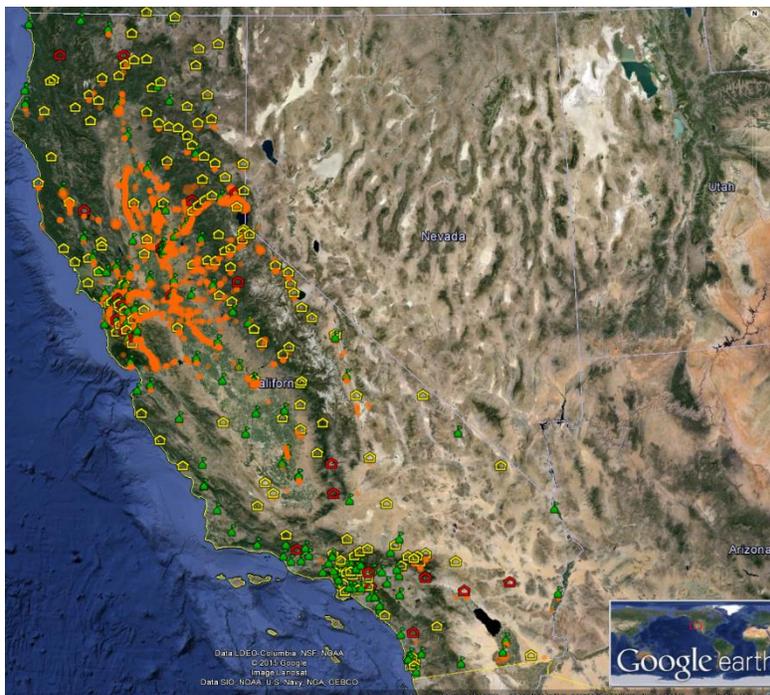
Figure D.19 – Idling location cluster map for January 2014 for all of California, from Google Earth

76
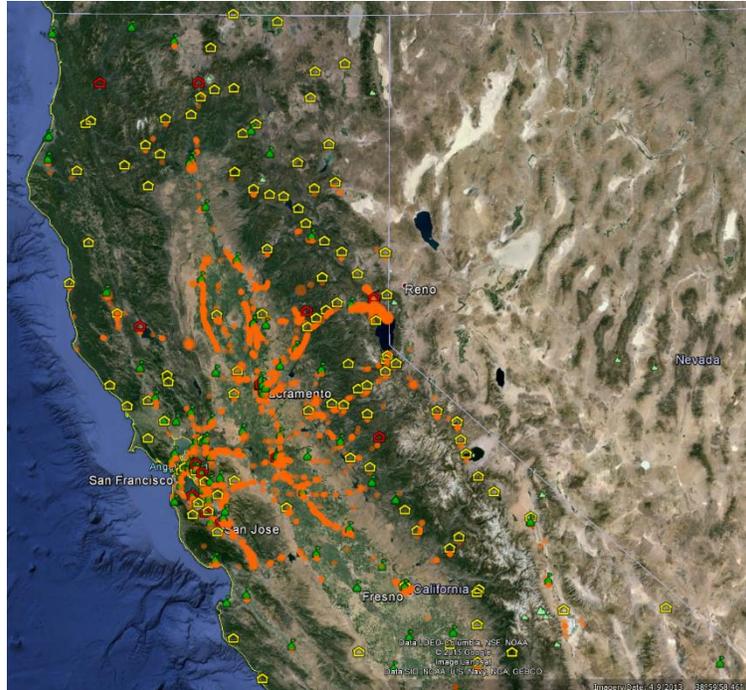
Figure D.20 – Idling location cluster map for January 2014 for northern California, from Google Earth
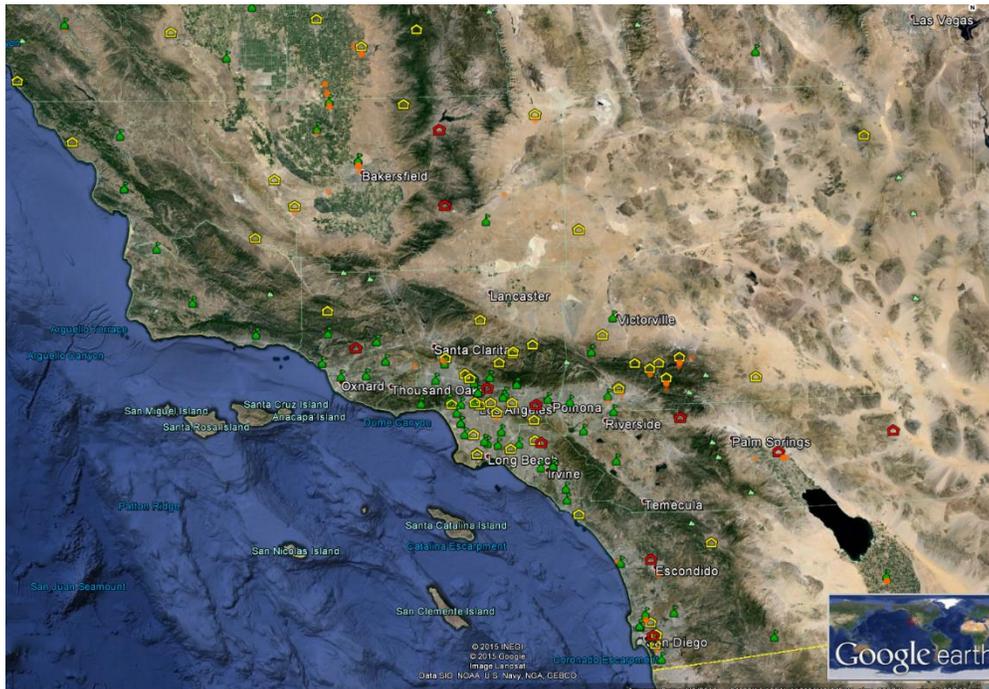
Figure D.21 – Idling location cluster map for January 2014 for southern California, from Google Earth

77
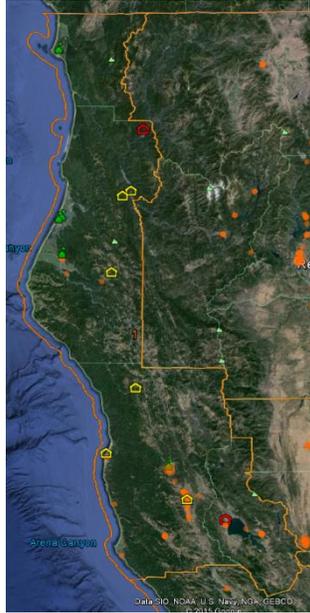
Figure D.22 – Idling location cluster map for January 2014 for Caltrans District 1, from Google Earth
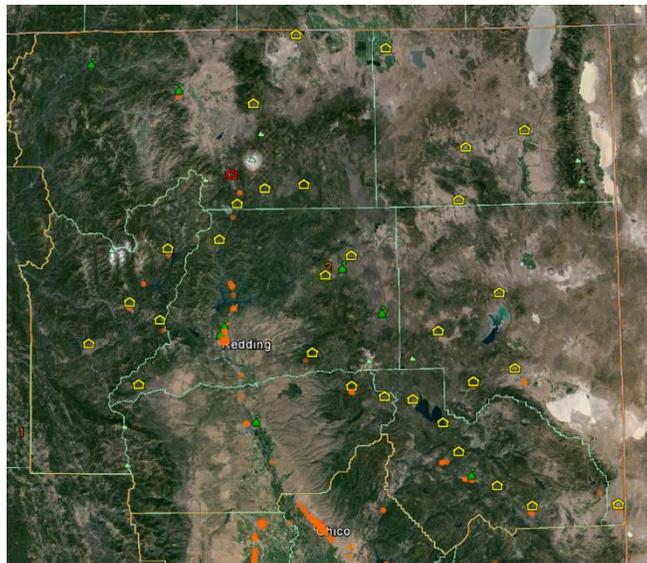
Figure D.23 – Idling location cluster map for January 2014 for Caltrans District 2, from Google Earth

78
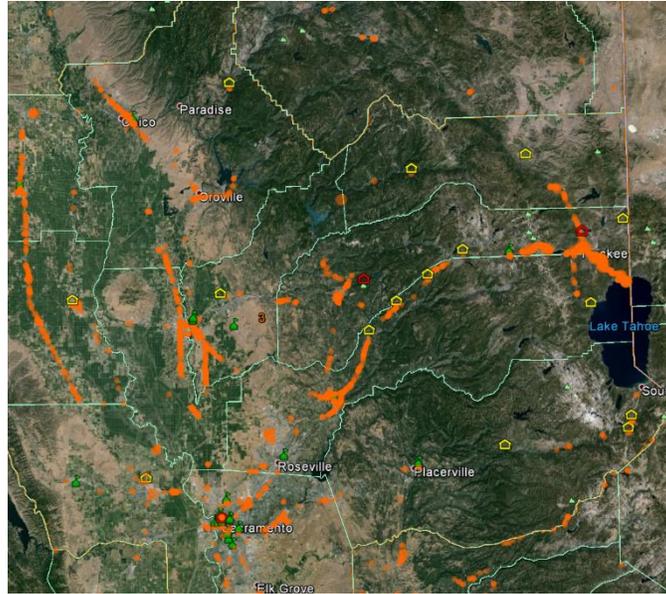
Figure D.24 – Idling location cluster map for January 2014 for Caltrans District 3, from Google Earth
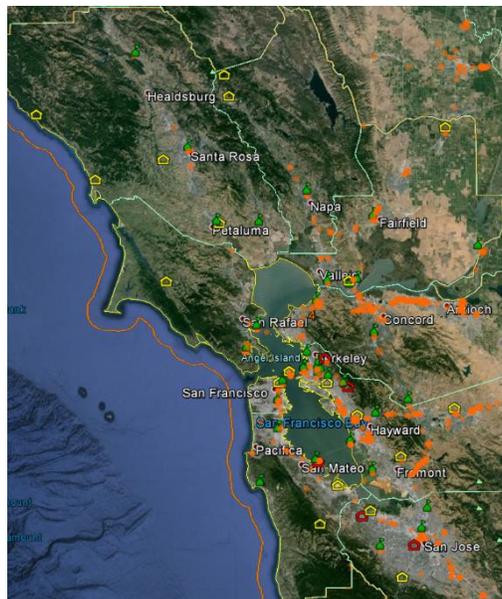
Figure D.25 – Idling location cluster map for January 2014 for Caltrans District 4, from Google Earth

Figure D.26 – Idling location cluster map for January 2014 for Caltrans District 5, from Google Earth

Figure D.27 – Idling location cluster map for January 2014 for Caltrans District 6, from Google Earth
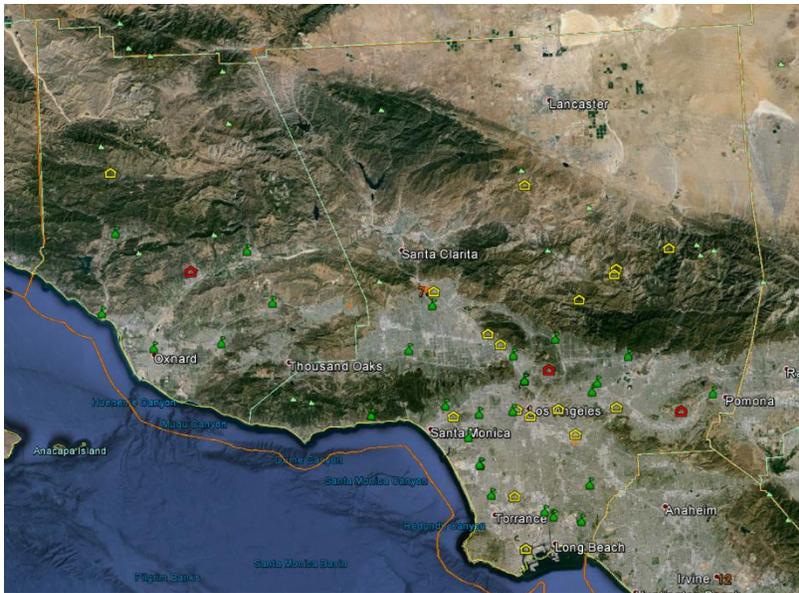
Figure D.28 – Idling location cluster map for January 2014 for Caltrans District 7, from Google Earth
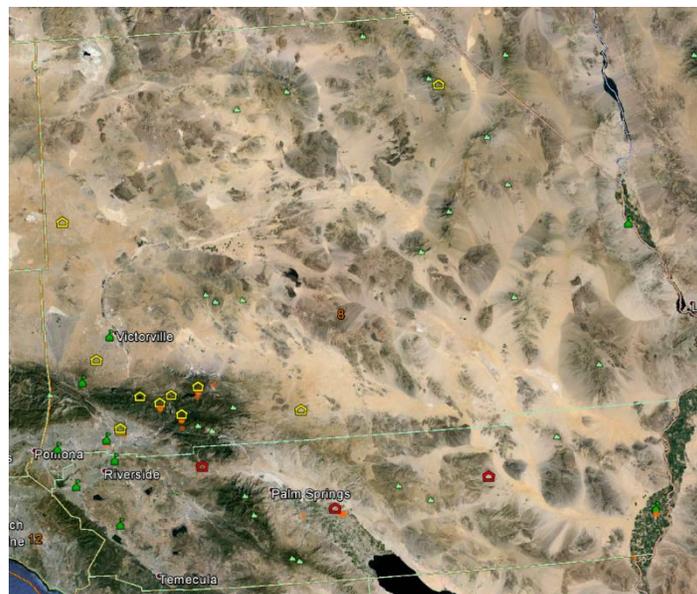
Figure D.29 – Idling location cluster map for January 2014 for Caltrans District 8, from Google Earth

81

Figure D.30 – Idling location cluster map for January 2014 for Caltrans District 9, from Google Earth
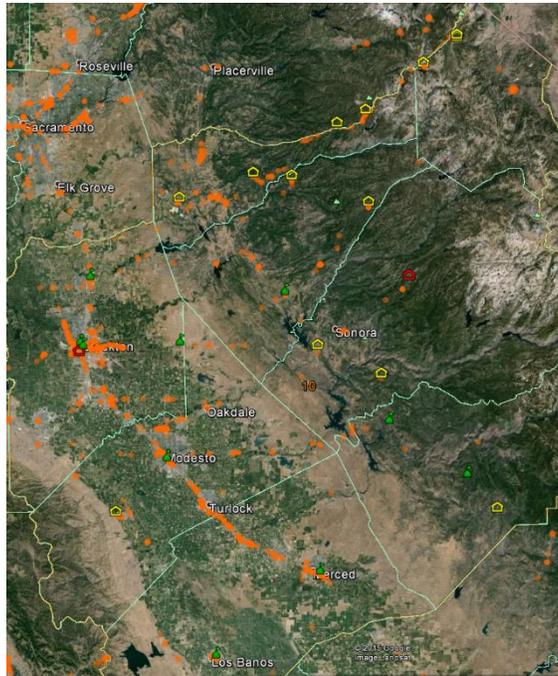
Figure D.31 – Idling location cluster map for January 2014 for Caltrans District 10, from Google Earth

82

Figure D.32 – Idling location cluster map for January 2014 for Caltrans District 11, from Google Earth
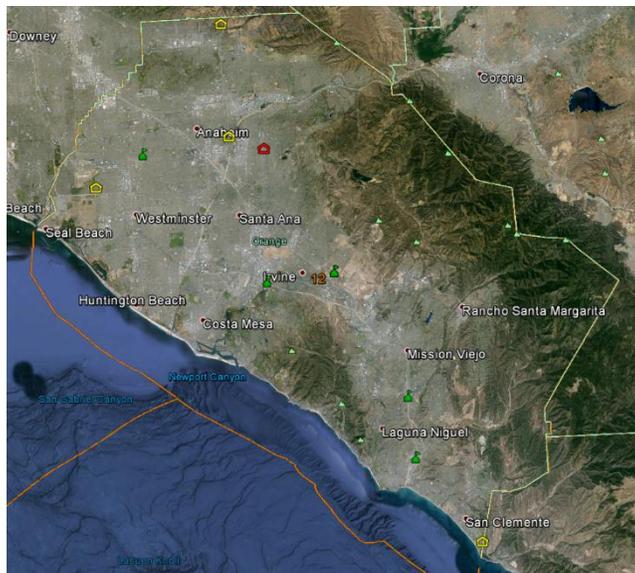
Figure D.33 – Idling location cluster map for January 2014 for Caltrans District 12, from Google Earth

APPENDIX E:
IDLING ANALYSIS CODE

The idling analysis Python code is provided in this appendix. The analysis code uses the *Tkinter* graphical user interface (GUI). *Tkinter* is a thin layer on top of *Tcl/Tk*, and is the default GUI environment for Python, i.e. it is included in the distribution by default.

The idling analysis code uses the *xlrd* Python module to read in the Networkfleet data from Microsoft Excel files, and the *xlwt* Python module to write output to Microsoft Excel files. These modules are not part of the standard python distribution, but can be obtained at http://www.python-excel.org. The code also uses numerous components from the built-in *pylab* module for various analysis, plotting, and visualization tasks.

Note that the Networkfleet data format changed numerous (6) times throughout the pilot study. The code presented in this appendix works for the final format of May 2014. The code presented in Appendices G and H are written in a general fashion, and can handle any of the six Networkfleet input file formats from December 2012 through May 2014.

Listing E.1: Idling analysis code idleAnalysis.py

```python
#!/usr/bin/python
"""

idleAnalysis.py

Idling analysis for Networkfleet Excel data format.

Ty A. Lasky
AHMCT Research Center
University of California, Davis

ver. 1.0: Initial function, 10/31/2012
"""

import xlrd
# see: http://www.python-excel.org/
import xlwt
# see: http://www.python-excel.org/
from pylab import hist, plot, xlabel, ylabel, title, figure, show
from pylab import close, legend, mean, std, median
from Tkinter import *
import Tkconstants
import os
import string

startRow = 15
threshold = 3.98
BINS = 50
outIndex = 0

def getEngineSize():
# Read in all data from Excel file, row at a time.
  for rownum in range(2,193):
    vid = sh2.cell(rownum,0).value
    engineSize[vid] = sh2.cell(rownum,3).value

def getAll():
# Read in all data from Excel file, row at a time.
  for rownum in range(startRow,sh.nrows):
    d = getRow(rownum)
    if d['idle']:
     t = float(d['stopMinutes'])
     if t > threshold:
       if d['vid'] not in idletime:
         idletime[d['vid']] = []
```

85

```
      # Append this time to both 'All' and particular vehicle
      idletime['All'].append(float(d['stopMinutes']))
      idletime[d['vid']].append(float(d['stopMinutes']))

def printRow(d):
  print "vid:",d['vid'],"idle:",d['idle'],"lat:", d['latitude']

def getRow(rownum):
# Get a single row of data using xlrd module. Return it as a dictionary
  global outIndex

  d = {'vid': sh.cell(rownum,1).value}

  s = sh.cell(rownum,8).value
  d['stopDur'] = s.replace("{Idle Stop}","")
  if "{Idle Stop}" in s or "(Idle Stop)" in s:
    d['idle'] = True
  else:
    d['idle'] = False

# up to March 2014, stopMinutes was col 10 (11th col), and was in minutes.
# April 2014, moved to col 18 (19th col), with units of hours

  stopHours = sh.cell(rownum,18).value
  # Handle blank cells in data:
  if stopHours == "":
    stopHours = "0"
  d['stopMinutes'] = float(stopHours) * 60
  d['latitude'] = sh.cell(rownum,13).value
  d['longitude'] = sh.cell(rownum,14).value
  if d['idle']:
    outIndex+=1
    sh3.write(outIndex,0,d['vid'])
    sh3.write(outIndex,1,d['latitude'])
    sh3.write(outIndex,2,d['longitude'])
    sh3.write(outIndex,3,d['stopMinutes'])
    if d['vid'] in engineSize:
      sh3.write(outIndex,4,engineSize[d['vid']])
      print d['vid'], d['latitude'], d['longitude'], d['stopMinutes'], engineSize[d['vid']]
    else:
      sh3.write(outIndex,4,'N/A')
      print d['vid'], d['latitude'], d['longitude'], d['stopMinutes'], "N/A"

  return d

def calcStats():
```

86

```
# Simple stats for entire data set.
  tsum = 0
  for t in idletime['All']:
    tsum += t
  print "Total idle time:",tsum,"minutes"
  print "Total",int(tsum / 60), "hours,",int(tsum%60), "minutes"
  print "Mean:",mean(idletime['All']),"minutes"
  print "Median:",median(idletime['All']),"minutes"
  print "St. Dev.:",std(idletime['All']),"minutes"

def doPlot(keysort,var):
# Plot number of events vs. idle time for selected vehicles.
  close('all')
  prob = {}
  bins = {}
  patches = {}
  for key in keysort:
    if var[key].get():
      prob[key], bins[key], patches[key] = hist(idletime[key],50)
  ylabel('Number of events')
  xlabel('Idle time (min)')
  title('Idle time histogram')
  figure()
  leg = []
  for key in prob:
    plot(bins[key][:-1:],prob[key],lw=3)
    leg.append(key)
  ylabel('Number of events')
  xlabel('Idle time (min)')
  title('Idle time')
  legend(leg)
  close(1)
  show()

def doHistogram():
# Histogram for entire data set.
  prob = {}
  bins = {}
  patches = {}
  figure()
  prob['All'], bins['All'], patches['All'] = hist(idletime['All'],BINS)
  ylabel('Number of events')
  xlabel('Idle time (min)')
  title('Idle time histogram, all vehicles')
  show()
```

```python
if __name__ == "__main__":

    inname = '../data/Stop and Idle Time - Idle 2014-06-02.xls'
    in2 = '../data/idling.xls'
    outHeat = '../data/idleHeat.xls'
    string.replace(inname,"/",os.sep)
    string.replace(in2,"/",os.sep)
    string.replace(outHeat,"/",os.sep)
    print "Opening",inname
    wb = xlrd.open_workbook(inname)
    wb2 = xlrd.open_workbook(in2)
    wb3 = xlwt.Workbook()

# Get first sheet
    sh = wb.sheet_by_index(0)
    sh2 = wb2.sheet_by_name(u'idling_summary')
    sh3 = wb3.add_sheet('idleHeat')
    sh3.write(0,0,'vehicle')
    sh3.write(0,1,'latitude')
    sh3.write(0,2,'longitude')
    sh3.write(0,3,'idleTime')
    sh3.write(0,4,'enginieSize')

    idletime = {}
    idletime['All'] = []
    print "Getting data"
    engineSize = {}
    getEngineSize()
    getAll()

    wb3.save(outHeat)

    keysort = []
    for key in idletime:
        keysort.append(key)
    keysort.sort()

    master = Tk()
    # options for buttons
    button_opt = {'fill': Tkconstants.BOTH, 'padx': 5, 'pady': 5}
    Button(master, text='Plot Data',command=lambda: doPlot(keysort,var)).pack()
    vsb = Scrollbar(orient="vertical")
    text = Text(master, width=20, height=40, yscrollcommand=vsb.set)
    vsb.config(command=text.yview)
    vsb.pack(side="right",fill="y")
    text.pack(side="top",fill="both",expand=True)
```

```
var = {}
c = {}
for key in keysort:
  var[key] = IntVar()
  c[key] = Checkbutton(master, text=key, variable=var[key])
  text.window_create("end", window=c[key])
  text.insert("end", "\n") # to force one checkbox per line

master.mainloop()

print "\n\n"
calcStats()
doHistogram()
```

APPENDIX F:
SPEEDING ANALYSIS CODE

The speeding analysis Python code is provided in this appendix.

The speeding analysis code uses the *xlrd* Python module to read in the Networkfleet data from Microsoft Excel files. This module is not part of the standard python distribution, but can be obtained at http://www.python-excel.org. The code also uses numerous components from the built-in *pylab* module for various analysis, plotting, and visualization tasks.

Note that the Networkfleet data format changed numerous (6) times throughout the pilot study. The code presented in this appendix works for the final format of May 2014. The code presented in Appendices G and H are written in a general fashion, and can handle any of the six Networkfleet input file formats from December 2012 through May 2014.

90

Listing F.1: Speeding analysis code speedAnalysis.py

```python
#!/usr/bin/python
"""

speedAnalysis.py

Speed analysis for Networkfleet Excel data format.

Ty A. Lasky
AHMCT Research Center
University of California, Davis

ver. 1.0: Initial function, 12/11/2012
"""


import xlrd
# see: http://www.python-excel.org/
import os
import string
from pylab import hist, plot, xlabel, ylabel, title, figure, show, close, legend, mean, std, median

startRow = 13
previousVID = 0
violations = []
durations = []
deltaSpeed = []
BINS = 50

def getAll():
  for rownum in range(startRow,sh.nrows):
    d = getRow(rownum)

def printRow(d):
  if 'numVio' in d:
    print "vid:",d['vid'],"numVio:",d['numVio']
  else:
    print
"vid:",d['vid'],"vioTime:",d['vioTime'],"vioDurn:",d['vioDurn'],'speed:',d['speed'],'posted:',d['postedSpeed'],'vioDist:',d['vioDist']

def getRow(rownum):
# Return row as dictionary
  global previousVID
  d = {}
  v = sh.cell(rownum,0).value
  if v != '':
```

```
    d['vid'] = v
    d['numVio'] = sh.cell(rownum,4).value
    violations.append([d['vid'],int(d['numVio'].encode('ascii','ignore'))])
    previousVID = v
  else:
    d['vid'] = previousVID
  d['vioTime'] = sh.cell(rownum,5).value
  d['vioDurn'] = sh.cell(rownum,6).value
  d['speed'] = sh.cell(rownum,7).value
  d['postedSpeed'] = sh.cell(rownum,8).value
  d['vioDist'] = sh.cell(rownum,9).value
  d['durnSeconds'] = durationToSeconds(d['vioDurn'])
  durations.append([d['vid'],d['durnSeconds']])
  if d['postedSpeed'] != "":
    d['deltaSpeed'] = int(d['speed'].encode('ascii','ignore')) -
int(d['postedSpeed'].encode('ascii','ignore'))
  else:
    # Assume 65 mph posted speed
    d['deltaSpeed'] = int(d['speed'].encode('ascii','ignore')) - 65
  deltaSpeed.append([d['vid'],d['deltaSpeed']])
  return d

def durationToSeconds(durUnicode):
  s = durUnicode.encode('ascii','ignore')
  hr = 0
  i = s.find('h')
  if i != -1:
    hr = int(s[0:i])
  min = 0
  i = s.find('m')
  if i != -1:
    if hr > 0:
      min = int(s[i-2:i])
    else:
      min = int(s[0:i])
  sec = 0
  i = s.find('s')
  if i != -1:
    sec = int(s[i-2:i])
  dur = hr*3600 + min*60 + sec
  return dur

def doDeltaSpeedHistogram():
# Histogram for entire data set.
  prob = {}
  bins = {}
```

92

```
  patches = {}
  # ref: http://muffinresearch.co.uk/archives/2007/10/16/python-transposing-lists-with-map-and-
zip/
  x = zip(*deltaSpeed)[1]
  figure()
  prob['All'], bins['All'], patches['All'] = hist(x,BINS)
  ylabel('Number of events')
  xlabel('Delta Speed (mph)')
  title('Speed - Posted Speed histogram, all vehicles')
  show()

def doDurationHistogram():
# Histogram for entire data set.
  prob = {}
  bins = {}
  patches = {}
  # ref: http://muffinresearch.co.uk/archives/2007/10/16/python-transposing-lists-with-map-and-
zip/
  x = zip(*durations)[1]
  figure()
  prob['All'], bins['All'], patches['All'] = hist(x,BINS)
  ylabel('Number of events')
  xlabel('Duration of violation (sec)')
  title('Duration histogram, all vehicles')
  show()

def doVioNumberHistogram():
# Histogram for entire data set.
  prob = {}
  bins = {}
  patches = {}
  # ref: http://muffinresearch.co.uk/archives/2007/10/16/python-transposing-lists-with-map-and-
zip/
  # Transpose the list (*), and get second row [1].
  x = zip(*violations)[1]
  figure()
  prob['All'], bins['All'], patches['All'] = hist(x,BINS)
  ylabel('Number of events')
  xlabel('Number violations')
  title('Violation histogram, all vehicles')
  show()

def calcStatsVioNumber():
# Simple stats for entire data set.
  x = zip(*violations)[1]
  print "Total Violations:",sum(x)
```

93

```
  print "Mean:",mean(x)
  print "Median:",median(x)
  print "St. Dev.:",std(x)

def calcStatsDeltaSpeed():
# Simple stats for entire data set.
  x = zip(*deltaSpeed)[1]
  print "deltaSpeed = Speed - Posted Speed"
  print "Total:",sum(x),"mph"
  print "Mean:",mean(x),"mph"
  print "Median:",median(x),"mph"
  print "St. Dev.:",std(x),"mph"

def calcStatsDuration():
# Simple stats for entire data set.
  x = zip(*durations)[1]
  print "Total duration:",sum(x),"seconds"
  print "Total",int(sum(x) / 60), "minutes,",int(sum(x)%60), "seconds"
  print "Mean:",mean(x),"seconds"
  print "Median:",median(x),"seconds"
  print "St. Dev.:",std(x),"seconds"

if __name__ == "__main__":
  inname = '../data/Speed violation data 2014-06-02.xls'
  string.replace(inname,"/",os.sep)
  print "Opening",inname
  wb = xlrd.open_workbook(inname)

# Get first sheet
  sh = wb.sheet_by_index(0)

  print "Getting data"
  getAll()
  calcStatsVioNumber()
  doVioNumberHistogram()
  print "\n"
  calcStatsDuration()
  doDurationHistogram()
  print "\n"
  calcStatsDeltaSpeed()
  doDeltaSpeedHistogram()
  print "\n"
```

94

APPENDIX G:
IDLING DATA TO CSV CODE

Python code for a utility to convert Networkfleet idling data (Excel format) into CSV is presented in this appendix.

This utility uses the *xlrd* Python module to read in the Networkfleet data from Microsoft Excel files. This module is not part of the standard python distribution, but can be obtained at http://www.python-excel.org. The code also uses the *csv* Python module to support CSV output.

Note that the Networkfleet data format changed numerous (6) times throughout the pilot study. The code presented in this appendix works for all six Networkfleet input file formats from December 2012 through May 2014. It does so by way of a list of six parameter value settings contained in the *parm* variable. The approach used herein could be easily ported to the idling and speeding analysis code of Appendices E and F.

95

Listing G.1: idleToCsv.py code listing for utility to convert Networkfleet input data into CSV output

```python
#!/usr/bin/python
"""
idleToCsv.py

Read stop and idle report Excel file, parse, output CSV

Vehicle Label, total # of trips, total idle event count, total idle duration (min or second),
Total drive duration or trip duration (min or second), Total Travel Distance

ver. 1.0: Initial function, 2/4/2015
"""

import xlrd
# see: http://www.python-excel.org/
import os
import string
import csv

idleThreshold = 12 * 60 # in minutes
tripThreshold = 12 * 60 # in minutes
timeAddition = 4 # in minutes

inData = (
  ('Stop and Idle Time 2013-01-02',0),
  ('Stop and Idle Time 2013-02-02',1),
  ('Stop and Idle Time 2013-03-02',1),
  ('Stop and Idle Time 2013-04-02',1),
  ('Stop and Idle Time 2013-05-02',1),
  ('Stop and Idle Time 2013-06-02',1),
  ('Stop and Idle Time 2013-07-02',1),
  ('Stop and Idle Time 2013-08-02',1),
  ('Stop and Idle Time 2013-09-02',2),
  ('Stop and Idle Time 2013-10-02',3),
  ('Stop and Idle Time 2013-11-02',3),
  ('Stop and Idle Time 2013-12-02',3),
  ('Stop and Idle Time - Idle 2014-01-02',4),
  ('Stop and Idle Time - Idle 2014-02-02',4),
  ('Stop and Idle Time - Idle 2014-03-02',4),
  ('Stop and Idle Time - Idle 2014-04-02',5),
  ('Stop and Idle Time - Idle 2014-05-02',5),
  ('Stop and Idle Time - Idle 2014-06-02',5),
)
```

```
parm = [
  {'startRow':25, 'vid':1, 'tripDur':5, 'stopDurText':8, 'idleDur':9, 'tripDist':10, 'lat':14, 'lon':15,
'durUnit':'minute'},
  {'startRow':26, 'vid':1, 'tripDur':5, 'stopDurText':8, 'idleDur':9, 'tripDist':10, 'lat':14, 'lon':15,
'durUnit':'minute'},
  {'startRow':26, 'vid':1, 'tripDur':6, 'stopDurText':9, 'idleDur':10, 'tripDist':11, 'lat':15, 'lon':16,
'durUnit':'minute'},
  {'startRow':19, 'vid':1, 'tripDur':6, 'stopDurText':9, 'idleDur':10, 'tripDist':11, 'lat':15, 'lon':16,
'durUnit':'minute'},
  {'startRow':15, 'vid':1, 'tripDur':6, 'stopDurText':9, 'idleDur':10, 'tripDist':11, 'lat':15, 'lon':16,
'durUnit':'minute'},
  {'startRow':15, 'vid':1, 'tripDur':17, 'stopDurText':8, 'idleDur':18, 'tripDist':9, 'lat':13, 'lon':14,
'durUnit':'hour'}
]

# Read in the full vehicle list:
wb = xlrd.open_workbook('vehicle_label.xlsx')
# Get first sheet
sh = wb.sheet_by_index(0)
vehList = []
for rownum in range(sh.nrows):
  vehList.append(str(sh.cell(rownum,0).value).strip())
vehList.sort()

def getAll(startRow, vid, tripDur, stopDurText, idleDur, tripDist, lat, lon, durUnit, filePrefix):
  csvfile = None
  vehData = dict()

  # Read in all data from Excel file, row at a time.
  for rownum in range(startRow,sh.nrows):
    d = getRow(rownum, vid, tripDur, stopDurText, idleDur, tripDist, lat, lon, durUnit)
    if (d['vehId'] in vehList):
      # Check for anomalies:
      if 'tripMinutes' in d:
        tripThresholdTime = d['tripDistance'] *60 / 80 - timeAddition
      if (('idleMinutes' in d and (d['idleMinutes'] > idleThreshold)) or
        (('tripMinutes' in d and d['tripMinutes'] > tripThreshold)) or
        (('tripMinutes' in d and d['tripMinutes'] < tripThresholdTime))):
        # is the file in place?
        if (csvfile is None):
          outname = '../data/out/' + entry[0] + '_anomalies' + '.csv'
          string.replace(inname,"/",os.sep)
          csvfile = open(outname, 'wb')
          csvWriter = csv.writer(csvfile, delimiter=',', quotechar='|',
quoting=csv.QUOTE_MINIMAL)
          header = ['Row', 'Vehicle Label', 'Idle Duration (min)',
```

97

```
         'Trip Duration (min)', 'Travel Distance (mi)', 'Latitude', 'Longitude', 'Trip Threshold
(min)']
        csvWriter.writerow(header)
      csvWriter.writerow([
        rownum+1,
        '="'+d['vehId']+'"',
        d['idleMinutes'] if 'idleMinutes' in d else '',
        d['tripMinutes'] if 'tripMinutes' in d else '',
        d['tripDistance'] if 'tripDistance' in d else '',
        d['lat'],
        d['lon'],
        tripThresholdTime
        ])
      continue
    if not (d['vehId'] in vehData):
      initDict = {'tripDurSum':0, 'tripDistSum':0, 'idleDurSum':0, 'tripCount':0, 'idleCount':0}
      vehData[d['vehId']] = initDict
    if d['idle']:
      vehData[d['vehId']]['idleCount'] += 1
      vehData[d['vehId']]['idleDurSum'] += d['idleMinutes']
    else:
      vehData[d['vehId']]['tripDurSum'] += d['tripMinutes']
      vehData[d['vehId']]['tripDistSum'] += d['tripDistance']
      vehData[d['vehId']]['tripCount'] += 1
  return vehData

def printRow(d):
  print('vid: %s, tripDur: %s, idle: %s, idleDur: %s, dist: %s'%(d['vid'],d['tripDur'],
    d['idle'], d['idleDur'], d['dist']))

def getRow(rownum, vid, tripDur, stopDurText, idleDur, tripDist, lat, lon, durUnit):
# Get a single row of data using xlrd module. Return it as a dictionary

  d = {'vehId': str(sh.cell(rownum,vid).value).strip()}
  s = sh.cell(rownum,stopDurText).value
  if "{Idle Stop}" in s or "(Idle Stop)" in s:
    d['idle'] = True
  else:
    d['idle'] = False

  if d['idle']:
    idleTime = sh.cell(rownum,idleDur).value
  # Handle blank cells in data:
    if idleTime == "":
      idleTime = "0"
    if durUnit == 'hour':
```

98

```
      d['idleMinutes'] = float(idleTime) * 60
    else:
      d['idleMinutes'] = float(idleTime)
  else: # a regular trip
    tripTime = sh.cell(rownum,tripDur).value
  # Handle blank cells in data:
    if tripTime == "":
      tripTime = "0"
    if durUnit == 'hour':
      d['tripMinutes'] = float(tripTime) * 60
    else:
      d['tripMinutes'] = float(tripTime)
    d['tripDistance'] = float(sh.cell(rownum,tripDist).value)
  d['lat'] = sh.cell(rownum,lat).value
  d['lon'] = sh.cell(rownum,lon).value
  return d

if __name__ == "__main__":

  for entry in inData:
    inname = '../data/' + entry[0] + '.xls'
    string.replace(inname,"/",os.sep)

    parmd = parm[entry[1]]

    wb = xlrd.open_workbook(inname)

    # Get first sheet
    sh = wb.sheet_by_index(0)

    print("\nGetting data from %s\n"%inname)
    vehData = getAll(parmd['startRow'], parmd['vid'], parmd['tripDur'],
      parmd['stopDurText'], parmd['idleDur'], parmd['tripDist'],
      parmd['lat'], parmd['lon'], parmd['durUnit'], entry[0])

    # Write out CSV:
    outname = '../data/out/' + entry[0] + '.csv'
    string.replace(outname,"/",os.sep)
    csvfile = open(outname, 'wb')
    csvWriter = csv.writer(csvfile, delimiter=',', quotechar='|', quoting=csv.QUOTE_MINIMAL)
    print('Saving %s\n'%outname)

    # Write header:
    header = ['Vehicle Label', 'Total Trips', 'Total Idle Events', 'Total Idle Duration (min)',
      'Total Trip Duration (min)', 'Total Travel Distance (mi)']
    csvWriter.writerow(header)
```

```
  for veh in vehList:
    if veh in vehData:
      csvWriter.writerow([
        '="'+veh+'"',
        vehData[veh]['tripCount'],
        vehData[veh]['idleCount'],
        vehData[veh]['idleDurSum'],
        vehData[veh]['tripDurSum'],
        vehData[veh]['tripDistSum']
        ])
    else:
      csvWriter.writerow([
        '="'+veh+'"',
        '',
        '',
        '',
        '',
        ''
        ])
```

100

APPENDIX H:
IDLING LOCATION MAPS GENERATION CODE

This appendix contains the code to generate the two forms of idling location maps presented in Chapter 4 and Appendix D.

The first listing provides the Python code for generating the idling cluster maps as a KML file.

The second listing provides the Python code for generating the HTML files to display the idling heat maps. This includes an HTML template listing which is used by the Python program. The resulting HTML code relies on two Javascript files: *heatmap.js* and *heatmap-gmaps.js*, both of which can be found at http://www.patrick-wied.at/static/heatmapjs/

The code for both programs uses the *xlrd* Python module to read in the Networkfleet data from Microsoft Excel files. This module is not part of the standard python distribution, but can be obtained at http://www.python-excel.org. The idling cluster KML generation program uses the *simplekml* module. This module is not part of the standard python distribution, but can be obtained at https://code.google.com/p/simplekml/

Note that the Networkfleet data format changed numerous (6) times throughout the pilot study. The code for the two map generators presented in this appendix works for all six Networkfleet input file formats from December 2012 through May 2014. It does so by way of a list of six parameter value settings contained in the *parm* variable. The approach used herein could be easily ported to the idling and speeding analysis code of Appendices E and F.

Listing H.1: Python code for generating the idling cluster maps as a KML file, idleDots.py

```
#!/usr/bin/python
"""
idleDots.py

Reads Networkfleet data, generates KML file with dots for idling, radius based on duration

ver. 1.0: Initial function, 2/20/2015
"""

import xlrd
# see: http://www.python-excel.org/
import os
import string
import simplekml

outDir = 'idlingDotKml/'

idleMin = 3
# idleMax = 12 * 60 # in minutes

month = ('','January','February','March','April','May','June','July','August',
  'September','October','November','December')

# ( input file prefix, parameter group, output file prefix)
inData = (
  ('Stop and Idle Time 2013-01-02',0,'201212'),
  ('Stop and Idle Time 2013-02-02',1,'201301'),
  ('Stop and Idle Time 2013-03-02',1,'201302'),
  ('Stop and Idle Time 2013-04-02',1,'201303'),
  ('Stop and Idle Time 2013-05-02',1,'201304'),
  ('Stop and Idle Time 2013-06-02',1,'201305'),
  ('Stop and Idle Time 2013-07-02',1,'201306'),
  ('Stop and Idle Time 2013-08-02',1,'201307'),
  ('Stop and Idle Time 2013-09-02',2,'201308'),
  ('Stop and Idle Time 2013-10-02',3,'201309'),
  ('Stop and Idle Time 2013-11-02',3,'201310'),
  ('Stop and Idle Time 2013-12-02',3,'201311'),
  ('Stop and Idle Time - Idle 2014-01-02',4,'201312'),
  ('Stop and Idle Time - Idle 2014-02-02',4,'201401'),
  ('Stop and Idle Time - Idle 2014-03-02',4,'201402'),
  ('Stop and Idle Time - Idle 2014-04-02',5,'201403'),
  ('Stop and Idle Time - Idle 2014-05-02',5,'201404'),
  ('Stop and Idle Time - Idle 2014-06-02',5,'201405'),
)
```

102

```
parm = [
  {'startRow':25, 'vid':1, 'tripDur':5, 'stopDurText':8, 'idleDur':9, 'tripDist':10, 'lat':14, 'lon':15,
'durUnit':'minute'},
  {'startRow':26, 'vid':1, 'tripDur':5, 'stopDurText':8, 'idleDur':9, 'tripDist':10, 'lat':14, 'lon':15,
'durUnit':'minute'},
  {'startRow':26, 'vid':1, 'tripDur':6, 'stopDurText':9, 'idleDur':10, 'tripDist':11, 'lat':15, 'lon':16,
'durUnit':'minute'},
  {'startRow':19, 'vid':1, 'tripDur':6, 'stopDurText':9, 'idleDur':10, 'tripDist':11, 'lat':15, 'lon':16,
'durUnit':'minute'},
  {'startRow':15, 'vid':1, 'tripDur':6, 'stopDurText':9, 'idleDur':10, 'tripDist':11, 'lat':15, 'lon':16,
'durUnit':'minute'},
  {'startRow':15, 'vid':1, 'tripDur':17, 'stopDurText':8, 'idleDur':18, 'tripDist':9, 'lat':13, 'lon':14,
'durUnit':'hour'}
]

# Settings for KML point scaling and color:
# Note that KML colors are Alpha-Blue-Green-Red
pointColor = '880066ff'
pointScale = 0.01
minSize = 0.5

# scale value list:
scaleList = [
(minSize,range(4)),
(minSize,[4]),
(minSize,[5]),
(minSize,[6]),
(minSize,[7]),
(minSize,[8]),
(minSize,[9]),
(minSize,[10]),
(minSize,range(11,21)),
(minSize,range(21,31)),
(minSize,range(31,41)),
(0.5,range(41,51)),
(0.6,range(51,61)),
(0.7,range(61,71)),
(0.8,range(71,81)),
(0.9,range(81,91)),
(1.0,[90])
] # last element is not a range, rather it is where the max scale is first applied.

# Read in the full vehicle list:
wb = xlrd.open_workbook('../vehicle_label.xlsx')
# Get first sheet
```

```python
sh = wb.sheet_by_index(0)
vehList = []
for rownum in range(sh.nrows):
  vehList.append(str(sh.cell(rownum,0).value).strip())
vehList.sort()

def getAll(startRow, vid, stopDurText, idleDur, lat, lon, durUnit, filePrefix):
  entries = []
  # Read in all data from Excel file, row at a time.
  for rownum in range(startRow,sh.nrows):
    d = getRow(rownum, vid, stopDurText, idleDur, lat, lon, durUnit)
    if (d['vehId'] in vehList and d['idle'] and d['idleMinutes'] >= idleMin):
      entries.append((d['lat'],d['lon'],d['idleMinutes'],d['vehId']))
  return entries

def getRow(rownum, vid, stopDurText, idleDur, lat, lon, durUnit):
# Get a single row of data using xlrd module. Return it as a dictionary.
# Columns are numbered starting at zero

  d = {'vehId': str(sh.cell(rownum,vid).value).strip()}
  s = sh.cell(rownum,stopDurText).value
  if "{Idle Stop}" in s or "(Idle Stop)" in s:
    d['idle'] = True
  else:
    d['idle'] = False
  if d['idle']:
    idleTime = sh.cell(rownum,idleDur).value
  # Handle blank cells in data:
    if idleTime == "":
      idleTime = "0"
    if durUnit == 'hour':
      d['idleMinutes'] = float(idleTime) * 60
    else:
      d['idleMinutes'] = float(idleTime)
  d['lat'] = sh.cell(rownum,lat).value
  d['lon'] = sh.cell(rownum,lon).value
  return d

def generateKml(data, filename, foldername):
  """Generates the output KML file."""

  kml = simplekml.Kml()
  print("Generating KML file {}".format(filename))

  # Create an array of shared styles and mapping ranges
  sharedStyle = []
```

104

```
   for scaleElem in scaleList:
     style = simplekml.Style()
     style.iconstyle.scale = scaleElem[0]
     style.labelstyle.color = 'ffffffff' # white
     style.labelstyle.scale = 0 # hide labels (.5 is ok size)
     style.iconstyle.icon.href = 'http://merlin.ahmct.ucdavis.edu/res/dot16.png'
     style.iconstyle.color = pointColor
     sharedStyle.append((style, scaleElem[1]))

   # Create a schema
   schema = kml.newschema(name='idling')
   schema.newsimplefield(name='Lon', type='float', displayname='Longitude')
   schema.newsimplefield(name='Lat', type='float', displayname='Latitude')
   schema.newsimplefield(name='Dur', type='int', displayname='Duration')
   schema.newsimplefield(name='VehId', type='string', displayname='Vehicle ID')

   folder = kml.newfolder(name = foldername)
   for event in data:
     pnt = folder.newpoint(coords=[(event[1], event[0])], name = event[3]) # (lon, lat), omitting
height, optional
     pnt.extendeddata.schemadata.schemaurl = schema.id
     # Select style based on duration:
     duration = event[2]
     # first, set to the max scale style:
     currentStyle = sharedStyle[-1][0]
     for elem in sharedStyle:
       if duration in elem[1]:
         currentStyle = elem[0]
     pnt.style = currentStyle
     pnt.extendeddata.schemadata.newsimpledata("Lon",event[1])
     pnt.extendeddata.schemadata.newsimpledata("Lat",event[0])
     pnt.extendeddata.schemadata.newsimpledata("Dur",event[2])
     pnt.extendeddata.schemadata.newsimpledata("VehId",event[3])
   kml.document.name = filename
   kml.save(filename)

if __name__ == "__main__":

  data = []
  for entry in inData:
   date = entry[2]
   inname = '../../data/' + entry[0] + '.xls'
   string.replace(inname,"/",os.sep)
   prefix = './idlingDotKml/' + date + 'idleDot'
   kmlOutname = prefix + '.kml'
   string.replace(kmlOutname,"/",os.sep)
```

```
parmd = parm[entry[1]]

wb = xlrd.open_workbook(inname)
# Get first sheet
sh = wb.sheet_by_index(0)

year = date[:4]
monthNum = int(date[-2:])
dateStr = '%s %s'%(month[monthNum],year)

print("\nGetting data from %s"%inname)
dotMap = getAll(parmd['startRow'], parmd['vid'],
  parmd['stopDurText'], parmd['idleDur'],
  parmd['lat'], parmd['lon'], parmd['durUnit'], entry[0])

# Write out idling dot KML data:
folderName = date
generateKml(dotMap, kmlOutname, folderName)
```

Listing H.2: Python code for generating the HTML files to display the idling heat maps, idleHeatMap.py

```python
#!/usr/bin/python
"""
idleHeatMap.py

Reads Networkfleet data, generates input for monthly idling heat map

Exported classes:

None

ver. 1.0.1: Generating html as well as csv, 2/18/2015
ver. 1.0: Initial function, 2/17/2015
"""

import xlrd
# see: http://www.python-excel.org/
import os
import string

template = 'template.html'

idleMin = 3
idleMax = 12 * 60 # in minutes

heatMax = 100

month = ('','January','February','March','April','May','June','July','August',
  'September','October','November','December')

# ( input file prefix, parameter group, output file prefix)
inData = (
  ('Stop and Idle Time 2013-01-02',0,'201212'),
  ('Stop and Idle Time 2013-02-02',1,'201301'),
  ('Stop and Idle Time 2013-03-02',1,'201302'),
  ('Stop and Idle Time 2013-04-02',1,'201303'),
  ('Stop and Idle Time 2013-05-02',1,'201304'),
  ('Stop and Idle Time 2013-06-02',1,'201305'),
  ('Stop and Idle Time 2013-07-02',1,'201306'),
  ('Stop and Idle Time 2013-08-02',1,'201307'),
  ('Stop and Idle Time 2013-09-02',2,'201308'),
  ('Stop and Idle Time 2013-10-02',3,'201309'),
  ('Stop and Idle Time 2013-11-02',3,'201310'),
  ('Stop and Idle Time 2013-12-02',3,'201311'),
```

107

```
  ('Stop and Idle Time - Idle 2014-01-02',4,'201312'),
  ('Stop and Idle Time - Idle 2014-02-02',4,'201401'),
  ('Stop and Idle Time - Idle 2014-03-02',4,'201402'),
  ('Stop and Idle Time - Idle 2014-04-02',5,'201403'),
  ('Stop and Idle Time - Idle 2014-05-02',5,'201404'),
  ('Stop and Idle Time - Idle 2014-06-02',5,'201405'),
)

parm = [
  {'startRow':25, 'vid':1, 'tripDur':5, 'stopDurText':8, 'idleDur':9, 'tripDist':10, 'lat':14, 'lon':15,
'durUnit':'minute'},
  {'startRow':26, 'vid':1, 'tripDur':5, 'stopDurText':8, 'idleDur':9, 'tripDist':10, 'lat':14, 'lon':15,
'durUnit':'minute'},
  {'startRow':26, 'vid':1, 'tripDur':6, 'stopDurText':9, 'idleDur':10, 'tripDist':11, 'lat':15, 'lon':16,
'durUnit':'minute'},
  {'startRow':19, 'vid':1, 'tripDur':6, 'stopDurText':9, 'idleDur':10, 'tripDist':11, 'lat':15, 'lon':16,
'durUnit':'minute'},
  {'startRow':15, 'vid':1, 'tripDur':6, 'stopDurText':9, 'idleDur':10, 'tripDist':11, 'lat':15, 'lon':16,
'durUnit':'minute'},
  {'startRow':15, 'vid':1, 'tripDur':17, 'stopDurText':8, 'idleDur':18, 'tripDist':9, 'lat':13, 'lon':14,
'durUnit':'hour'}
]

# Read in the full vehicle list:
wb = xlrd.open_workbook('../vehicle_label.xlsx')
# Get first sheet
sh = wb.sheet_by_index(0)
vehList = []
for rownum in range(sh.nrows):
  vehList.append(str(sh.cell(rownum,0).value).strip())
vehList.sort()

def getAll(startRow, vid, stopDurText, idleDur, lat, lon, durUnit, filePrefix):
  entries = []
  # Read in all data from Excel file, row at a time.
  for rownum in range(startRow,sh.nrows):
    d = getRow(rownum, vid, stopDurText, idleDur, lat, lon, durUnit)
    if (d['vehId'] in vehList and
        d['idle'] and
        d['idleMinutes'] >= idleMin and
          d['idleMinutes'] <=  idleMax):
      entries.append((d['lat'],d['lon'],d['idleMinutes']))
  return entries

def getRow(rownum, vid, stopDurText, idleDur, lat, lon, durUnit):
# Get a single row of data using xlrd module. Return it as a dictionary.
```

108

```
# Columns are numbered starting at zero

 d = {'vehId': str(sh.cell(rownum,vid).value).strip()}
 s = sh.cell(rownum,stopDurText).value
 if "{Idle Stop}" in s or "(Idle Stop)" in s:
   d['idle'] = True
 else:
   d['idle'] = False
 if d['idle']:
   idleTime = sh.cell(rownum,idleDur).value
 # Handle blank cells in data:
   if idleTime == "":
     idleTime = "0"
   if durUnit == 'hour':
     d['idleMinutes'] = float(idleTime) * 60
   else:
     d['idleMinutes'] = float(idleTime)
 d['lat'] = sh.cell(rownum,lat).value
 d['lon'] = sh.cell(rownum,lon).value
 return d

if __name__ == "__main__":

 for entry in inData:
   date = entry[2]
   inname = '../../data/' + entry[0] + '.xls'
   string.replace(inname,"/",os.sep)
   prefix = './idlingHeatMaps/' + date + 'idleHeat'
   outname = prefix + '.csv'
   string.replace(outname,"/",os.sep)
   htmlOutname = prefix + '.html'
   string.replace(htmlOutname,"/",os.sep)
   templateFP = open(template,'r')
   templateStr = templateFP.read()
   templateFP.close()

   parmd = parm[entry[1]]

   wb = xlrd.open_workbook(inname)
   # Get first sheet
   sh = wb.sheet_by_index(0)

   year = date[:4]
   monthNum = int(date[-2:])
   dateStr = '%s %s'%(month[monthNum],year)
```

109

```
  print("\nGetting data from %s"%inname)
  heatMap = getAll(parmd['startRow'], parmd['vid'],
    parmd['stopDurText'], parmd['idleDur'],
    parmd['lat'], parmd['lon'], parmd['durUnit'], entry[0])

  # Write out heat map data as CSV, and HTML:
  if heatMap:
    fp = open(outname, 'w')
    fpHtml = open(htmlOutname,'w')
    dataStr = ''
    print('Saving %s'%outname)
    fp.write('latitude, longitude, count\n')
    for row in heatMap:
      fp.write('%s, %s, %s\n'%row)
      dataStr += '\t'*6 + '{lat:%s, lng:%s, count:%s},\n'%row
    fp.close()
    print('Saving %s'%htmlOutname)
    fpHtml.write(templateStr%(dateStr, dateStr, heatMax, dataStr))
    fpHtml.close()
  else:
    print('No heat map data found for %s. No output generated.'%inname)
```

Listing H.3: HTML template used by idleHeatMap.py, template.html

```html
<html>
        <head>
                <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
                <title>Idling, %s</title>
                                <meta name="viewport" content="width=device-width, initial-scale=1.0">
                                <style>
                                        body, html {
                                                margin:0;
                                                padding:0;
                                                font-family:Arial;
                                        }
                                        h1 {
                                                margin-bottom:10px;
                                        }
                                        #main {
                                                position:relative;
                                                width:1020px;
                                                padding:20px;
                                                margin:auto;
                                        }
                                        #heatmapArea {
                                                position:relative;
                                                float:left;
                                                width:1000px;
                                                height:1000px;
                                                border:1px dashed black;
                                        }
                                </style>
                <script type="text/javascript"
src="http://maps.google.com/maps/api/js?sensor=false"></script>
        </head>
        <body>
                <div id="main">
                        <h1>Idling Time in Minutes for %s</h1>
                        <div id="heatmapArea">
                        </div>
                </div>
                <script type="text/javascript" src="heatmap/heatmap.js"></script>
                <script type="text/javascript" src="heatmap/heatmap-gmaps.js"></script>
                <script type="text/javascript">

                        var map;
                        var heatmap;
```

111

```
                        window.onload = function(){

                                var myLatlng = new google.maps.LatLng(38, -120);
                                var myOptions = {
                                  zoom: 6,
                                  center: myLatlng,
                                  mapTypeId: google.maps.MapTypeId.SATELLITE,
                                  disableDefaultUI: false,
                                  scrollwheel: true,
                                  draggable: true,
                                  navigationControl: true,
                                  mapTypeControl: false,
                                  scaleControl: true,
                                  disableDoubleClickZoom: false
                                };
                                map = new
google.maps.Map(document.getElementById("heatmapArea"), myOptions);

                                heatmap = new HeatmapOverlay(map, {"radius":3, "visible":true,
"opacity":60});

                                var testData={
                                        max: %d,
                                        data: [
%s

                                        ]
                                };


                                // this is important, because if you set the data set too early, the
latlng/pixel projection doesn't work
                                google.maps.event.addListenerOnce(map, "idle", function(){
                                        heatmap.setDataSet(testData);
                                });
                        };
                </script>
        </body>
</html>
```

112