



Vehicle-to-Vehicle Network Based Collaborative Navigation System

A thesis

submitted by

Xuan Xiao

In partial fulfillment of the requirements
for the degree of

Master of Science

in

Mechanical Engineering

TUFTS UNIVERSITY

August 2011

© 2011, Xuan Xiao

Advisor: Jason Rife

ABSTRACT

This research develops a Vehicle-Based Augmentation System (VBAS) for GPS that has the capabilities of Ground-Based Augmentation Systems (GBAS) but without fixed ground reference stations. The proposed system targets low-cost automotive applications. Existing augmented GPS systems which can provide centimeters level of accuracy are not suitable for automotive applications due to their high infrastructural cost (e.g. one reference station is needed at approximately every 20 km for GBAS). In contrast, by setting up a vehicle-to-vehicle (V2V) network, the proposed system can improve positioning accuracy simply by sharing measurement data among vehicles in the network. The cost of “infrastructure upgrade” will essentially be free, since V2V communication equipment and GPS receivers are expected to become standard equipment on automobiles in the near future. With assistance of such system, the expected accuracy of GPS could reach the 20-centimeter level, which is critical for automotive applications such as automated driving, automated lane keeping, and highway collision avoidance.

To build this system, a new algorithm for differentiating GPS measurements will be investigated. A vehicle-to-vehicle (V2V) network-based collaborative navigating method will also be tested to enable the new algorithm. Simulations, as well as a physical test system will be set up to demonstrate the capabilities of the new algorithm.

ACKNOWLEDGEMENTS

To my grandpa, I could not see you one last time, but I know you will be pleased to see I gain my Master's Degree. Thank you for your blessing.

Thank you to Professor Rife. Without your advice, I could not finish this project. I will remember your thousand-page post-doc notes because it reminds me that being a Ph.D. is not just doing what I feel interested and passionate but the organizational work is extremely important. Thank you again for sharing your knowledge.

Thank you to members in Club 204. Thank you for teaching me English, introducing me American culture, and sharing similar experience of childhood! You are the coolest guys I met in America!

Thank you to my parents and family for your financial and mental support along my way of graduate study.

Thank you to Lisa for encouraging me every time I need you. ☺

ABSTRACT.....	II
ACKNOWLEDGEMENTS	III
LIST OF TABLES	VI
LIST OF FIGURES.....	VII
ACRONYMS.....	IX
CHAPTER 1: INTRODUCTION.....	1
1.1. BACKGROUND	1
1.2. MOTIVATION FOR COLLABORATIVE NAVIGATION	2
1.3. THESIS CONTRIBUTIONS	3
1.4. THESIS OVERVIEW	4
CHAPTER 2: ALGORITHM DEVELOPMENT.....	6
2.1. REVIEW OF STANDARD GPS SOLUTION	6
2.1.1. <i>Basic position estimate model in GPS.....</i>	6
2.1.2. <i>Networked GPS measurements and its limitation.....</i>	9
2.2. COLLABORATIVE APPROACH FOR GPS POSITION SOLUTION.....	13
2.2.1. <i>Fusing lane-boundary and GPS measurements.....</i>	13
2.2.2. <i>Estimating satellite-specific GPS error with lane-boundary sensor.....</i>	16
2.2.3. <i>Multi-User Estimation Problem with Sharing Corrections.....</i>	22
2.3. SIMULATION.....	23
2.3.1. <i>Simulation Parameters</i>	23
2.3.2. <i>Single-User Simulations</i>	26
2.3.3. <i>Collaborative-Navigation Simulations.....</i>	27
2.3.4. <i>Decentralized Method Simulations.....</i>	30
CHAPTER 3: EXPERIMENTAL VERIFICATION OF ALGORITHM	32
3.1. GPS RECEIVER AND SOFTWARE	32
3.1.1. <i>SIRF Star chipset GPS receiver</i>	32
3.1.2. <i>Introduction of SiRFDemo.....</i>	36
3.1.3. <i>Procedure of Logging Data on SiRFDemo.....</i>	44
3.2. GROUND REFERENCE AND MAP SOURCES	45
3.3. EXPERIMENTS AND RESULTS.....	46
3.3.1. <i>Experiment Design and Setup.....</i>	46
3.3.2. <i>Experiment Routine</i>	53
3.3.3. <i>Experiment Results</i>	57
3.3.4. <i>Results Discussion and Error Analysis.....</i>	62
CHAPTER 4: CONCLUSION.....	66
4.1. THESIS CONTRIBUTIONS	66
4.2. SIGNIFICANCE AND FUTURE WORK.....	67

APPENDIX A: IMPLEMENTATION OF DATASETS LOGGED BY SIRFDEMO69

APPENDIX B: MATLAB PROGRAMS73

 B.1. STANDARD GNSS SOLUTION FOR SINGLE USER..... 73

 B.2. COLLABORATIVE NAVIGATION SOLUTION 75

 B.3. LOGGED FILE READING FUNCTION..... 77

 B.4. FUNCTION FOR EXTRACTING MEASUREMENTS OF A SPECIFIC SECOND..... 79

 B.5. FUNCTIONS OF CALCULATING USER POSITION..... 80

 B.6. SATELLITE POSITION ADJUSTING FUNCTION 81

 B.7. FUNCTION FOR ROTATING USER POSITION FROM ENU-FRAME TO ROAD-FRAME..... 81

BIBLIOGRAPHY.....83

LIST OF TABLES

TABLE I. LOCATIONS OF SIMULATED CARS	23
TABLE II. SIRF BINARY MESSAGES – OUTPUT MESSAGE LIST (PARTIAL EXCERPT)	33
TABLE III. MESSAGE ID 2: MEASURED NAVIGATION DATA OUTPUT (PARTIAL EXCERPT).....	34
TABLE IV. MESSAGE ID 7: CLOCK STATUS DATA (RECEIVER’S CLOCK).....	34
TABLE V. MESSAGE ID 28: NAVIGATION LIBRARY MEASUREMENT DATA (PARTIAL EXCERPT).....	35
TABLE VI. MESSAGE ID 30: NAVIGATION LIBRARY SV (SPACE VEHICLE) STATE DATA.....	36
TABLE VII. LATITUDE-LONGITUDE COORDINATES OF REFERENCE LINES ON COLLEGE AVENUE	49
TABLE VIII. LATITUDE-LONGITUDE COORDINATES OF REFERENCE LINES ON COUSENS PARKING LOT	50
TABLE IX. LATITUDE-LONGITUDE COORDINATES OF REFERENCE LINES ON BOSTON AVENUE	51
TABLE X. DATASETS FROM COLLEGE AVENUE, DIRECTION: SOUTHWEST TO NORTHEAST.....	55
TABLE XI. DATASETS FROM COLLEGE AVENUE, DIRECTION: NORTHEAST TO SOUTHWEST.....	56
TABLE XII. DATASETS FROM COUSENS PARKING LOG, DIRECTION: SOUTHWEST TO NORTHEAST.....	56
TABLE XIII. DATASETS FROM COUSENS PARKING LOG, DIRECTION: NORTHEAST TO SOUTHWEST.....	56
TABLE XIV. DATASETS FROM BOSTON AVENUE, DIRECTION: NORTHWEST TO SOUTHEAST	56
TABLE XV. DATASETS FROM BOSTON AVENUE, DIRECTION: SOUTHEAST TO NORTHWEST	56

LIST OF FIGURES

FIGURE 1. GEOMETRY FOR CAMERA-BASED LANE-BOUNDARY SENSING.....	15
FIGURE 2. SATELLITE AZIMUTH AND ELEVATION ANGLES FOR SIMULATION	24
FIGURE 3. CARS' POSITION AND ROAD PARAMETERS FOR SIMULATION.....	25
FIGURE 4. SIMULATED ERRORS FOR SINGLE-USER NAVIGATION ALGORITHMS.....	27
FIGURE 5. LATERAL ACCURACY FOR COLLABORATIVE NAVIGATION WITH A VARYING NUMBERS OF USERS EQUIPPED WITH LANE-BOUNDARY SENSORS.....	29
FIGURE 6. LATERAL ACCURACY FOR DECENTRALIZED COLLABORATIVE NAVIGATION ALGORITHM WITH A VARYING NUMBERS OF USERS EQUIPPED WITH LANE-BOUNDARY SENSORS.....	31
FIGURE 7. GPS RECEIVER	32
FIGURE 8. EXAMPLE OF MESSAGE ID 28.....	35
FIGURE 9. DATA SOURCE SETUP DIALOGUE BOX	37
FIGURE 10. MAIN USER INTERFACE	37
FIGURE 11. "SETUP" MENU	38
FIGURE 12. RECEIVER MODEL SELECTING WINDOW	38
FIGURE 13. "VIEW" MENU.....	39
FIGURE 14. "ACTION" MENU	40
FIGURE 15. LOGGING DATA FILE	40
FIGURE 16. INITIALIZING DATA SOURCE.....	40
FIGURE 17. "NAVIGATION" MENU	41
FIGURE 18. SELECTING DGPS MODE	41
FIGURE 19. SELECTING DGPS SOURCE	41
FIGURE 20. "POLL" MENU	41
FIGURE 21. MAIN USER INTERFACE	42
FIGURE 22. HOTKEYS ON MAIN UI	43
FIGURE 23. SIGNAL STRENGTH COLOR CODING.....	43
FIGURE 24. RELATIVE POSITIONS ON A MAP VIEW	44
FIGURE 25. GOOGLE MAPS "LATLNG MARKER" TOOL.....	45
FIGURE 26. TWO-CAR NETWORKED CASE.....	47
FIGURE 27. BICYCLE LANE MARKER AND LANE BOUNDARY ON COLLEGE AVENUE.....	49
FIGURE 28. LANE MARKER AND LANE BOUNDARY IN COUSENS PARKING LOT	50
FIGURE 29. LANE MARKER AND LANE BOUNDARY ON BOSTON AVENUE	50
FIGURE 30. EXAMPLE OF LATITUDE-LONGITUDE POSITIONS OF COLLEGE AVENUE (CREATED FROM GOOGLE MAPS)	52
FIGURE 31. EXPERIMENT HARDWARE SETUP AND ROUTINE DEMONSTRATION.....	53
FIGURE 32. EXPERIMENT ROUTE ON COLLEGE AVENUE	54
FIGURE 33. EXPERIMENT ROUTE IN SOUTHWEST CORNER OF COUSENS PARKING LOT	54
FIGURE 34. EXPERIMENT ROUTE ON BOSTON AVENUE.....	55
FIGURE 35. POSITION RESULTS COMPARISON (FROM ONE DATASET TAKEN ON COLLEGE AVENUE).....	58

FIGURE 36. DATA FOR STANDARD DEVIATION ANALYSIS..... 60
FIGURE 37. STANDARD DEVIATION OF LATERAL ERRORS (COLLEGE AVENUE)..... 61
FIGURE 38. " STANDARD DEVIATION OF LATERAL ERRORS (COUSENS PARKING LOT)..... 61
FIGURE 39. STANDARD DEVIATION OF LATERAL ERRORS (BOSTON AVENUE) 61
FIGURE 40. POSITION ERROR INTRODUCED BY LATERAL CORRECTION ESTIMATE ct 64

ACRONYMS

ECEF:	Earth-Center Earth-Fixed
GBAS:	Ground-Based Augmented System
GNSS:	Global Navigation Satellite Systems
GPS:	Global Positioning System
LAAS:	Local-Area Augmented System
NMEA:	National Marine Electronics Association
SBAS:	Space-Based Augmented System
SV:	Space Vehicle
V2V:	Vehicle to Vehicle
VBAS:	Vehicle-Based Augmented System
WAAS:	Wide-Area Augmented System

CHAPTER 1: INTRODUCTION

1.1. BACKGROUND

To date, navigation is no longer only important to marine and aviation [1] [2]. It becomes increasingly important to ground vehicle applications [3] [4]. An essential area of ground vehicle navigation is safety-related applications, such as lane departure warning systems. Currently, most of these applications rely on Global Navigation Satellite Systems (GNSS), which employ artificial satellites to navigate globally. A typical and popular GNSS is the Global Positioning System (GPS). GPS is initially for defense purpose. But not long after its operation, it became freely available for civilian use [5]. However, civilian users have limited positioning accuracy mainly due to economic reasons. In order to improve GPS accuracy while without increasing the manufacturing price of a commercial receiver, position corrections are created and broadcast by differential GPS (DGPS). These augmented GPS can be categorized into two classes: Ground-Based Augmentation Systems (GBAS) and Space-Based Augmentation Systems (SBAS).

GBAS employs ground fixed reference antennas to create differential position corrections. Based on the known and precisely surveyed position of the reference antenna, discrepancy between the estimated position and the known position of the antenna is transform to differential corrections. It is also because of the ground fixed reference, GBAS can only cover an area with up to 20 km in radius. An example of GBAS is FAA's Local-Area Augmentation System (LAAS) [6], which is developed to assist aircraft landing and taking off under poor visibility. For ground vehicle applications, however, GBAS is not practical since it will require numerous ground reference stations, which are expensive to build, in order to cover all the roadways, nationwide.

On the other hand, SBAS fills the requirement of wide cover range. SBAS collects correction information from the ground reference stations, and then transmits the corrections to users distributed in a wide area. FAA's Wide-Area Augmentation System (WAAS) [7] is an augmented

system based on the concept of SBAS. Typically, SBAS can cover an entire continent. However, it is not hard to see that, although SBAS can cover widely, its accuracy is poor comparing to that of GBAS. Thus there is no doubt that SBAS does not meet the accuracy requirement for automotive applications, especially for those safety-critical applications.

1.2. MOTIVATION FOR COLLABORATIVE NAVIGATION

A novel collaborative navigation solution is proposed in this thesis to address the need of accurate and safety-critical automotive applications. Collaborative navigation is an emerging field that achieves navigation benefits by networking measurements among multiple users. An example of a collaborative navigation scenario is automotive navigation. Automotive collaborative navigation will soon be possible because of rapid developments in vehicle-to-vehicle (V2V) communication systems, which will provide 2-way communication via ad hoc networks among multiple cars. In concept, it will soon be possible to share navigation data among a large number of vehicles in proximity.

Collaborative navigation has significant potential benefit for future automated driving applications, which will require high accuracy, worldwide. Automated driving has already been demonstrated under controlled conditions, notably during the DARPA Grand Challenge [8] and Urban Challenge [9]. To obtain the high accuracy necessary for automated driving, demonstration systems have relied either on expensive sensors, like rotating laser range finders [10] or on costly infrastructure, such as fixed reference antennas to support precise differential GNSS navigation [11]. For automated driving to be commercialized globally, new navigation methods will be required that provide high accuracy positioning using low cost sensors and little or no physical infrastructure (such as a physical network of differential reference stations).

Relatively little collaborative navigation research has focused on common-mode GNSS error removal. To date, the major focus in collaborative navigation research has been the fusion of

GNSS measurements with inter-receiver ranging measurements and/or inertial measurements [12]-[14] to ensure availability under degraded conditions, such as when satellite signals are obstructed by jamming, rugged terrain, or tall buildings. Collaborative navigation has also been shown to enhance localization accuracy in an urban environment with severe multipath [15]. Non-GNSS collaborative navigation methods have also been studied, for instance, to perform wireless-network-based positioning of a cluster of automobiles [16].

To remove common-mode GNSS error, this study proposes a novel solution that deploys additional camera-based lane-boundary sensor to serve as a source of differential correction. In addition to the lane-boundary sensor, V2V network is employed to transfer the differential correction. Due to the mobility of this system, it is called a Vehicle-Based Augmentation System (VBAS).

Nowadays, camera-based sensors are mainly used in aiding reverse parking and lane departure warning systems for some luxury cars [17]. In the applications for lane departure warning system, specific approach is developed to detect lane boundary of a roadway [18]-[22]. In the research of V2V network, most of the focus is on networking architecture and protocol [23]-[28]. Relatively little research combines camera-based lane-boundary sensor and V2V network with GPS to remove common-mode error so that to improve positioning accuracy.

This thesis study fills in one of these vacancies. By deploying camera-based lane-boundary sensor, this study expects to establish a mobile GBAS, which has the accuracy of GBAS but also has the coverage of SBAS due to its mobility. To achieve the mobility function, V2V network is indispensable. Through V2V network, differential GNSS correction created with assistance of lane-boundary sensor can be transferred along with traffic, thus benefits all the users covered by this network.

1.3. THESIS CONTRIBUTIONS

In the interest of achieving a GBAS capability, the primary goal of this thesis study is to develop and experiment a novel approach to remove common-mode GNSS errors by networking GNSS and camera-based lane-boundary sensors. To summarize, the contribution of this thesis work includes:

- 1) Development of two versions of collaborative navigation algorithm – centralized and decentralized approach. These are the essential algorithms that are used throughout the study.
- 2) Design and implementation of physical experiment to test novel collaborative navigation algorithm. This relates the theoretical work to practical systems and demonstrates the capabilities of collaborative algorithm.

1.4. THESIS OVERVIEW

Chapter 2 firstly reviews the standard GNSS solution briefly. After having a basic background of GNSS positioning method, this chapter develops the collaborative navigation algorithm. In order to address different needs, the chapter introduces two versions of the algorithms. In the end of Chapter 2, there are simulations and the corresponding results. These simulations and results show the advantage of the novel collaborative navigation approach.

Chapter 3 mainly discusses about experiment. The necessary content about the experimental hardware and software is arranged in the first section. The second section of this chapter briefly introduces how to obtain reference positions for experiment. In the last section of Chapter 3, details of experiment design and deployment are introduced. For the simplicity of experiment, camera-based lane-boundary sensor is surrogated. Instead, ruler and an alternative method are used to detect the lane boundary. A discussion of this alternative method is also in the last section. Finally, in the end of Chapter 3, experimental results and relevant discussions are presented.

Finally, Chapter 4 concludes the thesis and outlines the future work.

CHAPTER 2: ALGORITHM DEVELOPMENT

In this chapter, the new collaborative navigation approach is discussed. The novel approach can be applied to any GNSS. To be specific, GPS is used as example to derive this approach. In this chapter, the first section reviews the approach of conventional GPS position solution. After having a basic knowledge about conventional solution, a novel collaborative solution for user position is introduced in the second section. Finally in the third section, simulations and their results will be discussed in order to show the capabilities of this new collaborative approach.

2.1. REVIEW OF STANDARD GPS SOLUTION

2.1.1. Basic position estimate model in GPS

One basic method to estimate position is to use linear pseudorange measurement model. For a user receiver has a view of K satellites in the sky, the pseudorange measurement model from the k^{th} satellite at epoch t (GPS time) is given by [29]

$$\begin{aligned} \rho^{(k)}(t) = & r^{(k)}(t, t - \tau) + c[\delta t_u(t) - \delta t^{(k)}(t - \tau)] + I^{(k)}(t) + T^{(k)}(t) \\ & + E^{(k)}(t) + \epsilon_\rho^{(k)}(t) \end{aligned} \quad (1)$$

where $k = 1, 2, \dots, K$, indicates the number of satellite rather than the order of differentiation. Each pseudorange measurement is related to the actual distance $r^{(k)}(t, t - \tau)$ between the receiver antenna at signal reception time t and the satellite antenna at signal transmission time $(t - \tau)$. Several factors cause the pseudorange to differ from the actual distance. These factors include the receiver clock offset $\delta t_u(t)$, the satellite clock offset $\delta t^{(k)}(t - \tau)$, the ionosphere delay $I^{(k)}(t)$, the troposphere delay $T^{(k)}(t)$, the satellite orbit estimation (or *ephemeris*) error $E^{(k)}(t)$, and additional errors such as multipath and thermal noise, which are lumped into the term $\epsilon_\rho^{(k)}(t)$. Receiver and satellite clock offsets are converted to distance units by multiplying the speed of light c .

For simplicity, common-mode errors such as satellite clock offset, ephemeris error and ionosphere and troposphere delays are lumped with $\epsilon_\rho^{(k)}(t)$, then the pseudorange measurement becomes

$$\rho^{(k)}(t) = r^{(k)}(t, t - \tau) + c \cdot \delta t_u(t) + \tilde{\epsilon}_\rho^{(k)}(t) \quad (2)$$

where the error term $\tilde{\epsilon}_\rho^{(k)}(t)$ denotes the combined effect of the residual errors.

If vector $\mathbf{x}(t) = (x, y, z)$ and $\mathbf{x}^{(k)}(t - \tau) = (x^{(k)}, y^{(k)}, z^{(k)})$ are used to represent the position of the receiver at the time of measurement and the position of the k^{th} satellite at the time of signal transmission, then the geometric range from receiver to satellite is

$$\begin{aligned} r^{(k)}(t, t - \tau) &= \sqrt{(x^{(k)} - x)^2 + (y^{(k)} - y)^2 + (z^{(k)} - z)^2} \\ &= \|\mathbf{x}^{(k)}(t - \tau) - \mathbf{x}(t)\| \end{aligned} \quad (3)$$

Thus Equation (2) can be written as

$$\rho^{(k)}(t) = \|\mathbf{x}^{(k)}(t - \tau) - \mathbf{x}(t)\| + b(t) + \tilde{\epsilon}_\rho^{(k)}(t) \quad (4)$$

where receiver clock offset term $c \cdot \delta t_u(t)$ is replaced by $b(t)$ with unit of meters. Because both second order form and square root are involved in Equation (4), it is obviously a nonlinear model. In order to linearize the equation, *Taylor Series expansion* is used; then *Newton-Raphson method* is applied iteratively to solve the set of K linear equations (which indicate K pseudorange measurements from the satellite constellation) for user position $\mathbf{x}(t)$ and user clock bias $b(t)$. It is assumed \mathbf{x}_0 and b_0 are the initial guesses of the receiver position and receiver clock bias, respectively. Hence the corresponding approximate pseudorange based on the initial guesses is

$$\rho_0^{(k)}(t) = \|\mathbf{x}^{(k)}(t - \tau) - \mathbf{x}_0(t)\| + b_0(t) \quad (5)$$

For simplicity reason, the explicit reference to time is dropped from here. By assuming that the true receiver position and true receiver clock bias to be $\mathbf{x} = \mathbf{x}_0 + \delta\mathbf{x}$ and $b = b_0 + \delta b$, where $\delta\mathbf{x}$ and δb are the unknown corrections to be applied to the initial estimates, a linear equation of disturbed pseudorange measurement is shown as

$$\begin{aligned}
\delta\rho^{(k)} &= \rho^{(k)} - \rho_0^{(k)} \\
&= \|\mathbf{x}^{(k)} - \mathbf{x}\| - \|\mathbf{x}^{(k)} - \mathbf{x}_0\| + (b - b_0) + \tilde{\epsilon}_\rho^{(k)} \\
&= \|\mathbf{x}^{(k)} - \mathbf{x}_0 - \delta\mathbf{x}\| - \|\mathbf{x}^{(k)} - \mathbf{x}_0\| + (b_0 + \delta b - b_0) + \tilde{\epsilon}_\rho^{(k)} \\
&\approx \|\mathbf{x}^{(k)} - \mathbf{x}_0\| - \frac{(\mathbf{x}^{(k)} - \mathbf{x}_0)}{\|\mathbf{x}^{(k)} - \mathbf{x}_0\|} \cdot \delta\mathbf{x} - \|\mathbf{x}^{(k)} - \mathbf{x}_0\| + \delta b + \tilde{\epsilon}_\rho^{(k)} \\
&= -\frac{(\mathbf{x}^{(k)} - \mathbf{x}_0)}{\|\mathbf{x}^{(k)} - \mathbf{x}_0\|} \cdot \delta\mathbf{x} + \delta b + \tilde{\epsilon}_\rho^{(k)} \\
&= -\mathbf{u}^{(k)} \cdot \delta\mathbf{x} + \delta b + \tilde{\epsilon}_\rho^{(k)}
\end{aligned} \tag{6}$$

where in the fourth line, the first two terms are the approximation from *Taylor Series*. In the above equation, $\mathbf{u}^{(k)}$ is the estimated line-of-sight unit vector directed from the initial estimate of the receiver position to the k^{th} satellite. Its elements are direction cosines of the vector drawn from the estimated receiver location to the satellite,

$$\mathbf{u}^{(k)} = \frac{1}{\|\mathbf{x}^{(k)} - \mathbf{x}_0\|} (x^{(k)} - x_0, y^{(k)} - y_0, z^{(k)} - z_0)^T \tag{7}$$

It should be noticed that $\mathbf{u}^{(k)}$ will be updated according to the new estimate position in each step of an iterative approach, such as *Newton-Raphson method*.

Above is the derivation of the equation for one satellite-receiver channel. In order to compute a solution for 4 unknowns (i.e. 3 components of receiver's position and 1 receiver's clock bias), at least 4 pseudorange measurements from different satellite-receiver channels are needed. Assume there are K ($K \geq 4$) satellites in the sky are viewed by the receiver, the set of K linear equations can be written in matrix notation as

$$\delta\boldsymbol{\rho} = \begin{bmatrix} \delta\rho^{(1)} \\ \delta\rho^{(2)} \\ \vdots \\ \delta\rho^{(K)} \end{bmatrix} = \begin{bmatrix} (-\mathbf{u}^{(1)})^T & 1 \\ (-\mathbf{u}^{(2)})^T & 1 \\ \vdots & \vdots \\ (-\mathbf{u}^{(K)})^T & 1 \end{bmatrix} \begin{bmatrix} \delta\mathbf{x} \\ \delta b \end{bmatrix} + \tilde{\boldsymbol{\epsilon}}_\rho \quad (8)$$

Writing the above equations more compactly, there is

$$\delta\boldsymbol{\rho} = \mathbf{G} \begin{bmatrix} \delta\mathbf{x} \\ \delta b \end{bmatrix} + \tilde{\boldsymbol{\epsilon}}_\rho \quad (9)$$

where

$$\mathbf{G} = \begin{bmatrix} (-\mathbf{u}^{(1)})^T & 1 \\ (-\mathbf{u}^{(2)})^T & 1 \\ \vdots & \vdots \\ (-\mathbf{u}^{(K)})^T & 1 \end{bmatrix} \quad (10)$$

is a $(K \times 4)$ matrix characterizing the receiver-satellite geometry. Thus it is referred as geometry matrix. So far, it is possible to solve a least-squares problem to obtain $[\delta\mathbf{x} \ \delta b]^T$, shown as below

$$\begin{bmatrix} \delta\mathbf{x} \\ \delta b \end{bmatrix} = \mathbf{G}^{-1} \delta\boldsymbol{\rho} \quad (11)$$

$$\begin{bmatrix} \hat{\mathbf{x}} \\ \hat{b} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}} \\ \hat{b} \end{bmatrix} + \begin{bmatrix} \delta\mathbf{x} \\ \delta b \end{bmatrix} \quad (12)$$

By iteratively solving $[\delta\mathbf{x} \ \delta b]^T$, new estimate of receiver position and clock bias $[\hat{\mathbf{x}} \ \hat{b}]^T$ can be updated in each iteration, until the change of new estimate drops below a preset threshold.

2.1.2. Networked GPS measurements and its limitation

In the previous subsection, errors are ignored when deriving the standard GPS solution. In this subsection, a potential error reduction approach is discussed. Despite the random noisy signal a user receives, a significant part of estimate errors derives from the spatially correlated GPS errors

(e.g. ionosphere delay and troposphere delay). To reduce the effect of spatially correlated errors, a straightforward thought is to combine all the pseudorange measurements of nearby users and cancel the effect of these errors. This technique, sometimes called “single-difference” GPS, is useful for producing accurate relative positions (between receivers). However, it is not possible to estimate spatially correlated GPS errors simply by sharing single-frequency pseudorange measurements among multiple users. This subsection briefly discusses this limitation of distributing GPS data over a network and motivates the introduction of an additional sensor (a camera-based lane-boundary sensor).

Given that pseudorange data for multiple receivers are shared over a network, the multi-receiver GPS solution is a straightforward extension of the conventional, single user solution. In the multi-user case, if full information is distributed among users, least-squares is applied to process all pseudorange measurements $\rho_n^{(k)}$, each associated with a particular receiver n and satellite k . The following equation is a model for this pseudorange measurement.

$$\rho_n^{(k)} = \|\mathbf{x}^{(k)} - \mathbf{x}_n\| + c[\delta t_n - \delta t^{(k)}] + I^{(k)} + T^{(k)} + E^{(k)} + \epsilon_n^{(k)} \quad (13)$$

This equation is very similar to Equation (1). It only differs in the explicit reference to the specific user n .

For the purposes of this study, it is useful to rewrite Equation (13) by clustering common-mode errors into two terms. A first term b_n describes the clock error for a particular receiver. This term will be called the *receiver-specific* error. The second term $a^{(k)}$ is the spatially correlated error for a particular satellite, which is common to all receivers in a local area. This term will be called the *satellite-specific* error. The modified equation is as below,

$$\rho^{(k)} = \|\mathbf{x}^{(k)} - \mathbf{x}\| + b_n + a^{(k)} + \epsilon^{(k)} \quad (14)$$

The receiver-specific term b_n is equal to the user clock error ($c\delta t_n$). The satellite-specific error includes the spatially correlated errors, which are very nearly equal for all users over a region of several kilometers in radius.

$$a^{(k)} = -c\delta t^{(k)} + I^{(k)} + T^{(k)} + E^{(k)} \quad (15)$$

In concept, it is possible to set up a least-squares problem that solves for an estimate $\hat{a}^{(k)}$ of the spatially correlated errors, while also obtaining an estimate $\hat{\mathbf{x}}_n$ for each user position and \hat{b}_n for each receiver-specific error. To set up such a least-squares problem, we can neglect the random noise terms $\varepsilon_n^{(k)}$ and linearize the pseudorange measurement model (14) in terms of the perturbed measurements $\delta\rho^{(k)}$ and states: $\hat{\mathbf{x}}_n$, \hat{b}_n , and $\delta\hat{a}^{(k)}$.

$$\delta\rho^{(k)} = (-\mathbf{u}^{(k)})^T \delta\hat{\mathbf{x}}_n + \delta\hat{b}_n + \delta\hat{a}^{(k)} \quad (16)$$

In this equation, the unit vector $\mathbf{u}^{(k)}$ is the estimated pointing vector from the user receivers to each satellite k , as shown in Equation (7). Users are assumed to be in close proximity, so this pointing vector is the same for all users. Let $\hat{\mathbf{x}}$ to be a position vector and represent one of the networked users, Equation (7) is rewritten as

$$\mathbf{u}^{(k)} = \frac{\mathbf{x}^{(k)} - \hat{\mathbf{x}}}{\|\mathbf{x}^{(k)} - \hat{\mathbf{x}}\|} \quad (17)$$

To estimate the states, pseudoranges (for N users and all K satellites) are compiled into a single vector \mathbf{y} .

$$\mathbf{y} = [\rho_1^{(1)} \quad \rho_1^{(2)} \quad \dots \quad \rho_1^{(K)} \quad \dots \quad \rho_N^{(1)} \quad \dots \quad \rho_N^{(K)}]^T \quad (18)$$

The corresponding state vector $\hat{\mathbf{z}}$ consists of the estimated positions and clock offset for all N users as well as the satellite-specific errors for all K satellites:

$$\hat{\mathbf{z}} = [\mathbf{x}_1^T \quad b_1 \quad \cdots \quad \mathbf{x}_N^T \quad b_N \quad a^{(1)} \quad \cdots \quad a^{(K)}]^T \quad (19)$$

Perturbations to the measurement vector (18) are related to perturbations of the state vector (19) by the linearized measurement model (16). These equations may be written in matrix form as follows.

$$\delta \mathbf{y} = \mathcal{G} \delta \hat{\mathbf{z}} \quad (20)$$

Here \mathcal{G} is a matrix of the coefficients for the linearized model. For the specific case in which all users view the same set of satellites, the \mathcal{G} matrix is:

$$\mathcal{G} = \left[\begin{array}{cccc|c} \mathbf{G} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{G} & \cdots & \mathbf{0} & \mathbf{I} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{G} & \mathbf{I} \end{array} \right] \quad (21)$$

where \mathbf{I} is the identity matrix and \mathbf{G} is the geometry matrix, as shown in Equation (10).

At first glance, this set of equations appears to be solvable, because the number of equations (NK) grows much more quickly than the number of unknowns ($4N+K$). In fact, regardless of the number of measurements, it is not possible to solve (20) for the states $\hat{\mathbf{z}}$, because the matrix \mathcal{G} is rank deficient.

One way to demonstrate the rank deficiency of \mathcal{G} is to linearly combine its rows, by subtracting the first block row from all other block rows, there is,

$$\mathcal{G}_{sd} = \left[\begin{array}{cccc|c} \mathbf{G} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \\ -\mathbf{G} & \mathbf{G} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -\mathbf{G} & \mathbf{0} & \cdots & \mathbf{G} & \mathbf{0} \end{array} \right] \quad (22)$$

Differencing matrix rows does not change matrix rank; hence the rank of (22) is the same as that of (21). The rank of the new matrix \mathcal{G}_{sd} (where the subscript “ sd ” refers to single difference) can

be inferred by examining each block row. The identity matrix at the end of the first block row ensures the row contributes a full set of N linearly independent equations. Subsequent block rows only solve for the relative position and time between two receivers; thus each block row after the first contributes only four linearly independent equations (In effect, each block row after the first represents a single-difference between two receivers, which removes all common-mode errors $a^{(k)}$). The total number of linearly independent equations in \mathcal{G}_{sd} is $4(N-1)+K$, not enough to solve for the total number of unknowns $(4N+K)$.

Thus, it is not possible to solve for the satellite-specific errors using only networked GPS pseudoranges. For clarity, the above analysis considers the specific case when all users see the same set of satellites; the rank deficiency problem can be demonstrated, however, even when users see different satellite sets. By extension, inter-receiver range measurements do not help to estimate the satellite-specific errors, since they are not linearly independent from single-difference GPS measurements.

2.2. COLLABORATIVE APPROACH FOR GPS POSITION SOLUTION

2.2.1. Fusing lane-boundary and GPS measurements

In order to enable estimation of satellite-specific errors among networked GPS receivers, it is necessary to add additional sensor measurements that are absolute (i.e., those that relate receiver position to Earth-fixed coordinates). As discussed in the previous section, relative measurements (i.e., those that relate the positions of multiple receivers to each other) do not aid in observing satellite-specific errors. The most common approach for obtaining absolute information is to employ a fixed-location, surveyed reference receiver in the network (as is the case in differential GPS applications). For automotive applications, however, it is anticipated that widespread deployment of surveyed reference antennas will be impractical.

Camera-based lane-boundary sensors can serve as an alternative source of earth-fixed positioning data. These sensors are attractive in that they are available at a relatively low-cost and have already been deployed in lane-departure warning systems for some luxury cars [17]. A camera-based lane boundary sensor functions by applying vision processing methods to detect lane-markers in a video sequence [30]. By fitting a parametric curve to the markers in each video image [18], it is possible to infer the heading and distance of the camera relative to the lane boundary. Given a sufficiently accurate database of surveyed lane-boundary locations [31], it is thus possible to obtain highly accurate absolute positioning information.

The major challenge in integrating a lane-boundary sensor is that it detects a *line feature* (or curve feature) rather than a *point feature*. In other words, the sensor directly measures the distance from a line (i.e. from the lane boundary), but cannot measure location along that line. Thus the camera's location along the length of the line is ambiguous. We can mathematically capture this ambiguity by defining the surveyed road contour in terms of a path integral s , which describes the distance traveled along the contour from a nominal reference point. As an example, if the n^{th} user observes a straight lane boundary, that boundary is a locus of points described by the function $\mathbf{r}_n(s)$, where s is the distance from a reference point \mathbf{R}_n in a direction parallel to the road. This direction is described by the road unit vector \mathbf{u}_r (as shown in Figure 1).

$$\mathbf{r}_n(s) = \mathbf{R}_n + s\mathbf{u}_r \quad (23)$$

It is further assumed that the road boundary model consists not only of a unit vector parallel to the road (\mathbf{u}_r), but also a transverse unit vector (\mathbf{u}_t), which lies in the plane of the road, perpendicular to the path of the road (see Figure 1).

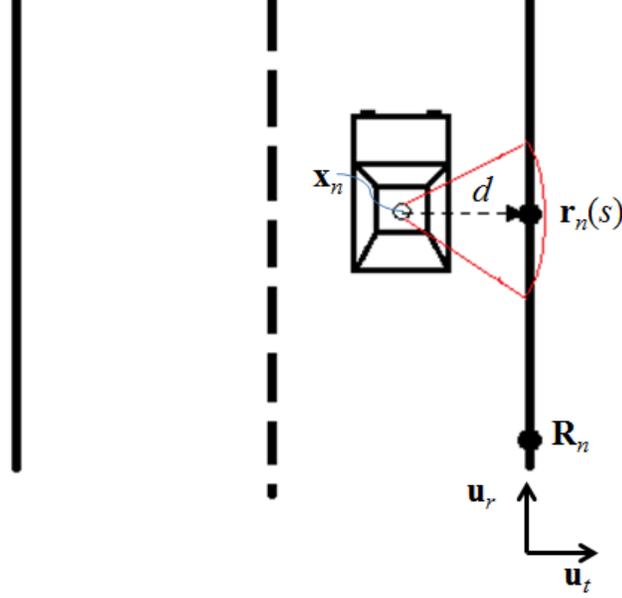


Figure 1. Geometry for Camera-Based Lane-Boundary Sensing

The camera sensor measures the perpendicular distance d from the camera to the lane boundary. Without loss of generality, it is assumed that the camera and GPS receiver are collocated (which is equivalent to assuming the distance between the two sensors is well calibrated). An expression relating the car position to the lane boundary contour is:

$$\mathbf{x}_n + d\mathbf{u}_t = \mathbf{r}_n(s) \quad (24)$$

Although the perpendicular line from the camera to the lane boundary intersects the lane boundary at an unknown path coordinate s , the distance from the camera to the lane boundary (in the \mathbf{u}_t direction) is not ambiguous. Note that since both Equation (23) and (24) are functions of along-road distance s , they are valid for a curved contour as well. Supposing in a straight line segment of a road, considering only the transverse component of the camera model (24), we can obtain the following measurement model.

$$d = (\mathbf{R}_n - \mathbf{x}_n)^T \mathbf{u}_t \quad (25)$$

Specifically, this equation is obtained by setting (23) equal to (24) and taking a dot product with \mathbf{u}_t .

It is straightforward to augment the GPS solution with the extra measurement equation given by (25). For instance, in the single user case, an augmented measurement vector \mathbf{y}_{lb} can be formed from the GPS measurements and the lane-boundary sensor measurement.

$$\mathbf{y}_{lb} = [\rho^{(1)} \quad \rho^{(2)} \quad \dots \quad \rho^{(K)} \quad d]^T \quad (26)$$

States can be obtained by iteratively solving the following linear measurement equations.

$$\delta \mathbf{y}_{lb} = \mathbf{G}_{lb} \begin{bmatrix} \delta \hat{\mathbf{x}} \\ \delta \hat{b} \end{bmatrix} \quad (27)$$

$$\mathbf{G}_{lb} = \begin{bmatrix} \mathbf{G} \\ -\mathbf{u}_t^T \quad 0 \end{bmatrix} \quad (28)$$

The last line of the augmented geometry matrix \mathbf{G}_{lb} provides additional geometric diversity, which further reduces the error in estimating user time offset and position, \hat{b} and $\hat{\mathbf{x}}$.

2.2.2. Estimating satellite-specific GPS error with lane-boundary sensor

This subsection shows that the additional measurement provided by a lane-boundary sensor can be used to estimate the satellite-specific error, which is common for all users operating in proximity. In this sense, lane-boundary measurements can be applied not only to improve the position estimate of a single receiver, as shown in the previous section; they can also be shared, in the context of collaborative GPS navigation, to improve the accuracy of all collaborating receivers.

2.2.2.1. Single-User Estimation Problem

The content below first describes that how the satellite-specific error terms $a^{(k)}$ can be estimated for an individual user by leveraging a lane-boundary measurement. Then the single-user case will be extended to multiple users in the next subsection.

In the single-user case, there are only K GPS measurements, so it is not possible to solve simultaneously for the user clock and position estimates (4 unknowns) and also for the satellite-specific error terms $a^{(k)}$ (K more unknowns). The lane-boundary measurement adds just one more equation, so it is still not possible to solve for all $K + 4$ unknowns. What the lane-boundary sensor *can* do is aid in solving for the component of the satellite-specific errors which causes the receiver position to move laterally toward (or away from) the lane boundary. For the k^{th} satellite, the contribution of the satellite-specific error to lateral motion can be obtained by dotting the pointing vector from the receiver to the satellite with that from the receiver to the lane boundary: $\mathbf{u}^{(k)} \cdot \mathbf{u}_t$. Clustering the dot products for all satellites together, we can obtain a vector \mathbf{t} , which describes the lateral components of the satellite-specific error for all satellites.

$$\mathbf{t} = \mathbf{G} \begin{bmatrix} \mathbf{u}_t \\ 0 \end{bmatrix} \quad (29)$$

In this equation, the satellite pointing vectors are embedded in the geometry matrix \mathbf{G} , defined by (10).

If the lane-boundary sensor is used to correct an estimate of a user's location, shifting it toward or away from the lane boundary, then the satellite-specific errors *in the lateral direction* can be corrected accordingly. To appreciate this effect, it is useful to rewrite satellite-specific error $a^{(k)}$ in terms of five components. Thus for all the satellites, there is,

$$\begin{bmatrix} a^{(1)} \\ a^{(2)} \\ \vdots \\ a^{(K)} \end{bmatrix} = \mathbf{G} \underbrace{\begin{bmatrix} \mathbf{u}_t \\ 0 \end{bmatrix}}_{\mathbf{t}} c_t + \mathbf{G} \begin{bmatrix} \mathbf{u}_r \\ 0 \end{bmatrix} c_r + \mathbf{G} \begin{bmatrix} \mathbf{u}_n \\ 0 \end{bmatrix} c_n + \mathbf{G} \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} c_b + \mathbf{a}_{null} \quad (30)$$

This equation emphasizes that, regardless of the number of satellites K (so long as K is at least four), the vector of satellite-specific errors can be represented in terms of four vectors that affect the navigation solution. In the equation above, these vectors shift the navigation solution in the

direction transverse to the lane boundary (\mathbf{u}_t) by a distance c_t , in the direction parallel to the lane boundary (\mathbf{u}_r) by a distance c_r , in the direction normal to the plane of the road (\mathbf{u}_n) by a distance c_n , and in the “direction” of the clock correction by a “distance” c_b . An additional $(K - 4)$ components of the satellite-specific error vector must exist, in order to form a complete basis for the vector space; however, these components, lumped into the vector \mathbf{a}_{null} , lie in the null space of the \mathbf{G} matrix, and hence these additional components do not affect the navigation solution.

The lane-boundary measurement makes it possible to correct the vehicle’s lateral position estimate, and hence, to estimate the transverse component of the satellite-specific error vector, which is $c_t \mathbf{t}$, as seen in (30). Since the vector \mathbf{t} can be computed based purely on geometry, it is possible to construct the lateral “differential correction” vector $c_t \mathbf{t}$ if the user estimates one new state in addition to position and time: \hat{c}_t . The corresponding state vector is $\hat{\mathbf{z}}_{lbt}$, where the subscript “*lbt*” indicates lane-boundary measurement used to create a transverse differential correction.

$$\hat{\mathbf{z}}_{lbt} = \begin{bmatrix} \hat{\mathbf{x}} \\ \hat{b} \\ \hat{c}_t \end{bmatrix} \quad (31)$$

This estimate is obtained from a measurement vector that includes pseudoranges and a lane-boundary measurement.

$$\mathbf{y}_{lbt} = \mathbf{y}_{lb} \quad (32)$$

The linear least squares equation is the following.

$$\delta \mathbf{y}_{lbt} = \mathbf{G}_{lbt} \delta \hat{\mathbf{z}}_{lbt} \quad (33)$$

$$\mathbf{G}_{lbt} = \begin{bmatrix} \mathbf{G} & \mathbf{t} \\ [-\mathbf{u}_t^T & 0] & 0 \end{bmatrix} \quad (34)$$

The modified geometry matrix \mathbf{G}_{lbt} directly incorporates the vector \mathbf{t} . This vector, given by (29), projects the satellite-specific error into the lateral direction.

Curiously, a consequence of the mathematical structure of (33) is that the user's lateral position depends entirely on the lane-boundary measurement. Any discrepancy between the pseudorange measurement and the lane-boundary measurement is implicitly assumed to be due to a satellite-specific error. Thus, the lateral position error for (33) is that of the lane-boundary sensor.

2.2.2.1. Multi-User Estimation Problem with Sharing Measurements

The previous subsection developed a method by which a vehicle-borne GPS receiver could produce a differential correction to remove satellite-specific errors, like ionosphere delay, troposphere delay, and satellite clock offset. This differential correction is very much like differential corrections created by a conventional differential GPS, except that (i) the correction applies only in the lateral direction and that (ii) no fixed reference station is needed to generate the correction!

This subsection develops a method that transfers common-mode (e.g. satellite-specific) error data from users equipped with lane-boundary sensors to unequipped users. Thus, even users not equipped with any sensor other than a conventional GPS receiver still benefit from common-mode GPS error removal.

A straightforward approach to solving the collaborative navigation problem is to fuse all pseudorange and lane-boundary sensor data together into one large least-squares estimation problem. For such a problem, the collaborative-navigation state vector is $\hat{\mathbf{z}}_{CN}$. Here the subscript "CN" refers to collaborative navigation.

$$\hat{\mathbf{z}}_{CN} = [\hat{\mathbf{x}}_1^T \quad \hat{b}_1 \quad \hat{\mathbf{x}}_2^T \quad \hat{b}_2 \quad \cdots \quad \hat{\mathbf{x}}_N^T \quad \hat{b}_N \quad \hat{c}_t \quad \hat{c}_r]^T \quad (35)$$

The multi-user state vector includes estimates for the positions and clock-offsets for all users as well as two coefficients for satellite-specific error correction, \hat{c}_t and \hat{c}_r . For the special case when all vehicles are traveling in a parallel direction (e.g. along the same straight road), only one of these coefficients (\hat{c}_t) would be a state. However, in general, two satellite-specific error coefficients, \hat{c}_t and \hat{c}_r , are introduced because different cars may be traveling in different directions. When two or more cars travel in different directions, the two ground-plane components of the satellite-specific error can both be estimated. Any two vectors which span the space of the ground plane can be introduced for this purpose; here in this formulation, two orthogonal vectors are chosen to describe the road-parallel and road-transverse directions for the first user ($\mathbf{u}_{r,1}$ and $\mathbf{u}_{t,1}$, respectively).

The vector of sensor data consists of the measurements for each user, including GPS pseudoranges and, if available, a lane-boundary measurement.

$$\mathbf{y}_{CN} = [\mathbf{y}_1^T \quad \mathbf{y}_2^T \quad \cdots \quad \mathbf{y}_N^T]^T \quad (36)$$

Here the size of the data set \mathbf{y}_n for each user n depends on the number of satellites K_n seen by that user and whether or not the user is equipped with a lane-boundary sensor.

$$\mathbf{y}_n = \begin{cases} [\rho_n^{(1)} & \rho_n^{(2)} & \cdots & \rho_n^{(K)} & d_n]^T & \text{equipped with LB sensor} \\ [\rho_n^{(1)} & \rho_n^{(2)} & \cdots & \rho_n^{(K)}]^T & \text{unequipped with LB sensor} \end{cases} \quad (37)$$

The linear least-squares equations used to iteratively obtain the estimated states from this set of measurements are:

$$\delta \mathbf{y}_{CN} = \mathbf{G}_{CN} \delta \hat{\mathbf{z}}_{CN} \quad (38)$$

The augmented geometry matrix takes the following form,

$$\mathbf{G}_{CN} = \left[\begin{array}{cccc|cc} \tilde{\mathbf{G}}_1 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{t}_1 & \mathbf{r}_1 \\ \mathbf{0} & \tilde{\mathbf{G}}_2 & \cdots & \mathbf{0} & \mathbf{t}_2 & \mathbf{r}_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \tilde{\mathbf{G}}_N & \mathbf{t}_N & \mathbf{r}_N \end{array} \right] \quad (39)$$

The augmented geometry matrix \mathbf{G}_{CN} is similar in form to the GPS-only matrix described by (21); however, \mathbf{G}_{CN} is no longer rank deficient. Each block row is characterized by a user geometry matrix $\tilde{\mathbf{G}}_n$ and by the corresponding matrix projections in the directions of the road-plane coordinates, \mathbf{t}_n and \mathbf{r}_n . The user geometry matrix resembles the conventional geometry matrix for GPS processing, as described by (10), augmented by an additional row when that user is equipped with a lane-boundary sensor. The form of the user geometry matrix for an equipped user is:

$$\tilde{\mathbf{G}}_n = \begin{bmatrix} (-\mathbf{u}^{(1)})^T & 1 \\ (-\mathbf{u}^{(2)})^T & 1 \\ \vdots & \vdots \\ (-\mathbf{u}^{(K)})^T & 1 \\ -\mathbf{u}_{t,n}^T & 0 \end{bmatrix} \quad (40)$$

Because the number of satellites viewed by each user may be different, the road-plane projection vectors, \mathbf{t}_n and \mathbf{r}_n , may vary in length for each user. Without loss of generality, these projection vectors are defined by using the road-plane unit vectors for the first user.

$$\mathbf{t}_n = \tilde{\mathbf{G}}_n \begin{bmatrix} \mathbf{u}_{t,1} \\ 0 \end{bmatrix}, \mathbf{r}_n = \tilde{\mathbf{G}}_n \begin{bmatrix} \mathbf{u}_{r,1} \\ 0 \end{bmatrix} \quad (41)$$

If all users observe the same set of visible satellites, then the first K rows are identical for all users for both the geometry matrix \mathbf{G}_{CN} and the road-plane projection vectors \mathbf{t}_n and \mathbf{r}_n . In the special case when all users are on the same straight road (or parallel roads), then only one planar correction (c_t) can be estimated, as the orthogonal component (c_r) becomes unobservable. In this

case it is necessary to delete the last column of \mathbf{G}_{CN} , which contains the along-road correction vectors \mathbf{r}_n , in order for a solution to be found.

2.2.3. Multi-User Estimation Problem with Sharing Corrections

The collaborative navigation algorithm described in the preceding subsection is a “centralized” algorithm in the sense that it processes measurements for many cars at the same time, by solving the least-squares problem (38). Solving this problem becomes increasingly computationally intensive as the number of collaborators increases. To limit the computational burden required of any one user, it is useful to develop a decentralized version of this algorithm. A solution for the decentralized algorithm is to share the computed corrections among networked users. Thus for users without lane-boundary sensor, there is no need to collect all the measurements and solve for a large size least-squares problem (38). The only thing they need to do is to apply the shared corrections into their own estimated pseudorange equation and perform a single user standard solution.

For example, in a case of two networked vehicles, which are assumed to be heading the same direction, the first car has lane-boundary sensor and estimates the lateral component \hat{c}_t of common-mode errors. With this parameter, the second car can multiply it by its transverse pointing vector \mathbf{t} , thus results in the lateral correcting terms for each satellite,

$$\begin{bmatrix} a_t^{(1)} \\ a_t^{(2)} \\ \vdots \\ a_t^{(K)} \end{bmatrix} = \mathbf{G} \underbrace{\begin{bmatrix} \mathbf{u}_t \\ 0 \end{bmatrix}}_{\mathbf{t}} c_t \quad (42)$$

In the above equation, the subscript t indicates the corrections are in the direction of unit vector \mathbf{t} . The necessity of assuming an identical direction is that, both cars have the same road parameters so that the second car can apply the correction without any transforming. For each user without lane-boundary sensor, applying these correcting terms in each of the estimate pseudorange

equations can compensate a portion of common-mode errors relative to that satellite. Using this estimate pseudorange to update the perturbed pseudorange, each user can achieve an improved position result only by solving a conventional single-user least-squares problem (Equation 43 to 45).

$$\hat{\rho}^{(k)} = \|\mathbf{x}^{(k)} - \hat{\mathbf{x}}\| + \hat{b} + \hat{a}_t^{(k)} \quad (43)$$

$$\delta\boldsymbol{\rho} = \boldsymbol{\rho}_{meas} - \hat{\boldsymbol{\rho}} \quad (44)$$

$$\begin{bmatrix} \delta\hat{\mathbf{x}} \\ \delta\hat{b} \end{bmatrix} = \mathbf{G}^{-1}\delta\boldsymbol{\rho} \quad (45)$$

2.3. SIMULATION

This section uses simulation to assess the performance of the multi-user collaborative navigation algorithm detailed in the previous section. The simulation is also used to characterize the benefits of lane-boundary sensors for single-user navigation. It is non-trivial to state again that the approach developed above is generic and can be applied to any GNSS. As an example, GPS is used in the simulation to demonstrate the results.

2.3.1. Simulation Parameters

The simulation consisted of a set of five cars, all within 1 km of each other, travelling along a straight roadway (Boston Avenue in Medford, MA, near the Tufts University campus). Locations of the cars (latitude and longitude) are summarized in Table I. All cars were at approximately the same altitude relative to the local sea level.

Table I. Locations of Simulated Cars

	Latitude(°)	Longitude(°)	Travel Direction
Car 1	42.40064	-71.11459	Southeast
Car 2	42.40020	-71.11334	Southeast
Car 3	42.39713	-71.10358	Southeast
Car 4	42.39931	-71.11083	Northwest
Car 5	42.39827	-71.10727	Northwest

GPS satellite positions were determined based on the satellite constellation visible on UTC hour 0:00:00 on 28 August 2006. (Satellite coordinates were obtained from NOAA sp3 data [32].) Figure 2 describes azimuth and elevation angles for all ten visible satellites on a polar diagram. For the purposes of the simulation, it was assumed that all ten satellites (each above a mask angle of 5°) were visible to all users.

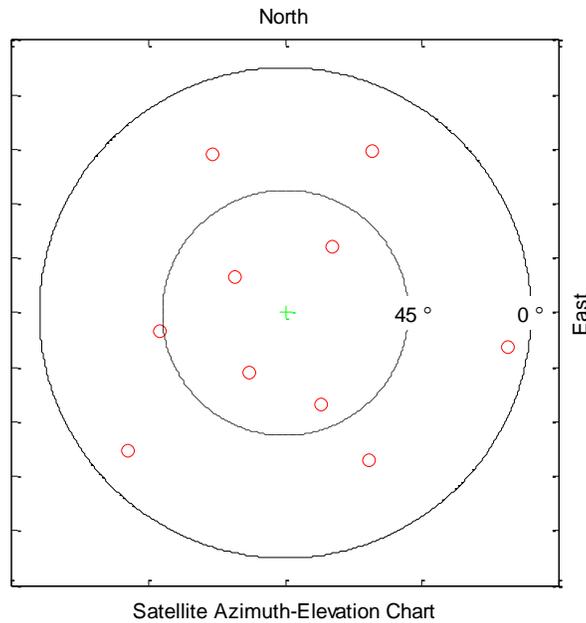


Figure 2. Satellite Azimuth and Elevation Angles for Simulation

The number of users equipped with lane-boundary sensors was variable. It was considered that there were six scenarios, ranging from one extreme (no vehicles equipped with lane-boundary sensors) to the other extreme (all five users equipped with lane-boundary sensors). The parameters describing the lane-boundary contour (\mathbf{R}_n , \mathbf{u}_r , and \mathbf{u}_t) were assumed to be the same for all five users (Figure 3), as the users were aligned on the same straight road. Moreover, it was assumed the road-boundary parameters were surveyed with high precision.

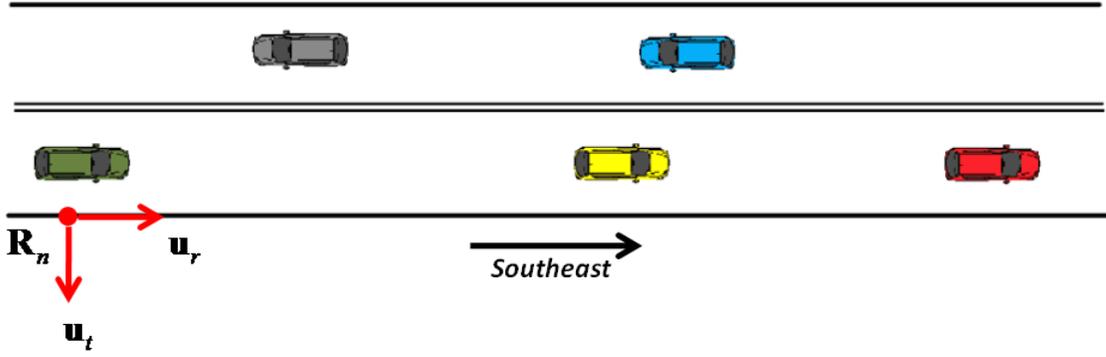


Figure 3. Cars' Position and Road Parameters for Simulation

GPS errors were modeled as random numbers. The error for each pseudorange measurement consisted of three terms: a user-specific error (e.g. user clock) that was the same for all measurements made by a particular user receiver n , a satellite-specific error (e.g. ionosphere, troposphere and satellite clock) that was the same for all user receivers viewing that satellite k , and a thermal noise and multipath error that was independent for each measurement. The user-specific error was modeled using a zero-mean Gaussian distribution with a standard deviation of 3 m; the satellite-specific error was modeled using a zero-mean Gaussian distribution with a standard deviation of 6 m summed with an independent uniform distribution with a minimum value of 0 m and a maximum of 25 m; the thermal noise and multipath error was modeled using a zero-mean Gaussian distribution with a standard deviation of 2 m.

The lane-boundary sensor error was also modeled as a random number. Specifically, it was assumed a camera-based lane boundary sensor features a zero-mean Gaussian error distribution with a standard deviation of 0.25 m. In this sense, the lane-boundary sensor was assumed to be substantially more accurate than GPS pseudoranges in the lateral direction (while providing no information at all about the user's position along the axis of the road).

For clarity, the simulation generates the results in a particular instant of time. In order to generate a statistical comparison at this instant, a Monte Carlo simulation was used, meaningful that the

error at that instant was evaluated using multiple trials, each with a different randomized vector of measurement error values. It is further assumed that all signals were transmitted instantaneously.

2.3.2. Single-User Simulations

To demonstrate the benefits of fusing the lane-boundary sensor with GPS measurements, this subsection first simulates a single-user equipped with both lane-boundary and GPS sensors. For this user receiver (at the Car 1 location listed in Table I), three different cases were considered. In the first case, user position was determined using only GPS, by iteratively solving the conventional linearized least-squares equation through an inverse of the geometry matrix of (10). In the second case, user position was determined using GPS pseudoranges and the measurement from a camera-based lane-boundary sensor, by iteratively solving (27). In the third case, user position was determined using both GPS pseudoranges and measurements from a camera-based lane-boundary sensor; however, the lateral component of the common-mode GPS errors was also determined as an additional state, by solving equation (33). These three cases will be referred to as the “GPS only”, “Fused GPS + Camera”, and “GPS + Camera + New state” formulations, respectively.

In order to compare these three formulations for determining single-user position, the simulation generates 1,000 sets of pseudoranges and lane-boundary measurements. For each of these trials, all three navigation algorithms are evaluated. Standard deviations (of the error over all 1,000 trials) for each of the three cases are illustrated in Figure 4. On the left side, the figure shows along-track errors (parallel to the road). In the middle, the figure shows lateral errors (transverse to the road). On the right side, the figure shows vertical errors (normal to the road).

As shown in the figure, the along-track and vertical errors are not affected by fusing the lane-boundary sensor data with the GPS measurements. However, lateral error is reduced from conventional GPS processing (first case) by fusing the camera-based lane-boundary data with

GPS measurements (second case). Specifically, error was reduced from 5.3 m standard deviation (first case) to 4.1 m standard deviation (second case).

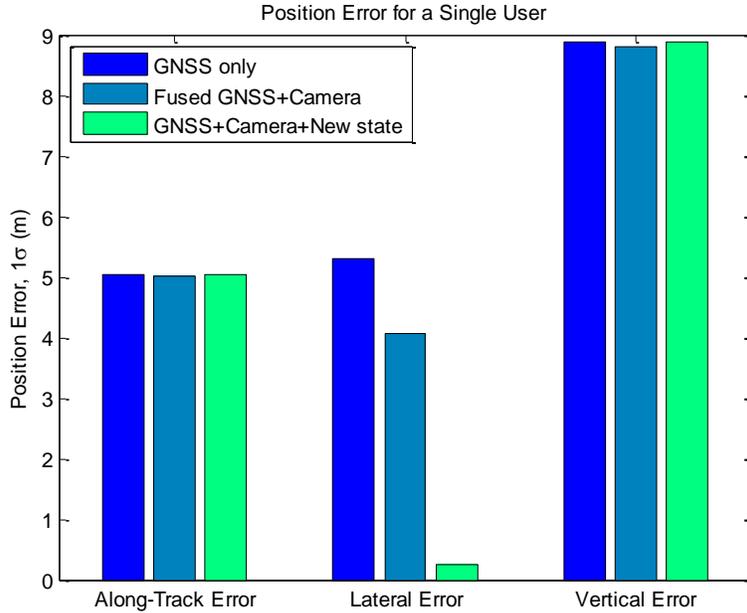


Figure 4. Simulated Errors for Single-User Navigation Algorithms

Additional reduction in error is achieved by augmenting the state vector to compute the satellite-specific error in the lateral direction. The resulting reduction in lateral position error is dramatic, with one-sigma error falling to 0.25 m in the third case. Essentially, because of the structure of (33), the lateral position of the car is determined entirely by its camera-based lane-boundary sensor, and hence the discrepancy between the GPS and lane-boundary sensors is “pushed” entirely into the estimation of the satellite-specific error coefficient c_l .

2.3.3. Collaborative-Navigation Simulations

The proposed collaborative navigation algorithm has the capability to generate differential GPS corrections for a set of mobile users, with no fixed reference antenna, as long as at least one of those users is equipped with a camera-based lane-boundary sensor. To demonstrate this capability, the simulation ran 1,000 trials to test the collaborative navigation approach described by equation (38). For each trial, the navigation equations for six different scenarios were solved. In the first

scenario, no users were equipped with a lane-boundary sensor. In the second scenario, the first user was equipped. In each additional scenario, one new user was assumed to be equipped with a lane-boundary sensor. Hence, in the final scenario, all five users were assumed to be equipped.

Since all users were assumed to be aligned on the same, straight road, lane-boundary data only aided in reducing lateral positioning error. Once again (as was in the single-user trials summarized in Figure 4), lane-boundary sensor data did not affect user position errors in the vertical and along-track directions.

Estimates of lateral position were significantly improved by fusing lane-boundary sensor data with GPS measurements. The standard deviation of position error in the lateral direction is plotted in Figure 5, as a function of the number of users equipped with lane-boundary sensors. As shown in the figure, there is a dramatic reduction in lateral positioning error **for all users** even when a single user is equipped with a lane-boundary sensor. Specifically, the lateral positioning error falls from 5.3 m (case of no user equipped) to 1.6 m (case of one user equipped). In effect, the lane-boundary sensor for the first user generates a differential correction which is distributed to reduce the GPS errors experienced by all other users. (For these simulations, communication of pseudorange and lane-boundary data is modeled as essentially instantaneous.) As more users are equipped, the estimate of the lateral differential correction improves, resulting in progressively better lateral-positioning accuracy with increased equipage. For the case in which all but the fifth user is equipped (the case labeled “4/1” in Figure 5), the standard deviation for the unequipped user’s lateral positioning error is 1.3 m. The lateral positioning error can be seen to converge toward the idealized case, in which the satellite-specific error is set to zero. This idealized case is shown as a dashed line in Figure 5.

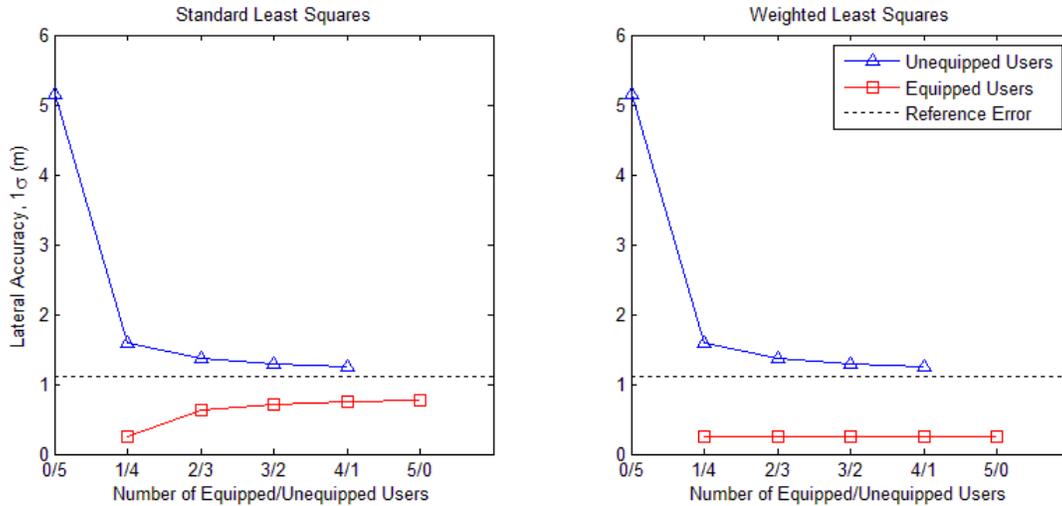


Figure 5. Lateral Accuracy for Collaborative Navigation with a Varying Numbers of Users Equipped with Lane-Boundary Sensors

Unexpectedly, the lateral position error for equipped users grows worse as the number of equipped users increases! This anomaly is an artifact of using a standard least-squares solution rather than a weighted least-squares solution. In the unweighted least-squares solution, the lateral error is lowest (for the equipped user) when only a single user has a lane-boundary measurement available. In this case, the least-squares solution computes the user’s lateral position using only the lane-boundary sensor data, as was the case in the single-user simulations illustrated in Figure 4. When lane-boundary measurements are made by more than one user, then the discrepancy between the lane-boundary measurements and the GPS measurements can no longer be “pushed” into the lateral component of the satellite-specific error c_t , because thermal noise and multipath introduce a different discrepancy for each equipped user. The greater the number of equipped users, the more thermal noise and multipath are “averaged out”. Hence the quality of the differential correction c_t improves, but the ability of the differential correction to remove thermal noise and multipath from individual user solutions is diminished.

An obvious remedy is to introduce a weighted least-squares solution. The weighted least-squares solution appropriately accounts for the accuracy of each individual sensor measurement. Since the lane-boundary sensors are much more accurate than any individual GPS pseudorange

measurement, the lane-boundary sensor data provides significantly more information for lateral position estimation. Hence, in weighted least-squares, each equipped user's lateral position is essentially determined by the lane-boundary sensor measurement. The lateral positioning accuracy is thus essentially flat for equipped users in weighted least-squares (see right plot of Figure 5).

The weighted least-squares solution used in this analysis was a diagonal matrix \mathbf{W} . The elements of the diagonal were $(12 \text{ m})^{-2}$ for each pseudorange and $(0.25 \text{ m})^{-2}$ for each lane-boundary measurement. To incorporate the weighting matrix, the least-squares solution to (38) was modified as follows.

$$\delta \hat{\mathbf{z}}_{CN} = (\mathbf{G}_{CN}^T \mathbf{W} \mathbf{G}_{CN})^{-1} \mathbf{G}_{CN}^T \mathbf{W} \delta \mathbf{y}_{CN} \quad (46)$$

2.3.4. Decentralized Method Simulations

The previous subsection simulates the algorithm by using the centralized method, i.e. each user receiver should solve for the augmented least-squares problem. This was shown in the previous simulation to be time consuming and computationally intensive. To address this problem, a decentralized approach is introduced in Section 2.2.3. This subsection shows the simulation results using the decentralized approach.

For comparison, the same scenarios of different number of cars equipped with lane-boundary sensors were simulated. To be specific, in each scenario, the user(s) with lane-boundary sensor would solve a linear least-squares problem (33) for state vector (31). For user(s) without lane-boundary sensor, average value of the shared correction c_t is applied to Equation (42). Then user's position will be solved according to Equation (43) to (45).

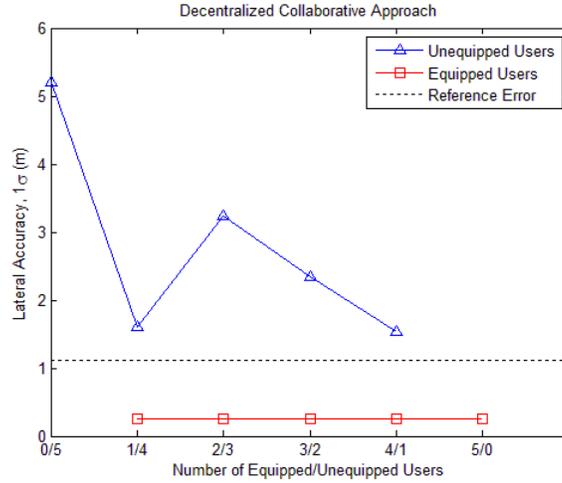


Figure 6. Lateral Accuracy for Decentralized Collaborative Navigation Algorithm with a Varying Numbers of Users Equipped with Lane-Boundary Sensors

As shown in the Figure 6, for the equipped users, the decentralized collaborative approach has the same accuracy as the case in centralized approach. For the unequipped users, this approach also reduces their lateral position error. However, in the case of 2 equipped users and 3 unequipped users, there is an unexpected higher error. Similar result shows in the case of 3 equipped users and 2 unequipped users. The reason for these unusual results is that, the decentralized approach does not take into account of the users' different geometry. In the centralized method, while it is computationally inefficient to solve a large size least-squares problem (38), it accounts for different users' geometry. Thus its position error is stable. In the decentralized method, however, unequipped users apply only the shared corrections to their solutions. While these corrections can reduce common-mode errors in the lateral direction, they introduce additional errors produced by those users who create and share the corrections. Example of these additional errors includes but not limit to multipath and receiver-related random noise.

Despite the uncertain additional errors of the decentralized approach, this method will be used to process the experimental data. The specific reason will be discussed in the experiment chapter.

CHAPTER 3: EXPERIMENTAL VERIFICATION OF ALGORITHM

Chapter 2 introduces the development and simulation of the novel collaborative navigation algorithm. In this chapter, experimental verification of the algorithm is the focus. In the following sections, Section 3.1 first talks about the GPS receiver that was used in this study and its corresponding software. Then the second section introduced the determination of ground reference. Finally in the last section, experiments and the corresponding results are discussed. An analysis of error is also presented in the last section.

3.1. GPS RECEIVER AND SOFTWARE

3.1.1. SIRF Star chipset GPS receiver

The GPS receiver used for this study is a product of GlobalSat Technology Corporation (Figure 7). It uses the GPS chipset SiRF Star III that manufactured by SiRF Technology Inc (owned by CSR now). Its model number is BU-353. The receiver has a compact design with built-in roof mount magnet, which makes it especially suitable for on-field, mobile working environment. Meanwhile, since the BU-353 uses USB interface to transfer both power and data, the installation of this receiver on a Windows PC is simple and quick.



Figure 7. GPS Receiver

After installing the BU-353 receiver on a Windows PC, a user would like to see its output. Unlike popular GPS products for personal vehicle drivers, there is no screen interface on this receiver to output its data or computed information. Thus third-party software is needed in order to capture the output data from the receiver and show information on a computer. Since the receiver uses SiRF Star III chipset, the experiments use the software called SiRFDemo, which is published by the same company SiRF Technology Inc, to capture, display and log the data from BU-353 receiver.

By default, SiRFDemo uses NMEA (National Marine Electronics Association) protocol to communicate with the receiver. NMEA protocol does not provide raw GPS measurements such as pseudoranges, satellites' positions and velocities and etc. However, for the purpose of this study, these raw GPS measurements are the very data that are desired from the receiver. In order to obtain these measurements, another protocol that is closer to the physical communication layer is desired. This purpose can be achieved by SiRFDemo. It has a neat function to switch the communication protocol between NMEA and SiRF Binary. SiRF Binary Protocol is the standard interface protocol of SiRF Star products. It defines two-way communication messages: input messages for the receiver and output messages from the receiver. The output messages include all the desired measurements such as pseudoranges. Below are tables that list the output messages that are useful for this study [33].

Table II. SiRF Binary Messages – Output Message List (partial excerpt)

Message ID	Name	Description
2	Measured Navigation Data	Position, velocity, and time
6	SW Version	Receiver software
7	Clock Status	Current clock status
14	Almanac Data	Response to poll
15	Ephemeris Data	Response to poll
19	Navigation Parameters	Response to poll
28	Nav. Lib. Measurement Data	Measurement data
29	Nav. Lib. DGPS Data	Differential GPS data
30	Nav. Lib. SV State Data	Satellite state data
31	Nav. Lib. Initialization Data	Initialization data

Among the listed messages in the above table, the information from message ID 2, 7, 28 and 30 is used to input into the algorithms and compute position results. Below is the detailed formatting information of each message. Notice that there is a portion of output messages attached in the Appendix A.

Table III. Message ID 2: Measured Navigation Data Output (partial excerpt)

Name	Bytes	Binary (Hex)		Unit	ASCII (Decimal)	
		Scale	Example		Scale	Example
Message ID	1 U		02			2
X-position	4 S		FFD6F78C	m		-2689140
Y-position	4 S		FFBE536E	m		-4304018
Z-position	4 S		003AC004	m		3850244
X-velocity	2 S	*8	0000	m/sec	V _x /8	0
Y-velocity	2 S	*8	0003	m/sec	V _y /8	0.375
Z-velocity	2 S	*8	0001	m/sec	V _z /8	0.125
Mode 1	1 D		04	Bitmap		4
HDOP	1 U	*5	0A		/5	2.0
Mode 2	1 D		00	Bitmap		0
GPS Week	2 U		036B			875
GPS TOW	4 U	*100	039780E3	sec	/100	602605.79
SVs in Fix	1 U		06			6

Information from Message ID 2 includes the receiver position and velocity results calculated by the receiver itself. Specifically, the second byte to the fifth byte of the message contains user position result in X-axis. The sixth byte to ninth byte and the tenth byte to thirteenth byte contain the position results in Y-axis and Z-axis, respectively. The information provided in Message ID 2 can be used as rough reference for results computed by the developed algorithm. It provides an approach to quickly check the estimated results and evaluate the quality of data sets.

Table IV. Message ID 7: Clock Status Data (receiver's clock)

Name	Bytes	Binary (Hex)		Unit	ASCII (Decimal)	
		Scale	Example		Scale	Example
Message ID	1 U		07			7
Extended GPS Week	2 U		03BD			957
GPS TOW	4 U	*100	02154924	sec	/100	349494.12
SVs	1 U		08			8
Clock Drift	4 U		00012231	Hz		74289
Clock Bias	4 U		00004728	ns		18216
Estimated GPS Time	4 U		14D4DAEF	ms		349493999

Data from Message ID 7 are also the results calculated by the receiver itself. For the use of experiment, the clock bias and estimated GPS time from this message are used to assist calculation.

Table V. Message ID 28: Navigation Library Measurement Data (partial excerpt)

Name	Bytes	Binary (Hex)		Unit	ASCII (Decimal)	
		Scale	Example		Scale	Example
Message ID	1 U		1C			28
Channel	1 U		00			0
Time Tag	4 U		000660D0	ms		135000
Satellite ID	1 U		15			20
GPS Software Time	8 Dbl		41740B0B48353F7D	sec		2.4921113696e+005
Pseudorange	8 Dbl		7D3F354A0B0B7441	m		2.1016756638e+007
Carrier Frequency	4 Sgl		89E98246	m/s		1.6756767578e+004
Carrier Phase	8 Dbl		A4703D4A0B0B7441	m		2.1016756640e+007
Time in Track	2 U		7530	ms		10600
Sync Flags	1 D		17			23

Message ID 28 contains the important pseudorange measurement information. For each satellite that is tracked by the receiver, there is one Message ID 28. For example, if there are 6 satellites observed by the receiver, there should be 6 sequential messages labeled ID 28, each one corresponding to one satellite measurement, indicated by Satellite ID. An example of Message ID 28 is shown in Figure 8. In this example, there are measurements from 3 satellites.

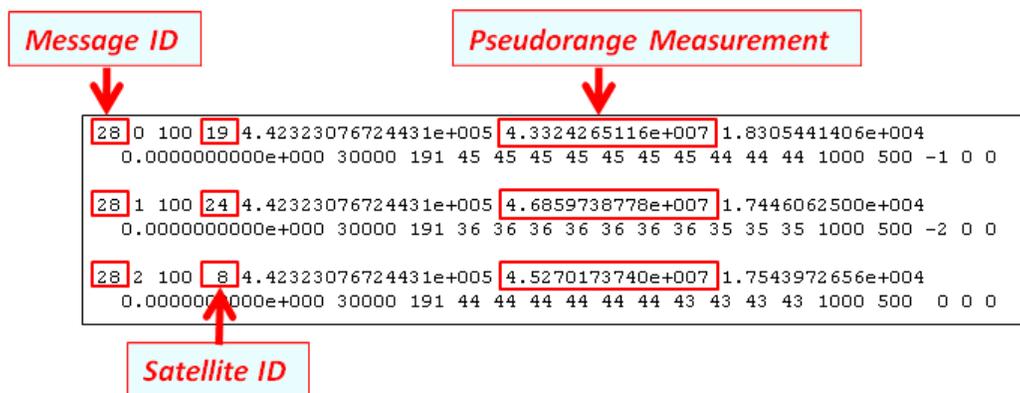


Figure 8. Example of Message ID 28

Table VI. Message ID 30: Navigation Library SV (Space Vehicle) State Data

Name	Bytes	Binary (Hex)		Unit	ASCII (Decimal)	
		Scale	Example		Scale	Example
Message ID	1 U		1E			30
Satellite ID	1 U		15			21
GPS Time	8 Dbl			sec		
Position X	8 Dbl			m		
Position Y	8 Dbl			m		
Position Z	8 Dbl			m		
Velocity X	8 Dbl			m/sec		
Velocity Y	8 Dbl			m/sec		
Velocity Z	8 Dbl			m/sec		
Clock Bias	8 Dbl			sec		
Clock Drift	4 Sgl		2C64E99D	Hz		744810909
Ephemeris Flag	1 D		01			1
Reserved	4 Sgl					
Reserved	4 Sgl					
Ionospheric Delay	4 Sgl		408906C8	m		1082721992

Information in Message ID 30 provides the position and velocity information of the entire satellite constellation. These data are used in Equation (5) in order to generate an estimate pseudorange.

3.1.2. Introduction of SiRFDemo

This section introduces how to setup SiRFDemo software to run the experiment. Only the functions that were used for the experiments are introduced here. For more detailed introduction about this software, readers can refer to the User Manual [34].

1) Startup

To start the software, user can simply double click the icon on desktop of a Windows computer. Then a window pops up asking the data source type (Figure 9). Data source can be outside source from a serial port connected to a signal generator, or a saved file on computer, or even random signal generated by user's computer. For the experiment, data is input through the serial port connected to a GPS receiver, such as BU-353. By default, the receiver uses NMEA protocol, and the corresponding baud rate is 4800. For SiRF Binary protocol, the software uses baud rate 57,600 to transfer data. After first time of configuration, the software can remember the setting.

Figure 9 shows the baud rate 57,600, indicating that the communication protocol was set as SiRF Binary protocol. The port number is the com port occupied by the receiver. User can find the com port number from control panel in Windows OS. By clicking OK in this window, the main user interface will appear (Figure 10).

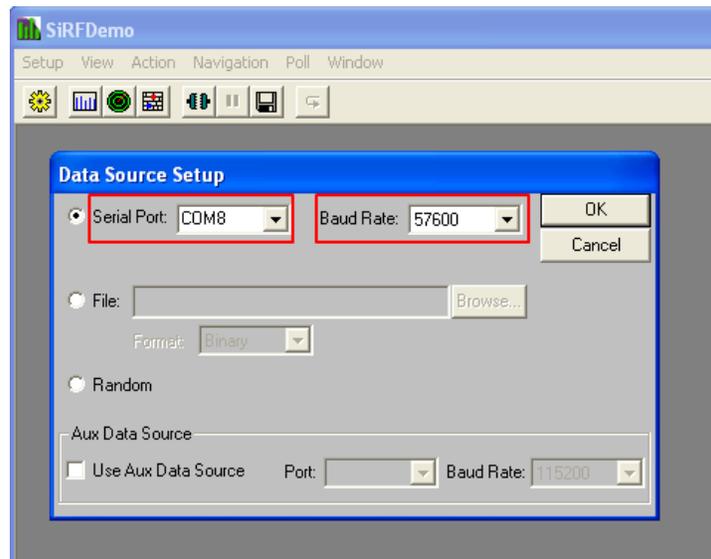


Figure 9. Data Source Setup Dialogue Box

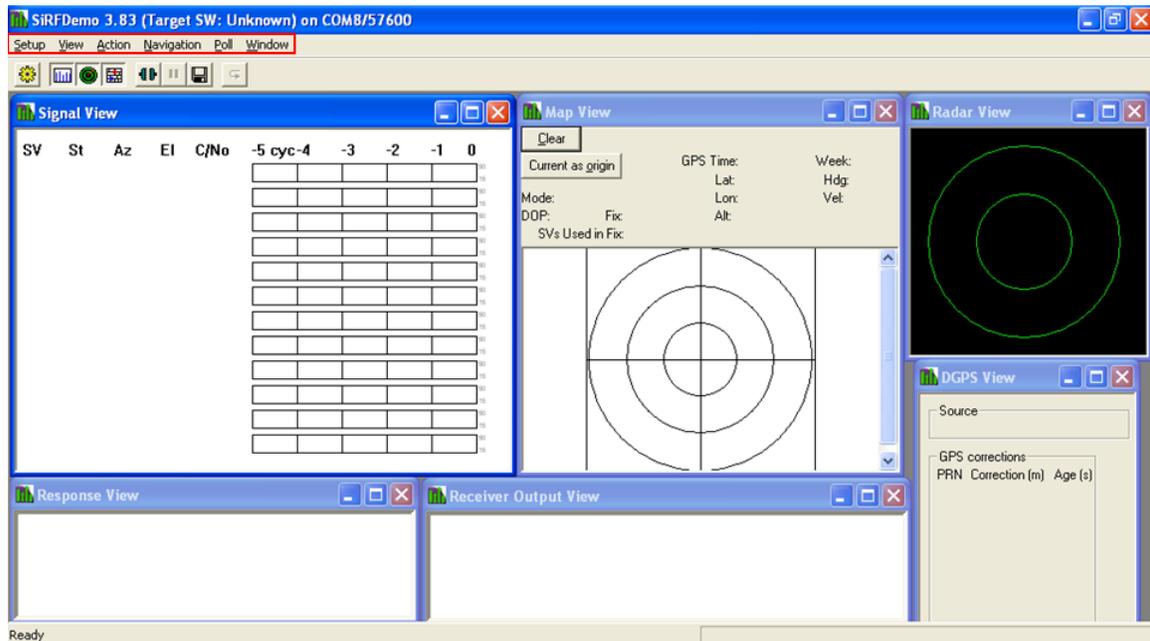


Figure 10. Main User Interface

2) Menu Bar

In the main user interface, there is a menu bar on the top. Under the “Setup” menu, user can click “Target S/W” to choose the model of receiver (Figure 11). By selecting the “Auto-detection” box in the “Select Target Software” window (Figure 12), software will detect the receiver model number automatically. In the same menu, user can click “Data Source” to change data source type. This will popup the same window as in the startup operation (Figure 9).

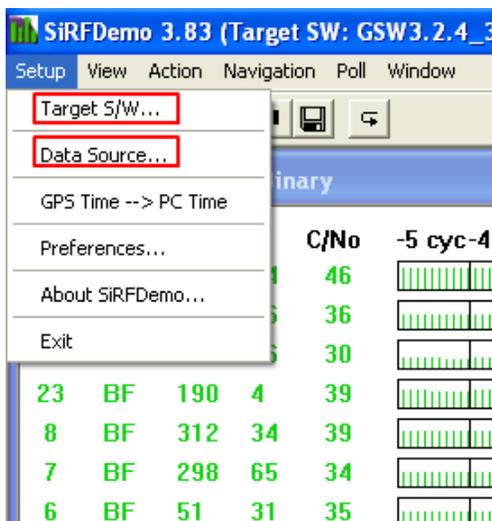


Figure 11. “Setup” Menu

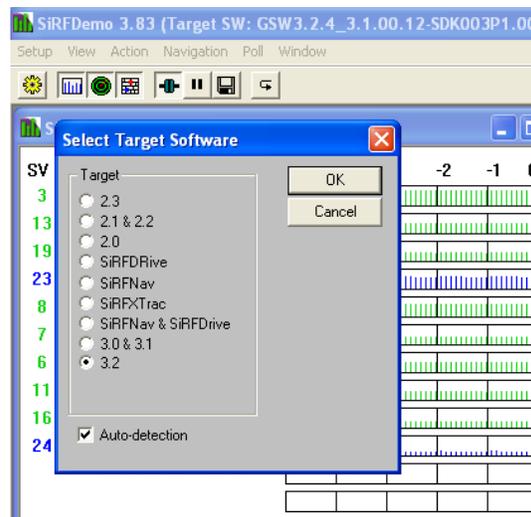


Figure 12. Receiver Model Selecting Window

Under the “View” menu (Figure 13), user can choose which windows are shown on the panel. Useful windows include: Signal View, Map View, Radar View, Response View, and Receiver Output View. Some detailed information of these view windows will be introduced later in this section.

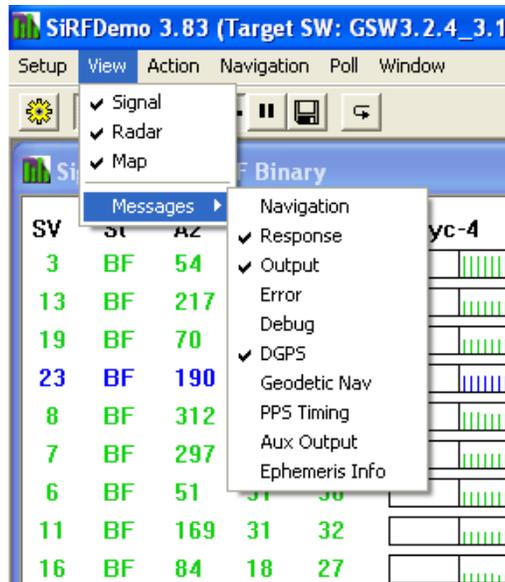


Figure 13. "View" Menu

The "Action" menu (Figure 14) includes some important functions for the experiment. First it is the "Open Data Source". The computer-receiver communication channel is established by clicking this. Thus computer can obtain receiver's data and show them on the main user interface. Then it is the "Open Log File". This enable user to store a data file in the computer. User can indicate saving path, logging data types (i.e. Message IDs) and even logging start and end time through the setting window (Figure 15). Next one is the "Initialize Data Source". By clicking this, a dialogue box will appear (Figure 16). User can thus choose different way to initialize the receiver. It is essential to initialize the data source every time the receiver configuration is changed.

The "Switch to NMEA Protocol" button is enabled if the software is using SiRF Binary protocol. In this case, the "Switch to SiRF Protocol" button is grayed, as Figure 14 shows. User can conveniently switch between NMEA protocol and SiRF Binary protocol via these two options. Because NMEA and SiRF Binary protocol require different baud rate to transfer data, it is necessary to "Synchronize Protocol & Baud Rate" every time user switches the protocol. Finally, the "Pause Display" option above the initialization option can "freeze" the data displaying in the

main user interface. It is useful to stop the output information “scrolling” and manually check the output.

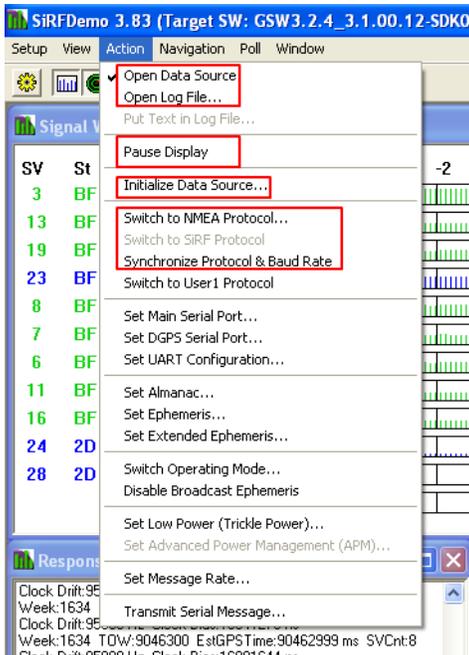


Figure 14. "Action" Menu

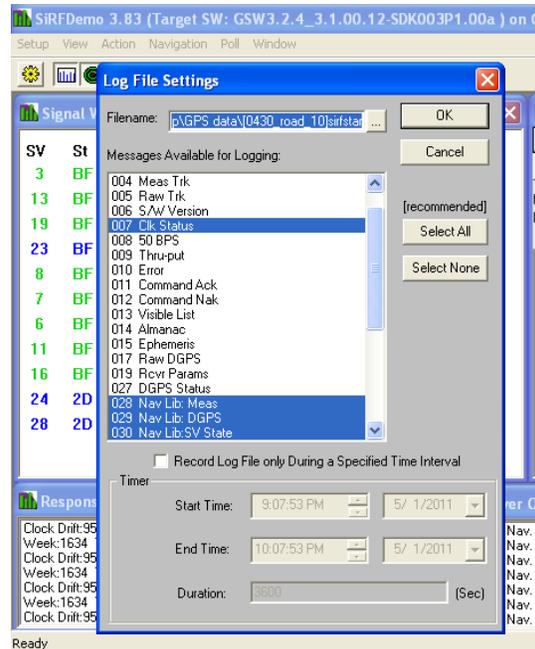


Figure 15. Logging Data File

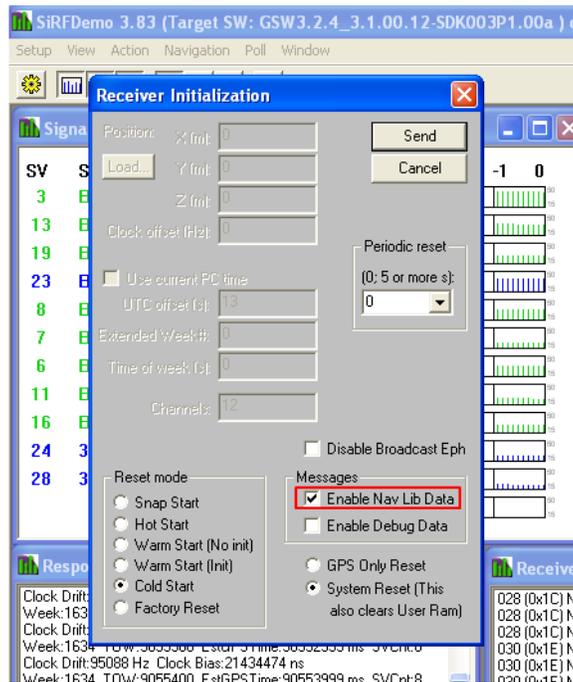


Figure 16. Initializing Data Source

User can choose DGPS (Differential GPS) mode and DGPS source under the “Navigation” menu (Figure 17). For this specific receiver, DGPS usually relies on SBAS-borne differential corrections. In the “DGPS Mode” dialogue box, user can choose one from the three modes including “Automatic”, “Exclusive” and “Never use” (Figure 18). In the experiment, “Never use” mode is selected to deactivate the DGPS function of the receiver since the proposed approach will calculate differential correction itself. In the “DGPS Source” window, user can choose different sources of differential corrections (Figure 19). Based on the same reason, “None” option is chosen for experiment.

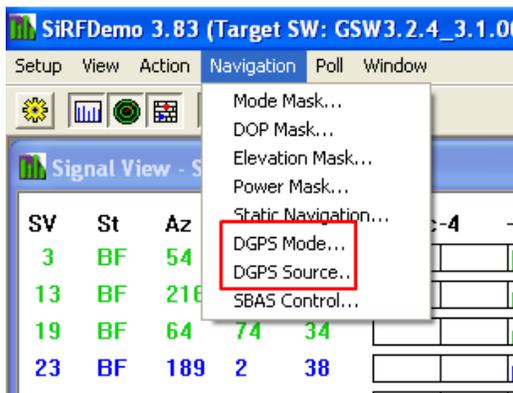


Figure 17. "Navigation" Menu

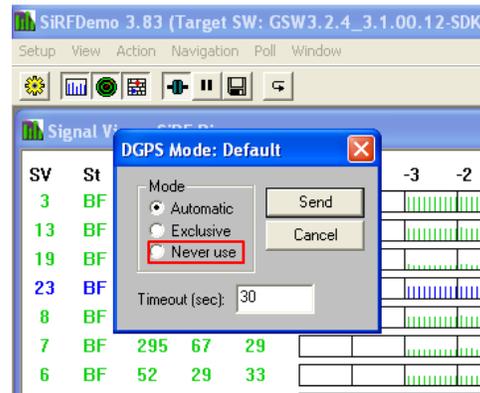


Figure 18. Selecting DGPS Mode

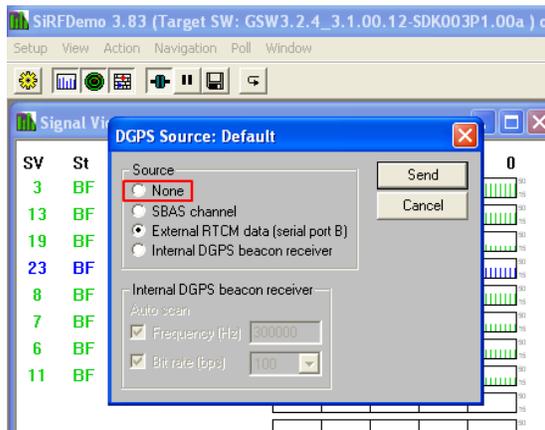


Figure 19. Selecting DGPS Source

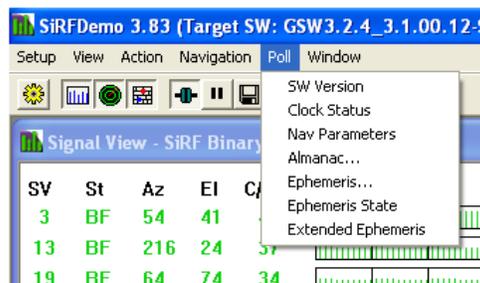


Figure 20. "Poll" Menu

In the “Poll” menu, user can manually acquire different information like receiver model number and ephemeris data (Figure 20). The output of “Almanac” and “Ephemeris” information will be saved into a text file. Other polled information will be output in the “Response View” window.

3) Main User Interface

Main user interface includes several separate windows showing receiver output data, a menu bar on top and some hotkey buttons under the menu bar (Figure 21). Hotkeys include “Data Source Setup”, “Signal View”, “Radar View”, “Map View”, “Connect to Data Source”, “Pause Display”, “Open Log File”, and “Init Receiver” (Figure 22). “Data Source Setup” is a shortcut to open data source setup window. If the “Signal View”, “Radar View” and “Map View” buttons are pressed, corresponding view windows will be shown in the main user interface. “Connect to Data Source” has the same function as that of “Open Data Source” under “Action” menu (Figure 14). When data source is connected, “Pause Display” and “Init Receiver” buttons become active. Thus users can either freeze the data display in main user interface or initialize data source. “Open Log File” is a hotkey to create data log file. An example of created log file can be found in the Appendix A.

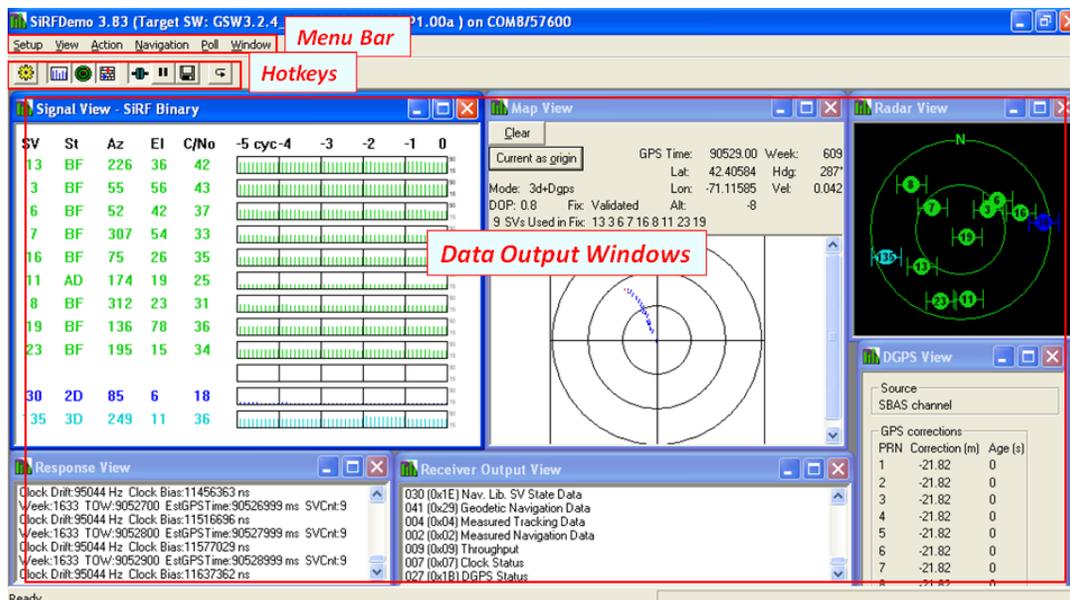


Figure 21. Main User Interface



Figure 22. Hotkeys on Main UI

In experiments, when computer is connected to the data source (i.e. “Connect to Data Source” button is pressed), data will start displaying in the interface (as shown in Figure 21). According to signal strength, “Signal View” window shows signal with different colors (Figure 23). Green color indicates the satellite signal is healthy and the receiver is tracking and receiving this signal. Blue color means unstable satellite signal and the receiver is only tracking this signal but does not use it to compute user position. Red color indicates the satellite is known from almanac information but not being tracked by the receiver. “Radar View” window has the same color-coding pattern and it shows the satellites scatter in user’s sky (shown in Figure 23).

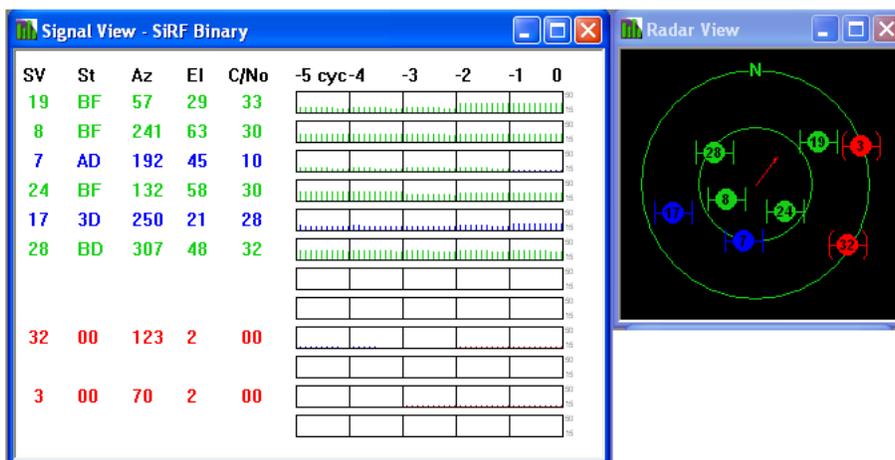


Figure 23. Signal Strength Color Coding

In the “Map View” window (Figure 24), by clicking the “Current as Origin” button, position calculated at the moment of clicking is used as the origin in the Map View. Then subsequent computed positions will be plotted relative to this origin. The scale of this Map View is set to be 10, 20 and 30 meters in radius for respective circle and each pixel represents 0.3 meter. This view is useful to have a brief idea of how accurate the receiver is. Or when there is no problem with the receiver, this is also helpful to determine whether it is a good condition for experiment.

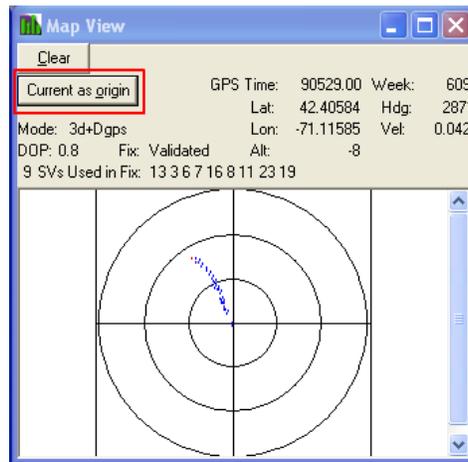


Figure 24. Relative Positions on a Map View

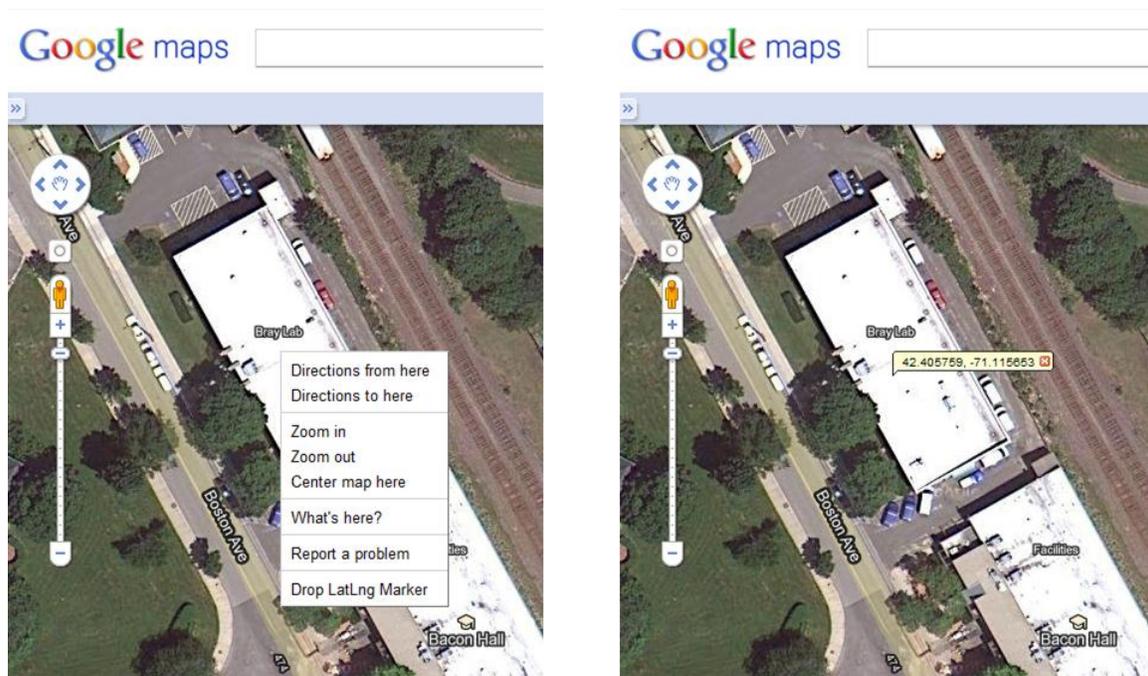
3.1.3. Procedure of Logging Data on SiRFDemo

In order to store the experimental data and save them as a text file, user can follow this simple procedure. After starting up the SiRFDemo (with proper serial port and baud rate settings, Figure 9), user click the “Connect to Data Source” hotkey in the Main UI (Figure 22). User should wait for a while until the receiver has good signal reception (as Figure 21 shows, all signals are in “green” condition). This usually will not be longer than 2 minutes. When logging the data, user should press the “Open Log File” hotkey in Main UI (Figure 22) or the same option under the “Action” menu (Figure 14). This will prompt a window shown as Figure 15. After setting desired filename (i.e. saving path) and the desired logging messages, user click “OK” in this window. When data is being logged, the “Open Log File” hotkey in the Main UI will be stayed “pressed”.

To stop logging data, user simply presses the pressed “Open Log File” hotkey again. The hotkey will be popped and the logging process is terminated.

3.2. GROUND REFERENCE AND MAP SOURCES

For experiment, it is necessary to know where the receiver exactly is, i.e. the ground truth reference of the user position. Also it is essential to have accurate data of the absolute position of the lane boundary. A well established map is a good source of such information. For this study, Google Maps with satellite images was used to obtain reference absolute data in latitude-longitude coordinates. Moreover, a feature, named “LatLng Marker”, in Google Maps Lab must be enabled in order to access useful data. When this feature is enabled, user can right click any point on Google Maps and choose to “Drop LatLng Marker” (shown as Figure 25).



*Figure 25. Google Maps "LatLng Marker" Tool
Left: Right-click on An Arbitrary Point to Show the Menu with "Drop LatLng Marker" Option
Right: A Latitude-Longitude Position Was Marked on the Google Maps*

Google Maps provides the useful latitude and longitude information. However, in order to transform the latitude-longitude coordinates into earth-center earth-fixed (ECEF) coordinates,

geoid height information is needed [29] [35]. In geography, geoid height can be calculated from latitude and longitude values according to different geoid models. MATLAB can perform this calculation based on Earth Gravitational Model 1996 (EGM96) [36].

3.3. EXPERIMENTS AND RESULTS

3.3.1. Experiment Design and Setup

The experiment is intended to test the collaborative algorithm in practices. In accordance with simulations, at least two cars with both GPS receiver and camera are needed to establish a V2V network. For the current stage, however, there is no capability to build two experimental cars. Thus only one set of equipments was used in the experiments. In order to be able to test the algorithm, some modifications were made to the experiment. Thus it can be run with only one set of equipment but still be able to simulate the two-car communication scenario. Here is the idea.

As Figure 1 shows, an important additional measurement is the lateral distance d measured from the receiver to the lane boundary. In a simple two-car networked case, two vehicles are on the same straight road and are separated by a distance D (Figure 26, top). For simplicity, they are heading to the same direction so that the lateral distance measurements from the lane boundary are on their same side if both of them have lane-boundary sensor. When one car has lane-boundary sensor (e.g. a camera) and the other one does not, lateral distance measurement from first car can be used for the second car without any change when they are in the same direction. Also in this scenario, lateral distance measurement from the first car can be used for the second car is because they are not far away from each other and the common-mode errors (e.g. ionosphere and troposphere error) are almost the same in a local area.

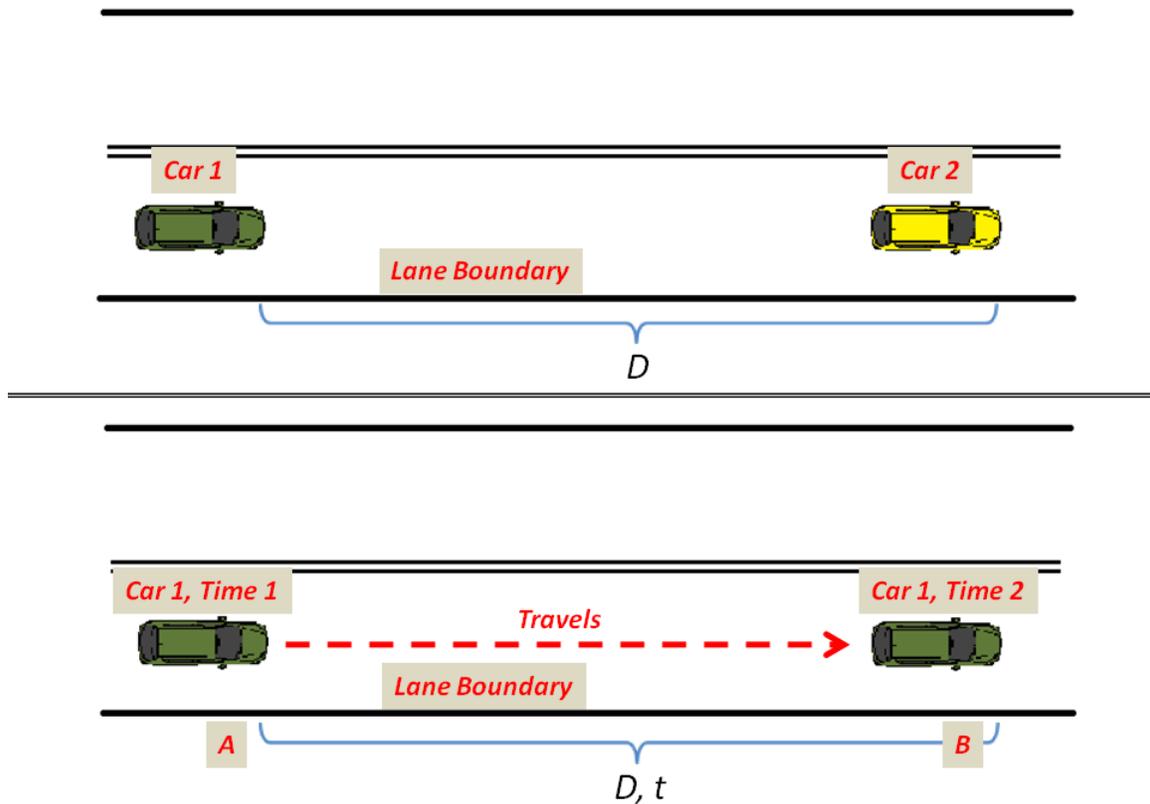


Figure 26. Two-Car Networked Case
 Top: Two Cars Separated by Distance D
 Bottom: One Car Travels Distance d and Spends Time t

Common-mode errors are quite the same for a short pass of time as well [29] [37] [38]. Thus lateral distance measured at point A of time 1 can be used for nearby points (e.g. point B) of a later time 2 to compensate the lateral component of common-mode errors. In Figure 26, the bottom one shows that there is only one car on a straight road way. After a short period of time t , it travels a distance of D , from point A to point B . Based on the time-correlated feature of common-mode errors, lateral distance measurement obtained at point A can be used at point B and it is still possible to reduce the errors in lateral direction. Therefore with only one set of equipments, it is possible to simulate a two-car case. Obviously, the only car has both lane-boundary and GPS sensors. When calculating the position at point B , however, it is assumed there is only the GPS measurement and only the pseudoranges are used for calculation. Thus it looks like at point B , the car loses its lane-boundary sensor.

Having this idea, experiment is possible to run with only one set of devices. However, it is necessary to state again that this thesis study mainly deals with positioning approach. It is possible to combine the vision detecting algorithm with positioning algorithm to process experimental data. The vision detecting algorithm, however, was not complete by the time the experiment was conducted. It was not consistent for detecting an arbitrary lane boundary. Thus camera lane-boundary sensor was not adopted in the experiment. Alternatively, a manual approach to measure lateral distance was developed for the experiment. This approach is introduced below.

Without a camera-based lane-boundary sensor, it is difficult to determine the lateral distance to lane boundary while doing the experiment. One possible way of addressing the problem is to use sand to track the route of receiver. For example, if a vehicle passes a road covered with sand, there is track of its tire left on the ground. By measuring distance from the track to the road edge, we can have a measurement of lateral distance. This measurement is rough, however, and it is not possible to have the whole road covered by sand. Another approach is to pre-measure the lateral distance. For example, two persons are tied by a rope and pulling the rope tightly. Then the distance of these two persons is the length of the tight rope. When doing the experiment, one of these two persons is walking on the lane boundary and the other one is walking in the road and holding the GPS receiver. Thus their walking tracks can be considered as the track of the lane boundary and the track of receiver, respectively. Then the desired lateral distance is known for each moment of the experiment. The drawbacks of this approach are that, 1) the two persons should make sure they keep the rope tight during the whole routine, and 2) they should try to move synchronously to avoid movement conflict. Therefore, it is difficult to practice the method.

For the pre-measured approach, however, there are potential other ways to pre-measure. On a road with explicit lane marker, if considering this lane marker as the lane boundary and let the receiver move directly on the lane marker, then we can think the lateral distance from the receiver

to the lane boundary is always **zero**. An example on College Avenue (the section in front of Halligan Hall) is shown in Figure 26. This pre-measured approach can adapt any case so long as there is explicit lane marker painted on the ground. Based on this criterion, Cousens Parking Lot and Boston Avenue are chosen to be experimental places besides College Avenue (Figure 27, Figure 28). Table VII – Table IX list the surveyed latitude-longitude values of reference points. The values are determined from Google Maps, as Section 3.2 shows.



Figure 27. Bicycle Lane Marker and Lane Boundary on College Avenue

Table VII. Latitude-Longitude Coordinates of Reference Lines on College Avenue

	Coordinates on Lane Marker/Lane Boundary
1	(42.407433, -71.116256)
2	(42.407530, -71.116162)
3	(42.407634, -71.116063)
4	(42.407755, -71.115948)
5	(42.407859, -71.115846)
6	(42.407971, -71.115740)
7	(42.408083, -71.115643)
8	(42.408187, -71.115535)



Figure 28. Lane Marker and Lane Boundary in Cousens Parking Lot

Table VIII. Latitude-Longitude Coordinates of Reference Lines on Cousens Parking Lot

	Coordinates on Lane Marker/Lane Boundary
1	(42.407334, -71.115775)
2	(42.407391, -71.115720)
3	(42.407453, -71.115661)
4	(42.407524, -71.115594)
5	(42.407585, -71.115535)
6	(42.407650, -71.115476)
7	(42.407705, -71.115421)
8	(42.407748, -71.115380)

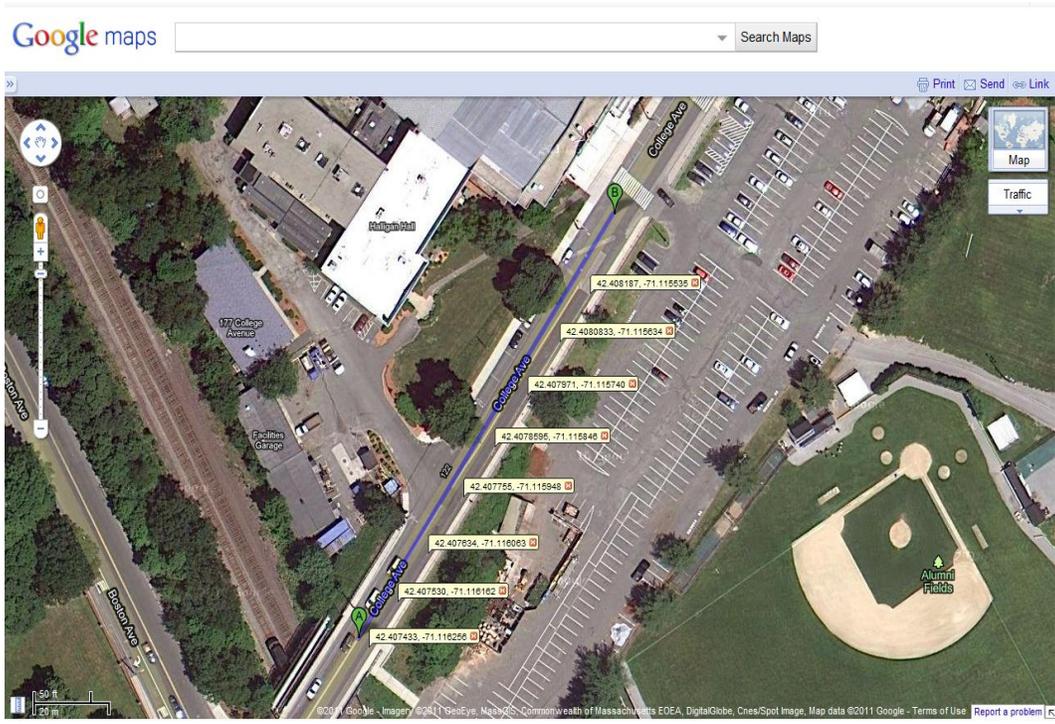


Figure 29. Lane Marker and Lane Boundary on Boston Avenue

Table IX. Latitude-Longitude Coordinates of Reference Lines on Boston Avenue

	Coordinates on Lane Marker/Lane Boundary
1	(42.406062, -71.116300)
2	(42.405970, -71.116232)
3	(42.405754, -71.116071)
4	(42.405543, -71.115910)
5	(42.405366, -71.115780)
6	(42.405077, -71.115564)
7	(42.404934, -71.115460)
8	(42.404704, -71.115290)
9	(42.404540, -71.115164)
10	(42.404387, -71.115049)
11	(42.403889, -71.114680)
12	(42.403778, -71.114600)
13	(42.403647, -71.114497)
14	(42.403514, -71.114397)

With the information in Table VII – Table IX, the road reference point \mathbf{R}_n and the transverse road unit vector \mathbf{u}_t in Equation (25) can be determined. As Figure 30 shows, values in Table VII – Table IX are a series points surveyed in Google Maps. Since the road way is straight, the road reference point \mathbf{R}_n can be simply chosen as the first surveyed point. For the transverse road unit vector \mathbf{u}_t , some calculations are needed. Based on the coordinates of surveyed points, we cannot obtain \mathbf{u}_t directly. However, the along-road unit vector \mathbf{u}_r can be determined directly from the surveyed coordinates – simply subtract the coordinates of point A from point B and divide the vectorial difference by its magnitude, and then it is the unit vector pointing from point A to point B. Thus transverse road unit vector \mathbf{u}_t can be obtained by rotating \mathbf{u}_r 90 degrees clock-wise. For clarity, the road parameters are expressed in earth-fixed, earth-center (ECEF) coordinate frame. The coordinates of lane marker is the reference for user’s position estimates calculated from experimental data.



*Figure 30. Example of Latitude-Longitude Positions of College Avenue (Created from Google Maps)
 ©2011 Google – Imagery ©2011 GeoEye, MassGIS, Commonwealth of Massachusetts EOE, DigitalGlobe, Cnes/Spot Image, Map data ©2011 Google*

When the road parameters are found, it is ready to conduct experiment. As demonstrated in Figure 31, the hardware for experiment includes a long post mounted with a GPS receiver on top and a small web book computer running Windows XP OS. During experiment, a person holds the long post and walks on the lane marker (as shown in the right part of Figure 31). In order to keep the GPS receiver right above the lane marker, the person should try to keep the post as stable as possible while he is walking. A way to determine whether the post is stable is to inspect the ground projection of the bottom. If the projection keeps inside the lane marker’s painting range, then the post is said to be stable. The post was chosen to be long is for easier inspecting the ground projection. In Figure 31, the holding point of the post is a static point in experiment. That means when the post swings, it swings about this point.

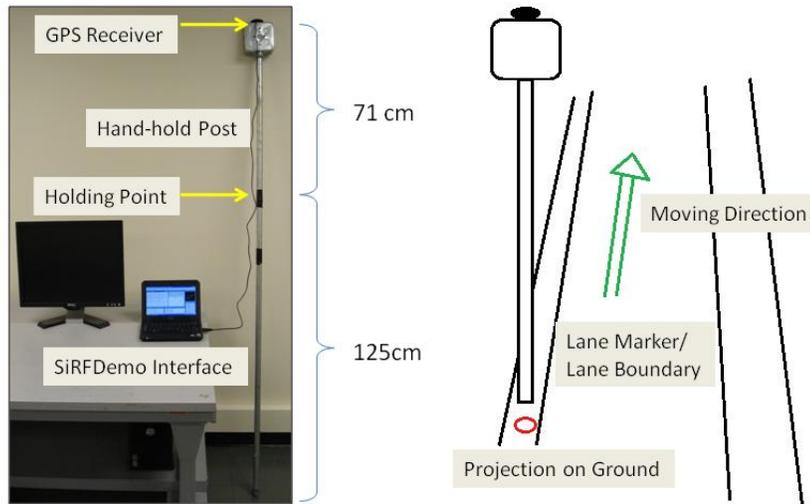


Figure 31. Experiment Hardware Setup and Routine Demonstration

For clarity, other experimental hardware setups were tried (e.g. computer and receiver were put on a handcart, and receiver was mounted on bicycle). However, those setups are not suitable although they sound more convenient. Because the experiment is designed to track the lane marker in order to have the lateral distance measurement; most of the time the experiment has to be implemented in traffic. Pushing a handcart is hard to response to oncoming traffic. Though riding a bicycle may be easier to avoid from traffic, it is difficult to track the lane marker precisely. Therefore the current setup was adopted for the experiment.

3.3.2. Experiment Routine

As mentioned in the previous subsection, experiments were taken place on 1) College Avenue, the section between Anthony Pagano Bridge and Cousens Gym (Figure 32); 2) Cousens Parking Lot, southwestern corner (Figure 33); and 3) Boston Avenue, the section between Bray Labs and St Clement School (Figure 34). All the experimental routes were chosen to be straight. As discussed in last subsection, a person holding the post and computer will walk on the lane marker along the road. Figure 32 – Figure 34 also shows the starting/end points of walking. In order to collect more data, experiments at each place were implemented in two directions. For example on College Avenue, experiments were run from Anthony Pagano Bridge to Cousens Gym and from

Cousens Gym to Anthony Pagano Bridge. To be specific, the correction will be created only at the starting/end points of experiment route. The rest of the points along the route will use this correction to assist position calculation.

- **College Avenue**

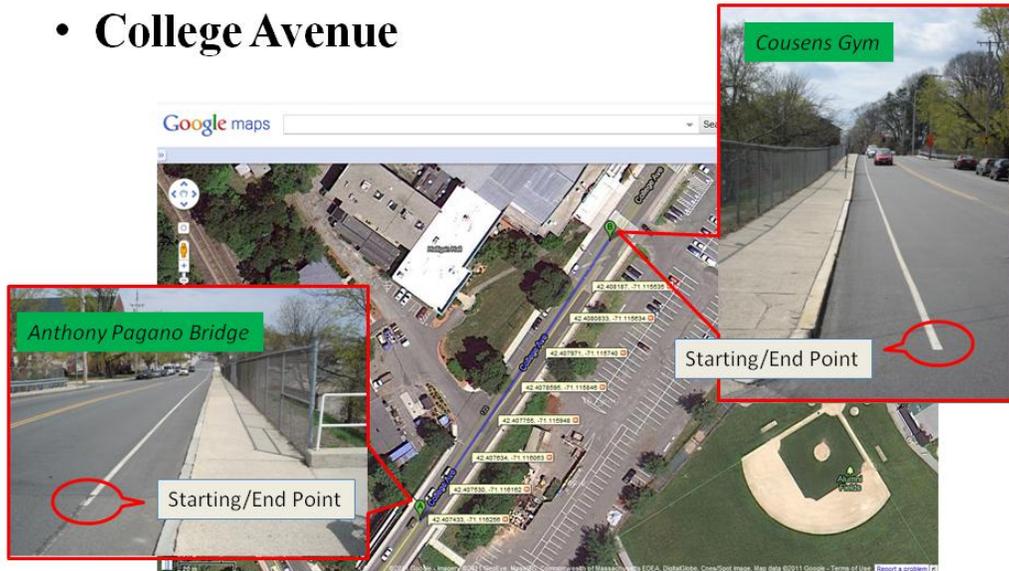


Figure 32. Experiment Route on College Avenue

- **Cousens Parking Lot**



Figure 33. Experiment Route in Southwest Corner of Cousens Parking Lot

- **Boston Avenue**

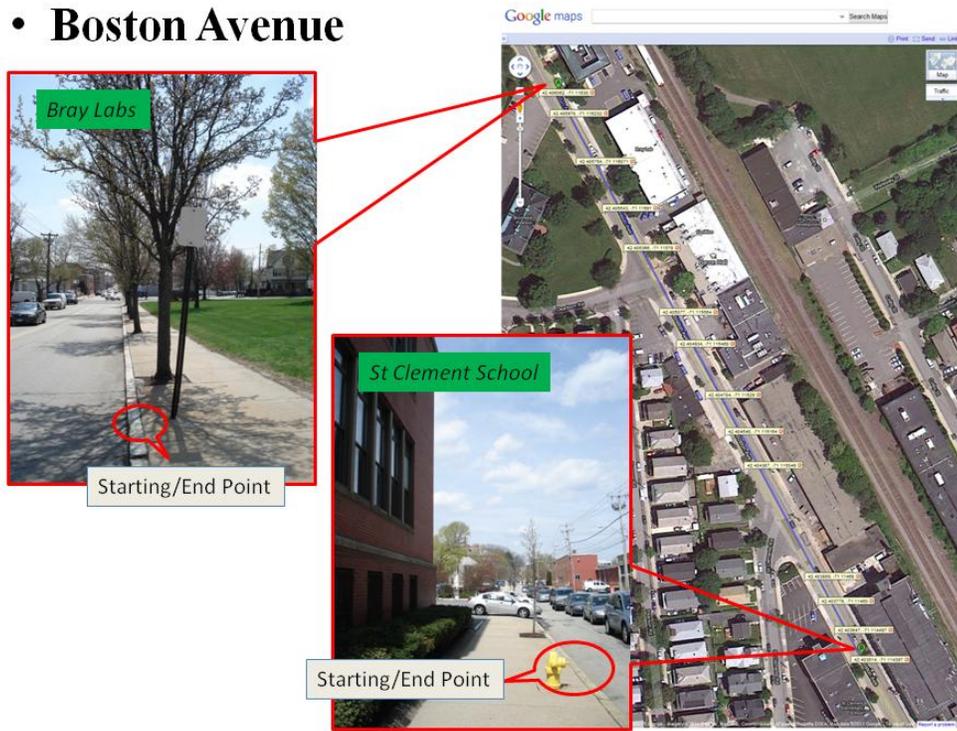


Figure 34. Experiment Route on Boston Avenue

Since the experiment is run on road and often with traffic, it is better that the experiment is done during night time with lower traffic. Lower traffic can reduce some multipath errors as well, because the passing-by vehicles can reflect a significant portion of satellite signals and thus introduce additional multipath error. Besides the relative lower traffic, atmosphere errors like ionosphere and troposphere are at valley in the evening [29]. Table X – Table XV lists the date and time of the datasets taken in experiments. The length of dataset in second indicates the size of each dataset. Since the GPS and its receiver work in frequency of 1 Hz, there will be one set of measurements in each second, corresponding to one data point or one position result for a user.

Table X. Datasets from College Avenue, Direction: Southwest to Northeast

Date	Start Time	End Time	Length (sec)
04-07-2011	22:51:48	22:53:54	126
04-17-2011	21:49:57	21:52:14	137
04-17-2011	21:55:31	21:58:04	153
04-18-2011	19:44:12	19:46:25	133
04-26-2011	22:49:26	22:51:30	124
04-26-2011	22:53:51	22:55:47	116

Table XI. Datasets from College Avenue, Direction: Northeast to Southwest

Date	Start Time	End Time	Length (sec)
04-17-2011	21:52:26	21:55:06	160
04-25-2011	21:13:52	21:15:49	117
04-26-2011	22:51:44	22:53:38	114
04-26-2011	22:56:01	22:57:55	114
04-30-2011	22:17:28	22:19:32	124
04-30-2011	22:22:12	22:24:14	122

Table XII. Datasets from Cousens Parking Log, Direction: Southwest to Northeast

Date	Start Time	End Time	Length (sec)
04-07-2011	23:01:40	23:02:51	71
04-18-2011	19:50:30	19:51:42	72
04-22-2011	20:24:27	20:25:42	75
04-22-2011	20:27:35	20:28:52	77
04-26-2011	20:26:12	20:27:22	70
04-26-2011	22:59:07	23:00:13	66
04-26-2011	23:01:46	23:02:50	64
04-30-2011	22:28:05	22:29:15	70

Table XIII. Datasets from Cousens Parking Log, Direction: Northeast to Southwest

Date	Start Time	End Time	Length (sec)
04-17-2011	22:04:03	22:05:27	72
04-18-2011	19:51:53	19:53:04	75
04-22-2011	20:25:53	20:27:05	77
04-26-2011	20:27:30	20:28:40	66
04-26-2011	23:00:21	23:01:31	64
04-26-2011	23:02:56	23:04:01	64
04-30-2011	22:29:25	22:30:35	70

Table XIV. Datasets from Boston Avenue, Direction: Northwest to Southeast

Date	Start Time	End Time	Length (sec)
04-17-2011	22:13:56	22:18:43	277
04-26-2011	23:07:52	23:12:05	253

Table XV. Datasets from Boston Avenue, Direction: Southeast to Northwest

Date	Start Time	End Time	Length (sec)
04-16-2011	14:09:26	14:13:56	270
04-26-2011	23:12:15	23:16:30	255

It should be stated that, the datasets listed above are only the data that have good quality for experiment analysis. More data were taken at these three places. However, due to factors which are unknown for current stage, those data cannot be included for further analysis.

3.3.3. Experiment Results

This subsection demonstrates the results of the experiments. Details of how to implement the experimental data to obtain these results is discussed in Appendix A, which also gives a sample of collected dataset. However, one thing should be stated clearly for the data processing. As mentioned in the end of Chapter 2, the decentralized collaborative approach will be used to process the experimental data. In the simulation section of Chapter 2, an important assumption for applying centralized method is that signal is transmitted instantaneously. Based on this assumption, if ignoring the computation time, there is no problem with using shared pseudorange measurements to solve for users' positions. In practice, however, since the satellites are always moving, pseudorange measurements of last second may introduce unpleasant big error into position result of this second. Especially for this experiment with only one set of equipments, satellites may move hundreds of meters during the time the receiver moves from point A to point B (Figure 26). Pseudorange measurements from point A can absolutely not be used for solving the position at point B.

On the other hand, the decentralized method does not rely on the instantaneity assumption. So long as the networked users are within a local area (e.g. 10 km radius), the correction created by the decentralized method can be applied by any user to solve its position, even there are a few minutes discrepancy; because the satellite-specific errors change slowly in time [29]. Corrections for compensating these errors can be useful for even a few hours. This is also one of the characteristics of GBAS.

To be specific, all the experimental data are processed as below.

For first data point, group all the pseudorange measurements $\rho^{(k)}$ and lateral distance measurement d , which is **zero**, into the measurement vector (32). The estimates of pseudorange and lateral distance can be computed from Equation (5) and (25), respectively. Subtract the

estimated pseudoranges and lateral distance from the measurement vector (32), Equation (33) is obtained. Solve Equation (33) for the estimates of the user position and clock bias as well as the lateral component of satellite-specific error (\hat{c}_t). The geometric matrix and the state vector are (34) and (31), respectively.

For the subsequent data points, assume there is no lateral distance measurement. However, there is shared correcting term \hat{c}_t . Apply this correcting term to Equation (42), and then solve for the position by using the decentralized approach, as Equation (43) to (45) shows.

Store the position results computed from all the data points. Since the computation is based on ECEF coordinates, computed data are transformed to “road coordinates” in order to demonstrate the errors in both along-road direction and transverse road direction.

Below is a result of College Avenue.

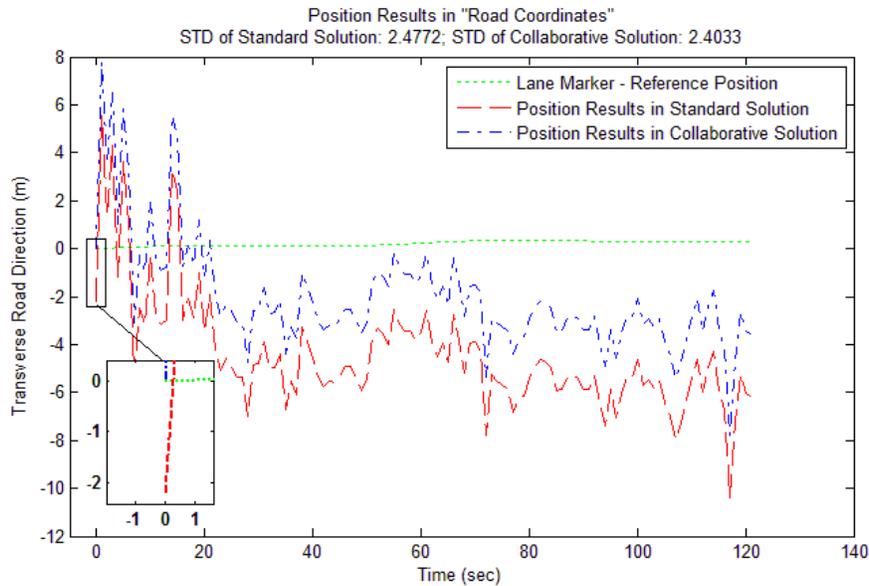


Figure 35. Position Results Comparison (from One Dataset Taken on College Avenue)

The above figure shows the position estimate results of dataset taken on 04-07-2011 in Table X. The position results are projected onto along-road, transverse road, and normal to road plane directions. The figure above shows only the results in the transverse road direction. Results in the

along-road direction are not improved by this method while results in the direction that normal to road plane are not considered in the current stage of study.

From Figure 35 of the transverse road direction, we can see the improvement of position result through the magnifying window. The red dash line is the position result calculated by the standard solution, it is seen to be about 2 m off from the reference, in transverse road direction. After applying the collaborative solution, the estimate position is “pushed” to be almost at the reference point, as the blue dash-dot line shows. Because the new state \hat{c}_t is only estimated at the first point, all the subsequent points will share this \hat{c}_t to correct their positions. As a result, they are all “pushed” a same distance toward the same direction, like that happens at the first point. This is seen on the graph that, the whole blue dash-dot line is “pushed” up for about 2 m. It is also seen that the blue dash-dot line is closer to the reference line than the red dash line does.

To assess the instantaneous accuracy of the decentralized method, absolute error was computed statistically for this method compared to that of a standard GPS solution. The error is computed as the Equation (47).

$$error = \hat{\mathbf{x}}_{road} - \mathbf{x}_{LM,road} \quad (47)$$

In the above equation, both the position estimate of user $\hat{\mathbf{x}}_{road}$ and the lane marker reference position $\mathbf{x}_{LM,road}$ are projected into “road” coordinates. To be specific, they are the direction of “along-road”, “perpendicular to road”, and “normal to road plane”, respectively.

In Section 2.3, multiple trails were generated for a fixed place and fixed time. Likewise, a “fixed” place can be picked from the datasets. Position estimates for that “fixed” place can be found from different datasets. By combining these estimates, a standard deviation analysis can thus be performed to provide a comparison with Section 2.3. For example in Table X, there are 6 datasets taken on different date or at different time. However, all the data were collected in the same moving direction (e.g. moving from southwest to northeast). Solve for the positions through all

these datasets, there will be 6 sets of position estimates. Then a “fixed” place (e.g. the middle point) can be picked from each set of these position estimates. Collect all the position estimates of the middle point, there are totally 6 estimates for this point (shown in Figure 36 with 4 data points).

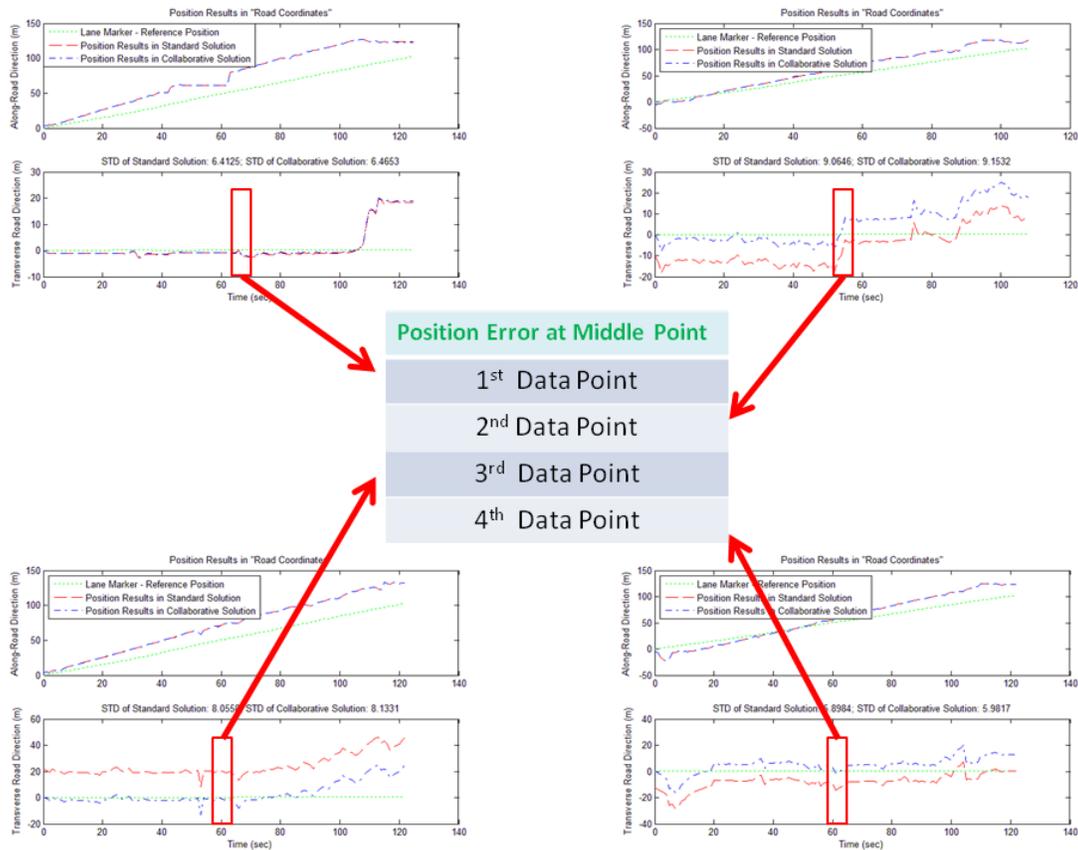


Figure 36. Data for Standard Deviation Analysis

As shown in Figure 4, accuracy in along-road direction would not be improved by this collaborative approach. Thus it is trivia to compare the results in this direction. Below are the standard deviation of estimate errors computed from all the datasets in Table X – Table XV. These errors are all in lateral direction.

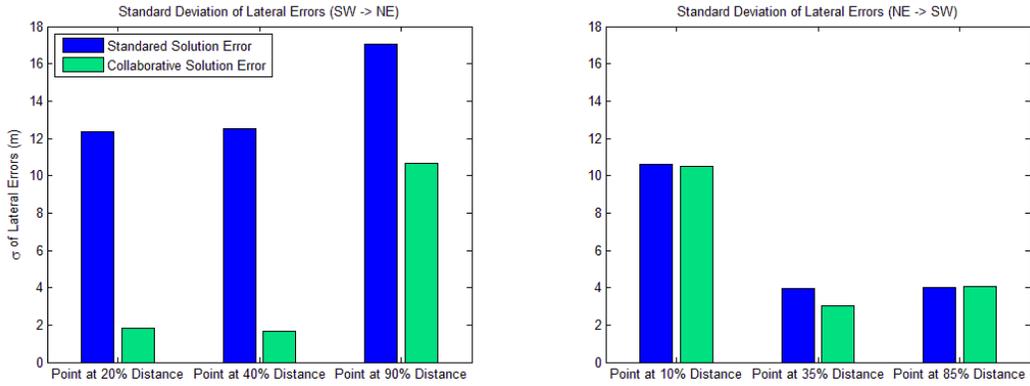


Figure 37. Standard Deviation of Lateral Errors (College Avenue)
 Left: Moving from Southwest to Northeast (Calculated from Table X, 6 Trials)
 Right: Moving from Northeast to Southwest (Calculated from Table XI, 6 Trials)

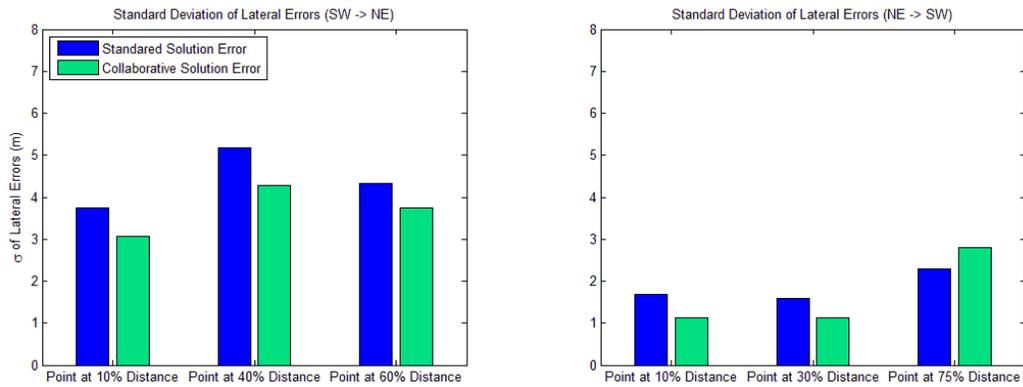


Figure 38. " Standard Deviation of Lateral Errors (Cousens Parking Lot)
 Left: Moving from Southwest to Northeast (Calculated from Table XII, 8 Trials)
 Right: Moving from Northeast to Southwest (Calculated from Table XIII, 7 Trials)

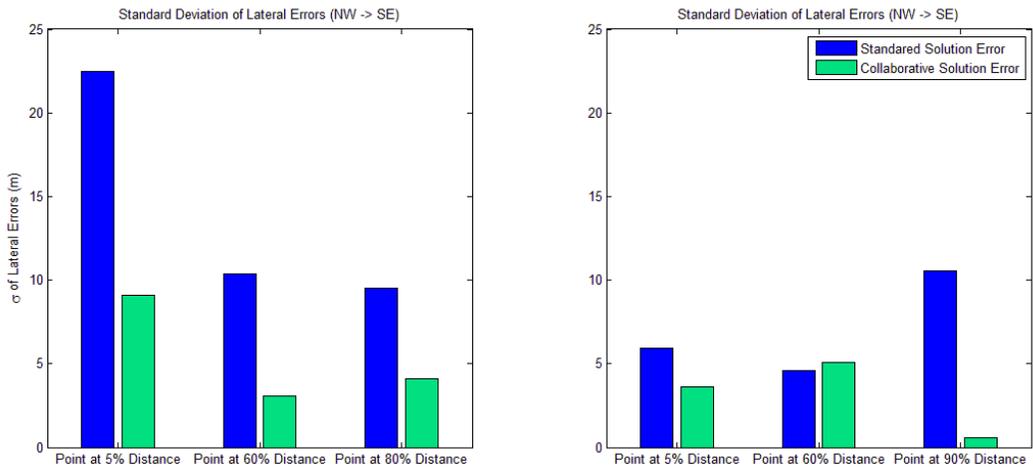


Figure 39. Standard Deviation of Lateral Errors (Boston Avenue)
 Left: Moving from Northwest to Southeast (Calculated from Table XIV, 2 Trials)
 Right: Moving from Southeast to Northwest (Calculated from Table XV, 2 Trials)

Figure 37 – Figure 39 shows the standard deviation for each group of datasets shown in Table X – Table XV. The y-axis of each graph is the value of 1-sigma error. The x-axis indicates the points that are analyzed. In the left chart of Figure 37, “Point at 20% Distance” means a point that is about 20% of the complete dataset’s length away from the first point. Take the first dataset in Table X for example. There are 126 data points with this dataset. So a “point at 20% distance” is the 25th point in this set. Rounding is used when determining the index of data point.

3.3.4. Results Discussion and Error Analysis

1) Results of College Avenue

The left chart in Figure 37 demonstrates a pattern very similar to the middle part of Figure 4. For the first two interested points in this chart, 1-sigma error drops from more than 10 meters to less than 2 meters, from the standard solution to the collaborative solution. It shows that the lateral correction worked very well for these two points. At the point of 90% distance, the error reduction is not as big as of the two previous points. This is because that as the user getting farther away from the correction source point, the effect of the correction is lower for this user. The correction created at the source point does not only include the correcting information, it also includes the unmodeled thermal and multipath error associated with the source point. Thus the correction is able to reduce the common-mode errors in lateral direction; it may also bring in those unexpected errors at the 90% distance point. It can be seen that for the results at 90% distance, the error of standard solution goes higher than those of the previous two. This can be a result of the multipath error introduced by the nearby tall building – Cousens Gym. Again, the correction from the source point cannot eliminate this multipath error. Thus the error of collaborative solution goes higher than previous two.

The bar chart on the right of Figure 37 shows the results in reverse direction on College Avenue. In this case, the correction is created at the place in front of Cousens Gym. There is not much

improvement seen in this graph. This can be the result of a small correction \hat{c}_t that is created. As seen in Figure 35, if the estimate of first point in standard solution happens to be very close to the reference, then the estimated correction \hat{c}_t will be very small. As a result, the estimates of subsequent points will not change much from standard solution to collaborative solution. Thus we can see the two results of both solutions are very close to each other.

2) Results of Cousens Parking Lot

Figure 38 demonstrates the results of data taken at Cousens Parking Lot. The experimental place is close to a wooden wall (within 3 m). The wall stands along the reference lane marker (i.e. experimental moving track). Thus there may be constant multipath error at this place. The results in Figure 38 correspond with this hypothesis – position estimates do not improve much from standard solution to collaborative solution since multipath error dominates the error source.

The point at 75% distance in the right chart of Figure 38 has a worse result in collaborative solution. This is due to the negative effect of correction c_t . An example is in Figure 40. Although the correction can improve the estimates for the first half of the dataset, it pushes the estimates further away for the second half. In this case, estimated \hat{c}_t becomes an error for the position estimate.

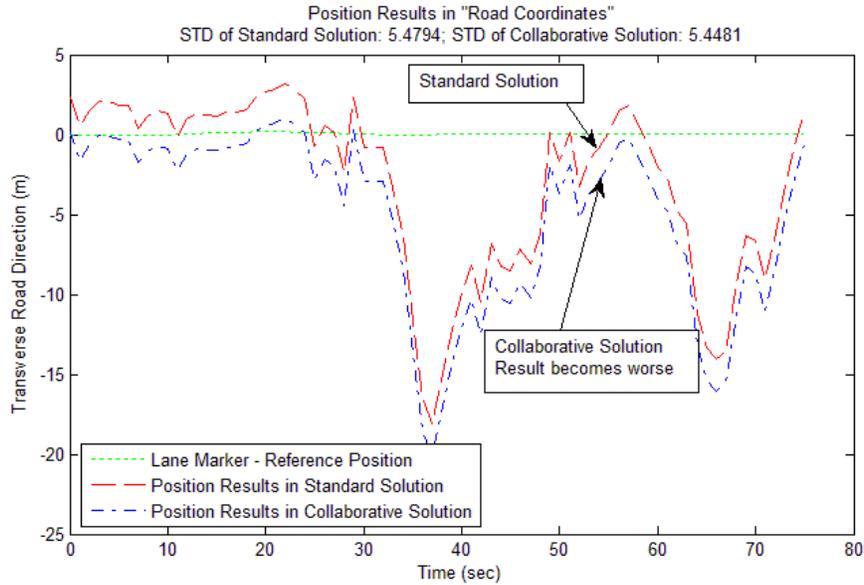


Figure 40. Position Error Introduced by Lateral Correction Estimate \hat{c}_t

3) Results of Boston Avenue

Results shown in Figure 39 are in effect not statistically significant, since there are only two data points for each group of the results (Table XIV and Table XV). Although there seems to have been improvement in the collaborative solution, it is uncertain what the result will be if the sample size becomes larger. As stated in Section 3.3.2, more data were taken on Boston Avenue. However, due to unknown factors, those data cannot be used for experiment analysis.

4) Error Analysis

Since the reference points are selected through Google Maps satellite images, there is error associate with image resolution. The highest resolution of Google Maps satellite image is about 5 cm/pixel. Also the cursor has pointing error on the computer screen. Taking account of all these factors, the total error from Google Maps is between 5 and 10 cm.

Other experimental error appears during the experiment. For example, the post will swing while it is hold by a walking person. As shown in Figure 31, the length from the holding point of the post to the bottom is about 125 cm. During experiment, the bottom of the post swings about ± 5 cm

away from medium at maximum. According to geometry, the top (with GPS receiver) swings about ± 3 cm at maximum. Thus the error introduced by the swinging post swinging is about ± 3 cm. This is in normal condition, excluding the occasions like wing turbulent or kicking by foot.

Include all the factors discussed above, there is at total about 8-13 cm of error associate with the ground reference. Compared to the standard deviation of the results listed in Figure 37 – Figure 39, the reference provides satisfied accuracy to validate the experiment results.

CHAPTER 4: CONCLUSION

4.1. THESIS CONTRIBUTIONS

This thesis investigates a novel collaborative navigation approach for solving positions of networked GNSS users. To date, little attention of collaborative navigation is on common-mode error removal. In filling up this gap, this study employs camera-based lane-boundary sensor and V2V network to create differential correction and remove common-mode error. To summarize, the contributions of this study include:

- 1) Development of two versions of collaborative navigation algorithm – centralized and decentralized approaches. Based on the discovery of the rank deficiency problem with networking the single difference GNSS solution, this study proposes an approach to solve multi-user's absolute position by introducing an additional camera-based lane boundary sensor. By using this lane-boundary sensor to measure user position relative to the absolute position of a lane boundary, this study develops an algorithm to estimate and remove the common-mode GNSS error in the direction perpendicular to the lane boundary. With the assistance of V2V network, multiple users can share the correction information and thus benefit from the lane-boundary sensor. In addition to developing a novel algorithm to resolve multi-user position estimation, the study expands the algorithm to two versions in order to cater to different occasions. The centralized version collects all available measurements including pseudoranges and lane boundary measurements to estimate all users' positions in one least-squares problem. It takes account of all users' geometry thus it is highly accurate. However, as the size of user group grows, this method becomes computationally inefficient. The other decentralized version addresses the inefficient problem while keeps a high computing accuracy. The

only drawback of the decentralized version is that it does not take account of the users' geometry. Thus it will sometimes introduce unexpected error to position estimate.

- 2) Design and implementation of physical experiment to test novel collaborative navigation algorithm. This thesis employs physical system to demonstrate the capabilities of the novel approach. In the absence of multi-receiver, the study designs an ad hoc experiment, which requires only one set of devices. With the special setup, the research collects abundant experimental data in three different places. In the requiring of ground reference positions, the study investigates a method of surveying ground positions on Google Maps. By comparing the experimental results to the ground reference, the study demonstrates the advantages of the new collaborative navigation approach. Besides relating the theoretical work to practical systems, the study thoroughly investigates a specific GPS receiver. This provides important first-hand experience of using the receiver. Thus following researches can conveniently absorb this experience and make good use for their own purpose.

4.2. SIGNIFICANCE AND FUTURE WORK

The work presented in this research will influence automotive navigation by fusing GNSS sensor with camera-based lane-boundary sensor and vehicle-to-vehicle network. It provides a novel and achievable approach to improve position estimate accuracy for on-road vehicles. This will enhance the development of both safety-critical and automatic-related automotive applications, such as lane departure warning systems, automatic lane keeping system, and automatic driving systems.

Future work for this study is classified into two branches: camera-based lane-boundary detecting, and vehicle-to-vehicle networking.

For camera-based lane-boundary detecting, robust detecting algorithm is required for the collaborative navigation system. Also, algorithm for transforming the lane-boundary information into lateral distance measurement is necessary. In addition, a platform that fuses the lane-boundary measurement and the GPS measurement is essential for the system being achieved for practical applications.

For vehicle-to-vehicle networking, communicating architecture and protocol need to be defined in order to effective communication. Many current researches are related to this aspect. It is hoped that, a vehicle-to-vehicle communication module would be constructed. Thus the collaborative navigation system can simply integrate this module to achieve the goal of sharing measurements and corrections.

APPENDIX A: IMPLEMENTATION OF DATASETS LOGGED BY SIRFDEMO

Data logged by SiRFDemo is saved as a text file. It cannot be read directly by MATLAB program, due to the `fgetl()` function rather than the `importdata()` is used to read the file. `fgetl()` function considers space as separator in a line of text file, while the logged data file uses comma (,) and colon (:) as separator, as shown in Fig Append - 1. Thus before open a read file in MATLAB, it is necessary to substitute all the separators with MATLAB readable separator – space. Moreover, the original data file has two blank lines – one in the beginning and the other in the end of file. The MATLAB code that is written to read the data file cannot ignore a blank line by itself. Thus another two steps are necessary to handle this problem. As shown in Fig Append - 1, for the first blank line, it is recommended to insert a string of characters: `Log file begin collecting data MM/DD/YY HH MM SS`. The “MM/DD/YY HH MM SS” is the file logging date and time, which can be determined from the first line of the logged file. The reason for this step is that, if a user logs the file using the “Record Log File only During a Specified Time Interval” Option shown in Figure 15, this line of character string is automatically added; when the file reading program was written, this option was always selected and therefore there was always a line of that information in the logged files at that time. For the second blank line, one should simply delete it, as Figure shows.

After the above implementation, we now have a ready-to-read file for MATLAB, as shown in Fig Append - 2. In the computation for user position estimate, the following data are used:

1) SV ID

These are the ID number associated with each satellite. The SV ID numbers are used for matching the pseudoranges from Message ID 28 to the satellite positions from Message ID 30.

2) Pseudorange Measurements

The pseudorange measurements are expressed in the unit of meter. These are the measurements described by Equation (2). In order to have a disturbed pseudorange vector (8), estimated pseudoranges are subtracted from these measurements, as Equation (6) shows.

3) SV Positions in ECEF Frame

These are the satellite's positions, which are also described in unit of meter, included in the satellite signals. As indicated in Equation (5), satellite position is required in order to have an estimate pseudorange. These positions information are used for this purpose. After having an estimate pseudorange, subtract (5) from (4), thus results in the disturbed pseudorange model (6). Group all the disturbed models from each satellite, there forms Equation (8). Moreover, the satellite's positions are also important to obtain the unit pointing vector (7).

The information from Message ID 2 – Receiver Computed User Position in ECEF Frame and the information from Message ID 7 – Receiver Computed Receiver Clock Bias are the results computed by the receiver itself. The position estimate from the receiver is in the unit of meter. It has accuracy of meter-level. The estimated receiver clock bias is estimated in nanosecond level. These data are also logged as a rough reference.

SiRFDemo Version 3.83 log file opened 04/07/11 22:51:48
SiRFDemo Build Date: Feb 16, 2005

Insert: Log file begin collecting data MM/DD/YY HH MM SS

Tx PC Time=1302231108.060
Tx: 0x80A2000284000084E0B3
Tx PC Time=1302231108.075

Substitute all colons(:) with one space

Substitute all commas (,) with one space

```
28,0,100,19,4.423230000000e+000,0.000000000000e+000,30000,191,45,45,45,45,45,45,45,44,44,44,1000,500,-1,0,0
28,1,100,24,4.423230000000e+000,0.000000000000e+000,30000,191,36,36,36,36,36,36,36,35,35,35,1000,500,-2,0,0
28,2,100,8,4.42323076724431e+005,4.5270173740e+007,1.7543972655e+004,0.000000000000e+000,30000,191,44,44,44,44,44,44,43,43,43,43,1000,500,0,0,0
28,3,100,6,4.42323076724431e+005,4.6121460758e+007,1.8700869141e+004,0.000000000000e+000,30000,191,41,41,41,41,41,41,41,40,40,40,1000,500,-1,0,0
28,4,100,10,4.423230000000e+000,0.000000000000e+000,30000,191,46,46,46,46,46,46,46,45,45,45,1000,500,-1,0,0
28,5,100,11,4.423230000000e+000,0.000000000000e+000,30000,191,39,39,39,39,38,38,38,38,38,38,1000,500,0,0,0
28,6,100,7,4.423230000000e+000,0.000000000000e+000,30000,191,46,46,46,46,46,46,46,45,45,45,1000,500,-1,0,0
28,7,100,16,4.42323076724431e+005,4.7358654230e+007,1.8654681641e+004,0.000000000000e+000,740,173,0,31,22,24,24,24,33,32,31,30,0,0,-297,50,0
28,8,100,3,4.42323076724431e+005,4.4369861888e+007,1.8657320313e+004,0.000000000000e+000,30000,191,44,44,44,43,43,43,43,43,43,1000,500,-2,0,0
28,9,100,28,4.42323076724431e+005,4.7693312992e+007,1.7646337891e+004,0.000000000000e+000,4540,173,24,24,24,24,23,23,23,1000,500,-2,50,0
28,10,100,13,4.42323076724431e+005,4.6209811682e+007,1.8754896484e+004,0.000000000000e+000,30000,191,41,41,41,41,41,41,40,40,40,1000,500,-2,0,0
30,19,4.423230000000e+005,1.0379226538e+007,-1.5034164322e+007,1.9156367130e+007,6.9328235176e+002,2.3493348764e+003,1.5022271321e+003,-1.2820612632e-004,
-4.9812424151e-012,1,0,0.0000000000e+000,0.0000000000e+000,1.5371084213e+000
30,24,4.423230000000e+005,1.3495267247e+007,-2.1257003632e+007,-7.9423713035e+006,9.5630555790e+002,-4.9022541888e+002,2.9432198718e+003,3.7983970557e-004,
1.0711371026e-012,1,0.0000000000e+000,0.0000000000e+000,3.6090416908e+000
30,8,4.423230000000e+005,-1.0207729948e+007,-1.0232466716e+007,2.2307301328e+007,2.0795139590e+003,-1.9010230035e+003,2.6768603519e+001,7.6975284153e-006,
2.6841633842e-013,1,0.0000000000e+000,0.0000000000e+000,2.3214020729e+000
30,6,4.423230000000e+005,1.6677400329e+007,6.3856867868e+005,2.0860291694e+007,8.5279280006e+002,2.4760748290e+003,-7.4176850794e+002,-2.9138072854e-004,
-1.1139510842e-010,1,0.0000000000e+000,0.0000000000e+000,2.6906158924e+000
30,11,4.423230000000e+005,1.1431824708e+007,-2.3742870150e+007,-1.3082573731e+006,5.3604033970e+002,1.3343784349e+002,3.0218482111e+003,-1.4086475046e-004,
-5.2634207756e-012,1,0.0000000000e+000,0.0000000000e+000,2.5172908306e+000
30,7,4.423230000000e+005,-1.1457502788e+006,-1.8074505700e+007,1.9357973695e+007,1.7644703649e+003,-1.7192034942e+003,-1.5208768714e+003,7.5993822076e-006,
3.8128322114e-013,1,0.0000000000e+000,0.0000000000e+000,1.5807491541e+000
30,16,4.423230000000e+005,2.4871631207e+007,-4.7927600503e+005,9.7591707174e+006,1.1391107657e+003,4.8350245682e+002,-2.8585829623e+003,-1.6353173621e-004,
-7.1696273050e-013,1,0.0000000000e+000,0.0000000000e+000,3.5962467194e+000
30,3,4.423230000000e+005,1.5268045798e+007,-4.2845731970e+005,2.0914968858e+007,1.1388449539e+003,2.5172404445e+003,-2.7779158159e+002,6.9903412360e-004,
1.1779106336e-012,1,0.0000000000e+000,0.0000000000e+000,2.1319568157e+000
30,28,4.423230000000e+005,-2.0411467581e+007,-1.3341861851e+007,1.1351369203e+007,1.3521846888e+003,1.7473855443e+002,2.7385533994e+003,3.0025767239e-005,
6.6131791883e-012,1,0.0000000000e+000,0.0000000000e+000,4.6640086174e+000
30,13,4.423230000000e+005,-4.8194749976e+006,-2.6038187079e+007,-1.7794465439e+006,2.3397990268e+002,1.5929578594e+002,-3.2319162729e+003,2.7598212309e-004,
-1.7294492661e-012,1,0.0000000000e+000,0.0000000000e+000,3.2444505692e+000
2,1526519,-4462729,4279123,0.000,0.000,0.125,132,0.8,2,606,44232300,10,19,24,8,6,11,7,16,3,28,13,0,0
Week:1630 TOW:44232300 EstGPSTime:442322999 ns SUCnt:10 Clock Drift:95250 Hz Clock Bias:76724453 ns
```

Log file closed 04/07/11 22:53:54

Delete Blank Row

Fig Append - 1. SiRFDemon Logged Data File - Initial Raw Data

SIRFDemo Version 3.83 log file opened 04/07/11 22 51 48

SIRFDemo Build Date Feb 16 2005

Log file begin collecting data 04/07/11 22 51 48

Inserted Line

Tx PC Time=1302231108.060

Tx 0xA0A2000284000084B0B3

Tx PC Time=1302231108.075

Tx 0xA0A2000298000098B0B3

```

28 0 100 19 4.42323076724431e+005 4.3324265116e+007 1.78005441406e+004 0.0000000000e+000 30000 191 45 45 45 45 45 45 44 44 44 1000 500 -1 0 0
28 1 100 24 4.42323076724431e+005 4.6859738778e+007 1.7445062500e+004 0.0000000000e+000 30000 191 36 36 36 36 36 36 35 35 35 1000 500 -2 0 0
28 2 100 8 4.42323076724431e+005 4.5270173740e+007 1.7543972656e+004 0.0000000000e+000 30000 191 44 44 44 44 44 44 43 43 43 1000 500 0 0 0
28 3 100 6 4.42323076724431e+005 4.6121460758e+007 1.8700869101e+004 0.0000000000e+000 30000 191 41 41 41 41 41 41 40 40 40 1000 500 -1 0 0
28 4 100 23 4.42323076724431e+005 4.8210378708e+007 1.8805039100e+004 0.0000000000e+000 30000 191 15 15 15 15 15 15 0 0 -1 50 0
28 5 100 11 4.42323076724431e+005 4.5428012703e+007 1.7493862500e+004 0.0000000000e+000 30000 191 38 38 38 38 38 38 1000 500 0 0 0
28 6 100 7 4.42323076724431e+005 4.3488016474e+007 1.7918191000e+004 0.0000000000e+000 30000 191 45 45 45 45 45 45 1000 500 -1 0 0
28 7 100 16 4.42323076724431e+005 4.7358654230e+007 1.8654681641e+004 0.0000000000e+000 740 173 0 31 22 24 24 24 33 32 31 30 0 0 -297 50 0
28 8 100 3 4.42323076724431e+005 4.4369861888e+007 1.8657320313e+004 0.0000000000e+000 30000 191 44 44 43 43 43 43 43 43 1000 500 -2 0 0
28 9 100 28 4.42323076724431e+005 4.7693312992e+007 1.7645337891e+004 0.0000000000e+000 4540 173 24 24 24 24 24 24 23 23 23 1000 500 -2 50 0
28 10 100 13 4.42323076724431e+005 4.6209811682e+007 1.8754895484e+004 0.0000000000e+000 30000 191 41 41 41 41 41 41 40 40 40 1000 500 -2 0 0
30 19 4.4232300000e+005 1.0379226538e+007 -1.5034164322e+007 1.9156367130e+007 5.9328235176e+002 2.3493348764e+003 1.5022271321e+003 -1.2820612632e-004
012 1 0 0000000000e+000 0 0000000000e+000 0 1.5371084213e+000
30 24 4.4232300000e+005 1.3495267247e+007 -2.1257003632e+007 -7.9423713035e+006 9.5690555790e+002 -4.9022541888e+002 2.9432198718e+003 3.7983970557e-004
012 1 0 0000000000e+000 0 0000000000e+000 3 6090416908e+000
30 8 4.4232300000e+005 -1.0207729948e+007 -1.0232466716e+007 2.2307301328e+007 2.0795139590e+003 -1.9010230035e+003 2.6768603519e+001 7.6975284153e-006
2.6841633842e-013 1 0 0000000000e+000 0 0000000000e+000 2 3214020729e+000
30 6 4.4232300000e+005 1.6677400329e+007 6.3856867868e+005 2.0860291694e+007 8.5279280006e+002 2.4760748290e+003 -7.4176850794e+002 -2.9138072854e-004
-1.1139510842e-010 1 0 0000000000e+000 0 0000000000e+000 2 6906158924e+000
30 11 4.4232300000e+005 1.1431824708e+007 -2.3742870150e+007 -1.3082573731e+006 5.3604033100e+002 -1.4086475046e-004
-5.2634207756e-012 1 0 0000000000e+000 0 0000000000e+000 2 5172908306e+000
30 7 4.4232300000e+005 -1.1457502788e+006 -1.8074505700e+007 1.9357973695e+007 1.7644703100e+002 -7.5993822076e-006
3.8128322114e-013 1 0 0000000000e+000 0 0000000000e+000 1 5807491541e+000
30 16 4.4232300000e+005 2.4871631207e+007 -4.7927600503e+005 9.7591707174e+006 1.1391107657e+003 4.8350245682e+002 -2.8585829623e+003 -1.6353173621e-004
-7.1696273050e-013 1 0 0000000000e+000 0 0000000000e+000 3 5962467194e+000
30 3 4.4232300000e+005 1.5268045798e+007 -4.2845731970e+006 2.0914968581e+007 1.1388449539e+003 2.5172404445e+003 -2.7779158159e+002 6.9903412360e-004
1.1779106336e-012 1 0 0000000000e+000 0 0000000000e+000 2 1319568157e+000
30 28 4.4232300000e+005 -2.0411467581e+007 -1.3341861851e+007 1.1351369203e+007 1.3521846888e+003 1.7473855443e+002 2.7385533994e+003 3.0025767239e-005
6.6131791883e-012 1 0 0000000000e+000 0 0000000000e+000 4 6640086179e+000
30 13 4.4232300000e+005 -4.8194749976e+006 -2.6038187079e+007 -1.7794465439e+006 2.0397990268e+002 1.5929578594e+002 -3.2319162729e+003 2.7598212309e-004
-1.7294492661e-012 1 0 0000000000e+000 0 0000000000e+000 3 2444505692e+000
2 1526519 -4462729 4279123 0.000 0.000 0.125 132 0.8 2 606 44232300 10 19 24 8 6 11 7 16 3 28 13 0 0
Week 1630 TOW 44232300 EtcGPSTime 442322999 ms SvcCnt 10 Clock Drift 95250 Hz Clock Bias 76724453 ns
Log file closed 04/07/11 22 53 54

```

SV* ID

Pseudorange Measurements

SV* ID

SV Positions in ECEF** Frame

Receiver Computed User Position - in ECEF Frame

Receiver Computed Receiver Clock Bias

Fig Append - 2. SIRFDemo Logged Data File - MATLAB Readable Data
*SV: Space Vehicle (i.e. Satellite); **ECEF: Earth-Center Earth-Fixed

APPENDIX B: MATLAB PROGRAMS

B.1. STANDARD GNSS SOLUTION FOR SINGLE USER

```
function [roadUsr roadLane time] = solveSN(filename,place,dir,startPt,endPt)
% This function compute user position by using standard GNSS solution.
% Output is projected into road-coordinate frame.
%% part 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Initialize Satellite Data and Parameters %%%%%%%%%%
sirfdata = sirfRead(filename);
step = length(sirfdata.svSta);
if nargin == 4
    endPt = step;
elseif nargin < 4
    startPt = 1;
    endPt = step;
end
sirfdata = datatruncate(sirfdata,startPt,endPt);

Tsamp = 1; %Set sampling time
step = length(sirfdata.svSta);
time = 0:Tsamp:(Tsamp*(step-1));
c = 2.99792458e8; %Speed of light
posMeas = zeros(step,3); %Position matrix for measured data
clkBias = zeros(step,1); %Clock bias matrix for measured data
posEst = zeros(step,4); %Position matrix for estimated data
ids = zeros(step,12); %Satellite ID matrix
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% part 2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Manipulation of Road Model and Lane Marker Model %%%%%%%%%%
switch place
    case 'ColAve'
        ColAveLnMarker %Read data file, output is llaLn (n*3)
        ColAveLnBoundary %Read data file, output is llaRd (n*3)
    case 'CousPark'
        CousParkLnMarker
        CousParkLnBoundary
    case 'BosAve'
        BosAveLnMarker
        BosAveLnBoundary
end

if dir == 0 %Determine if it's reverse direction: 0 - Reverse
    llaLn = flipud(llaLn);
    llaRd = flipud(llaRd);
end

% Convert road and lane marker model from LLA to ENU coordinates
l = size(llaRd,1);
for i = 1:l
    temp = wgs11a2enu(llaRd(i,1),llaRd(i,2),llaRd(i,3),...
        llaRd(1,1),llaRd(1,2),llaRd(1,3));
    enuRd(i,1:3) = temp';
end
l = size(llaLn,1);
for i = 1:l
    temp = wgs11a2enu(llaLn(i,1),llaLn(i,2),llaLn(i,3),...
        llaLn(1,1),llaLn(1,2),llaLn(1,3));
    enuLn(i,1:3) = temp';
end
```

```

% Interpolate values for lane marker model
tSize = (l-1)/(step-1);
enuLnInterp = interp1(1:l,enuLn,1:tSize:1,'spline');

% Create unit vector of road model
% For creating ground ference in ENU/road coordinates
% No collaborative navigation solution is used, thus no vector in XYZ-frame
%Along-road direction vector
vecRdENU = [enuRd(end,1) enuRd(end,2) 0] - [enuRd(1,1) enuRd(1,2) 0];
%Along-road direction unit vector
u_rRdENU = vecRdENU/sqrt(sum(vecRdENU.^2));
%Transverse direction vector
u_tRdENU = [1 -u_rRdENU(1)/u_rRdENU(2) 0];
%Transverse direction unit vector
u_tRdENU = u_tRdENU/sqrt(sum(u_tRdENU.^2));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% part 3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Standard Navigating %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Compute results in standard procedure
for t = 1:step
    sat = pullData(sirfdata,t); %Pull satellite pos & pseudorange info
    %Store sat ID of sats being tracked
    ids(t,:) = [sat.ID' zeros(1,12-length(sat.ID))];
    xMeas = sirfdata.rcvPos{t}.x; %User pos calculated by receiver
    yMeas = sirfdata.rcvPos{t}.y;
    zMeas = sirfdata.rcvPos{t}.z;
    %Store receiver calculated positions
    posMeas(t,1:3) = [xMeas yMeas zMeas];
    %Store receiver calculated clock bias
    clkBias(t) = sirfdata.clkSta{t}.clkBias*1e-9*c;
    if t > 1
        posEst(t,:) = posEst(t-1,:); % Use new est value for ini guess
    end
    %Calculate user pos using lateral distance measurement
    posEst(t,:) = calUsrPos(sat);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% part 4
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Convert Position Results into ENU and LLA Coordinates %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
enuUsrEst = zeros(step,3);
%Reference of ENU conversion: First point of lane marker
latRef = llaLn(1,1);
lonRef = llaLn(1,2);
altRef = llaLn(1,3);
for t = 1:step
    %Filter results
    if t > 3
        if abs(posEst(t,1) - posEst(t-1,1)) > 50
            posEst(t,:) = sum(posEst(t-3:t-1,:),1)/3;
        end
    end
    %Convert all the results into ENU coordinates
    %Refer to first point of lane marker
    enuUsrEst(t,:) = wgsxyz2enu(posEst(t,1:3)',latRef,lonRef,altRef)';
end

% Convert into "road" coordinates
roadUsr = rot2road(u_rRdENU,enuUsrEst);
roadLane = rot2road(u_rRdENU,enuLnInterp);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

B.2. COLLABORATIVE NAVIGATION SOLUTION

```

function [roadUsr roadLane time]=solveCN(filename,place,dir,latD,startPt,endPt)
% This function compute user position by using
% collaborative navigation solution.
% Output is projected into road-coordinate frame.
%% part 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Initialize Satellite Data and Parameters %%%%%%%%%%
sirfdata = sirfRead(filename);
step = length(sirfdata.svSta);
if nargin == 5
    endPt = step;
elseif nargin < 5
    startPt = 1;
    endPt = step;
end
sirfdata = datatruncate(sirfdata,startPt,endPt);

Tsamp = 1; %Set sampling time
step = length(sirfdata.svSta);
time = 0:Tsamp:(Tsamp*(step-1));
c = 2.99792458e8; %Speed of light
posMeas = zeros(step,3); %Position matrix for measured data
clkBias = zeros(step,1); %Clock bias matrix for measured data
posEst = zeros(step,5); %Position matrix for estimated data
ids = zeros(step,12); %Satellite ID matrix
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% part 2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Manipulation of Road Model and Lane Marker Model %%%%%%%%%%
switch place
    case 'ColAve'
        ColAveLnMarker %Read data file, output is llaLn (n*3)
        ColAveLnBoundary %Read data file, output is llaRd (n*3)
    case 'CousPark'
        CousParkLnMarker
        CousParkLnBoundary
    case 'BosAve'
        BosAveLnMarker
        BosAveLnBoundary
end

if dir == 0
    llaLn = flipud(llaLn);
    llaRd = flipud(llaRd);
end

% Convert road and lane marker model from LLA to ENU coordinates
l = size(llaRd,1);
for i = 1:l
    temp = wgslla2enu(llaRd(i,1),llaRd(i,2),llaRd(i,3),...
        llaRd(1,1),llaRd(1,2),llaRd(1,3));
    enuRd(i,1:3) = temp';
end
l = size(llaLn,1);
for i = 1:l
    temp = wgslla2enu(llaLn(i,1),llaLn(i,2),llaLn(i,3),...
        llaLn(1,1),llaLn(1,2),llaLn(1,3));
    enuLn(i,1:3) = temp';
end

% Interpolate values for lane marker model
tSize = (l-1)/(step-1);
enuLnInterp = interp1(1:l,enuLn,1:tSize:1,'spline');

```

```

% Create unit vector of road model
%Along-road direction vector
vecRdENU = [enuRd(2,1) enuRd(2,2) 0] - [enuRd(1,1) enuRd(1,2) 0];
%Along-road direction unit vector
u_rRdENU = vecRdENU/sqrt(sum(vecRdENU.^2));
%Transverse direction vector
u_tRdENU = [1 -u_rRdENU(1)/u_rRdENU(2) 0];
%Transverse direction unit vector
u_tRdENU = u_tRdENU/sqrt(sum(u_tRdENU.^2));

% Convert unit vectors into XYZ coordinates
%Along-road direction unit vector
u_tRdXYZ = rotenu2xyz(u_tRdENU', llaRd(1,1), llaRd(1,2))';
%Transverse direction unit vector
u_rRdXYZ = rotenu2xyz(u_rRdENU', llaRd(1,1), llaRd(1,2))';
%Up direction unit vector
u_upRdXYZ = cross(u_tRdXYZ,u_rRdXYZ);

% Set reference road position
refRd = wgslla2xyz(llaRd(1,1),llaRd(1,2),llaRd(1,3))';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% part 3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Collaborative Navigating %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Compute results for first time step
for t = 1:1
    sat = pullData(sirfdata,t); %Pull satellite pos & pseudorange info
    %Store sat ID of sats being tracked
    ids(t,:) = [sat.ID' zeros(1,12-length(sat.ID))];
    xMeas = sirfdata.rcvPos{t}.x; %User pos calculated by receiver
    yMeas = sirfdata.rcvPos{t}.y;
    zMeas = sirfdata.rcvPos{t}.z;
    %Store receiver calculated positions
    posMeas(t,1:3) = [xMeas yMeas zMeas];
    %Store receiver calculated clock bias
    clkBias(t) = sirfdata.clkSta{t}.clkBias*1e-9*c;
    if t>1
        posEst(t,:) = posEst(t-1,:); % Use new est value for ini guess
    end
    %Calculate user pos using lateral distance measurement
    posEst(t,:) = calUsrPos(sat,latD,u_tRdXYZ,refRd);
end

% Share the correction computed at first step to the following time steps
corr_t = posEst(1,end);
for t = 2:step
    sat = pullData(sirfdata,t); %Pull satellite pos & pseudorange info
    %Store sat ID of sats being tracked
    ids(t,:) = [sat.ID' zeros(1,12-length(sat.ID))];
    xMeas = sirfdata.rcvPos{t}.x; %User pos calculated by receiver
    yMeas = sirfdata.rcvPos{t}.y;
    zMeas = sirfdata.rcvPos{t}.z;
    %Store receiver calculated positions
    posMeas(t,1:3) = [xMeas yMeas zMeas];
    %Store receiver calculated clock bias
    clkBias(t) = sirfdata.clkSta{t}.clkBias*1e-9*c;
    if t>1
        posEst(t,:) = posEst(t-1,:); % Use new est value for ini guess
    end
    %Calculate user pos using lateral correction shared by step 1
    posEst(t,1:4) = calUsrPos(sat,corr_t,u_tRdXYZ);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%% part 4
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Convert Position Results into ENU and LLA Coordinates %%%%%%%%%%
enuUsrEst = zeros(step,3);
%Reference of ENU conversion: First point of lane marker
latRef = llaLn(1,1);
lonRef = llaLn(1,2);
altRef = llaLn(1,3);
for t = 1:step
    %Filter results
    if t > 3
        if abs(posEst(t,1) - posEst(t-1,1)) > 50
            posEst(t,:) = sum(posEst(t-3:t-1,:),1)/3;
        end
    end
    %Convert all the results into ENU coordinates
    %Refer to first point of lane marker
    enuUsrEst(t,:) = wgsxyz2enu(posEst(t,1:3)',latRef,lonRef,altRef)';
    if t > 1
        if abs(enuUsrEst(t,1)-enuUsrEst(t-1,1)) > 20
            %enuUsrEst(t,:) = enuUsrEst(t-1,:);
        end
    end
end

% Convert into "road" coordinates
roadUsr = rot2road(u_rRdENU,enuUsrEst);
roadLane = rot2road(u_rRdENU,enuLnInterp);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

B.3. LOGGED FILE READING FUNCTION

```

function sirfdata = sirfRead(fname)
fid = fopen(fname);
t = 1;
chnl = 0;
chnlD = 0;
sv = 0;
logEnd = 0;
while ~feof(fid)
    tline = fgetl(fid);
    switch tline(1)
        case '2'
            if tline(2)=='8'
                %ID 28: Nav Lib meas data
                chnl = chnl + 1;
                scanvec = sscanf(tline,...
                    '%d%d%d%f%f%f%d%d%d%d%d%d%d%d%d%d%d%d%d%d');
                % 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4
                data.meas{t}(chnl,1:4) = ...
                    [scanvec(4) scanvec(5) scanvec(6) scanvec(7)];
                % 4:Sat ID 5:GPS time 6:pr 7:Carrier Fq
                sirfdata.svSta{t}.tx{chnl}.ID = data.meas{t}(chnl,1);
                sirfdata.svSta{t}.tx{chnl}.pr = data.meas{t}(chnl,3);
                sirfdata.svSta{t}.tx{chnl}.cFq = data.meas{t}(chnl,4);
                sirfdata.svSta{t}.tx{chnl}.GPSSoftTime = data.meas{t}(chnl,2);
                sirfdata.svSta{t}.tx{chnl}.x = 0;
                sirfdata.svSta{t}.tx{chnl}.y = 0;
                sirfdata.svSta{t}.tx{chnl}.z = 0;
                sirfdata.svSta{t}.tx{chnl}.clkBias = 0;
                sirfdata.svSta{t}.tx{chnl}.ionDelay = 0;
                sirfdata.svSta{t}.tx{chnl}.DGPS = 0;
                sirfdata.svSta{t}.DGPS{chnl}.ID = 0;
                sirfdata.svSta{t}.DGPS{chnl}.corr = 0;
            end
        end
    end
end

```

```

elseif tline(2)==' '
    %ID 2: Meas Nav Data
    scanvec = sscanf(tline,...
        '%d%f%f%f%f%f%d%d%d%f%d%d%d%d%d%d%d%d%d%d');
    %   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4
    sirfdata.rcvPos{t}.x = scanvec(2);
    sirfdata.rcvPos{t}.y = scanvec(3);
    sirfdata.rcvPos{t}.z = scanvec(4);
    sirfdata.rcvPos{t}.dx = scanvec(5);
    sirfdata.rcvPos{t}.dy = scanvec(6);
    sirfdata.rcvPos{t}.dz = scanvec(7);
elseif tline(2)=='9'
    %ID 29: DGPS Data
    chnID = chnID + 1;
    scanvec = sscanf(tline,'%d%d%d%f%f%f%f');
    %   0 1 2 3 4 5 6 7 8
    sirfdata.svSta{t}.DGPS{chnID}.ID = scanvec(2);
    sirfdata.svSta{t}.DGPS{chnID}.corr = scanvec(5);
end
case '3'
    %ID 30: sv state
    sv = sv + 1;
    scanvec = sscanf(tline,'%d%d%f%f%f%f%f%f%f%f%f%f');
    %   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4
    data.svState{t}(sv,1:11) = [scanvec(2) scanvec(3) scanvec(4)...
        scanvec(5) scanvec(6) scanvec(7) scanvec(8) scanvec(9)...
        scanvec(10) scanvec(11) scanvec(15)];
    i = 1; found = 0;
    while ~found
        if scanvec(2) == sirfdata.svSta{t}.tx{i}.ID
            found = 1;
        else
            i = i+1; if i>chnl; break; end
        end
    end
    sirfdata.svSta{t}.tx{i}.GPStime = data.svState{t}(sv,2);
    sirfdata.svSta{t}.tx{i}.x = data.svState{t}(sv,3);
    sirfdata.svSta{t}.tx{i}.y = data.svState{t}(sv,4);
    sirfdata.svSta{t}.tx{i}.z = data.svState{t}(sv,5);
    sirfdata.svSta{t}.tx{i}.xdot = data.svState{t}(sv,6);
    sirfdata.svSta{t}.tx{i}.ydot = data.svState{t}(sv,7);
    sirfdata.svSta{t}.tx{i}.zdot = data.svState{t}(sv,8);
    sirfdata.svSta{t}.tx{i}.clkBias = data.svState{t}(sv,9);
    sirfdata.svSta{t}.tx{i}.ionDelay = data.svState{t}(sv,end);
case 'W'
    %ID 7: clock status
    scanvec = sscanf(tline,'%s%d%s%d%s%d%s%s%d%s%s%s%d%s');
    %   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
    sirfdata.clkSta{t}.gpsWeek = scanvec(5);
    sirfdata.clkSta{t}.gpsTOW = scanvec(9);
    sirfdata.clkSta{t}.estTOW = scanvec(20);
    sirfdata.clkSta{t}.svCnt = scanvec(28);
    sirfdata.clkSta{t}.clkDrft = scanvec(39);
    sirfdata.clkSta{t}.clkBias = scanvec(51);

    %Initial to 0
    chnID = 0;
    chnID = 0;
    sv = 0;
    t = t+1;
case 'L'
    %Start & end time
    if ~logEnd

```

```

        scanvec = sscanf(tline, '%s%s%s%s%s%d%d%d');
        sirfdata.info.logBegin = [char(scanvec(27:34)) ', ' ...
            num2str(scanvec(35)) ':' num2str(scanvec(36)) ':' ...
            num2str(scanvec(37))];
        logEnd = 1;
    else
        scanvec = sscanf(tline, '%s%s%s%s%d%d%d');
        sirfdata.info.logClose = [char(scanvec(14:21)) ', ' ...
            num2str(scanvec(22)) ':' num2str(scanvec(23)) ':' ...
            num2str(scanvec(24))];
    end
end
end
fclose(fid);

```

B.4. FUNCTION FOR EXTRACTING MEASUREMENTS OF A SPECIFIC SECOND

```

function sat = pullData(sirfdata, eph)
svSta = sirfdata.svSta{eph}.tx;
dGPSsource = sirfdata.svSta{eph}.DGPS;
numSat = length(svSta);
count = 0;
tmp = svSta;
for i = 1:numSat %Filter out the sat w/ zero coordinate info
    if svSta{i}.x ~= 0
        if svSta{i}.ID <= 32
            count = count + 1;
            tmp(count) = svSta(i);
        end
    end
    try
        pRang = svSta{i}.pr;
    catch exception
        count = count - 1;
    end
end
svSta = tmp(1:count);
numSat = length(svSta);
numDGPS = length(dGPSsource);
for i = 1:numSat
    for j = 1:numDGPS
        if svSta{i}.ID == dGPSsource{j}.ID
            svSta{i}.DGPS = dGPSsource{j}.corr;
        end
    end
end
%%%Initial all the variables%%%
ID = zeros(numSat,1);
GPSSoftTime = zeros(numSat,1);
GPStime = zeros(numSat,1);
pr = zeros(numSat,1);
xyz = zeros(numSat,3);
xyzdot = zeros(numSat,3);
clkBias = zeros(numSat,1);
ionDelay = zeros(numSat,1);
DGPS = zeros(numSat,1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i = 1:numSat
    ID(i) = svSta{i}.ID;
    GPSSoftTime(i) = svSta{i}.GPSSoftTime;
    GPStime(i) = svSta{i}.GPStime;
    pr(i) = svSta{i}.pr;
    xyz(i,:) = [svSta{i}.x svSta{i}.y svSta{i}.z];

```

```

xyzdot(i,:) = [svSta{i}.xdot svSta{i}.ydot svSta{i}.zdot];
clkBias(i) = svSta{i}.clkBias;
ionDelay(i) = svSta{i}.ionDelay;
DGPS(i) = svSta{i}.DGPS;
end
sat.ID = ID;
sat.pr = pr;
sat.xyz = xyz;
sat.xyzdot = xyzdot;
sat.clkBias = clkBias;
sat.ionDelay = ionDelay;
sat.GPStime = GPStime;
sat.GPSSoftTime = GPSSoftTime;
sat.DGPS = DGPS;

```

B.5. FUNCTIONS OF CALCULATING USER POSITION

```

function usrPosEst = calUsrPos(sat,latParam,uTrans,refRd)
%
% This function calculates the user position using either
% 1) Standard procedure; OR
% 2) Modified procedure using lateral distance measurement; OR
% 3) Modified procedure using lateral corrections shared by other(s)
%
%   sat: Satellite information
% latParam: Lateral parameter: distance measurement or corrections
% uTrans: Unit vector in the transverse direction of the road
%   refRd: Reference position of road

satPos = adjustSatPos(sat);
numSat = length(sat.pr);
c = 2.99792458e8; %Speed of light
switch nargin
case 1 %Standard Solution
usrPosEst = zeros(1,4);
for k = 1:50
estClkBias = usrPosEst(4);
estPos = usrPosEst(1:3);
estPrVec = satPos - ones(numSat,1)*estPos;
estPr = sqrt(sum(estPrVec.^2,2));
G = [-estPrVec./repmat(estPr,[1 3]) ones(numSat,1)];
rangeEst = estPr + estClkBias - sat.clkBias*c;
dPr = sat.pr - rangeEst;
dUsr = G\dPr;
usrPosEst = usrPosEst + dUsr';
if all(abs(dUsr)<1e-5); break; end
end
case 3 %Solution using Differential Corrections
usrPosEst = zeros(1,4);
for k = 1:50
estClkBias = usrPosEst(4);
estPos = usrPosEst(1:3);
estPrVec = satPos - ones(numSat,1)*estPos;
estPr = sqrt(sum(estPrVec.^2,2));
G = [-estPrVec./repmat(estPr,[1 3]) ones(numSat,1)];
vecLat = G(:,1:3)*uTrans';
corr = vecLat*latParam;
rangeEst = estPr + estClkBias - sat.clkBias*c + corr;
dPr = sat.pr - rangeEst;
dUsr = G\dPr;
usrPosEst = usrPosEst + dUsr';
if all(abs(dUsr)<1e-5); break; end
end
end

```

```

case 4 %Solution using Lateral Measurement
usrPosEst = zeros(1,5);
for k = 1:50
    estClkBias = usrPosEst(4);
    estPos     = usrPosEst(1:3);
    estPrVec   = satPos - ones(numSat,1)*estPos;
    estPr      = sqrt(sum(estPrVec.^2,2));
    G          = [-estPrVec./repmat(estPr,[1 3]) ones(numSat,1)];
    vecLat     = G(:,1:3)*uTrans';
    rangeEst   = estPr + estClkBias - sat.clkBias*c + ...
        vecLat*usrPosEst(end);
    dPr        = sat.pr - rangeEst;
    dLat       = latParam - dot(uTrans,refRd-usrPosEst(1:3));
    dMeas      = [dPr; dLat];
    Gmod       = [G vecLat; -uTrans 0 0];
    dUsr       = Gmod\dMeas;
    usrPosEst  = usrPosEst + dUsr';
    if all(abs(dUsr)<1e-5); break; end
end
otherwise
warning('InputError:InvalidParam',...
    'Please input uTrans or both uTrans and refRd');
End

```

B.6. SATELLITE POSITION ADJUSTING FUNCTION

```

function posAfter = adjustSatPos(sat)
%
% This function adjusts the satellites' positions
% due to the earth rotation

c = 2.99792458e8; %(m/sec) Speed of light
wE = 7.2921150e-5; %(rad/sec) Angular velocity of Earth rotation
tTransmission = sat.GPSSoftTime - sat.pr/c; %Signal transmission time
tTravel = sat.GPStime - tTransmission; %Signal travelling time
adjMat = repmat(tTravel,1,3).*sat.xyzdot;
posAfter = sat.xyz - adjMat;
numSat = length(sat.pr);
for n = 1:numSat
    rotMat = [cos(wE*tTravel(n)) sin(wE*tTravel(n)) 0;...
        -sin(wE*tTravel(n)) cos(wE*tTravel(n)) 0;...
        0 0 1];
    posTemp = rotMat*posAfter(n,:);
    posAfter(n,:) = posTemp';
end

```

B.7. FUNCTION FOR ROTATING USER POSITION FROM ENU-FRAME TO ROAD-FRAME

```

function matOut = rot2road(ur,matIn)
%
% This function rotates the input matrix according to
% the road direction vector
% The output is [Along-road Transverse Up]
%
% ur: road direction vector
% matIn: input matrix

alpha = atan2(ur(2),ur(1));
n = size(matIn,1);
matOut = zeros(n,3);
for i = 1:n

```

```
beta = atan2(matIn(i,2),matIn(i,1));  
r = sqrt(matIn(i,2).^2 + matIn(i,1).^2);  
phi = beta - alpha;  
% Along-road Transvers Normal-to-road  
matOut(i,:) = [r*cos(phi) -r*sin(phi) matIn(i,3)];  
end
```

BIBLIOGRAPHY

- [1] E.G.R. Taylor, *The Haven-Finding Art: A History of Navigation from Odysseus to Captain Cook*, American Elsevier Pub. Co., 1971.
- [2] National Imagery and Mapping Agency, *The American Practical Navigator: Bowditch*, Paradise Cay Pub, 2010.
- [3] L. Jacobson, *GNSS Markets and Applications*, Artech House, 2007.
- [4] R. Heft, The Navigation Market for Road Vehicles, Proc. IEEE Position, Location and Navigation Symposium 1998, pp. 109-114.
- [5] C. Pellerin, United States Updates Global Positioning System Technology: New GPS Satellite Ushers in a Range of Future Improvements, *www.America.gov*, 2006.
- [6] P. Enge, Local Area Augmentation of GPS for the Precision Approach of Aircraft, *Proc. IEEE* 1999, vol. 87, no. 1, pp. 111-132.
- [7] P. Enge, T. Walter, S. Pullen, C. Kee, Y. Chao and Y. Tsai, Wide Area Augmentation of the Global Positioning System, *Proc. IEEE* 1996, vol. 84, no. 8, pp. 1063-1088.
- [8] S. Thrun, M. Montemerlo, *et al.*, Stanley, the robot that won the DARPA Grand Challenge, *J. Robotic Systems*, 23(9):661-692, 2006.
- [9] C. Urmson, J. Anhalt, *et al.*, Autonomous driving in urban environments: Boss and the Urban Challenge, *J. Field Robotics*, 25(8):425-466, 2008.
- [10] M. Montemerlo, J. Becker, *et al.*, Junior: the Stanford entry in the Urban Challenge, *J. Field Robotics*, 25(9):569-597, 2008.
- [11] M. Maile and L. Delgrossi, Cooperative intersection collision avoidance system for violations (CICAS-V) for avoidance of violation-based intersection crashes, *Proc. Enhanced Safety Vehicles (ESV) Conference*, Paper No. 09-0118, 2009.
- [12] F. Berefelt, B. Boberg, J. Nygard, P. Stromback and S.-L. Wirkander, Collaborative GPS/INS navigation in urban environment, *Proc. ION NTM* 2004, pp. 1114-1125.
- [13] D. A. Grejner-Brzezinska, C. K. Toth, L. Li, J. Park, X. Wang, H. Sun, I. J. Gupta, K. Huggins and Y. F. Zheng. Positioning in GPS-challenged environments: Dynamic sensor network with distributed GPS aperture and inter-nodal ranging signals, *Proc. ION GNSS* 2009, p. 111-123.
- [14] S. Wu, J. Kaba, S. Mau and T. Zhao, Distributed multi-sensor fusion for improved collaborative GPS-denied navigation, *Proc. ION ITM* 2009, pp. 109-123.
- [15] N. Drawil and O. Basir. Vehicular collaborative technique for location estimate correction, *Proc. IEEE Vehicular Technology Conference*, 2008.
- [16] S. Čapkun, M. Hamdi and J.-P. Hubaux, GPS-free positioning in mobile ad hoc networks, *Cluster Computing*, 5(2):157-167, 2002.
- [17] Mobileye, About Mobileye, **Error! Hyperlink reference not valid.**, 2010.
- [18] C. Mario and J. Rife, Integrity monitoring of vision-based automotive lane detection methods, *Proc. ION GNSS* 2010.

- [19] C. Mario, Integrity Analysis for Aviation and Automotive Applications, *Master's Thesis, Tufts University* 2011.
- [20] G. Cario, A. Casavola, G. Franze and M. Lupia, Predictive Time-to-Lane-Crossing Estimation for Lane Departure Warning Systems, *National Highway Traffic Safety Administration*. Pap. No. 09-0312, 2009.
- [21] J.C. McCall and M.M. Trivedi, Performance Evaluation of a Vision Based Lane Tracker Designed for Driver Assistance Systems, *Proc. IEEE Intelligent Vehicles Symposium 2005*, pp. 153-158.
- [22] J.M. Clanton, D.M. Bevly and A.S. Hodel, A Low-Cost Solution for an Integrated Multisensor Lane Departure Warning System, *IEEE Transactions on Intelligent Transportation Systems*, 10(1): 47-50, March 2009.
- [23] H. Wu, R. Fujimoto and G. Riley, Analytical Models for Information Propagation in Vehicle-to-Vehicle Networks, *Proc. IEEE Vehicular Technology Conference*, pp. 4548-4552 Vol. 6, 26-29 Sept. 2004.
- [24] W. Chen and S. Cai, Ad Hoc Peer-to-Peer Network Architecture for Vehicle Safety Communications, *Communications Magazine, IEEE* , vol.43, no.4, pp. 100- 107, April 2005.
- [25] S. Goel, T. Imielinski and K. Ozbay, Ascertaining Viability of WiFi Based Vehicle-to-Vehicle Network for Traffic Information Dissemination, *Proc. IEEE Intelligent Transportation Systems*, pp. 1086- 1091, 3-6 Oct. 2004.
- [26] T. Kosch, C. Schwingenschlogl and L. Ai, Information Dissemination in Multihop Inter-Vehicle Networks, *Proc. IEEE Intelligent Transportation Systems*, pp. 685- 690, 2002.
- [27] S. Biswas, R. Tatchikou and F. Dion, Vehicle-to-Vehicle Wireless Communication Protocols for Enhancing Highway Traffic Safety, *Communications Magazine, IEEE* , 44(1): 74- 82, Jan. 2006.
- [28] M. Sun, W. Feng, T. Lai, K. Yamada, H. Okada and K. Fujimura, GPS-Based Message Broadcasting for Inter-Vehicle Communication, *International Conference on Parallel Processing 2000*, pp.279.
- [29] P. Misra and P. Enge, *Global Position System: Signals, Measurements, and Performance*, Ganga Jamuna Press, 2006.
- [30] C. Jung and C. Kelber. A lane departure warning system based on a linear-parabolic lane model. *IEEE Intelligent Vehicle Symposium*, 2004, pp. 891-895.
- [31] D. Grejner-Brzezinska, C. Toth, and Q. Xiao, Real-time tracking of highway linear features, *Proc. ION GPS 2000*, pp. 1721-1728.
- [32] NGS, Precise GPS Orbits, **Error! Hyperlink reference not valid.**, 2010.
- [33] SiRF Technology Inc., SiRF Binary Protocol Reference Manual, *Revision 2.4*, November 2008.
- [34] SiRF Technology Inc., SiRFDemo User Guide, *Revision 1.2*, March 2006.
- [35] C.E. Barton and C.Z. Tarlowksi, Geomagnetic, Geocentric, and Geodetic Coordinate Transformations, *Computer & Geosciences*, 17(5): 669-678, 1991.
- [36] F.G. Lemoine, S.C. Kenyon, et al., The Development of the Joint NASA GSFC and NIMA Geopotential Model EGM96, *NASA Technical Paper*, July 1998

- [37] J. Seo, Overcoming Ionospheric Scintillation for Worldwide GPS Aviation, *Ph.D. Dissertation, Stanford University*, 2010.
- [38] J.D. Holmes, R.L. Meyer and Texas Instruments Inc Plano, *Ionospheric Delay Measurements Using GPS Satellite Signals*, Ft. Belvoir Defense Technical Information Center, 03 MAY 1990.