



Division of Research &  
Innovation

# Expediting Vehicle Infrastructure Integration (EVII): Where the Rubber Meets (and Talks to) the Road

Final Report

Report CA06-0714  
December 2008

# **Expediting Vehicle Infrastructure Integration (EVII): Where the Rubber Meets (and Talks to) the Road**

**Final Report**

**Report No. CA06-0714**

**December 2008**

*Prepared By:*

California PATH Program  
University of California, Berkeley  
Richmond Field Station  
1357 South 46<sup>th</sup> Street  
Richmond, CA 94804

*Prepared For:*

California Department of Transportation  
Division of Research and Innovation, MS-83  
1227 O Street  
Sacramento, CA 95814

## **DISCLAIMER STATEMENT**

**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 or the Federal Highway Administration. This publication does not constitute a standard, specification or regulation. This report does not constitute an endorsement by the Department of any product described herein.**

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

1. REPORT NUMBER  CA06-0714	2. GOVERNMENT ASSOCIATION NUMBER	3. RECIPIENT'S CATALOG NUMBER
4. TITLE AND SUBTITLE  Expediting Vehicle Infrastructure Integration (EVII): Where the Rubber Meets (and Talks to) the Road	5. REPORT DATE  October 2006	6. PERFORMING ORGANIZATION CODE
	8. PERFORMING ORGANIZATION REPORT NO.  N/A	
7. AUTHOR(S) Pravin Varaiya University of California PATH, Richmond Field Station, Bldg. 452 1357 South 46th Street Richmond, CA 94804	10. WORK UNIT NUMBER  11. CONTRACT OR GRANT NUMBER 65A0191	
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of California PATH, Richmond Field Station, Bldg. 452 1357 South 46th Street Richmond, CA 94804	13. TYPE OF REPORT AND PERIOD COVERED Final Report UCB-ITS-PRR-2006-20  14. SPONSORING AGENCY CODE	
12. SPONSORING AGENCY AND ADDRESS  California Department of Transportation Division of Research and Innovation, MS-83 1227 O Street, Sacramento, CA 95814		
15. SUPPLEMENTAL NOTES		
16. ABSTRACT This research effort between Caltrans, California PATH and DaimlerChrysler RTNA that demonstrated two potential VII services, one in traffic data probes and another with safety, using real cars and on Caltrans roadways. It presages an operational test and a deployment, and it lays the groundwork for technical and institutional know-how that will be leveraged onto the much larger VII California effort. As such, this project explores and resolves key engineering issues associated with the point deployment of these services in a realistic setting, California roadways and the first-ever private vehicle (owned by an automobile manufacturer's laboratory, DaimlerChrysler Research Technology North America) that "talked" to the roadside, with the roadside backhaul interfacing into an existing Caltrans database and archival application, the Performance Measurement System. In the end, the project demonstrated that Caltrans could do VII and create VII champions, within and outside of Caltrans.		
17. KEY WORDS Vehicle-infrastructure integration (VII), Dedicated Short Range Communications (DSRC), Performance Measurement System (PeMS), traffic probes, backhaul, road condition monitoring, tire slip, onboard unit (OBU), onboard equipment (OBE), roadside unit (RSU), roadside equipment (RSE).	18. DISTRIBUTION STATEMENT No restrictions. This document is available to the public through the National Technical Information Service, Springfield, VA 22161	
19. SECURITY CLASSIFICATION (of this report)  Unclassified	20. NUMBER OF PAGES 62	21. PRICE

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

## **Expediting Vehicle Infrastructure Integration (EVII)**

**Xuanming Dong, Kang Li,  
Jim Misener, Pravin Varayia,  
Wenbing Zhang**

**California PATH Research Report  
UCB-ITS-PRR-2006-20**

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

for the facts and the accuracy of the data presented herein. The contents do not  
This

Final Report for RTA-65A0191-15739

October 2006

ISSN 1055-1425



**EXPEDITING VEHICLE INFRASTRUCTURE  
INTEGRATION (EVII)**

**RTA-65A0191-15739**

Xuanming Dong  
Kang Li  
Jim Misener  
Pravin Varayia  
Wenbing Zhang





# Table of Contents

Abstract .....	vii
Keywords .....	vii
1.0 Background .....	1
2.0 Traffic Probe Application for EVII .....	3
2.1. Introduction .....	3
2.2 System Architecture.....	3
2.2.1 EVII Equipment .....	3
2.2.2 Software Components .....	5
2.2.3 Protocol Stack.....	6
2.3 Software Functionality.....	7
2.3.1 RSU Collector.....	7
<b>2.4 EVII-PeMS Adaptor .....</b>	<b>8</b>
<b>2.5 Traffic Probe Application.....</b>	<b>9</b>
<b>2.6 Data Interface.....</b>	<b>9</b>
2.6.1 From OBU to RSU Collector.....	9
2.6.4 GPS-FDM Convertor .....	13
<b>2.6.6 Development of Application Packages.....</b>	<b>20</b>
3.0 Development of Road Condition Evaluation Methods.....	22
<b>3.1 Introduction .....</b>	<b>22</b>
<b>3.2 Overview.....</b>	<b>24</b>
<b>3.3 Slip-Based Tire/Road Friction Estimator .....</b>	<b>28</b>
3.3.1 Approximation of the Tire-Road Friction Coefficient .....	31
3.3.2 Slip Calculation.....	33
3.3.3 Algorithm for Slippery Road Classification .....	34
<b>3.4 Rough Road Surface Detector .....</b>	<b>38</b>
<b>3.4.1 Frequency Domain Analysis of Wheel Speed Signals .....</b>	<b>38</b>
<b>3.4.2 Road Surface Texture Evaluation Method.....</b>	<b>39</b>
<b>3.5 Experimental Results.....</b>	<b>41</b>
<b>3.6 Development of Application Program Package .....</b>	<b>44</b>
5.0 Conclusions .....	47
References .....	49
Appendix A: DaimlerChrysler Vehicle Preparation.....	51
<b>Task 1 Requirement Definition .....</b>	<b>51</b>
GPS Data .....	51
Vehicle CAN bus data.....	53
Probe Data Format.....	54
<b>Task 2 Vehicle Data Interface / Test Data Provision.....</b>	<b>55</b>
Test Data Provision.....	55
DSRC Capable OBU and Vehicles.....	57
Data Recording HW and SW .....	58
Hardware .....	58
Software:.....	60
<b>Task 3 Roadside Infrastructure Definition and Prototype Implementation .</b>	<b>61</b>



## List of Figures

Figure 1. EVII Equipment Element.....	5
Figure 2. EVII Software Elements.....	6
Figure 3. Protocol Stack .....	7
Figure 4. EVII Probe Message Format (from emerging SAE J2735).....	10
Figure 5. GPS and FDM Coordinates .....	14
Figure 6. GPS-FDM Converter.....	15
Figure 7. Converter Sliding Window .....	16
Figure 8. Determine Movement Direction.....	17
Figure 9. User Interface.....	19
Figure 10. Schematic plot of $\mu$ -s curves for different surface.....	30
Figure 11. Concept system for the slip-based friction estimation.....	31
Figure 12. Flowchart of the algorithm for slippery level classification. ....	36
Figure 13. Nonlinear Curve Fitting Example .....	38
Figure 14. Raw Wheel Speed Signal vs. Filtered Wheel Speed Signal in Time and Frequency Domains.....	39
Figure 15. Road Surface Coarseness Evaluation .....	41
Figure 16 . Slip Calculation Using Wheel Speed.....	42
Figure 17. Slip Calculation Using GPS Speed and Wheel Speed.....	43
Figure 18. Surface Coarseness Evaluation Results .....	43
Figure 19. Probe Vehicle Moving Path and Road Condition Classification .....	44
Figure A-20 Mercedes E500 and Dodge Magnum cars used in the EVII project .....	57
Figure A-21 Simple schematic of OBU .....	58
Figure A-22 OBU hardware configuration detail .....	59
Figure A-23 Photo of vehicle onboard equipment.....	60
Figure A-24 EVII OBU software control and data flow schematics .....	61

## List of Tables

Table 1. List of developed MATLAB programs.....	45
Table 2. Generated C++ DLL Program Packages Converted from Original MATLAB M Files .....	45



## **Abstract**

This research effort between Caltrans, California PATH and DaimlerChrysler RTNA that demonstrated two potential VII services, one in traffic data probes and another with safety, using real cars and on Caltrans roadways. It presages an operational test and a deployment, and it lays the groundwork for technical and institutional know-how that will be leveraged onto the much larger *VII California* effort. As such, this project explores and resolves key engineering issues associated with the point deployment of these services in a realistic setting, California roadways and the first-ever private vehicle (owned by an automobile manufacturer's laboratory, DaimlerChrysler Research Technology North America) that "talked" to the roadside, with the roadside backhaul interfacing into an existing Caltrans database and archival application, the Performance Measurement System. In the end, the project demonstrated that Caltrans can do VII and create VII champions, within and outside of Caltrans.

## **Keywords**

Vehicle-infrastructure integration (VII), Dedicated Short Range Communications (DSRC), Performance Measurement System (PeMS), traffic probes, backhaul, road condition monitoring, tire slip, onboard unit (OBU), onboard equipment (OBE), roadside unit (RSU), roadside equipment (RSE).



## 1.0 Background

The national interest in Vehicle Infrastructure Integration (VII) is the result of a convergence of interests precipitated by three events: the emergence of wireless technologies, the incipient Dedicated Short Range Communications (DSRC), and the promise of a revolution of ITS services given that DSRC implementation of wireless technologies can indeed be widespread. There is currently significant discussion with vehicle manufacturers, US DOT, and AASHTO, with a national program the likely result. During the next several months organizational and policy issues with such an effort will be a primary topic.

Importantly, VII can impact safety. Through the years there has been much research and development with these systems, field operational tests have been conducted or are underway, and a host of products are now available to drivers [1]-[6]. To date, a hallmark of active safety systems has been autonomous operation.

More recently, the concept of cooperative systems has gained further acceptance, as the revolution and proliferation of wireless communications has captured the attention and interest of infrastructure owner-operators, as well as vehicle manufacturers. With the allocation of DSRC bandwidth and the present IEEE effort in establishing standards [7], the concept of vehicle-roadside cooperative systems, now referred to in the United States as Vehicle-Infrastructure Integration (VII), is generating significant attention [8]. The idea that vehicles, these days equipped with several hundred sensors, the real possibility of Global Positioning System becoming an additional and ubiquitous sensor, and a means to send sensed information off the Controller Area Network (CAN) bus to the infrastructure (and also from the infrastructure to the CAN bus) has manifold, revolutionary applications in Intelligent Transportation Systems.

But what about in-the-ground implementation? This effort directly addresses this topic, as it gives Caltrans and its PATH research partner – in addition to the DaimlerChrysler Research and Technology North America (DCRTNA) – a very significant “leg up” on VII: it describes a project between Caltrans, California PATH and DCRTNA that demonstrated two potential VII services, one in traffic data probes and another with safety, using real cars and on Caltrans roadways. This applied research project presages an operational test and a deployment; hence, it’s name: Expedited VII.

This project was conducted in seven tasks:

- Task 1. Requirements Definition
- Task 2. Vehicle Data Interface / Test Data Provision
- Task 3. Roadside Infrastructure Definition and Prototype Implementation
- Task 4. Traffic Application
- Task 5. Safety Application
- Task 6. Application Demonstrations
- Task 7. Final Report

Tasks 4 (*Traffic Application* – or as is more precisely described by the end-of-project report, *Traffic Probe Application*) and 5 (*Safety Application* – as described herein, *Development of Road Condition Evaluation Methods*) are at the core of this effort. Tasks 1 (*Requirements Definition*), 2 (*Data Interface*) and 3(*Roadside Infrastructure Definition and Prototype Implementation*) provided, respectively, the general requirements, specific data interface requirements, and the roadside and onboard units with wireless transceivers and ancillary equipment, plus connectivity to existing backhaul to the Performance Measurement System (PeMS). Tasks 4 and 5 were highlighted during Task 6 (*Application Demonstrations*).



## **2.0 Traffic Probe Application for EVII**

### *2.1. Introduction*

Travel time and traffic speed between predetermined freeway locations provides important information to travelers, infrastructure owner/operators, and public safety personnel. A number of techniques such as magnetic loop detectors, video monitors, or radar detection have been used to measure traffic speed and travel time. The incipient Dedicated Short Range Communications (DSRC) technology provides a new technology to aid in the estimation of travel time and traffic speed.

The traffic application software developed for EVII feeds GPS data records from the traffic probe vehicles into the PeMS system, produces vehicle travel time between two predetermined freeway locations in a format that is compatible with that provided by the MTC toll tag readers, and then allows users to access the travel time via web browsers over the internet.

### *2.2 System Architecture*

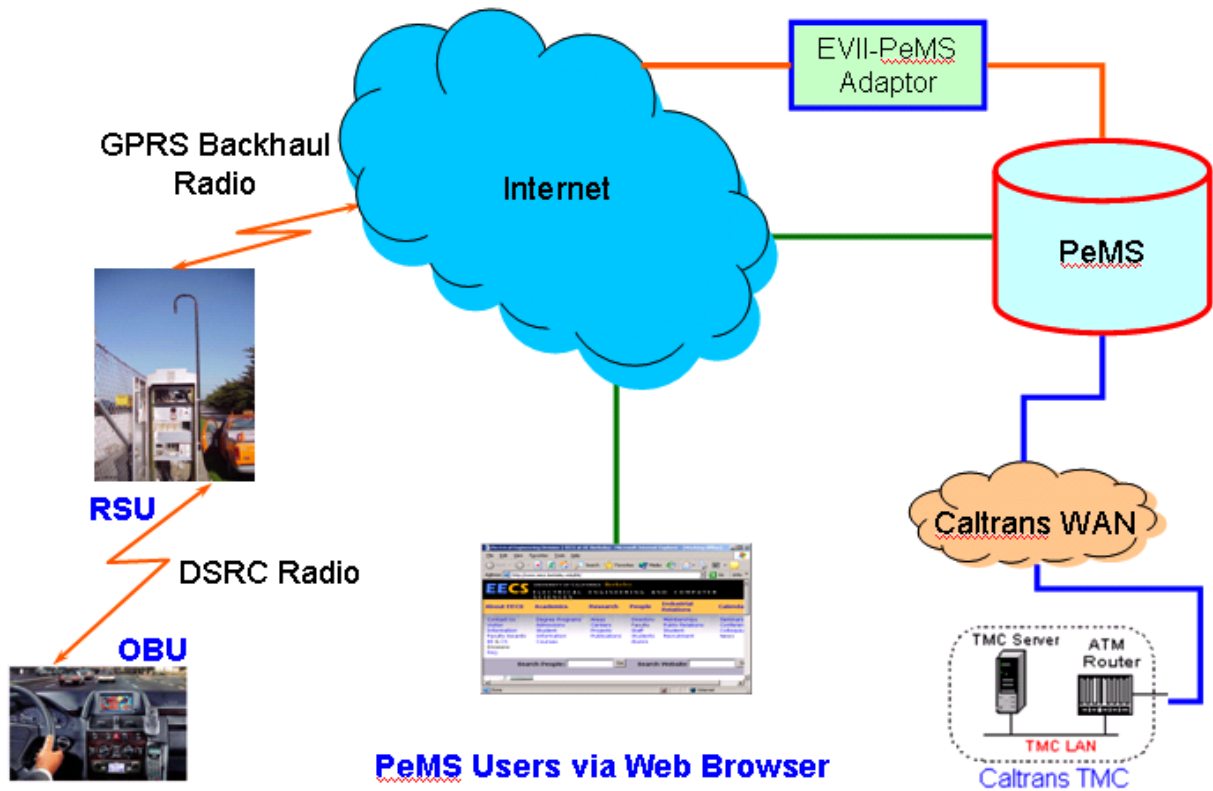
#### 2.2.1 EVII Equipment

An overview of the EVII equipment can be found in Figure 1.

A traffic probe vehicle with a 5.9 GHz DSRC transceiver (or On-Board Unit, OBU) and a GPS system is used to regularly generate probe records. Based on its memory and link capacity, the OBU also stores probe records temporarily until a connection with a RSU (Roadside Unit) is detected.

A RSU is a computer system equipped with a 5.9 GHz DSRC transceiver, an interface to the backbone networks (GPRS modem), application-oriented intelligent software, and enough storage capacity. In our demo, the RSU receives traffic probe data records from the OBU via DSRC radio link. Then it stores, aggregates and forwards the probe records to the backbone networks via GPRS backhaul radio, which for this project is provided by Cingular.

The PeMS server is the destination of traffic probe data records from OBUs. It is fully accessible from Internet. It also stores other data from Caltrans Wide Area Network (WAN) that is connected to road equipment used to collect traffic data. Since the Cingular GPRS service enables an internet connection, probe records from OBUs are forwarded by routers to reach the PeMS server. Therefore, any computer with an internet connection can be used to read the traffic speed and travel time information via standard web browsers.



**Figure 1. EVII Equipment Element**

2.2.2 Software Components

There are four primary software components in the system: (i) Onboard Unit (OBU) Sender, (ii) Roadside Unit (RSU) Collector, (iii) EVII-PeMS Adaptor, and (iv) Performance Measurement Systems (PeMS) Traffic Probe Application. All are shown in Figure 2.

The OBU Sender was developed by DaimlerChrysler RDCX. It sends broadcast UDP packets over Dedicated Short Range Communications DSRC wireless links to the RSU that it is associated with. The payload portion of those UDP packets is defined in the emergent probe message set (described in Appendix A, along with the entire vehicle-based development). Although the message set defines many types of messages, we are only interested in Traffic Probe Message and the Traffic Safety Message that reports slippery road conditions.

The RSU Collector knows the PeMS server and is responsible for collecting all traffic probe messages from OBUs, aggregating them, and forwarding them to the PeMS server reliably via the GPRS backhaul radio and the internet.

The EVII-PeMS Adaptor collects the traffic probe data records from RSU, converts all GPS data into FDM data, and stores them into PeMS database. In addition, EVII-PeMS Adaptor estimates the traffic speed and travel time based on the received traffic probe data records, and updates the PeMS database.

The PeMS Traffic Probe Application is a Common Gateway Interface (CGI) program that accepts web users' queries and returns intuitive graphical travel time and traffic speed information.

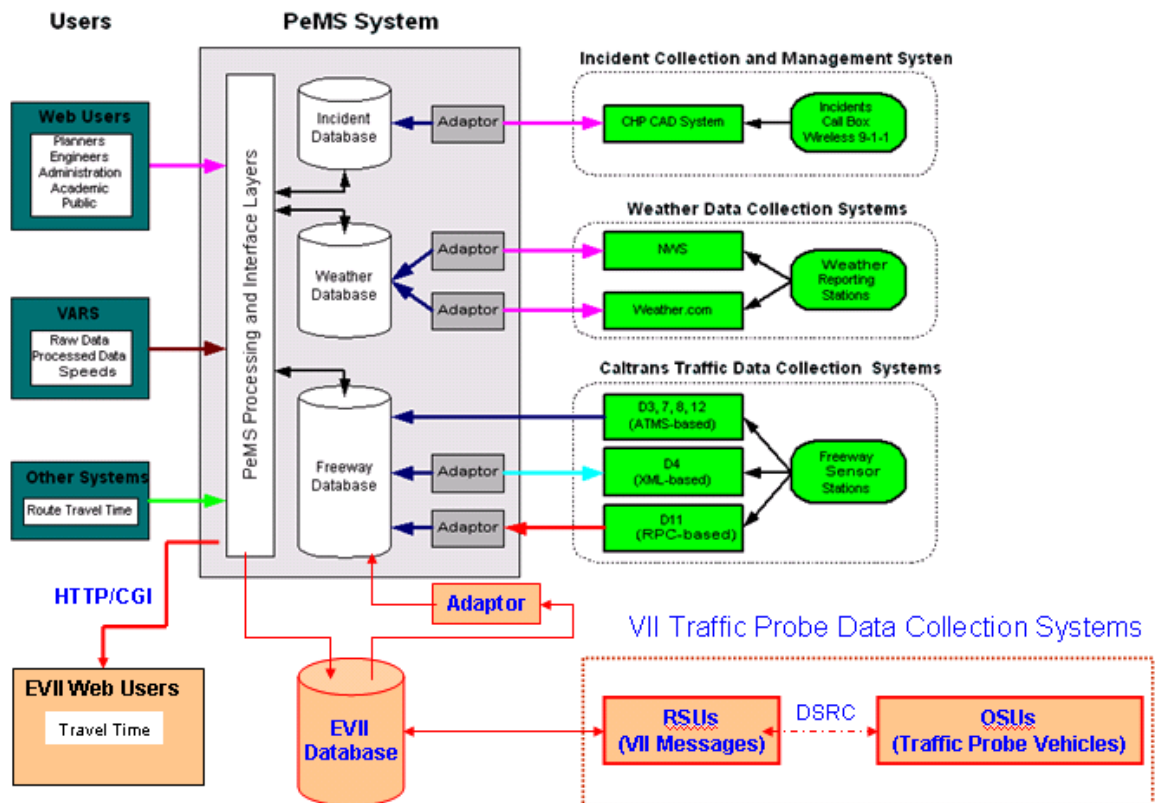
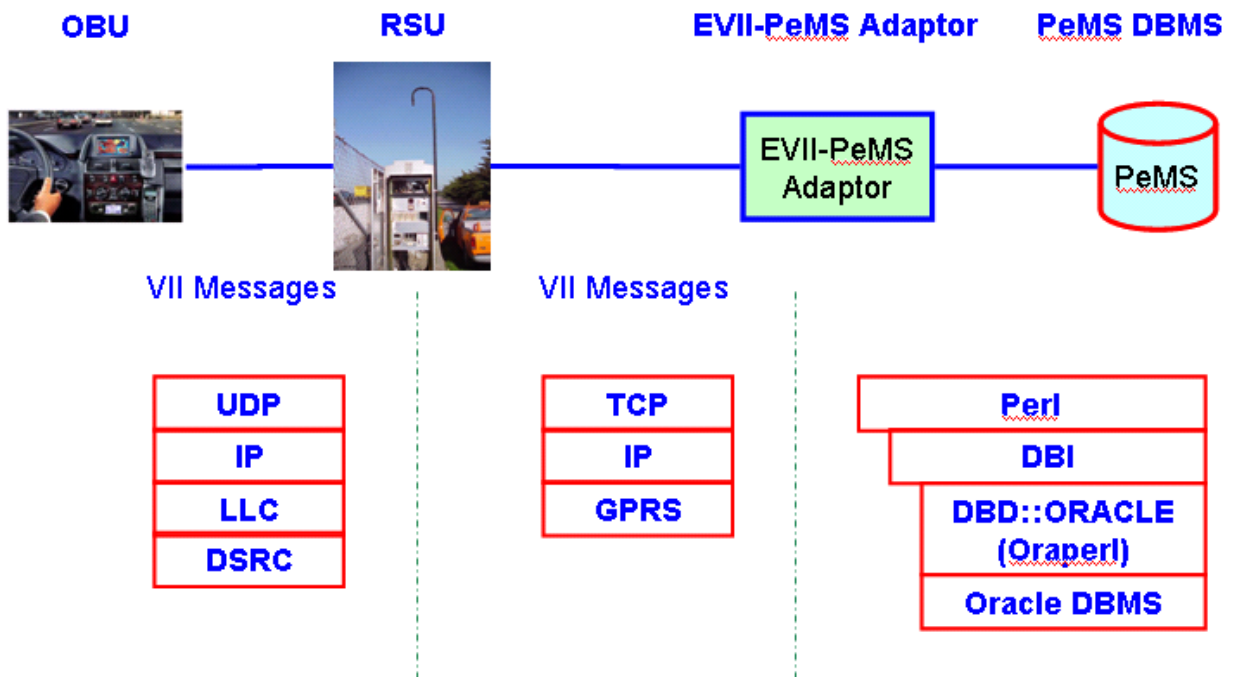


Figure 2. EVII Software Elements

### 2.2.3 Protocol Stack

The protocol stack between different software entities is shown in Figure 3. The UDP-based protocol between OBUs and RSUs helps to increase the throughput over the DSRC wireless link and combat mobility inherent with OBUs. A reliable TCP protocol is used between RSUs and the EVII-PeMS Adaptor to avoid packet loss. However, since the GPRS backhaul is slow and unstable, we put more intelligence in nodes closer to the PeMS server, and have developed a special protocol between RSU and the EVII-PeMS Adaptor.



**Figure 3. Protocol Stack**

## 2.3 Software Functionality

### 2.3.1 RSU Collector

The main function of the RSU Collector is to parse the broadcast UDP packets from OBUs, classify them, and then send required data to the PeMS Adaptor via UDP packets. Upon the arrival of the UDP packet, the main procedures followed by the RSU Collector include:

1. Parse the message based on predefined message formats. This includes Traffic Safety Messages and Traffic Probe Messages.

2. Extract useful information to the EVII-PeMS Adaptor and cache all recent probe records so that the number of records is large enough to write into a file with predetermined size.
3. Since the live time of a connection between an OBU and a RSU is short, the arrival of traffic probe data records is bursty. After the RSU Collector receives the first data record from a new connection, it can detect the completion of transmission for this connection. Once the data records from current OBU are ready, the RSU Collector initializes the TCP transmission to the EVII-PeMS Adaptor.
4. After the TCP transmission to EVII-PeMS Adaptor is finished, go to step 1.

#### ***2.4 EVII-PeMS Adaptor***

The main function of the EVII-PeMS adaptor is to parse the traffic probe data records from RSUs, classify them, and then store required data to the corresponding tables in PeMS database. Upon the arrival of a file from RSUs, the procedure followed by the EVII-PeMS adaptor is:

1. Check the receiving file. Keep checking if there is no new record.
2. If new records are detected, wait for a stabilizing period so that recent transmission from RSUs will be fully completed.
3. Either store new traffic probe records immediately to the PeMS database via embedded SQL commands; or cache them and store them to the PeMS database periodically (for example, every 30 seconds). Note that the traffic safety data should be stored in EVII\_TRAFFIC\_SAFETY table and the traffic probing data should be stored in EVII\_TRAFFIC\_PROBE table.
4. Convert all traffic probe records with GPS coordinates into records with FDM coordinates, and store them into database table.
5. Estimate the traffic speed and travel time for predetermined freeway segments, and store them into database table.

6. Go to step 1.

## ***2.5 Traffic Probe Application***

The entry point of Traffic Probe Application is a web link:

[http://pemscs.eecs.berkeley.edu/~xjdong/traffic\\_probing.html](http://pemscs.eecs.berkeley.edu/~xjdong/traffic_probing.html)

Once at this page, users navigate the GUI interface to select a source freeway location and a destination freeway location by clicking the two points in the map or finding the locations in the lists. Then the two locations are validated and sent to the PeMS server. The Perl CGI program developed for Traffic Probe Application in the PeMS server parses the query commands, and generates a new page that contains the estimated traffic speed and travel time, and sends the new web page back to the user.

## ***2.6 Data Interface***

Since the system consists of several independent software entities, their main interface is the message or packet format exchanged among them.

### **2.6.1 From OBU to RSU Collector**

The payload portion of broadcast UDP packets follows strictly the definition in the message set. Here the definition for the traffic probing data in the message set is given in Figure 4.

	A	B	C	D	E	F	G	H	I	J	K	L	M	
1	<b>Probe Message</b>													
2														
3	Del	8 (msb)	7	6	5	4	3	2	1(lsb)					
4	1	Application ID [1]												
5	2	Message Version Number (0 - 255)												
6	3	Length of Message in Bytes (0 - 65,535)												
7	4	Length of Message including headers												
8	5	Number of milliseconds since Jan. 1, 2004 at 00:00:00 UTC												
9	6	"												
10	9	UTC Time LSByte (LSBit = 0.001 seconds)												
11	10	Time Precision [2]												
12	11	Positioning Precision [5]												
13	12	Longitude of center of vehicle MSByte												
14	13	"												
15	15	Longitude LSByte (LSBit = 10 <sup>-7</sup> decimal degrees; signed)												
16	16	Latitude of center of vehicle MSByte												
17	17	"												
18	19	Latitude LSByte (LSBit = 10 <sup>-7</sup> decimal degrees; signed)												
19	20	Heading MSByte												
20	21	Heading (LSBit = 0.01 degrees; signed; 0 degrees = North)												
21	22	Vehicle Speed MSByte												
22	23	Vehicle Speed LSByte (LSBit = 0.01 m/s; unsigned)												
23	24	Heading Precision [8]												
24	25	Speed Precision [3]												
25	26	Altitude - meters above sea level (unsigned), offset by 1 km (value of 0=1 km below sea level), LSBit=0.1meter												
26	27	"												
27	28	"												
28	29	Vehicle Type [4]												
29	30	Seconds since last Probe Message sent												
30	31	"												
31	32	Longitudinal Acceleration (LSBit = 0.01 m/s <sup>2</sup> ; signed)												
32	33	"												
33	34	Number of Vehicle Device Status Fields [22]												
34	35	"												
35	36	Vehicle Status Device Type 1 [23]												
36	37	"												
37	38	Vehicle Status Device Value 1												
38	39	"												
39	40	Vehicle Status Device Type n [23]												
40	?	"												
41	?	Vehicle Status Device Value n												
42	????	"												
43	:	Snapshot 2												
44	:	"												
45	????	"												
46	:	Snapshot n												
47	???	"												

Color Codes for Context
Standard Message Header
Probe Message Header
Snapshot Data

Note:  
 Snapshots are collections of vehicle data at one point in time. They occur at time intervals that increase with the vehicle speed. They also occur when vehicle speed drops to zero and when the vehicle starts moving again. Additionally snapshots occur when certain parameters shown in red (see Note 23) exceed a threshold. These thresholds have not yet been defined.

Figure 4. EVII Probe Message Format (from emerging SAE J2735)

### 2.6.2 From RSU Collector to PeMS Adaptor

It is a text file in which each line represents a traffic probe record. A typical file is shown below:

```

.....
evii_demo, 2005/10/06-12:33:12, -122.165330, 37.711624, 7.500000, 36.708125, 282.100000
evii_demo, 2005/10/06-12:33:15, -122.165917, 37.711698, 7.800000, 37.066034, 269.600000
evii_demo, 2005/10/06-12:33:18, -122.166498, 37.711646, 7.500000, 38.676628, 259.100000
evii_demo, 2005/10/06-12:33:21, -122.167068, 37.711542, 6.500000, 37.066034, 260.200000
evii_demo, 2005/10/06-12:33:25, -122.167646, 37.711589, 5.200000, 35.097530, 293.700000
evii_demo, 2005/10/06-12:33:28, -122.168140, 37.711835, 5.100000, 41.651754, 304.700000
.....

```



The first column is the name of the probe vehicle, and the second column is the generation time of current record. The third, fourth and the fifth column represent the latitude, longitude, and altitude, respectively. The sixth and seventh columns are the speed and heading information. Each column is delimited by a coma. A new line indicates the end of each data record.

### 2.6.3 From PeMS Adaptor to PeMS Tables

The traffic probe data records with GPS coordinates are stored in table EVII\_TRAFFIC\_PROBE\_GPS. The structure of EVII\_TRAFFIC\_PROBE\_GPS table is:

```
SQL> desc EVII_TRAFFIC_PROBE_GPS
```

Name	Null?	Type
VEHICLE_ID	NOT NULL	CHAR(20)
VEHICLE_TIME	NOT NULL	DATE
LONGITUDE	NOT NULL	NUMBER(10,6)
LATITUDE	NOT NULL	NUMBER(10,6)
ALTITUDE	NOT NULL	NUMBER(10,6)
SPEED		NUMBER(5,2)
HEADING		NUMBER(5,2)
PEMS_TIME	NOT NULL	DATE

```
SQL>
```

After the GPS-FDM conversion, the traffic probe data records with FDM coordinates are stored in table EVII\_TRAFFIC\_PROBE\_FDM. The structure of the EVII\_TRAFFIC\_PROBE\_FDM table is:

```
SQL> desc EVII_TRAFFIC_PROBE_FDM
```

Name	Null?	Type
VEHICLE_ID		CHAR(20)
VEHICLE_TIME	NOT NULL	DATE
FREEWAY	NOT NULL	NUMBER(3)

```

DIRECTION                                NOT NULL VARCHAR2 (1)
POSTMILE                                  NOT NULL NUMBER (8, 3)
SPEED                                      NUMBER (5, 2)
HEADING                                    NUMBER (5, 2)
PEMS_TIME                                  NOT NULL DATE

```

SQL>

The estimation of travel time for predetermined freeway segments are stored in table EVII\_TRAFFIC\_PROBE\_ROUTE. The structure of EVII\_TRAFFIC\_PROBE\_ROUTE table is shown below:

SQL> desc EVII\_TRAFFIC\_PROBE\_ROUTE

Name	Null?	Type
ROUTE_ID	NOT NULL	NUMBER (3)
FREEWAY_BEGIN	NOT NULL	NUMBER (3)
DIRECTION_BEGIN	NOT NULL	VARCHAR2 (1)
POSTMILE_BEGIN	NOT NULL	NUMBER (8, 3)
FREEWAY_END	NOT NULL	NUMBER (3)
DIRECTION_END	NOT NULL	VARCHAR2 (1)
POSTMILE_END	NOT NULL	NUMBER (8, 3)
TRAVEL_TIME	NOT NULL	NUMBER (8)
TRAVEL_MILE	NOT NULL	NUMBER (8, 3)
PEMS_TIME	NOT NULL	DATE

SQL>

(It is worth mentioning that traffic safety data records are stored in table EVII\_TRAFFIC\_PROBE\_SAFETY, whose structure is shown below:

SQL> desc EVII\_TRAFFIC\_PROBE\_SAFETY

Name	Null?	Type
VEHICLE_ID		CHAR (20)
DETECTION_TIME	NOT NULL	DATE
LONGITUDE	NOT NULL	NUMBER (10, 6)

```

LATITUDE                                NOT NULL NUMBER(10,6)
ALTITUDE                                NOT NULL NUMBER(10,6)
SAFETY_LEVEL                            NOT NULL NUMBER(1)
SAFETY_VALUE                            NUMBER(1)
PEMS_TIME                               NOT NULL DATE

SQL>)

```

The estimation of traffic speed for predetermined freeway locations are stored in table EVII\_FDM\_DEMO\_POINT, whose structure is shown below:

```
SQL> desc EVII_FDM_DEMO_POINT
```

```

Name                                     Null?    Type
-----
POINT_ID                                NOT NULL NUMBER(3)
VEHICLE_TIME                            NOT NULL DATE
FREEWAY                                  NOT NULL NUMBER(3)
DIRECTION                                NOT NULL VARCHAR2(1)
POSTMILE                                  NOT NULL NUMBER(8,3)
SPEED                                     NUMBER(5,2)
HEADING                                   NUMBER(5,2)
PEMS_TIME                               NOT NULL DATE

SQL>

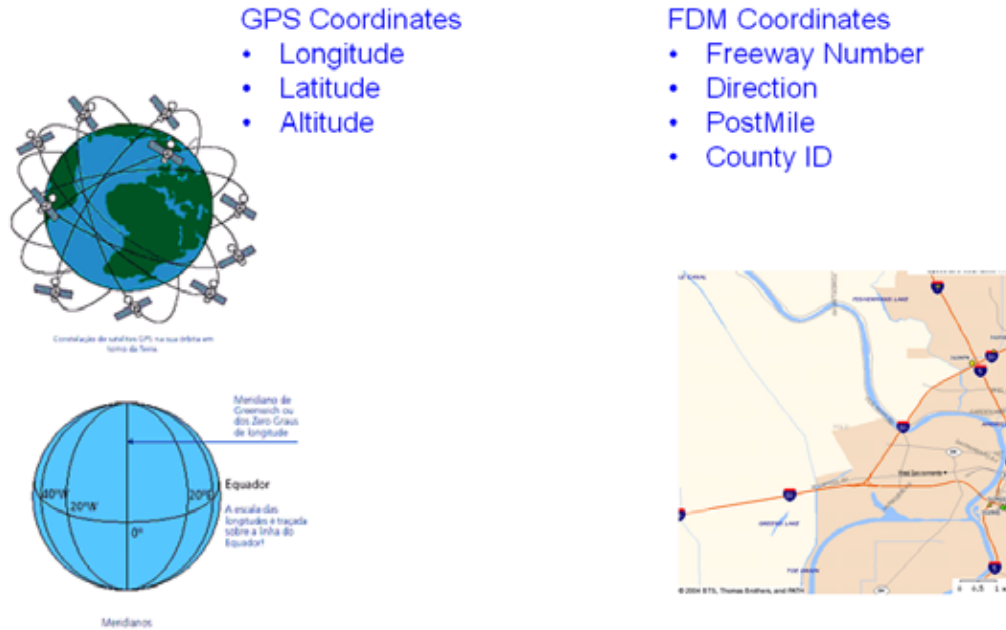
```

Whenever a user accesses the entry web page, [http://pemscs.eecs.berkeley.edu/~xjdong/traffic\\_probing.html](http://pemscs.eecs.berkeley.edu/~xjdong/traffic_probing.html), the CGI program extracts all predetermined freeway locations in table EVII\_FDM\_DEMO\_POINT and generates a web page to display all these points in the map.

#### 2.6.4 GPS-FDM Convertor

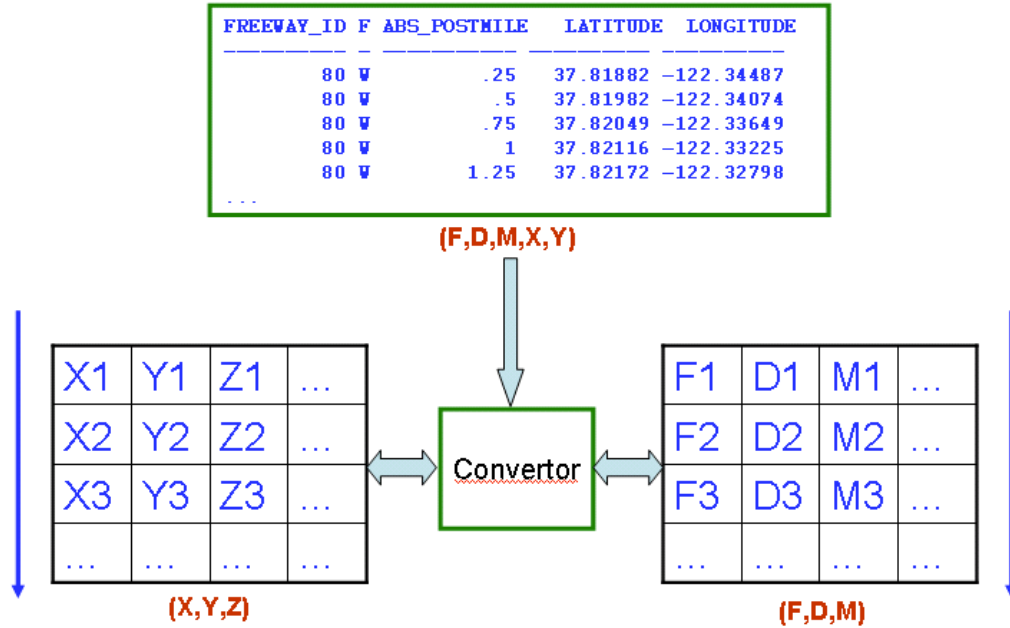
Most GPS receivers can report local information based on latitude, longitude, and altitude, as shown in Figure 5.

# GPS Coordinates and FDM Coordinates



**Figure 5. GPS and FDM Coordinates**

These GPS coordinates are recorded in angular units of degrees, minutes, and seconds. GPS receivers may display coordinates in degrees with minutes to four decimal places. In addition, GPS provides UTC time information instead of local time. However, MTC tag data and PeMS data records have different format. They give local time and location information based on freeway, direction, and postmile (so called FDM coordinates). We developed the software to convert GPS records into FDM records, illustrated in Figure 6.



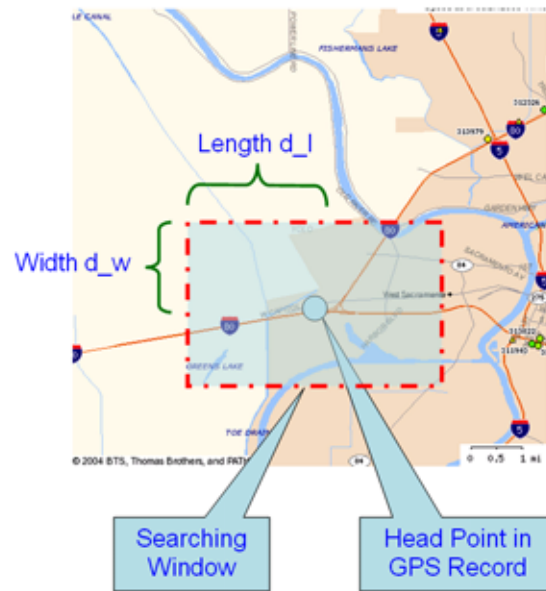
**Figure 6. GPS-FDM Converter**

The conversion algorithm is based on a freeway map with both FDM coordinates and segments with lat-long coordinates in the PeMS system. The conversion algorithm has three major functions:

1. Determine the Highway number
2. Determine the moving Direction
3. Determine the Post-Mileage

The main idea of the conversion algorithm is based on a sliding window, shown as in Figure 7. Assuming that the altitude doesn't change dramatically, for any given GPS point  $\langle X_i, Y_i \rangle$ , the conversion software maps it to a FDM point  $\langle F_i, D_i, M_i \rangle$ . Each point should be processed in the context of recent point history.

## From GPS to FDM: Sliding Window Algorithm



**Figure 7. Converter Sliding Window**

First, we define a rectangle location window whose center point is current GPS point with length  $2*d_l$  and width  $2*d_w$ . It is easy to find all pre-recorded map points in PeMS database.

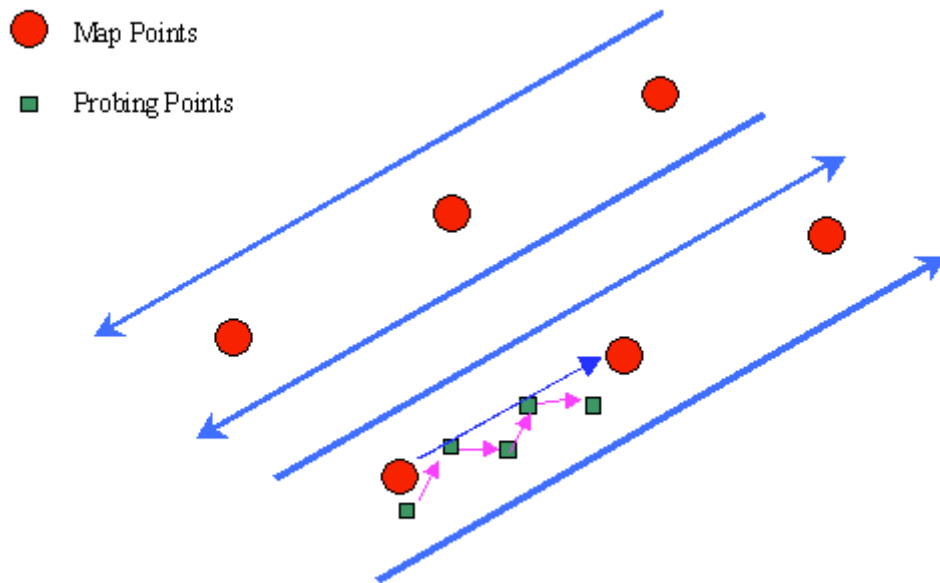
To perform Function 1, we take  $N_s$  sample GPS records from incoming points. For each of those sample points, we:

- get all map points within its rectangle window,
- count the number of points for each found freeway number in the rectangle area of the map, and
- choose the freeway number that most points belong.

However, it is not always true that the probing freeway has the most points in the rectangle area of the map. In that case, we need to determine the freeway number of each sample GPS record by the same way, and then choose the freeway number that most

sample GPS record has. The distance of those sample GPS records should be long enough, for example, 0.2 mile, to avoid wrong decisions.

After Function 1 has been done, we obtain the freeway number. For each GPS record, we can now find the closest two map points along the designated increasing direction from PeMS database, defined as upper map point  $\langle F_u, D_u, M_u, C_u, X_u, Y_u \rangle$  and lower map point  $\langle F_l, D_l, M_l, C_l, X_l, Y_l \rangle$  respectively. For any freeway, the possible directions are also known, which are strongly related to the reference vector between the upper map point and the lower map point. If we calculate the motion vector between last GPS record and current GPS record, we'll find current moving direction by matching the reference vector and the motion vector, as shown in Figure 8. Therefore, Function 2 can be done.



**Figure 8. Determine Movement Direction**

For both Function 1 and Function 2, we do not need to search the map points in the sliding window from the map database for each GPS record. Because of the high location correlation between successive GPS records, the sliding window may be reused for Function 3. The basic steps of Function 3 include:

1. Use  $d_l$  and  $d_w$  to read all  $\langle F, D, M, C, X, Y \rangle$  that satisfies  $X_i - d_l < X < X_i + d_l$  &  $Y_i - d_w < Y < Y_i + d_w$  &  $F=f$  &  $D=d$  into memory from Database
2. Order the two-dimensional array by  $M$
3. Compare  $\langle X, Y \rangle$  to all found records and return the closest two records  $\langle F_u, D_u, M_u, C_u, X_u, Y_u \rangle$  and  $\langle F_l, D_l, M_l, C_l, X_l, Y_l \rangle$
4. If  $\langle X, Y \rangle$  is within the two points, Do straight-line interpolation between FDM points:  $M_i = m_l + 0.25 * (X_i - X_l) / (X_u - X_l)$
5. Get a new GPS record from input records, compare  $X, Y$  to neighboring arrays (4 records), and return the closest two records  $\langle F_u, D_u, M_u, C_u, X_u, Y_u \rangle$  and  $\langle F_l, D_l, M_l, C_l, X_l, Y_l \rangle$
6. If  $(X, Y)$  is within the two points, go to step 4;  
Otherwise go to step 5
7. If  $(X, Y)$  is not within the two points,  
go to step 1 to searching the database again using current record as the head record.

Let  $O(DB)$  be the computation complexity of the SQL database access and  $N_s$  be the number of sample GPS points, the computation complexity of the GPS-FDM conversion algorithm is  $O(N_s * DB)$ . Let  $Step_l$  be the latitude degree per 0.25 mile and  $Step_w$  be the longitude degree per 0.25 mile, the storage complexity of the GPS-FDM conversion algorithm is  $O((d_l / step_l) * d_w / step_w)$ . What we implemented for EVII project is just a reference GPS-FDM conversion algorithm. There are many ways to refine this algorithm so that the overall system performance will be greatly improved.



## 2.6.5 Testing

DCRTNA provided the Probe Vehicle and the program to generate and transmit traffic probe data records.

The RSU was installed at the Berkeley Highway Lab (BHL) by PATH personnel. The demonstration concept has a DCRTNAprobe vehicle with DSRC OBU travels from US101@Dumbarton to BHL. In this demonstrated concept, the probe vehicle generates a probe record every 100 meters. Once a RSU is detected, the probesends all records to the RSU.

The RSU sends all probe data to PeMS server. PeMS stores, interprets, and analyze probe data. The traffic speed and travel time information can be browsed by any internet user, as shown in Figure 9.

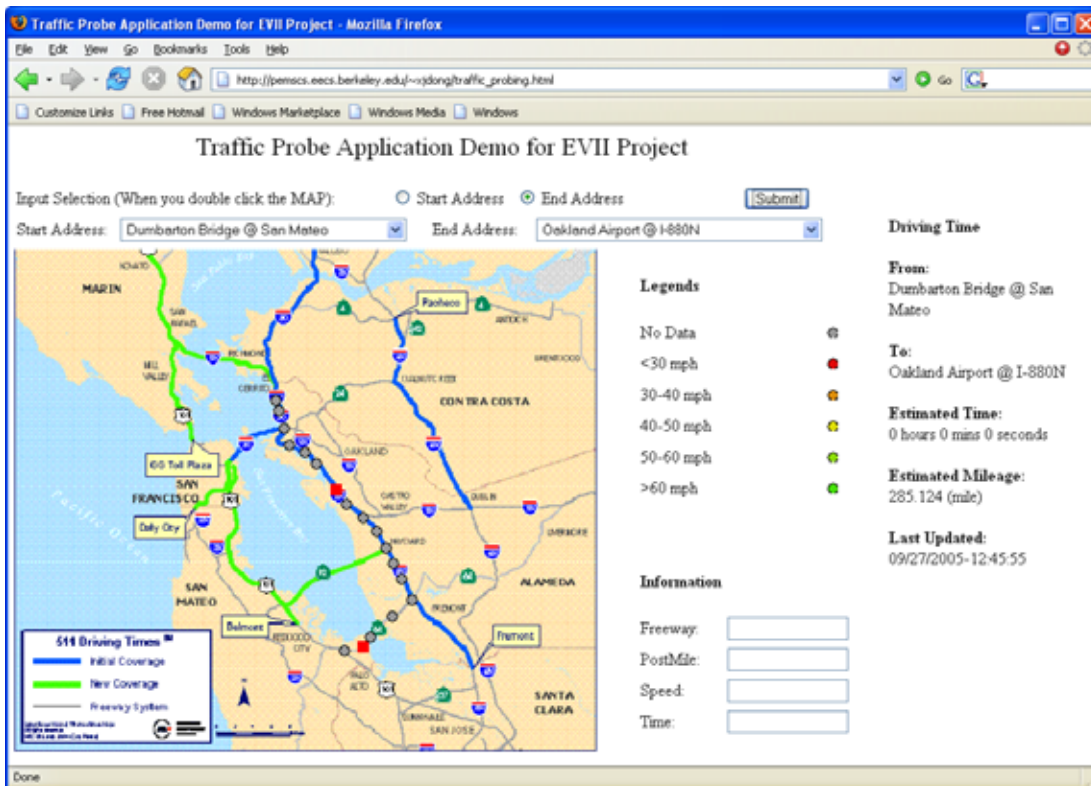


Figure 9. User Interface

### **2.6.6 Development of Application Packages**

The EVII computational programs are written in C and Perl, for a linux operating system. All C programs can be compiled successfully using GNU C compiler. The Perl interpreter must first be installed. SSH and SCP packages must also be installed in both RSU and PeMS server.

The files in the PeMS server include:

```
~/public_html  
blank.html  
cgi-index.html  
estimate_time.html  
evii_search.html  
FDM_data.txt  
get_page.pl  
GPS_data.txt  
index.html  
legend_0.png  
legend_1.png  
legend_2.png  
legend_3.png  
legend_4.png  
legend_5.png  
legend_star_begin.gif  
legend_star_end.gif  
Path_Map.jpg  
point_input.html  
rfs_google.gif  
rfs_map.jpg  
safety_display.html  
simulation.gif  
simulation.jpg  
slippery_display.html  
test.html  
traffic_probe.html
```

cgi-bin  
evii\_daemon.pl  
fdm\_process.pl  
fdm\_record.txt  
feed\_pems  
get\_page.pl  
gps\_fdm.pl  
gps\_record.bad  
gps\_record.txt  
parse\_gps.awk  
readme.txt  
sample.txt  
sample\_2.txt  
travel\_time.pl  
upload\_fdmctl  
upload\_fdm.log  
upload\_gpsctl  
upload\_gps.log  
gps\_come.bak  
gps\_come.txt  
gps\_demo\_record.txt  
gps\_demo\_second\_record.txt  
rsu\_daemon.pl

The files in the RSU include:

~/RSU\_EVII  
evii.cpp  
gps\_come.bak  
gps\_come.txt  
gps\_gps.txt  
gps\_record.txt  
Makefile  
null.txt  
rsu\_daemon.pl

## 3.0 Development of Road Condition Evaluation Methods

### 3.1 Introduction

This section presents an on-board road condition monitoring system, which was developed to fulfill the need of safety application in EVII. Experimental results demonstrate that the developed road condition monitoring system is capable of distinguishing road condition into three slippery levels and detecting rough surface in close to real-time. This system may be applicable to general production vehicles due given that there is no dedicated sensor required, and given further development, it may be robust and easy to calibrate.

The system is able to continuously evaluate and classify road condition in terms of slipperiness and coarseness. The road surface could be classified into four grades, normal ( $\mu_{\max} \geq 0.5$ ), slippery ( $0.3 \leq \mu_{\max} < 0.5$ ), very slippery ( $\mu_{\max} < 0.3$ ) and rough surface (gravel). The mechanism of this safety application is that once a slippery road is detected by a probe vehicle equipped with this system, a warning message with accurate GPS position can be transmitted from the probe vehicle to road side equipment (RSE) and further be relayed to following vehicles as well as traffic management center (TMC) via DSRC. Hence, the safety of road users can be improved with the aid of this cooperative or potential VII active safety system.

We address three main topics:

1. development of road condition evaluation methods,
2. sensor fusion and
3. generation of deployable application program.

Topic 1 mainly deals with two estimation problems: one with evaluation of slippery extent of road surface and the other with detection of rough road surface. The proposed method to tackle the former problem is a slip-based approach; however, it did not follow

the original slip-based approach in literature, which estimates the slope of slip curve to approximate maximum friction coefficient. The old approach has two well-known problems with calibration and robustness, which motivate the development of new strategy in this work.

Our approach utilizes nonlinear curve fitting method to directly estimate the maximum tire-road friction coefficient using the so-called “Magic Formula.” The characteristic of the relationship between friction coefficient and slip, i.e. that the value of maximum friction coefficient  $\mu_{\max}$  varies significantly with different surfaces, but its corresponding slip value  $\bar{A}_{\max}$  does not vary much, is exploited in the road condition classification algorithm. By direct estimation of maximum available friction coefficient, a parameterized algorithm with physically meaningful thresholds was developed to resolve the aforementioned two weaknesses in conventional slip-based approach. For detection of rough road surface, a separate classifier based on the variance of filtered wheel speed signal is implemented and validated.

Topic 2 details the software-based integration of on-board sensors with GPS measurements and signal processing for all measurements. Since modern GPS speed measurement, which utilizes Doppler shift effect, can provide a very accurate reference speed whose stochastic error can be as small as  $1\sigma = 5\text{cm/sec}$ , and the availability of GPS on production vehicles is increasing, GPS speed was used for slip calculation during braking in this study. Besides, we also utilized GPS speed-based heading angle to calibrate the slowly time-varying bias of yaw rate sensor. As a consequence, the synchronization and integration of on-board sensors with GPS measurements becomes a must due to GPS time-delay with respect to on-board sensors and their different error characteristics.

Topic 3 describes the development of deployable application program, whose core program was converted from post processing MATLAB codes. Due to incompatible data format in C++ and MATLAB, an interface program for data conversion was developed.

In this research, we developed a system, which is able to continuously evaluate road condition in terms of slipperiness and roughness in close to real-time. A condition was that the system was to be generally applicable to production vehicles with moderate calibration. However, there is no existing road condition evaluation method which can fulfill aforementioned requirements, although research which studies the tire-road friction estimation problems exists in literature. The main difficulties of applying those methods such as slip-based  $\mu_{\max}$  estimation, model-based friction estimation, acoustic approach, cause-based approaches to our system include (1) model dependence (2) robustness problem (3) calibration problem and (4) reliance on dedicated sensors. These four issues would pose serious problems in real practice. Therefore, this research was motivated to find a more suitable road condition estimation method, which can prevent those problems.

This report details the development of an on-board road condition monitoring system equipped on the probe vehicle, which provides warning messages for following vehicles through an indeterminate subset of RSE when detecting anomalous road conditions – a definite application of VII. The developed on-board road condition monitoring system is able to continuously evaluate and classify road condition in terms of extent of slipperiness and coarseness in close real-time. The evaluation work reported here includes two parts:

- Prediction of the maximum tire-road friction coefficient  $\mu_{\max}$ .
- Analysis of the texture of road surface.

Given input from each of these two parts, road condition is classified into either three levels of slipperiness: “normal”, “slippery”, and “very slippery” or “rough” surfaces.

### **3.2 Overview**

We study two estimation problems, tire-road friction estimation and evaluation of road surface roughness. Based on the estimation research results, a road condition classification algorithm was proposed. This algorithm is parameterized in physically

meaningful thresholds; this feature makes the developed system has the merit of easy calibration and also preserves the capability of self-calibration for future research.

The method for prediction of  $\mu_{\max}$  is the so-called slip-based approach, which is effect-based friction estimation [9]. As described in this paper, the advantage of a slip-based approach over existing alternatives such as cause-based, acoustic and tire-tread deformation approaches is that it does not require additional dedicated sensors. Moreover, although cause-based approaches demonstrate very high accuracy in prediction of  $\mu_{\max}$ , they tend to lose accuracy when conditions deviate from the conditions under which they were trained [10]. Note that slip-based approaches have problems with robustness and calibration; this gave motivation to this research. Since the developed system is expected to be generally applicable albeit with some modification, robustness and calibration would become critical issues in real practice. The following remedies were used to tackle these issues:

- 1) High accuracy of prediction of  $\mu_{\max}$  is not our primary interest as opposed to, for instance, automatic vehicle system controller design [11]; rather, distinguishing road conditions at a discrete set of levels of slipperiness is our goal. Hence, if we can approximate the  $\mu$  versus slip curve, our objective of classifying slippery road conditions may be met. Given the freedom to relax the accuracy in computation of  $\mu$ , i.e. using the rigorous definition to calculate friction coefficient,

$$\mu = \frac{\sqrt{F_x^2 + F_y^2}}{N_z} \quad (1)$$

where  $F_x$ ,  $F_y$  and  $N_z$  are longitudinal, lateral, and normal forces acting on the tire,  $\mu$  can be respectively approximated by equations (2) and (3) in acceleration case and braking case:

$$\mu \approx 0.2 \times a \quad (2)$$

$$\mu \approx 0.1 \times a \quad (3)$$

As a consequence, there is no need for an involved procedure to obtain vehicle/tire parameters such as engine map, vehicle mass, tire mass, suspension spring constant,

and suspension damping constant. These were required in [9],[12]-[14] for computing frictional and normal forces but again, not for our focused application.

- 2) Instead of using the most-often chosen linear regression approach to infer  $\mu_{\max}$  from the  $k$ - $\mu_{\max}$  correlation, where  $k$  is the slope of the linear fit to  $\mu$  versus slip data [9],[11]-[13], a nonlinear curve fitting approach was adopted for following two reasons:
- (i)  $\mu_{\max}$  can be estimated directly by nonlinear curve fitting, whereas using a linear regression approach gives rise to the problem with correlating slope  $k$  and the maximum friction coefficient  $\mu_{\max}$ . In reference [9], a comparison based on the results of [12], [15]-[17] shows that the  $k$ - $\mu_{\max}$  correlation has quite a large variation depending on different vehicle/tire combinations. Thus, this  $k$ - $\mu_{\max}$  correlation is not robust and not universally applicable.
  - (ii) Since the road condition classifier is desired to be more sensitive to the variation of road condition than that of vehicle/tire combination, linear regression approach pose serious problems in this regard for real practice. Thus, we propose a parameterized classification algorithm, which makes use of the nonlinear regression slip curve, such that the associated thresholds in the algorithm can be easily calibrated to fit each vehicle/tire combination.

Although the nonlinear regression approach tends to underestimate the value of  $\mu_{\max}$  especially when friction demands are low to medium [9]-[11], this limitation does not pose a serious problem for our classification, as explained further in this section.

As for the detection of rough surfaces, we developed a separate system which utilizes four highpass filters in parallel, mainly for identifying gravel roads. Due to the random contribution of the rough surface texture, the wheel speed measurement contains the information of “true wheel speed” and “random noise.” For very coarse surfaces such as gravel, the presence of random noise in the measurement is prominent, i.e. wheel speed signals on gravel roads are noisy. Thus, the proposed method is based on signal process technique and analysis of the observed noise.



We also address some problems with sensor fusion and signal processing. The main topics include (1) synchronization/integration of all on-board sensors (2) synchronization of GPS with on-board sensors and (3) calibration of bias error in yaw rate sensor using a Kalman filter. Sensor fusion plays an important role in this work, since we utilize GPS speed-based measurements for slip calculation and calibration of yaw rate bias, while GPS receiver uses an independent clock from on-board sensors'. Besides, there exists a time delay in GPS measurements compared to on-board sensor readings, and the sample rates of all sensors are not uniform. Therefore, the first two topics become critical issues in this work. Moreover, the intrinsic bias error existing in common inertial sensors such as accelerometer and gyro is inevitable and must be consistently calibrated in real-time. In this system, a Kalman filter based on a kinematic model using GPS heading angle measurement was utilized to calibrate the gyro bias.

We then present the development of the deployable program converted from post processing MATLAB codes. In addition, an extra interface program responsible for transforming data formats and exchanging data between the on-board PC's OS and the application program package, which was a C++ dynamic linking library, was addressed in this chapter.

The experimental results were then demonstrated. In field tests, the road condition monitoring system demonstrated the capability of distinguishing different slippery and coarse road surfaces including sand, dirt and gravel from asphalt roads.

Finally, we present conclusions and suggestions for future work. The advantages and drawbacks of this developed system— both in the theoretical and the practical domains are discussed. At the end of the thesis, several appendices discuss equipment and developed software that are important but are too involved in the main text.

### 3.3 Slip-Based Tire/Road Friction Estimator

The tire-road friction estimation problem has been extensively studied in literature, and the proposed methods can roughly be divided into “cause-based” approaches and “effect-based” approaches. The main idea of “cause-based” strategies is to try to measure factors that result in changes in tire-road friction and then attempt to predict  $\mu_{\max}$  based on past experience or friction models. On the other hand, “effect-based” approaches try to approximate  $\mu_{\max}$  based on the measurable effects on the vehicle or tire dynamics caused by the change of tire-road friction, namely, these approaches try to extrapolate the limit friction using available data.

Slip-based tire-road friction estimation is one of the effect-based approaches. The main advantage of this method compared to other alternatives is that there is no dedicated sensor required; it only needs a set of accessible measurements from production vehicle sensors, such as wheel speed, vehicle yaw rate and GPS speed measurements, etc. This merit makes this approach a favorable choice for the development of an on-board road condition monitoring system in EVII project, in which we want to realize the concept of using production vehicles as road condition probes. However, two main problems, i.e. robustness and calibration, with this approach pose serious difficulties for real practice. As a result, we intend to modify the former slip-based approach by (1) reducing the model dependence and (2) developing parameterized algorithm with physically meaningful thresholds to resolve these two problems.

For tire-road maximum friction coefficient  $\mu_{\max}$  estimation problem, the chosen method follows the so-called slip-based approach. The slip is defined as the relative difference of a driven wheel’s circumferential velocity,  $w_w r_w$ , and its absolute translational velocity,  $v_w$ :

$$s = \begin{cases} \frac{w_w r_w - v_w}{w_w r_w}; & (\text{vehicle is in traction, } w_w > 0) \\ \frac{v_w}{w_w r_w}; & (\text{vehicle is in braking, } v_w > 0) \end{cases} \quad (4)$$

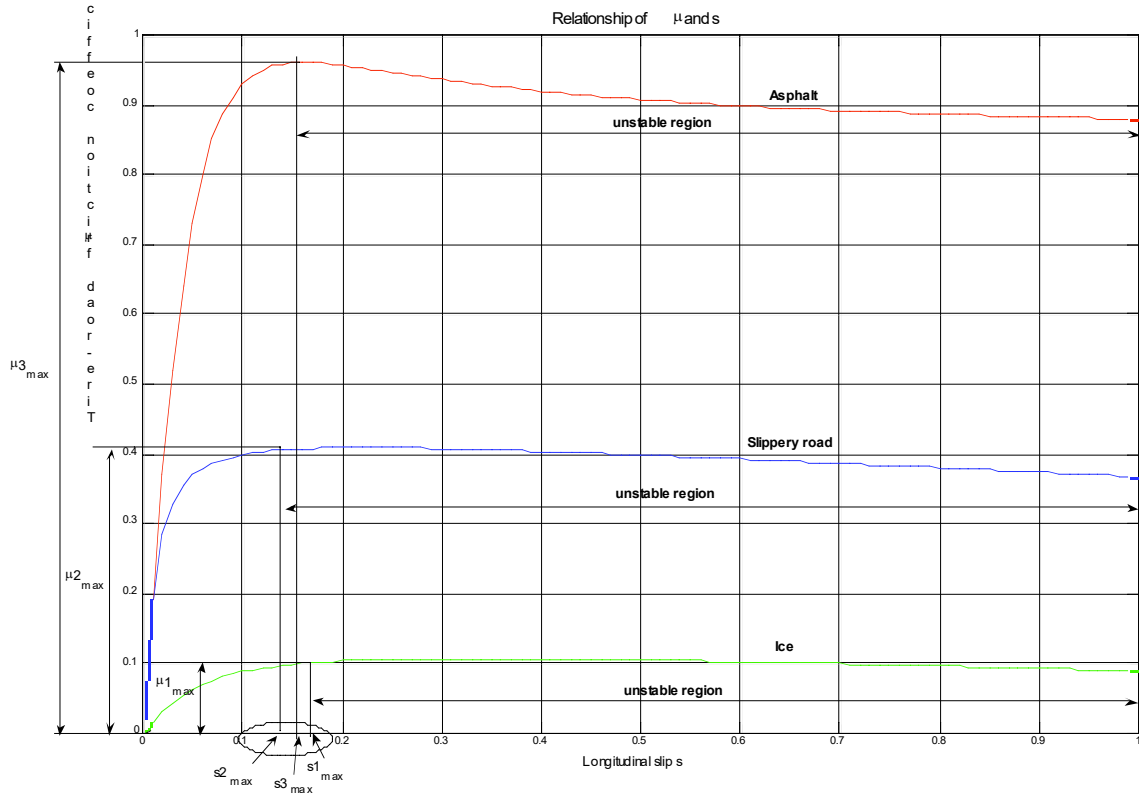
Where  $w_w$  is the wheel’s angular velocity, and  $r_w$ , is the effective wheel radius. By this definition, slip is between 0 and 1. Because only longitudinal friction force and slip are

considered in this work, the friction coefficient,  $\mu$ , is defined as the longitudinal tire-road friction force,  $F_x$ , normalized by the normal force,  $N$ :

$$\mu = \frac{F_x}{N} \quad (5)$$

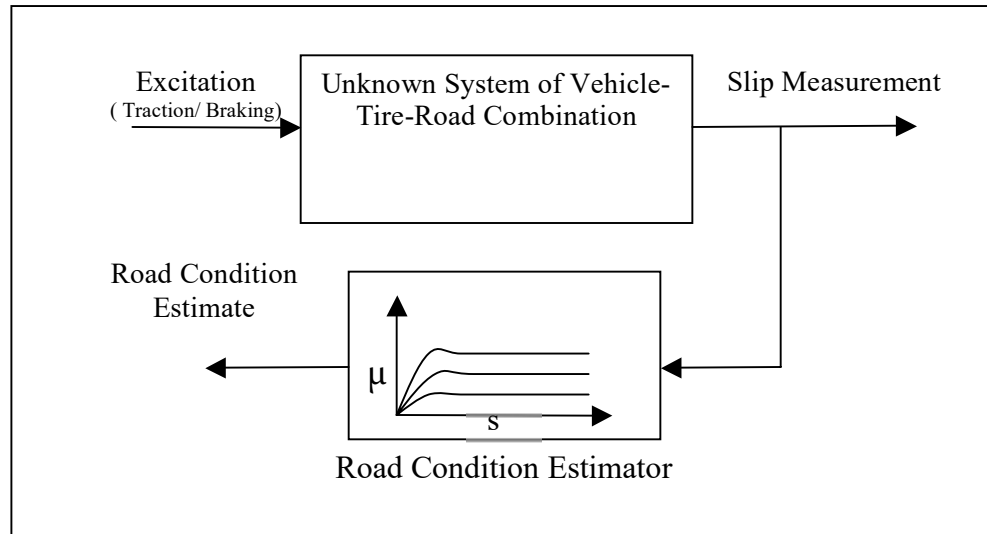
A schematic of  $\mu$ - $s$  curves for different surfaces is plotted in Figure 10. This plot depicts important characteristics of the relationship between friction coefficient and slip, which is exploited in our method for slippery road detection and classification in conjunction with reasonable assumptions for the use of probe vehicles. The three  $\mu$ - $s$  curves were generated by using the so-called ‘‘Magic Formula’’ proposed by Bakker et al. (1987). Characteristics of the  $\mu$ - $s$  curves include:

- 1) The friction coefficient,  $\mu = \mu(s)$ , is initially an increasing function of slip  $s$ . As  $\mu$  increases to the maximum value for each tire-road combination, the corresponding slip value reaches its critical point, denoted as  $s_{max}$ . Because slip is greater than this critical point, the curve enters the unstable region and  $\mu$  starts decreasing as  $s$  keeps increasing until  $s = 1$ , i.e. the wheel is purely rotating or locked in traction and braking case, respectively.
- 2) While the maximum friction coefficient  $\mu_{max}$  varies significantly for different tire-road combinations, the variation of the critical slip value  $s_{max}$  is relatively small. This phenomenon can be easily observed in the fitted curves of field test data. This observation provides a very important clue for the algorithm design of the road condition classifier. As illustrated in Fig.1, suppose the critical slip values,  $s1_{max}$ ,  $s2_{max}$ , and  $s3_{max}$ , corresponding to three slippery level surfaces are bounded in the elliptical region, and assume this boundary can be found in collected data. Then for the same friction demand, greater than a threshold set at  $\mu \bar{A} 0.1$ , it is possible to distinguish road surfaces, given slip values. Taking an ideal case as an example, if the friction demand  $\mu$  is about 0.1 and the main distribution of slip is greater than 0.1, the road surface is slippery enough to be classified as ice. On the other hand, if the friction demand is very high during braking, say over 0.5, and the measured slip is below 0.1, the friction level is classified as asphalt.



**Figure 10. Schematic plot of  $\mu$ - $s$  curves for different surface.**

Figure 11 depicts the concept of slip-based approach for road condition estimation. Borrowing from [12], traction and braking forces are regarded as excitation to the unknown system, which includes varying road conditions and a slowly time varying vehicle-tire combination. Compared with the variance of road condition, the variance of the vehicle-tire model can be assumed to be negligible over short time intervals. However, the estimator still needs to be robust to model uncertainty and can be calibrated as needed.



**Figure 11. Concept system for the slip-based friction estimation.**

Based on above assumptions for the vehicle-tire-road system and the aforementioned characteristics of  $\mu$ -s relationship, the road condition estimator can be used to evaluate and classify the slippery level when sufficiently high excitation is applied. Namely, if there is no traction/braking force applied to the system, no road condition estimate will be made. Therefore, it is possible that a probe vehicle cannot detect a slippery segment during neutral maneuver. However, this estimation gap could be filled by other probe vehicles. From a global point of view, each probe vehicle plays the role as a “moving sensor” of the sensor network. Each probe vehicle can provide and share the useful information, allowing the possibility of the loss of detecting slippery segment to be compensated.

### 3.3.1 Approximation of the Tire-Road Friction Coefficient

In this work, only the longitudinal wheel slip and longitudinal acceleration are taken into account, allowing lateral tire-road friction forces to be neglected. Parameters such as wheel rolling resistance, aerodynamic drag force, and road grade, are assumed to be neglected. The relationship between acceleration/deceleration of the vehicle and traction/braking force can be derived using Newton’s Second Law. Moreover, using the so-called “bicycle model,” the tire-road friction coefficient can be approximated as a function of longitudinal acceleration only.

### In Traction

Since the adopted probe vehicle is rear wheel drive, and we assume the wheel rolling resistance can be neglected,

$$ma_x \approx F_{xr} \quad (6)$$

where  $m$  is the vehicle mass including all wheels,  $a_x$  is the longitudinal acceleration and  $F_{xr}$  is the total traction forces applied on rear wheels. Suppose the normal forces acting on the front and rear tires are equivalent, and the sum of them is approximately equal to the weight of the vehicle:

$$N_f \approx N_r \quad (7)$$

$$N_f + N_r \approx mg \quad (8)$$

$$N_r \approx 0.5mg \quad (9)$$

By definition, the friction coefficient is the traction force normalized by the normal force:

$$\mu = \frac{F_{xr}}{N_r} \quad (10)$$

Therefore, friction coefficient can be approximated by the longitudinal acceleration:

$$\mu \approx \frac{a_x}{0.5g} \approx 0.2a_x \quad (11)$$

### In Braking

Since all four wheels are braking simultaneously before ABS engages, we can similarly approximate friction coefficients for all wheels as follows,

$$\mu = \frac{F_{xr} + F_{xf}}{N_r + N_f} \approx \frac{ma_x}{2 \times 0.5mg} = \frac{a_x}{g} \approx 0.1a_x \quad (12)$$

### 3.3.2 Slip Calculation

Based on the definition in equation (4), slip calculation can be divided into two cases, (1) in traction and (2) in braking. For traction case, assuming two-wheel drive vehicles, the difference between front and rear wheel speed is normalized by the driven wheel speed to approximate longitudinal wheel slip. For braking case, GPS speed measurement was utilized to be the reference speed under the assumptions described below.

#### (1) In traction:

Since the adopted experimental vehicle is rear wheel drive, the slip at front wheels is assumed negligible compared with the slip at rear wheels in traction. So the translational speed of front wheels serve as the reference speed,  $v_w$ , in equation (4):

$$s = \frac{W_{w,r}r_{w,r} - v_{w,f}r_{w,f}}{W_{w,r}r_{w,r}} \quad (13)$$

#### (2) In braking:

The reference speed  $v_w$  is calculated from the GPS speed measurement  $v_{GPS}$  and gyro yaw rate,  $\omega_{yaw}$ . Due to the intrinsic drifting error in the gyro measurement, the bias in yaw rate must be calibrated in real time. Besides, the time delay between GPS measurements and other on-board sensors also need to be taken into account. The associated sensor fusion technique will be presented later.

The formula for slip calculation in braking is the same as equation (4), but the reference speed  $v_w$  is calculated as follows:

$$v_w = v_A + \omega \times L_{A/W} \quad (14)$$

where  $v_A$  represents the GPS speed measurement assuming the GPS antenna is installed on top of the vehicle's C.G.,  $\bar{A}$  is the calibrated yaw rate, and  $L_{A/W}$  is the vector pointing from GPS antenna to the wheel hub. Since only yaw rate sensor is available on the experimental vehicle,  $\bar{A}$  does not take pitch and roll rates into account.

### 3.3.3 Algorithm for Slippery Road Classification

Instead of directly using the  $\mu$ -s relationship to design the algorithm for slippery road classification, we approximate  $\mu$  using acceleration/deceleration of the vehicle by equations (11) and (12). Therefore, the associated thresholds of  $\mu$  in the algorithm were replaced by available acceleration/deceleration of the vehicle. This approximation can be interpreted by the following inequality equation:

$$|a_x|_{\max} = \max \frac{|F_{x_{r1}} + F_{x_{r2}} + F_{x_{f1}} + F_{x_{f2}}|}{m} \leq \mu_{\max} g_x \quad (15)$$

where  $F_{x_{ij}}$  is the longitudinal force at the  $ij$ th wheel,  $m$  is the mass of the vehicle,  $g$  is the gravitational constant, and  $\mu_{\max}$  is assumed to be the same at all tires. Hence, the measured maximum acceleration gives a lower bound to the maximum friction coefficient  $\mu_{\max}$ . On the other hand, the maximum friction coefficient gives an upper bound to the maximum available acceleration. Therefore, if the measured maximum available deceleration is  $0.5g$  in one segment of highway, the maximum friction coefficient can be approximated to be at least greater than  $0.5$  using equation (12).

Figure 12 shows the proposed algorithm for classification of road surface's slipperiness. In this algorithm, there are three slip thresholds  $S_1 - S_3$ , three acceleration thresholds,  $A_1 - A_3$ , and one data number threshold  $N$ . With two-wheel drive vehicles, the relation between equations (11) and (12) dictates that deceleration thresholds are double acceleration thresholds. These values were obtained from field tests with our experimental apparatus. This classification algorithm is explained below:

Step 1: Under normal driving conditions on high friction surfaces like asphalt (Grade 1), the slip values are usually below 3%. This step should address most of the classification needs for driving on the highway, given good weather and mild maneuvers. This keeps the computational load low.



Step 2: This step identifies exceptionally slippery surfaces as ice and oil patches. When detecting anomalously high slip values while the acceleration/ deceleration is small, the outcome is that surfaces are judged as the most slippery level, Grade 3.

Step 3: This step precludes cases of poor excitation without prominent slip values from further evaluation. Due to poor excitation, namely low friction demand, the differences between slip values on asphalt and slippery road are ambiguous. Therefore, Grade 0 is given; this connotes inconclusive data for classification.

Step 4: This step checks if the number of data points during each excitation interval  $I(i)$  is enough for curve fitting, where the time interval  $I(i)$  is defined as the duration between two zero acceleration point:

$$I(i) [T_i, T_{i+1}], i = 1, 2, 3, \dots \Rightarrow \begin{cases} a_x(T_i) = 0 \\ \forall a_x(t_i) \neq 0 \text{ if } t_i \neq T_i \end{cases} \quad (16)$$

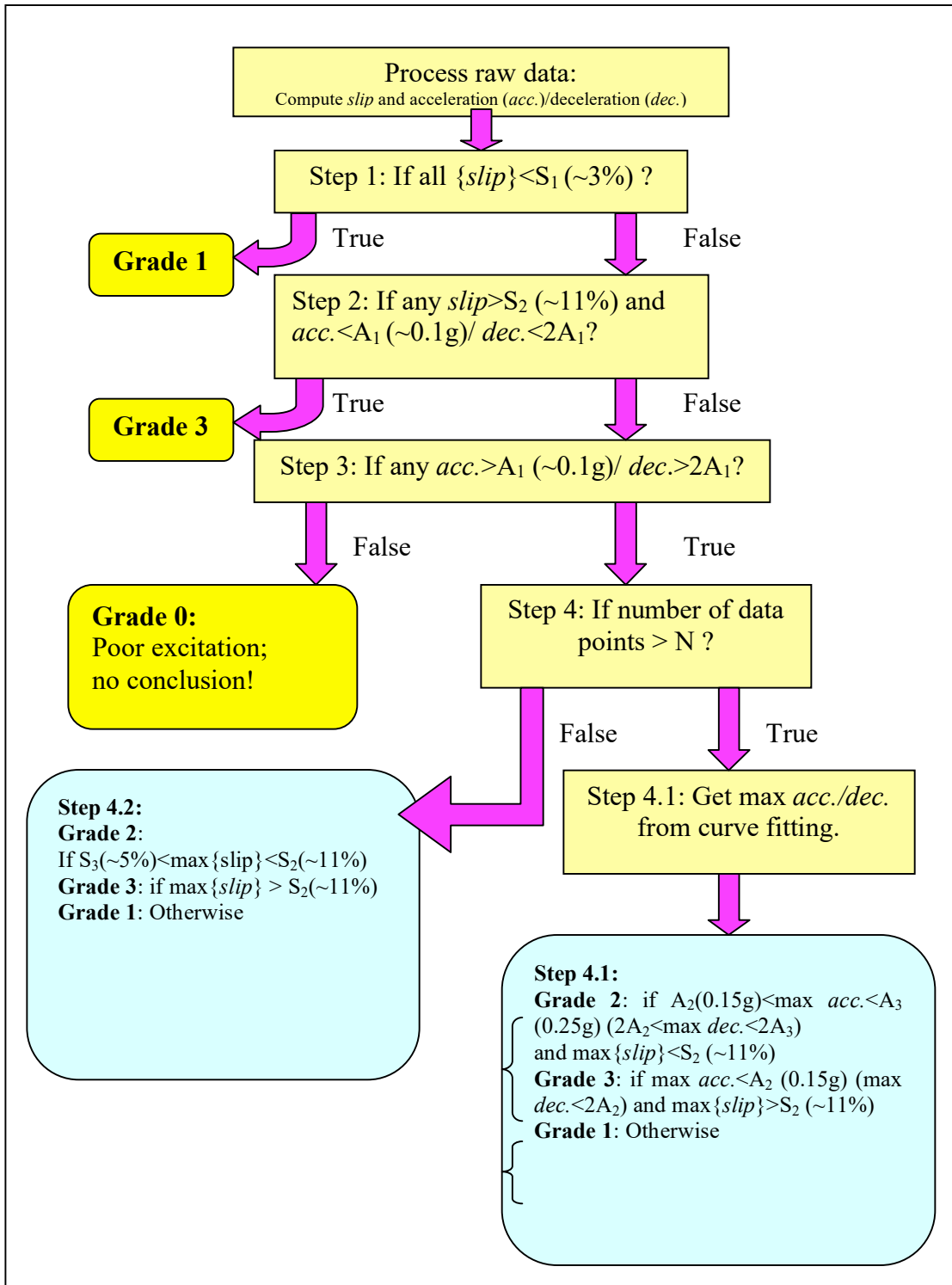


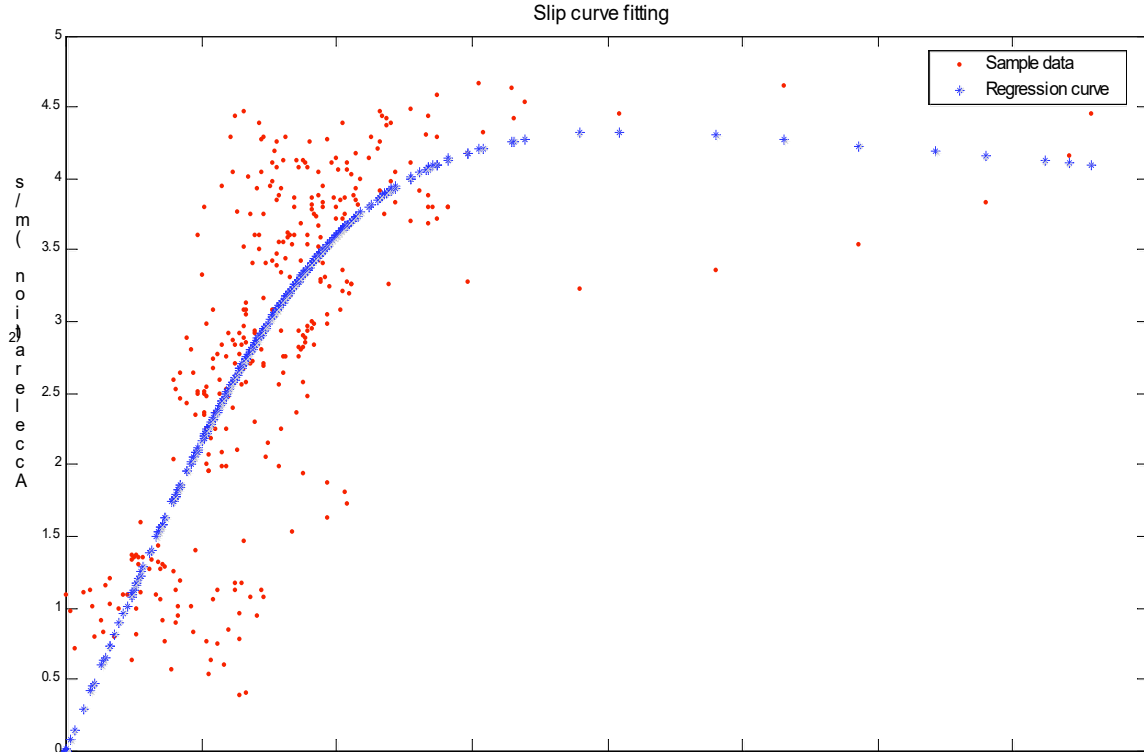
Figure 12. Flowchart of the algorithm for slippery level classification.

Since the adopted tire-road friction model for curve fitting is the so called “Magic Formula”,  $F_x = C_1 \sin(C_2 \tan^{-1}(C_3(1 - C_4)S + C_4 \times \tan^{-1}(C_3 S)))$  (17)

which has four parameters,  $C1—C4$ . The minimum number of data points is set to four. However, for better curve fitting results, this threshold  $N$  could be set to a greater number such that the fitted curve can have better noise rejection performance, i.e. this fitted curve can avoid being misled by some extreme points if the clustered mainstream points are relatively more than those “bad points” due to measurement noise.

Step 4.1: If excitation is high enough and the number of data points is also sufficient for curve fitting, the maximum available acceleration/deceleration is approximated by a nonlinear curve fit to the Magic Formula. Based on the approximated maximum available acceleration/deceleration and measured maximum slip value, the road condition is classified into three levels, Grade 1—3. Figure 13 demonstrates one example of the nonlinear curve fitting of data to the Magic Formula.

Step 4.2: This step deals with the case when the quantity of data is insufficient for curve fitting, but excitation is high enough. This is mainly for use in braking due to relatively lower update rate of GPS measurements (5Hz) compared with wheel speed sensor (50Hz).



**Figure 13. Nonlinear Curve Fitting Example**

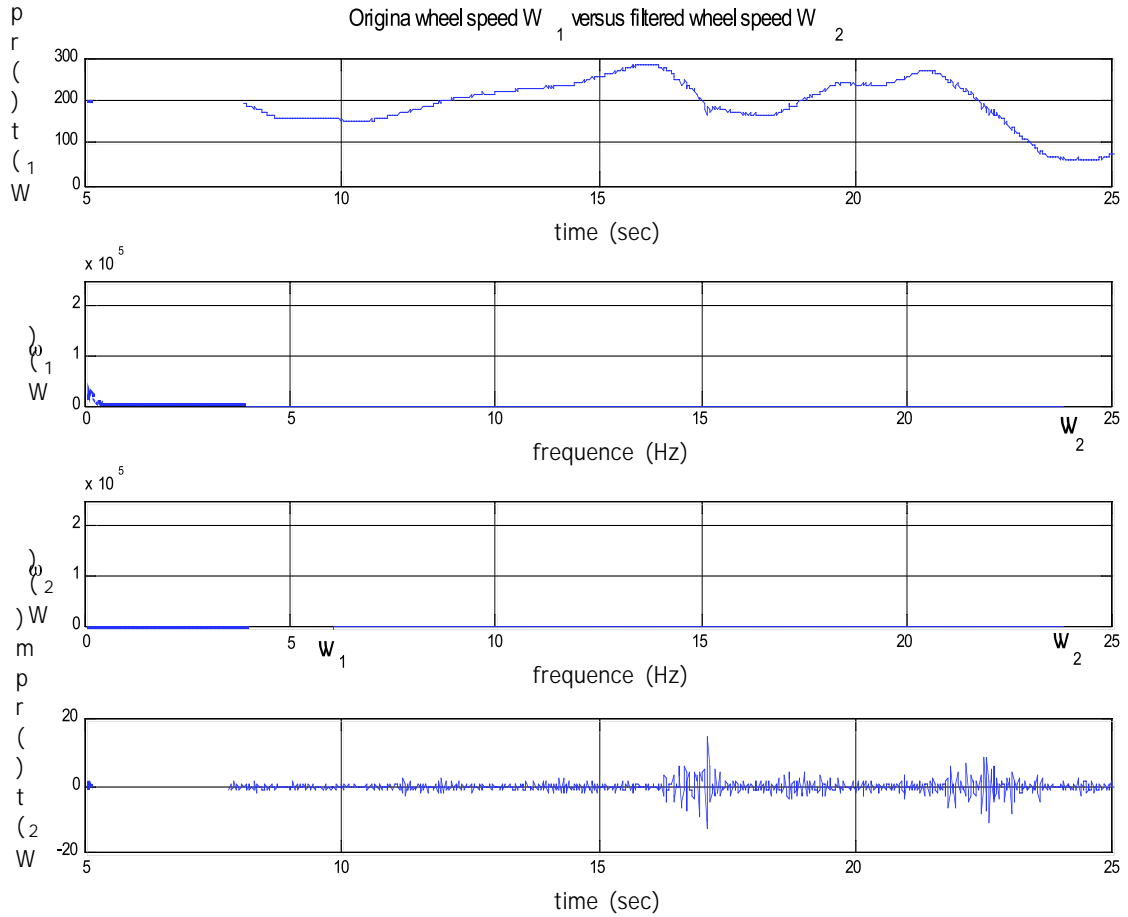
### **3.4 Rough Road Surface Detector**

Due to the rough texture of gravel surfaces, it is difficult to distinguish gravel from asphalt using only a slip-based approach. Therefore, a road condition classifier based on frequency domain signal process was developed.

#### **3.4.1 Frequency Domain Analysis of Wheel Speed Signals**

Figure 14 gives the plot of raw wheel speed signal versus filtered wheel speed signal in time and frequency domains. The vehicle began on an asphalt road and entered into a gravel region after the 16th second. The effect of the surface coarseness on the wheel speed signal,  $W_1(t)$  is clearly observed. The second plot is the frequency response of the original wheel speed signal,  $W_1(\bar{A})$ , and its response above  $w_2$  (where  $w_2$  is Nyquist frequency) is not shown. By filtering out the low frequency portion ( $\bar{A} < w_1$ ), the filtered signal,  $W_2(t)$ , (shown in the bottom plot) manifests the random contribution made by the

surface roughness and shows the potential use of its variance to evaluate surface coarseness.



**Figure 14. Raw Wheel Speed Signal vs. Filtered Wheel Speed Signal in Time and Frequency Domains**

### **3.4.2 Road Surface Texture Evaluation Method**

Borrowing from [12], the freely-rolling wheel angular velocity  $w_w$  has the following relationship with its longitudinal translational speed:

$$w_w = v_w / r_w + n \quad (18)$$

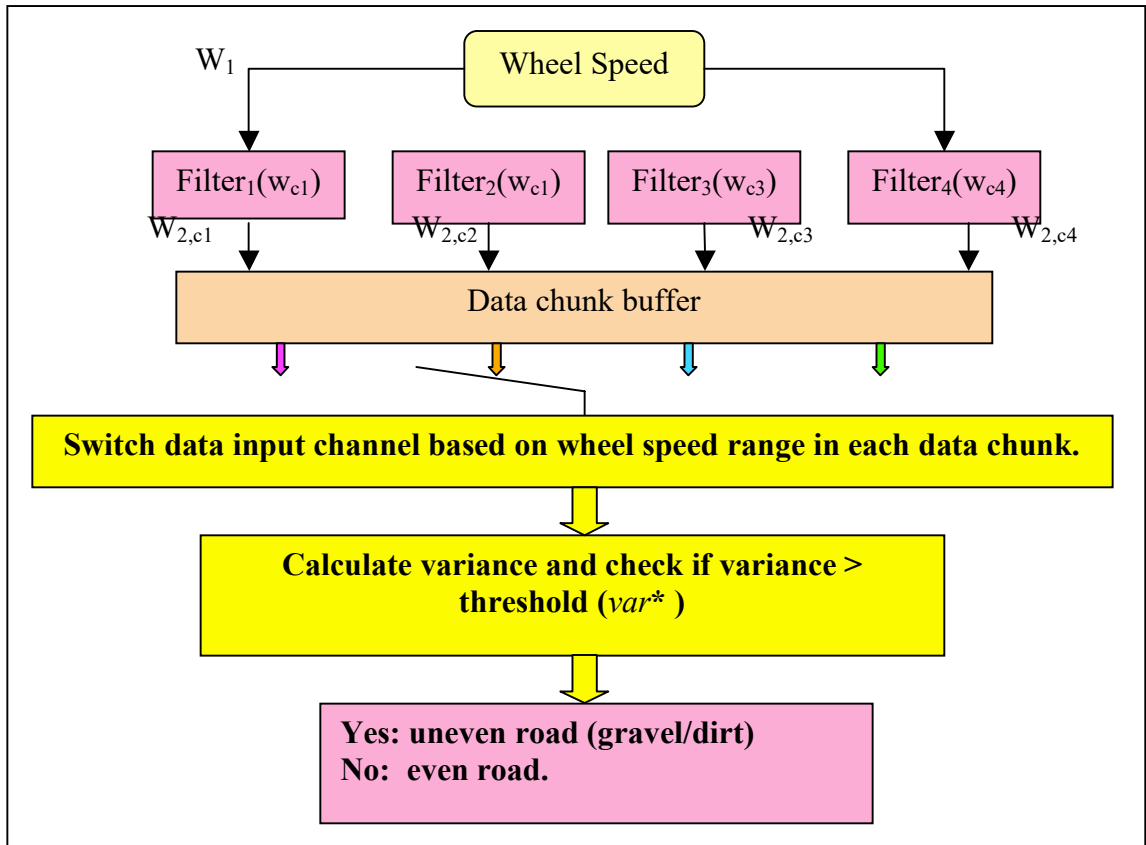
where  $r_w$  is effective tire radius, and  $n$  is measurement noise. On a perfectly even road surface  $w_w$  is equivalent to  $v_w/r_w$ . This model can be leverage to allow the variance of the filtered wheel speed signal to be an indicator of surface coarseness.

Since vehicle body has relatively larger inertia than the wheel, its dynamics has relatively lower frequency dominant mode, i.e. the wheel dynamics and vehicle body dynamics evolve on significantly different time scales. Thus using equation (18), if the wheel speed signal  $w_w$  is high-pass filtered with a cutoff frequency  $w_1$  covering the main spectrum of  $v_w/r_w$ , then the filtered signal is mainly composed of the random contribution of surface roughness and can be used to evaluate the surface coarseness.

However, the choosing the cutoff frequency for the high-pass filter is difficult. It was found that a single pair of cutoff frequency plus variance threshold cannot work well for the full range of speeds. Therefore, our surface roughness estimator utilizes four high-pass filters in parallel and one variance threshold. The associated cutoff frequency,  $w_{c1}$ — $w_{c4}$ , and the variance threshold,  $var^*$ , must be found from experiments.

As shown in Figure 15, the surface texture evaluation process is described:

1. Wheel speed data is fed into four high-pass filters with four different cutoff frequencies,  $w_{c1} = 6$  Hz,  $w_{c2} = 7$  Hz,  $w_{c3} = 12$  Hz,  $w_{c4} = 16.5$  Hz, in parallel.
2. Filtered data from each filter (channel) is grouped into small chunks in the buffer, and each chunk contains 50 data samples. (The adopted wheel speed sensor has 50Hz update rate.)
3. Based on the original wheel speed associated with the time interval of each chunk, the switcher takes data from a suitable channel. The speed thresholds are 200 rpm, 300 rpm, and 400 rpm. For example, if the maximum wheel speed during the  $i^{\text{th}}$  time interval  $(t_i, t_{i+1})$  is 180 rpm, the switcher takes the filtered data from the first filter/channel.
4. Variance is calculated and checked for each data chunk, so the surface coarseness evaluation is made for every one second.
5. The variance threshold  $var^*$  was found to be 2.25 from field tests.



**Figure 15. Road Surface Coarseness Evaluation**

### 3.5 Experimental Results

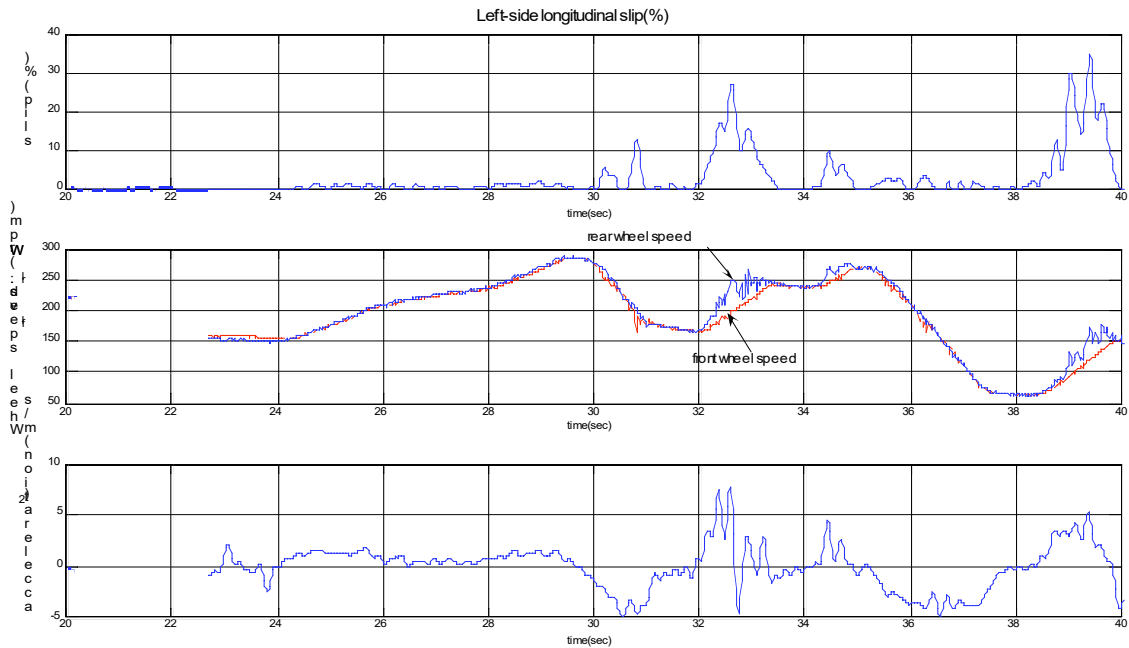
Figures 16 - 19 describe one test conducted at the UC Berkeley's Richmond Field Station. In the test, the vehicle began by moving on a concrete road covered with sand, and it entered into a gravel area and traveled on that for around 10 seconds.

Figure 16 shows wheel slip values using left two wheel speed sensor measurements, and Figure 17 shows wheel slip on two rear wheels using the GPS speed measurement and two rear wheel speed sensors. Both of these two plots show that sandy road (20~30 sec) could be identified and classified as second slippery level road using slip-based approach since slip values tend to be greater than 3% while acceleration/deceleration is small. However, it is difficult to distinguish gravel road (after 30 sec) from asphalt road using slip-based approach in that both of slip value and acceleration/deceleration can be very

large due to the characteristics of tire-road friction on rough texture of gravel road surface.

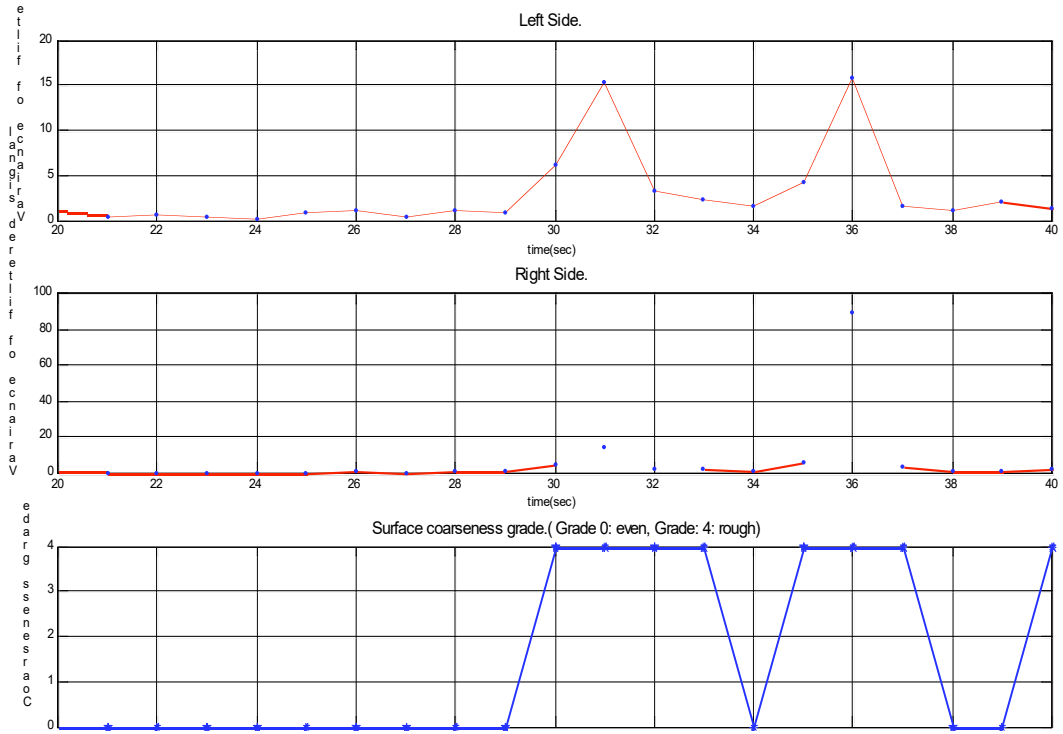
Figure 19 displays the surface coarseness evaluation results. Between 30 - 40 second, there are two spikes in the variance value, so it demonstrates that the variance of filtered wheel speed increases in correspondence with the increase of surface roughness.

On the left plot of Figure 19, the surface slipperiness classification results were displayed along with the vehicle's moving path by means of GPS position. It indicates that gravel roads are not distinguishable from asphalt in terms of slipperiness; however, it is easy to detect gravel roads using the variance of filtered wheel speed approach. This is shown on the right plot of Figure 19. The slippery sandy road was detected quite easily from the start point as shown on the left plot.



**Figure 16 . Slip Calculation Using Wheel Speed**

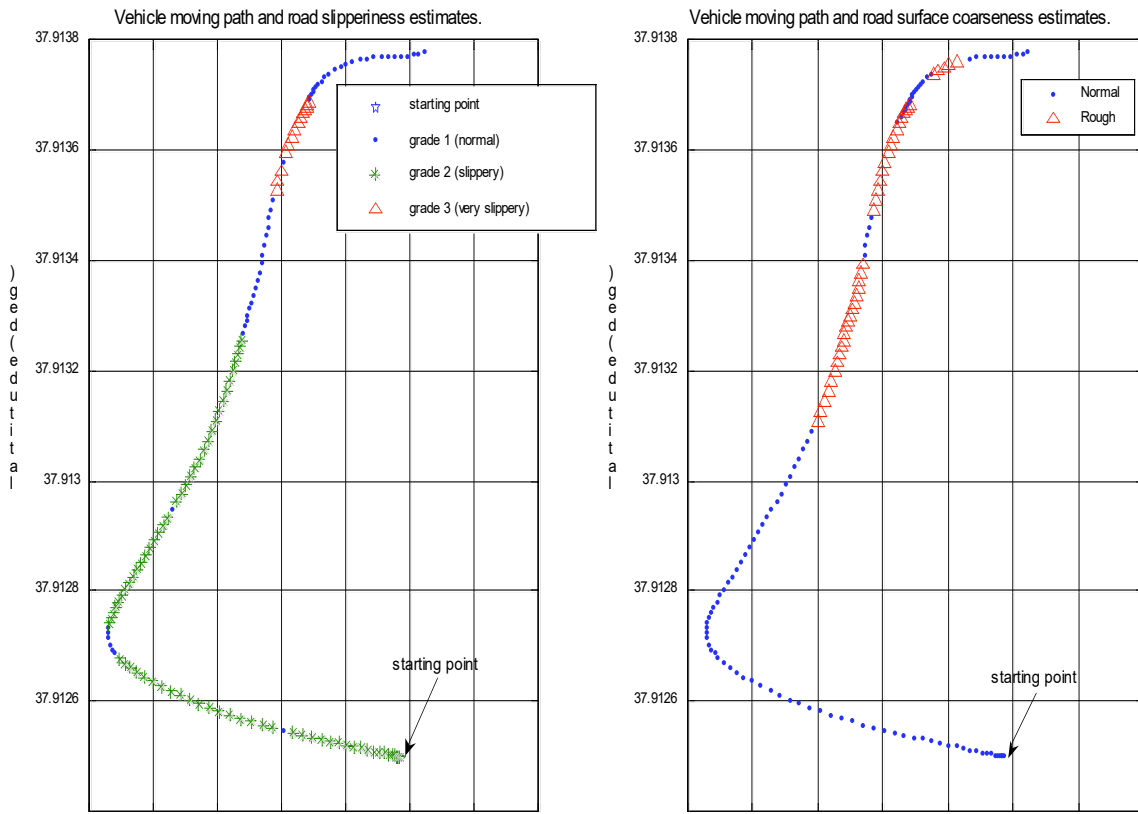




**Figure 17. Slip Calculation Using GPS Speed and Wheel Speed**



**Figure 18. Surface Coarseness Evaluation Results**



**Figure 19. Probe Vehicle Moving Path and Road Condition Classification**

### 3.6 Development of Application Program Package

In this work, the software development was first conducted in MATLAB, and the proposed algorithms for road condition classification were verified in a post-processing manner. As shown in Table 1, there are ten MATLAB programs developed in this work. After validating the algorithms, these M files were automatically converted to a C++ DLL package, shown in Table 2. However, due to the compatibility problem with the data formats in C++ and MATLAB, an interface program called SafetyDriver.cpp was developed.

This interface program is responsible for communicating between the road condition monitoring program and its higher-level master program. Therefore, the developed

application program only executes computation when requested by the higher-level master program. The main advantage of this kind of interaction is that the master program is able to decide when and how many application programs will be executed at any time. Since the developed road condition monitoring system is one of the applications implemented on the probe vehicle, it is very important to assure every application program will not interrupt one another.

**Table 1. List of developed MATLAB programs**

	<b>Program name</b>	<b>Function</b>
<b>1</b>	primary.m	Main program of the application software package.
<b>2</b>	import_data.m	Synchronize and integrate all data.
<b>3</b>	slip_acc.m	Calculate longitudinal slip in traction.
<b>4</b>	slip_brak.m	Calculate longitudinal slip in braking.
<b>5</b>	tire_radius.m	Estimate tire radius.
<b>6</b>	long_acceleration.m	Numerically compute the longitudinal acceleration.
<b>7</b>	filter_model_response.m	Filter
<b>8</b>	fun_slip_curve.m	Magic formula for nonlinear regression.
<b>9</b>	curve_fitting.m	Nonlinear regression program.
<b>10</b>	coarseness_analysis.m	Evaluate coarseness of road surface.

**Table 2. Generated C++ DLL Program Packages Converted from Original MATLAB M Files**

	<b>File name</b>
<b>1</b>	libprimary.cpp
<b>2</b>	libprimary.dll
<b>3</b>	libprimary.ctf

<b>4</b>	libprimary.lib
<b>5</b>	libprimary.h
<b>6</b>	libprimary.exp
<b>7</b>	libprimary.exports
<b>8</b>	libprimary_mcc_component_data.c
<b>9</b>	mccExcludedFiles.log

## 5.0 Conclusions

Two elements were demonstrated: a probe applications which tied vehicle speed data to an internet user, utilizing EVII-installed OBU and RSU and the existing PeMS backhaul and infrastructure, then two subelements – slip-based road condition and also surface roughness – comprised methods in road condition evaluation.

The EVII demonstration of the probe application shows great promise to supplement Caltrans-owned data base – PeMS – with VII- or DSRC-based probe data. Similar promise was shown with road condition monitoring system, since it demonstrated capability of detecting slippery and rough surfaces on line, or in real time.

In sum, the EVII project explored and sought resolution to key engineering issues associated with the point deployment of these services in a realistic setting. As such, it did pave the way for VII and the VII California project by in effect jump-starting technological work. Indeed, this project also demonstrated that Caltrans can do VII and create VII champions, within and outside of Caltrans.



## References

- [1] “Synthesis Report: Estimation of Target Vehicular Crashes and Potential Countermeasures.” Research Report, U.S. Department of Transportation, DOT-VNTSA-NHTSA-95-4, Volpe National Transportation Center, June 1995.
- [2] Godbole, Datta N., Raja Sengupta, James Misener, Natalia Kourjanskaia, and James Bert Micheal, “Benefit Evaluation of Crash Avoidance Systems”, Presented at TRB Annual Meeting, 1998.
- [3] Ioannou P.A. and C.C. Chien, “Autonomous Intelligent Cruise Control”, *IEEE Transactions on Vehicular Technology*, Vol. 43, No. 4, 1994, pp. 1125-1135.
- [4] Reichart G., R. Haller, and K. Naab, “Driver Assistance: BMW Solutions for the Future of Individual Mobility”, In *Proceedings of ITS World Congress*, Orlando, October 1996.
- [5] Jerry Woll, “Radar Based Adaptive Cruise Control for Truck Applications”, SAE Paper No. 973184, Presented at SAE *International Truck and Bus Meeting and Exposition*, Cleveland, Ohio, November 1997.
- [6] “Automotive Collision Avoidance System Field Operation Test: Phase I Interim Report”, U.S. Department of Transportation, DOT-HS-809-453, MAY 2002.
- [7] <http://www.leearmstrong.com/Dsrc/DSRCHomeset.htm>
- [8] <http://www.its.dot.gov/press/initiatives4.htm>
- [9] Steffen Müller, Michael Uchanski, and Karl Hedrick, “Estimation of the Maximum Tire-Road Friction Coefficient,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 125 pp. 607-617, December 2003.
- [10] U. Eichhorn and J. Roth, 1992, “Prediction and Monitoring of Tyre/Road Friction,” XXIV FISTA Congress, London, GB, 2:67-74, June 7-11.” Safety, the Vehicle, and the Road.”
- [11] Jingang Yi, “A Fault Tolerant Longitudinal Control and Tire/ road Friction Estimation System for Automated Highway Systems (AHS),” Ph.D. dissertation, Dept. Me. Eng., University of California, Berkeley, 2002.
- [12] Fredrik Gustafsson, “Slip-based Tire-Road Friction Estimation,” *Automatica*, vol. 33, No. 6, pp. 1087-1099, 1997.
- [13] C. Lee, K. Hedrick, and K. Yi, “Real-Time Slip-Based Estimation of Maximum Tire-Road Friction Coefficient,” *IEEE/ASME Transactions on Mechanics*, vol. 9, No. 2, June 2004.

- [14] M. Uchanski, "Road Friction Estimation for Automobiles Using Digital Signal Processing Methods," Ph.D. dissertation, Dept. Me. Eng., University of California, Berkeley, 2001.
- [15] W. Hwang, and B. Song, 2000, "Road Condition Monitoring System Using Tire-Road Friction Estimation," *Proceedings of AVEC 2000, Ann Arbor, Michigan*, pp. 437-442, August 22-24.
- [16] Yi, K., K. Hedrick, and S.-C. Lee, "Estimation of Tire-Road Friction Using Observer Based Identifiers," *Vehicle System Dynamics*, 31, pp. 233-261, 1999.
- [17] S. Müller, M. Uchanski, and K. Hedrick, "Slip-Based Tire-Road Friction Estimation During Braking," *Proceeding of IMECE' 01 (2001 asme International Engineering Congress and Exposition)*, November 11-16, 2001.
- [18] D. M. Bevly, J. C. Gerdes and C. Wilson, " The Use of GPS Based Velocity Measurements for Measurement of Sideslip and Wheel Slip," *Vehicle System Dynamics*, vol. 38, No. 2, pp. 127-147, 2002.



## Appendix A: DaimlerChrysler Vehicle Preparation

From November 2004 through October 2005, DaimlerChrysler Research and Technology North America, Inc. (DCRTNA) carried out research works as defined in the “Statement of Work” of the EVII project contract between U.C. Berkeley and DCRTNA (SUBAGREEMENT NO. SA4742).

This appendix documents DCRTNA’s efforts in the EVII project.

### *Task 1 Requirement Definition*

Two types of applications were identified and chosen for demonstration by the EVII project: traffic application and safety application. DCRTNA worked closely with PATH in defining data elements, data format and interface standard that were needed for the applications and demonstrations in the EVII project.

### GPS Data

Traffic application in the EVII project was mainly based on GPS data. Standard GPS data comes in different formats that serve different purposes. NMEA (National Marine Electronics Association) 0183 (or NMEA for short) is a combined electrical and data specification for communication between marine electronics and also, more generally, GPS receivers. What are of interests for traffic applications normally include UTC time, position (longitude, latitude, altitude), position error estimate, speed and heading. Among the various GPS NMEA messages, we used a combination of RMC and GGA messages in the project.

GGA data is essentially fix data that provide 3D location and accuracy data. The format of a GGA message looks like this:

```
$GPGGA,hhmmss.ss,llll.ll,a,yyyyy.yy,a,x,xx,x.x,x.x,M,x.x,M,x.x,xxxx*hh
```

1 = UTC of Position

- 2 = Latitude
- 3 = N or S
- 4 = Longitude
- 5 = E or W
- 6 = GPS quality indicator (0=invalid; 1=GPS fix; 2=Diff. GPS fix)
- 7 = Number of satellites in use [not those in view]
- 8 = Horizontal dilution of position
- 9 = Antenna altitude above/below mean sea level (geoid)
- 10 = Meters (Antenna height unit)
- 11 = Geoidal separation (Diff. between WGS-84 earth ellipsoid and mean sea level. -=geoid is below WGS-84 ellipsoid)
- 12 = Meters (Units of geoidal separation)
- 13 = Age in seconds since last update from diff. reference station
- 14 = Diff. reference station ID#
- 15 = Checksum

RMC is recommended minimum specific GPS/TRANSIT data, and its format is:

`$GPRMC,hhmmss.ss,A,llll.ll,a,yyyy.yy,a,x.x,x.x,ddmmyy,x.x,a*hh`

- 1 = UTC of position fix
- 2 = Data status (V=navigation receiver warning)
- 3 = Latitude of fix
- 4 = N or S
- 5 = Longitude of fix
- 6 = E or W
- 7 = Speed over ground in knots
- 8 = Track made good in degrees True
- 9 = UT date
- 10 = Magnetic variation degrees (Easterly var. subtracts from true course)
- 11 = E or W

12 = Checksum

If only one NMEA format can be chosen for traffic application, RMC will be the good choice, as it provides most of the needed information, such as time, longitude, latitude, speed and heading. In the EVII project, we also used GGA data, which additionally provides altitude and GPS error estimate. The two messages can be synchronized with the UTC time.

We used a CSI MAX GPS receiver in the project, which provides DGPS GGA and RMC with 1~2 m positioning accuracy at 5 Hz.

### Vehicle CAN bus data

Vehicle dynamic parameters are available from vehicle CAN buses, and can be used for road safety applications, such as road friction identification. We provided a set of parameters that was used in PATH's road friction model. Some of the parameters that we initially provided, such as the four wheel's suspension levels, engine speed, current gear, ESP status were not used in PATH's final model. The final set of parameters that we provided for road friction modeling is shown in the following table. Please note the vehicle speed in the table is obtained from the vehicle bus. The vehicle speed signals from the CAN bus occur at a higher frequency than from GPS and are more accurate when the vehicle is static or at low speed.

Parameter	Unit	Range	Signal Interval
Wheel_speed_front_left	1/Min	0 – 8191	21ms
Wheel_speed_front_right	1/Min	0 – 8191	21ms
Wheel_speed_rear_left	1/Min	0 – 8191	21ms
Wheel_speed_rear_right	1/Min	0 – 8191	21ms
Lateral_acceleration	m/s <sup>2</sup>	-10.24 – 10.08	14ms
Longitudinal_acceleration	m/s <sup>2</sup>	-10.24 – 10.08	14ms
Steering_wheel_angle	Degree	-2048 - 2047	10ms
Vehicle_speed	Km/h	0-511.99	21ms
Raw_Signal_yaw_rate	Degree/s	-327.68 – 327.66	21ms
Throttle_Position	%	0 – 100	20ms
Brake_torque_level	Nm	0 – 12282	21ms

--	--	--	--



*Probe Data Format*

UDP broadcasting was chosen as the means of DSRC communication between OBU and RSU. The OBU and RSU need to agree on a common format so that they can understand each other.

In parallel with the EVII project, both DCRTNA and California PATH were engaged in the VII California project, in which probe message set formats went through a definition and standardization processe. The final message set is now submitted to SAE for standardization. EVII serves as the first real project in which the VII message sets were successfully tested and used.

The VII California probe data message uses binary encoding and segments the probe data into snapshots. In the EVII project, each GPS data point ( a combined NMEA GGA and RMC ) was encoded in a snapshot. In EVII, the “number of vehicle device status fields” was set to zero for probe data applications. The format of the probe data message is shown in the following table.

Vehicle dynamics parameters could have been encoded in the VII California probe message format if they were intended to be transmitted to the RSU’s. But the quantity of the message will be huge due to the very high frequency of the message. It was decided the road friction modeling is done in the onboard unit, and only the result of the road friction classification is transmitted to the RUS. The result of road friction classification from the OBU calculation can be encoded as a vehicle status parameter in a snapshot in the VII message format.

Probe Message								
Bit	8 (msb)	7	6	5	4	3	2	1 (lsb)
1	Application ID [1]							
2	Message Version Number (0 - 255)							
3	Length of Message in Bytes (0 - 65,535)							
4	Length of Message including headers							
5	Number of milliseconds since Jan. 1, 2004 at 00:00:00 UTC							
:	"							
9	UTC Time LSByte (LSBit = 0.001 seconds)							
10	Time Precision [2]							
11	Positioning Precision [5]							
12	Longitude of center of vehicle MSByte							
:	"							
15	Longitude LSByte (LSBit = $10^{-7}$ decimal degrees; signed)							
16	Latitude of center of vehicle MSByte							
:	"							
19	Latitude LSByte (LSBit = $10^{-7}$ decimal degrees; signed)							
20	Heading MSByte							
21	Heading (LSBit = 0.01 degrees; signed; 0 degrees = North)							
22	Vehicle Speed MSByte							
23	Vehicle Speed LSByte (LSBit = 0.01 m/s; unsigned)							
24	Heading Precision [8]							
25	Speed Precision [9]							
26	Altitude - meters above sea level (unsigned), offset by 1 km (value of 0=1 km below sea level), LSBit=0.1meter							
27								
28								
29	Vehicle Type [4]							
30	Seconds since last Probe Message sent							
31								
32	Longitudinal Acceleration (LSBit = 0.01 m/s <sup>2</sup> ; signed)							
33								
34	Number of Vehicle Device Status Fields [22]							
35	Vehicle Status Device Type 1 [23]							
36								
37	Vehicle Status Device Value 1							
38								
39	Vehicle Status Device Type n [23]							
40								
?	Vehicle Status Device Value n							
?								
????	Snapshot 2							
:								
:								
????	Snapshot n							
:								

**Figure A-1. Probe Message**

***Task 2 Vehicle Data Interface / Test Data Provision***

Test Data Provision

There are two phases of test data provisioning. The first phase was at the early stage of the project during which a large amount of probe data were recorded in files on computer disks and

then provided to the PATH for road friction and traffic model development. In this phase, the emphasis was on trying to get an adequate amount data of various probe parameters under different weather and road conditions. We equipped a couple of RTNA cars and let people drive it for their regular commute between San Francisco and Palo Alto, and around the Palo Alto area.

GPS and CAN data were stored in files. The format of a data log file is:

```
Computer_system_time, Sensor type (CAN or GPS), Sensor_time list_of {parameter_name  
parameter_value}
```

Here is an example of the file:

```
Mon Jun 06 17:26:53.557665 2005, CAN, time 887 Wheel_speed_front_left 0  
Wheel_speed_front_right 0  
Mon Jun 06 17:26:53.597723 2005, CAN, time 4388 Brake_torque_level 12285  
Wheel_speed_rear_left 0 Wheel_speed_rear_right 0  
Mon Jun 06 17:26:53.567680 2005, CAN, time 912 Lateral_acceleration -0.24  
Mon Jun 06 17:27:47.855742 2005, GPS, secs 1949.48 lon -122.1493962 lat 37.41497667 alt  
17.5 hdop 1.9
```

The recorded data files were provided to PATH for road friction model development and validation. The GPS data is also valuable for traffic research and applications. About three Gigabytes of probe data were recorded and delivered to PATH over the course of the project.

The second phase of data provisioning was to support the end-to-end drive-by test and onboard road friction modeling. The focus was on the integration of the OBU with RSU and onboard road friction model validation and testing. We were successful in the end-to-end probe data transmission. A road friction model was also successfully developed by PATH and was tested on PATH road, based on the vehicle data provided by RTNA OBU.

## DSRC Capable OBU and Vehicles

Two EVII onboard data-recording units were developed and installed in RTNA vehicles for test data provisioning. Denso DSRC radios were connected to the onboard units in order to transmit recorded data to roadside units.

During the course of the EVII project, four RTNA vehicles participated in the data collection duties: a Mercedes E-5000, a Dodge Durango, a Dodge Magnum and a Jeep Cherokee (see Figure A-20). The Dodge Magnum, with its onboard data recording and DSRC transmitting equipment, was provided to PATH for onsite traffic and safety application testing for two separate weeks.

In October 2005 during the final EVII demonstration, The Dodge Durango, equipped with onboard equipment, was driven from Palo Alto to PATH at Richmond, and GPS data was collected every 50 meters in the VII California format. At the RSU along Intersection 80 in Berkeley, onboard traffic data was successfully relayed to the roadside unit through DSRC.

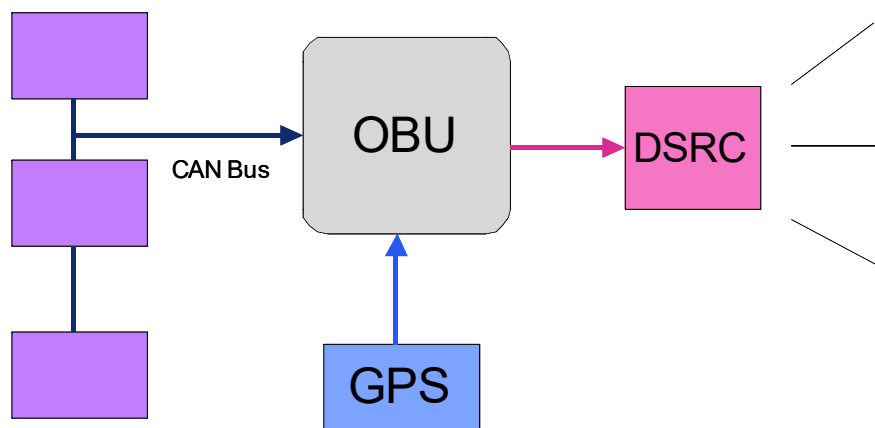


**Figure A-20 Mercedes E500 and Dodge Magnum cars used in the EVII project**

## Data Recording HW and SW

### Hardware

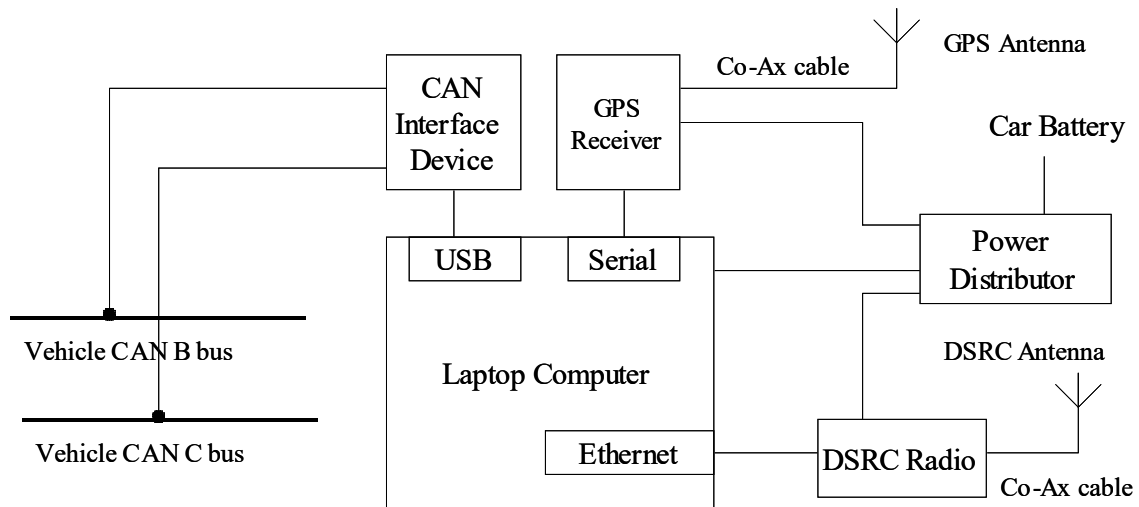
The EVII Data recording system developed by DCRTNA is composed of three functional components: data (CAN and GPS) interface devices, computer, and data transmitting device (DSRC radio), as illustrated in Figure A-21.



**Figure A-21 Simple schematic of OBU**

A more detailed description of the hardware is given in Figure A-22.





**Figure A-22 OBU hardware configuration detail**

We have used the following onboard hardware in the EVII project (Figure A-23):

- IBM Thinkpad T23 laptop
- Denso DSRC radio (Wireless Radio Module)
- Vector CanCaseXL for vehicle CAN interface
- CSI MAX DGPS receiver



**Figure A-23 Photo of vehicle onboard equipment**

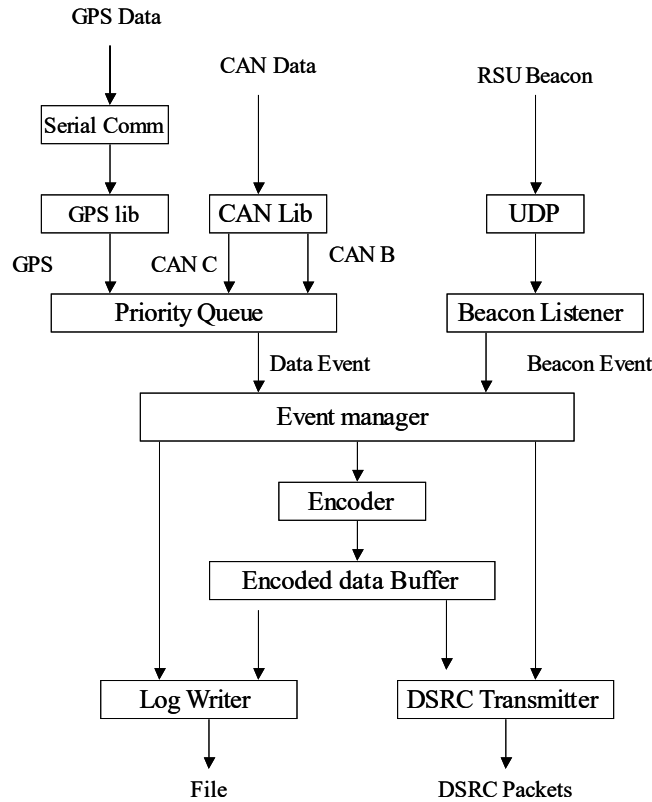
#### Software:

The software system for EVII project was developed on the Windows operating system using Microsoft Visual C++ .NET 2003. The reason to choose the Windows OS is because we wanted to use the Vector CAN device, which provides both high speed CAN (CAN C) and low speed CAN (CAN B). The Vector CAN device has very easy-to-use libraries, and has built-in filters for CAN messages. More importantly, it has its internal clock which can timestamp the CAN message, and this turns out to be very important to the road friction modeling. The Vector CAN device is only supported under the Windows OS.

The EVII onboard system includes software for two applications, traffic probe data collection and road friction related data recording. In addition, it has DSRC listening and transmission components. The onboard computer gets GPS data from the computer's Serial interface, and calculates the distance traveled by the vehicle based on the GPS position data. It records a GPS point about every 50 meters. The onboard computer also monitors CAN activities on both the

high speed and low speed CAN buses through a USB interface, and selectively records CAN parameters that are important for road surface friction modeling.

The control and data flow of the software logic is depicted in Figure A-24.



**Figure A-24 EVII OBU software control and data flow schematics**

***Task 3 Roadside Infrastructure Definition and Prototype Implementation***

DCRTNA supported PATH for RSU prototype implementation and testing.

- DCRTNA worked with PATH in reaching a common data and communication interface between OBU and RSU.
- DCRTNA and PATH worked closely in integration tests both on the traffic and safety applications.

- DSRC-capable DCRTNA vehicles were used by PATH for end-to-end performance test of the whole system.

Here is what happens during a drive-by scenario:

1. Caltrans RSU sends an “I am Here” beacon message at 5 Hz when it is in operation.
2. DCRTNA Vehicle starts data collection, and listens for RSU beacon.
3. DCRTNA OBU records a traffic data point every 50 meters, converts it in the VII California message format.
4. OBU gets beacon, starts sending all messages in computer buffer.
5. OBU repeats sending multiple times to enhance the chance of getting through.
6. RSU gets probe messages, forwards to PeMS database.
7. OBU leaves RSU range, clears message buffer, and starts recording new data.

This drive-by scheme turned out to be very effective, and a very similar version was used for the ITS World Congress VII California probe data demonstration.

During the RSU/OBU integration phase, several DSRC transmission schemes were tested. Based on the test result, the best way of transmitting DSRC messages were used in the EVII demo. This approach involves a pause of 5 milliseconds between consecutive DSRC messages, and repetition of DSRC transmission for a couple of times when in range. Based on our test result, the data loss rate is below 10 percent during the final demo drive-by, in which our vehicle was traveling at about 60 miles per hour on Interstate 80 when passing the EVII RSU.