Report No. FHWA-KS-03-7
FINAL REPORT

Global Positioning System Data Integration and Development of a Three-Dimensional Spatial Model of the Kansas Highway Network

Dr. Shing I. Chang
Kansas State University

JUNE 2004

KANSAS DEPARTMENT OF TRANSPORTATION

Division of Operations
Bureau of Materials and Research

1 Report No.
FHWA-KS-03-7

2 Government Accession No.

3 Recipient Catalog No.

4 Title and Subtitle
GLOBAL POSITIONING SYSTEM DATA INTEGRATION AND DEVELOPMENT OF A THREE-DIMENSIONAL SPATIAL MODEL OF THE KANSAS HIGHWAY NETWORK

5 Report Date
June 2004

6 Performing Organization Code

7 Author(s)
Dr. Shing I. Chang

8 Performing Organization Report No.

9 Performing Organization Name and Address
Kansas Department of Transportation
(address same as #12)

10 Work Unit No. (TRAIS)

11 Contract or Grant No.
C1294

12 Sponsoring Agency Name and Address
Kansas Department of Transportation
Bureau of Materials and Research, Research Unit
2300 Southwest Van Buren Street
Topeka, Kansas 66611-1195

13 Type of Report and Period Covered
Final Report
August 2001 to July 2003

14 Sponsoring Agency Code
RE-0299-01

15 Supplementary Notes
For more information write to address in block 9.

16 Abstract

This proposal addresses a number of issues concerning the integration of various data collected on the Kansas Highway System through the development of a spatial (3D) model. Currently, KDOT collects and maintains spatial data on the Kansas highway system from the yearly road condition survey and video log survey. These surveys utilize differentially corrected Global Positioning Systems (GPS) systems cross-referenced to the traditional KDOT linear reference system (LRS) derived from distance measuring instruments (DMI) on board the vehicles. Preliminary development of a spatial model has been performed internally to KDOT resulting in a 2D model (latitude and longitude, but not altitude) for approximately 75% of the state highway system. This project extends the 2D model into a 3D model by including altitude GPS coordinates using a B-spline fitting model. The proposed 3D model shall serve as a cross referencing system between project stationing, project numbers, LRS, and other methods of defining physical locations. It will also serve to enable spatial analysis of the system for such items as sub-standard stopping sight distance, sub-standard horizontal curves, and the correlation of accidents with highway geometry.

This report summarizes technical achievements of this project in two parts. Part 1 describes the data cleaning, curving fitting algorithms and geometric modeling using B-spline techniques. The proposed algorithm is capable of modeling over 90% of Kansas highway with high accuracy. Control points – 3D data coordinates for B-spline fitting were generated and stored in a database. The proposed 3D models are validated by comparing the altitudes of the 3D models with those obtained from original highway designs. Part 2 outlines the procedures of generating sub-standard stopping sight distance. The entire procedure was automated, right from the importing of the data from the control point database to transferring the results into the two tables of the result database.

17 Key Words
3D Model, Global Positioning System, GPS, and Stopping Sight Distance

18 Distribution Statement
No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161

19 Security Classification (of this report)
Unclassified

20 Security Classification (of this page)
Unclassified

21 No. of pages
76

22 Price

Form DOT F 1700.7 (8-72)

ABSTRACT

This proposal addresses a number of issues concerning the integration of various data collected on the Kansas Highway System through the development of a spatial (3D) model. Currently, KDOT collects and maintains spatial data on the Kansas highway system from the yearly road condition survey and video log survey. These surveys utilize differentially corrected Global Positioning Systems (GPS) systems cross-referenced to the traditional KDOT linear reference system (LRS) derived from distance measuring instruments (DMI) on board the vehicles. Preliminary development of a spatial model has been performed internally to KDOT resulting in a 2D model (latitude and longitude, but not altitude) for approximately 75% of the state highway system. This project extends the 2D model into a 3D model by including altitude GPS coordinates using a B-spline fitting model. The proposed 3D model shall serve as a cross referencing system between project stationing, project numbers, LRS, and other methods of defining physical locations. It will also serve to enable spatial analysis of the system for such items as sub-standard stopping sight distance, sub-standard horizontal curves, and the correlation of accidents with highway geometry.

This report summarizes technical achievements of this project in two parts. Part 1 describes the data cleaning, curving fitting algorithms and geometric modeling using B-spline techniques. The proposed algorithm is capable of modeling over 90% of Kansas highway with high accuracy. Control points – 3D data coordinates for B-spline fitting were generated and stored in a database. The proposed 3D models are validated by comparing the altitudes of the 3D models with those obtained from original highway designs. Part 2 outlines the procedures of generating sub-standard stopping sight distance. The entire procedure was automated, right from

the importing of the data from the control point database to transferring the results into the two tables of the result database.

PART 1

DATA CLEANING, CURVE FITTING AND GEOMETRIC MODELING OF HIGHWAYS

Final Report

Prepared by

Dr. Shang I. Chang
Associate Professor

David Ben-Arich
Associate Professor

Guoxi Zhang
Research Associate

Venkatachalam Chokalingam
Research Associate

All with Kansas State University

A Report on Research Sponsored By

THE KANSAS DEPARTMENT OF TRANSPORTATION
TOPEKA, KANSAS

KANSAS STATE UNIVERSITY
MANHATTAN, KANSAS

June 2004

1. Problem Statements

The Kansas Department of Transportation has used GPS data in the past five years to map the state highway system. Complicating factors in this effort are the following facts. First, the data collection process repeats the same highway segments every year, and sometimes multiple times within a year. Second, the data collection is performed by a van that drives on the high way to collects a GPS reading once every second. Third, the data may be collected in either directions of a highway, and finally, KDOT is responsible for approximately 11 thousand miles of highway. Thus, this mapping effort results in a huge amount of GPS data, repeated for the same highway several times with a noticeable inconsistency. This inconsistency is attributed to the van driving in both directions (there can be 200 feet distance between the lanes), seasonal bias, satellite and system inherent inaccuracy and a somewhat arbitrary driving pattern of the van (e.g. the van can pass other vehicles thus change lanes, get off the road for refueling or taking logs, etc). In addition, the altitude measures of the GPS data are especially prone to high inaccuracies.

The purpose of this research is to sort this enormous amount of data and end up with a reliable geometric representation of the highway. In order to do so, the project consists of the following steps:

1. Convert the GPS latitude and longitude into a Cartesian coordinate system.
2. Sort the data by a novel grid-clustering algorithm.
3. Clean the data from repetitions and inaccuracies. This task obviously was performed by a computer due to the large amount of data.
4. Generate a concise and accurate geometric representation of the highway.

2. Converting the GPS Data

A position on the Earth is uniquely defined in the GPS data by longitude ?, latitude F and altitude above sea level. To build a 3D model, it is necessary to transform the terrestrial coordinates to Cartesian coordinates (x, y, z). Using z as the elevation the problem becomes the conversion of terrestrial coordinates (?, F) on the curved surface of the Earth to planar Cartesian coordinates (x, y).

Several map projection methods exist but the choice of the most suitable map projection technique is as much an art as a science.

2.1 Selection of the Type of Map Projection

The model of the Earth is considered a spheroid or a sphere, whose curved surface cannot be converted to a plane surface without distortion. A map projection can preserve one or several characteristics of the surface at the cost of distorting other characteristics. According to the type of characteristic maintained, map projection can be classified into three kinds: equal area projection, conformal projection and conventional projection.

Equal area projection maintains the ratio of area on the Earth to corresponding areas on the planar surface. The area in the planar surface is true, but angles and distances can be greatly distorted. Conformal projection preserves the shape figures of Earth on the planar surface. Angles are also not distorted. However, it introduces distortion in area or size.

In the conventional projection, neither area nor shapes are maintained. They are designed to keep specific characteristic to remain true. For example, the azimuthal equidistance projection preserves the azimuth and distances from the origin.

In this project the shape and angle of roads are important, and therefore conformal projection is chosen.

2.2 Selection of Projection Surface

To transform a position (P) on the Earth surface to a corresponding point P' on a planar surface, three kinds of projection surfaces are applied: planar, conical and cylindrical. The planar projection surface also called azimuthal projection is shown in Figure 1(a).

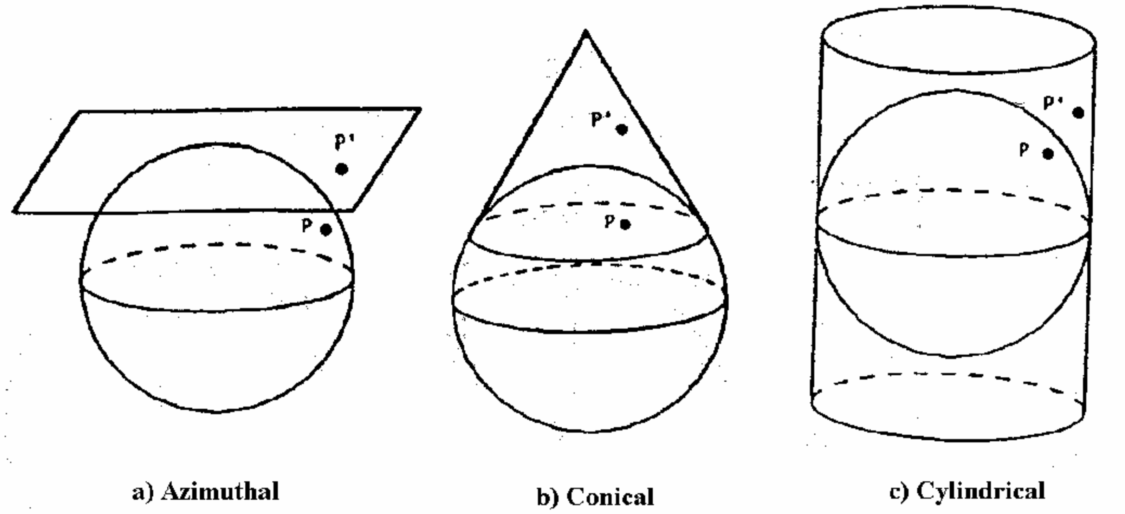

Figure 1 Projection surfaces

Azimuthal projection behaves well in high latitude from ±60° to ±90°. Figure 1(b) displays conical projection, which is appropriate for mid latitude from ±30° to ±60°. The cylindrical projection is shown in Figure 1(c), which is more suitable for equatorial area, that is, latitude from -30° to +30°.

Projection surfaces may have different orientations which result in different projection techniques. An azimuthal projection is called polar projection, equatorial projection or oblique projection if the azimuthal plane surface is tangent to the pole, the equator or any other point of the model of the Earth respectively. Similarly, the cylindrical projection surface is called regular, transverse or oblique when the mapping cylindrical surface is tangent to the equator, the pole or other points on the Earth separately.

Besides, as shown in figure 2, projection surfaces may be tangent or secant to the model of the Earth. As a result, more projection techniques are produced. Due to the fact that Kansas is in the middle of the nation and in mid latitude, a conical projection surface was selected.

Suggested projection technique: Lambert conformal projection with two standard parallels

There are many conical or oblique cylindrical projection techniques for the mid latitude area, such as Lambert conformal projection with two standard parallels (conical projection surface secant to the model of the Earth at two parallel circles), Mercator projection (regular, transverse or oblique cylindrical projection surface), etc. But for local area like the state of Kansas, they nearly have the same accuracy. Therefore in this KDOT project we decided to use Lambert conformal projection with two standard parallels.

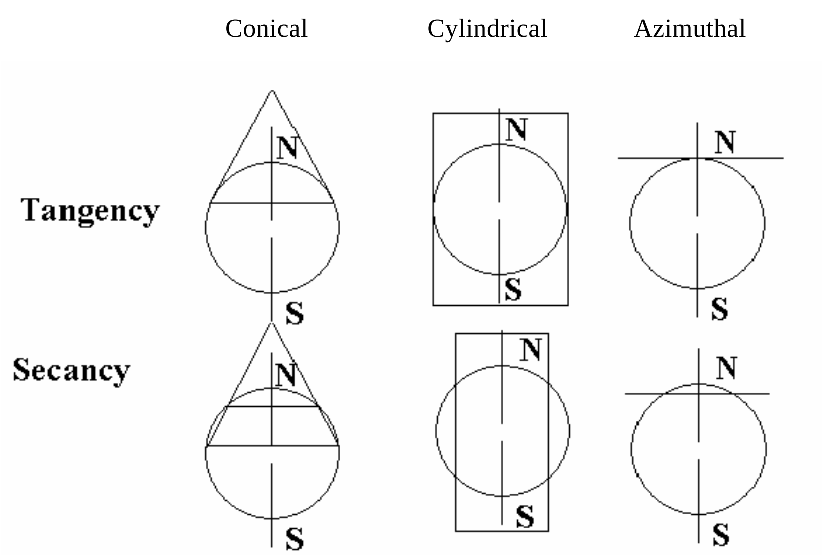

Figure 2 Tangency and secancy of the mapping surfaces

Figure 3 shows the Lambert conformal projection with two standard parallels has mapping conical surface secant to the Earth surface. It has traditionally been used for aircraft navigation charts for mid latitude area.

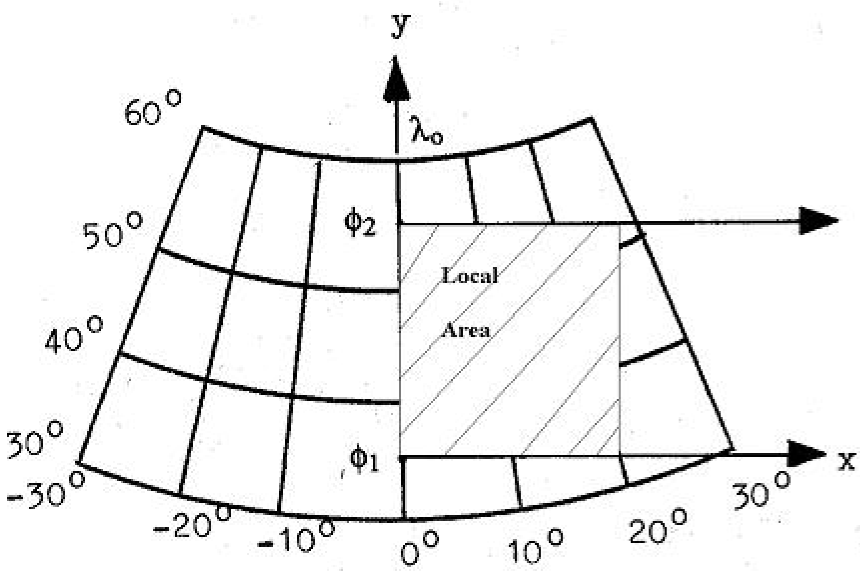

FIGURE 3 Lambert conformal projection, two standard parallels.

The following equations are used to transform (λ, φ) to (x, y):

(1) Constant of the cone

$$\sin \phi_0 = \frac{\ln\left(\dfrac{\cos \phi_1}{\cos \phi_2}\right)}{\ln\left[\dfrac{\tan(\pi/4 - \phi_1/2)}{\tan(\pi/4 - \phi_2/2)}\right]}$$

(2) Difference of longitude

$$\Delta\lambda = \lambda - \lambda_0$$

(3) Polar angle

$$\theta = \Delta\lambda \sin \phi_0$$

(4) Auxiliary function

$$y = \frac{R \cos f_1}{\sin f_0 \left[\tan\left(\frac{p}{4} - \frac{f_1}{2}\right)\right]^{\sin f_0}}$$

(5) Polar radius to origin

$$r1 = y \left[\tan\left(\frac{p}{4} - \frac{f_1}{2}\right)\right]^{\sin f_0}$$

(6) Polar radius to latitude φ

$$r = y \left[\tan\left(\frac{p}{4} - \frac{f}{2}\right)\right]^{\sin f_0}$$

(7) Cartesian plotting coordinates

$$x = S r \sin q$$

$$y = S(r1 - r \cos q)$$

In the above equations:

ϕ_1, ϕ_2: latitudes of the starting points and ending points of the area to be transformed.λ_0: longitude of the starting points of the area.

(λ,ϕ): longitude and latitude of a position in Terrestrial coordinates

Origin: (λ_0,ϕ_1) corresponding to (0,0) in Cartesian coordinates.

S: scale, that is, the ratio of a true length distance on the map to the equivalent distance on the Earth.

3. Characteristics of Raw GPS Data

GPS data are more accurate in latitude and longitude than in altitude. Figure 4 depicts the raw GPS data for elevation, measured along a segment of Highway 177 in Riley County. This data reflects the duplications in collecting the data and the errors and biases that occur in each such sampling run.

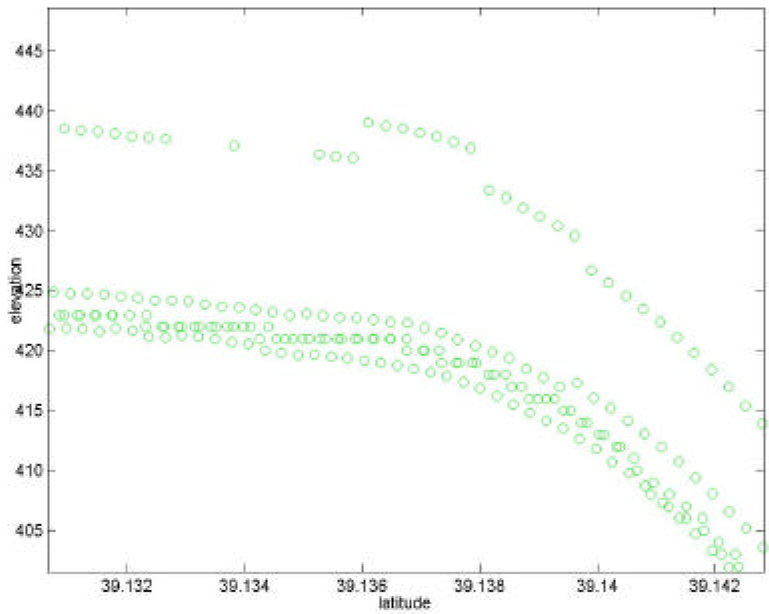

Figure 4: Raw GPS Data Showing Elevation

4. Data Cleaning, Smoothing and Curve Fitting

The following algorithm has been developed to clean the data and fit curves to the data.

The purpose of the algorithm is to combine the various data streams together, remove outliers and generate correct control points. The control points are used by the B-Spline representation of the data.

Step 1. Use "Lambert conformal projections with two standard parallels" map projection method to transform terrestrial coordinates to Cartesian coordinates.

Step 2. The two-dimensional data, i.e., longitude and latitude, are separated into different directions because every highway has one main direction (Easy-West or North-South). Then the data points are sorted by increasing mileages along the main direction.

Step 3. Grid-clustering approach to sorting the data

1. Sort the data in one direction of a highway by increasing time. These data are then called time-sorted-data.

2. For the time-sorted-data, let random variable dx represent the difference between every two consecutive points in x direction and dy denote that in y direction. Calculate their mean value $\overline{dx}$ and $\overline{dy}$.

3. The shape of grids is square. The length of a grid is $\sqrt{\overline{dx}^2 + \overline{dy}^2}$.

4. Put the data into a suitable grid that is represented by grid(index_grid_x, index_grid_y). For data point (x, y, z),

$$index\_grid\_x = \left[\frac{x - \min(x)}{\sqrt{\overline{dx}^2 + \overline{dy}^2}} \right] + 1, \quad index\_grid\_y = \left[\frac{y - \min(y)}{\sqrt{\overline{dx}^2 + \overline{dy}^2}} \right] + 1,$$

where function $[x]$ represent the largest integer less than x. Several data points may fall into one grid. In this case, calculate the average value of x, y, z and heading direction in that grid to denote the x value, y value, z value and heading direction of that grid.

5. Let the "seed" represent the grid with smallest index_grid_x and index_grid_y, find its nearest neighbor grids, then choose the next "seed" that is the grid whose heading direction is closest to the heading direction of the current "seed". In this way, find all the grids until the furthest point from the starting "seed". The data in each grid are sorted data. These data are called grid_sorted_data.

Step 4. For the grid_sorted_data, calculate the elevation difference between every two consecutive points, $dz = z(i) - z(i-1), i = 2,3,4,...$.

Step 5. Clean the data in the first run. Begin with the first point of the grid_sorted_data, if $dz = z(i) - z(i-1) > 5$, throw point i away and continue to the next point. The cleaned data are called first_run_data.

Step 6. Do a robust second run cleaning for the first_run_data.

Normally every consecutive 30 points form a fundamental curve (vertical curve or horizontal curve) or a straight line. In some rare cases, they constitute a combination of fundamental curves. A parabolic function has been found suitable to represent the vertical curve and straight line forms (Mannering and Kilareski, 1990). Compare the range of x and that of y. If the range of x is larger than that of y, use least squares regression method to fit z versus x and y versus x; or else, fit z versus y and x versus y. The least squares fitting algorithm and function can be seen in Matlab tutorial (MathWorks, 2000). Calculate r_z, the residual of z and r_{xy}, the residual of x or y.

Calculate the mean μ and standard deviation s of residuals. Let μ_z and s_z denote the mean and standard deviation of residuals of elevation; μ_{xy} and s_{xy} denote those of x or y.

Remove outliers. Least squares regression is sensitive to messy data. Removing the data with large residuals makes the fitting more robust (Rousseuw and Leroy, 1987). Here two sigma control limit is used to detect outliers. For any point, if $m_z - 2s_z \le r_z \le m_z + 2s_z$ and $m_{xy} - 2s_{xy} \le r_{xy} \le m_{xy} + 2s_{xy}$, then this point is considered as a good point. If the above condition is not satisfied, remove the point away. As a result, outliers are thrown away and good cleaned data are produced.

Use least squares regression to fit the good cleaned data again. In the fitted curve, select four points with same interval of x or y. These four points are called second run cleaned data representing the road segment constituted by the 30 data points.

Repeat the above steps for all segments and generate the second run cleaned data.

Step 7. Use B-spline to fit the data. Cubic B-spline has the continuity of order two. Use the second run cleaned data as control points to generate cubic B-spline curves (Mortenson,1985). Suppose the number of control points is n. Control Point sequence $P_1P_1P_1P_2P_3P_4 \ldots .P_nP_nP_n$ generate the curve passing through point P_1 to point P_n. The curve between P_j and P_{j+1} $(1 \le j \le n-1)$ is given by:

$$(x(t), y(t), z(t)) = \frac{1}{6}\begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_{j-1} & y_{j-1} & z_{j-1} \\ x_j & y_j & z_j \\ x_{j+1} & y_{j+1} & z_{j+1} \\ x_{j+2} & y_{j+2} & z_{j+2} \end{bmatrix}$$

where $t \in [0,1]$.

5. Results of the Algorithm

Figure 5 shows the fitted B-Spline for the road segment of highway 177. The fitted curve goes through the points smoothly and removes many biased points. The B-Spline also allows continuity of 2 degree between the curve segments (continuity of contact points and slope).

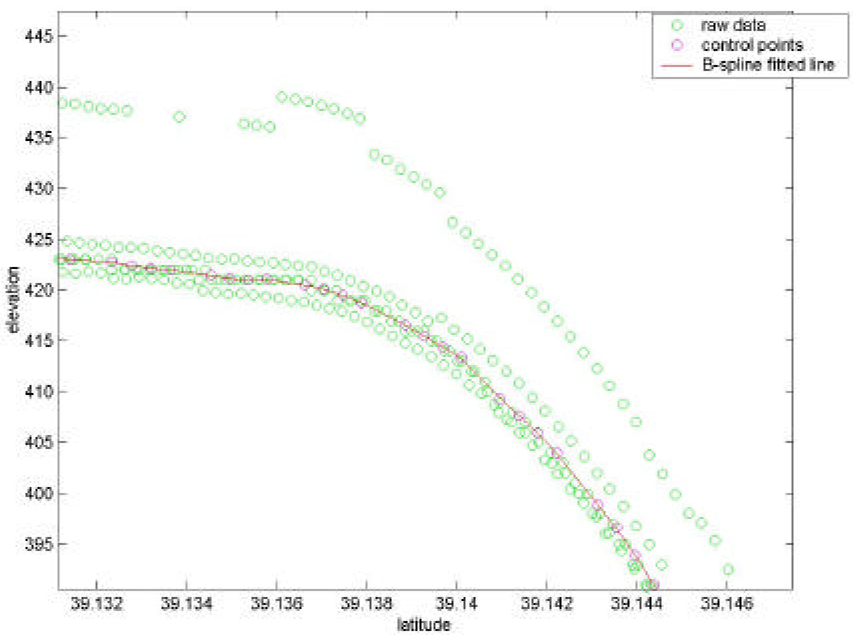

Figure 5: Data Cleaning and Curve Fitting Result

Figure 6 shows a comparison between the B-Spline model of the data and a piecewise polynomial approximation without continuity constraints at the boundaries of the segments. The figures clearly show the superior representation of the B-Spline model over the polynomial approximation without continuity constraints at the boundaries of the segments.

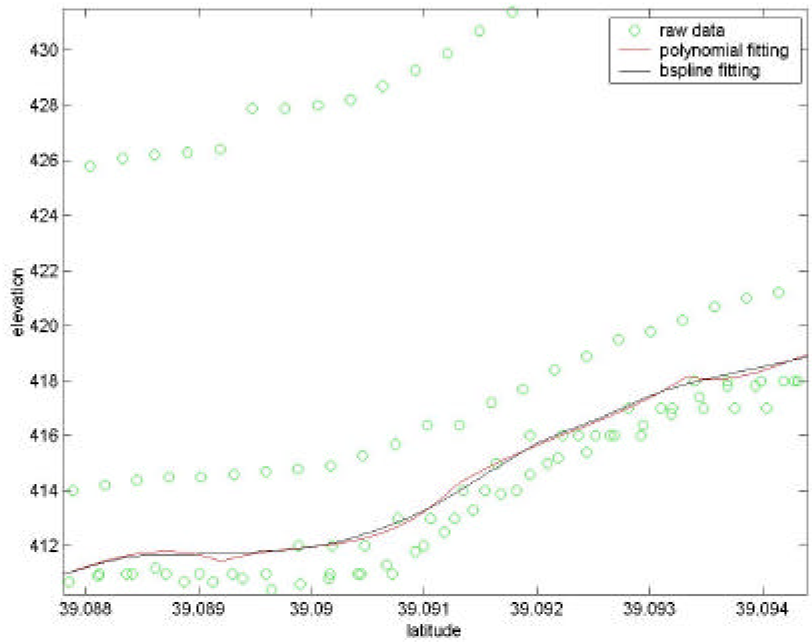

Figure 6: Comparison of Piecewise Polynomial Fitting and B-Spline Approximation

6. Accuracy of this algorithm

Figure 7 demonstrates the accuracy of the generated B-Spline representation of the highway. This is done by comparing the B-Spline representation generated from GPS data, and the geographical data generated by the highway design team. The two curves are very close to each other, so the algorithm's result is good.

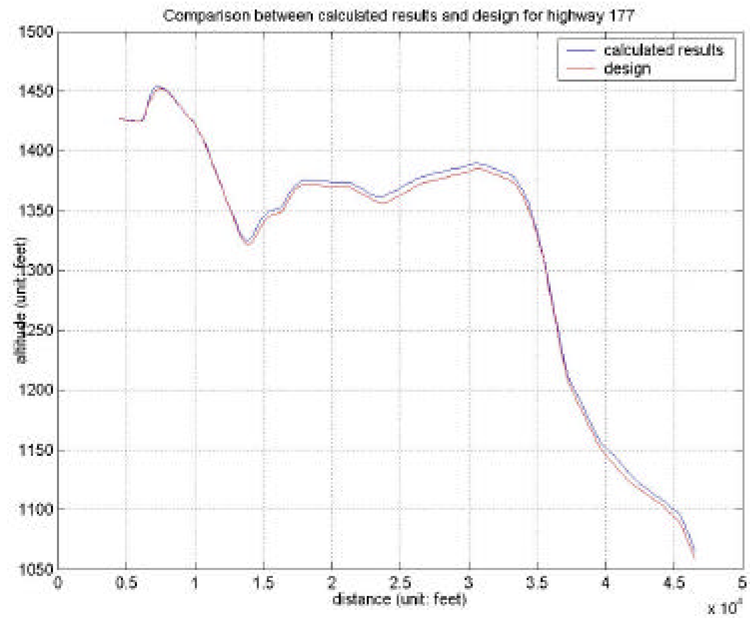

Figure 7: Comparison of Piecewise Polynomial Fitting and B-Spline Approximation

7. Error Analysis

In figure 5, let error at each point= elevation from fitting result - elevation from design at the same point, bias=mean(error at each point in the highway) and adjusted error (variance) = error at that point - bias. The biases for some highways are shown in the following table.

Table 1: Bias for Highway 177, 113 and 283

Highway	177 in Riley	113 in Riley	283 in Hodgeman
Bias (meter)	1.24	5.18	-0.16

The cumulative distribution function (cdf) for adjusted error (variance) of highways in the above table is described in Figure 8. It indicated that 80% of adjusted errors are within 0.5 meter for Highway 113, 1 meter for Highway 177 and 1.2 meter for Highway 283. The adjusted error is so small that the error can be neglected.

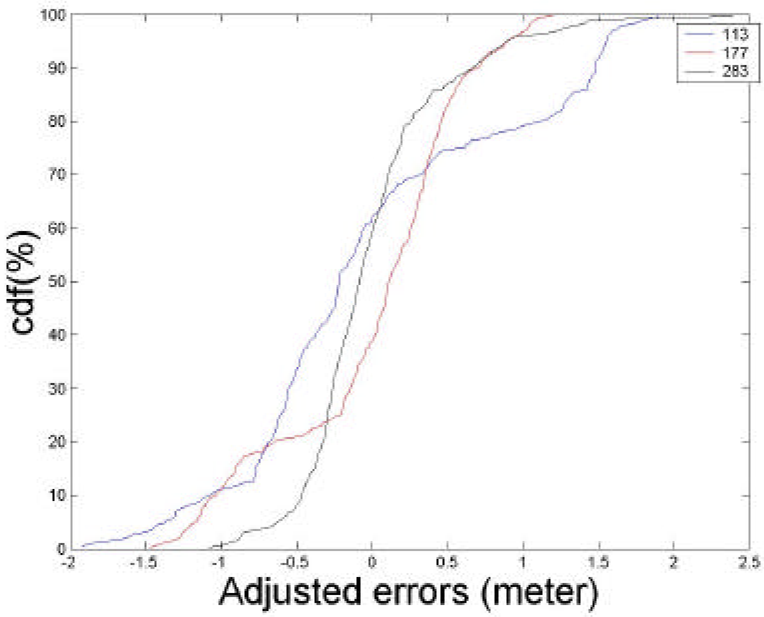

Figure 8: Cumulative Distribution Function for Highway 113, 177 and 283

8. Conclusion

1. 90% of highways in Kansas can be fitted by our algorithm
2. Accuracy of the algorithm: The cumulative distribution function shows that results of our algorithm is accurate.
3. This model provides a variety of applications: stopping sight distance analysis, passing sight analysis, verification of design, etc.

REFERENCES

1. Frederick Pearson ?, *Map Projections: Theory and Application*, CRC Press, Inc, 1990.
2. Matlab Mapping Toolbox reference.
3. Pearson F., 1990, *Map Projections: Theory and Application*, CRC Press, Inc.
4. Mannering F.L., and Kilareski, W.P., 1990, *Principles of Highway Engineering and Traffic Analysis*, John Wiley & Sons.
5. MathWorks, *Using Matlab (version 6)*, 2000.
6. Mortenson, M. E., 1997, *Geometric Modeling*, Second Edition, John Wiley & Sons.
7. Rousseuw P.J., and Leroy, A.M., 1987, *Robust Regression and Outlier Detection*, John Wiley & Sons.

PART 2

AUTOMATION OF CALCULATING STOPPING SIGHT DISTANCE FOR VERTICAL ALIGNMENTS FROM GLOBAL POSITIONING SYSTEM DATA

Final Report

Prepared by

Dr. Margaret J. Rys
Associate Professor

Girish Nehate
Research Assistant

Samson Namala
Research Assistant

All with Kansas State University

A Report on Research Sponsored By

THE KANSAS DEPARTMENT OF TRANSPORTATION
TOPEKA, KANSAS

KANSAS STATE UNIVERSITY
MANHATTAN, KANSAS

June 2004

ABSTRACT

Sight distance is a key element in highway geometric design. When the sight distance falls below the distance required for bringing the vehicle to a complete stop, it becomes unsafe for driving. Identifying such segments of zones (stoppers) becomes crucial to ensure safety on highways. A model for evaluating sight distance is developed for vertical alignments. Parametric equations in the form of cubic b-splines are used to represent the highway surface and sight obstructions, including tangents (grades), horizontal curves, and vertical curves. The available sight distance is found analytically by examining the intersection between the sight line and the elements representing the highway surface and sight obstructions. A profile of available sight distance can be established and used to evaluate sight distance deficiency. The available sight distances and the required sight distances are calculated. Any changes in the parameters or conditions can be accommodated. The entire procedure has been automated and the identification of stoppers has been done for all the highways in the state of Kansas.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. Introduction

Stopping sight distance is a major element in the safe operation on any highway. Sight distance calculations are often done considering horizontal and vertical geometry separately. Of course, there are situations where the combined effect of the two dimensions is more restrictive than when either is considered separately (Lovell, 1999).

Sight distance has been recognized as a key element in highway geometric design. Therefore, extensive research work has been directed at determining the required and available sight distance. The American Association of State Highway and Transportation Officials (AASHTO) presents mathematical formulas to determine the available sight distance on simple horizontal and vertical alignments (AASHTO, 2001). Further research has been conducted and analytical models that can deal with complex horizontal and vertical alignments have been developed (Hassan and Easa, 1995). However, most of the stoppers in real life occur due to vertical geometry of the highways, therefore the calculations on sight distance due to vertical geometry are of interests to the highway designers.

The SSD model has two components, available sight distance (SSD_{avail}) and required stopping sight distance (SSD_{req}). SSD_{avail} is equal to the distance along the highway within which the driver is able to perceive the existence of an unexpected object, of a specific height, stationary or slowly moving ahead. SSD_{req} is equal to the distance traversed by the vehicle, from the instance the driver sights an object necessitating a stop, to the instant the vehicle is fully immobilized. At any location, SSD_{avail} should be greater than or at least equal to the corresponding SSD_{req}; otherwise there will exist a segment of highway with SSD restriction (Taiganidis, 1998).

Sight distance is defined as the length of the roadway ahead that is visible to the driver (AASHTO, 2001). From a geometric design standpoint, the minimum sight distance available on a roadway should be long enough to enable a vehicle traveling at design speed to stop before reaching a stationary object in its path. Although greater length is desirable, sight distance at each and every point along the highway should not be below a minimum value that is required for a below-average driver or vehicle to stop (AASHTO, 2001).

AASHTO defines the stopping sight distance (SSD) as the sum of two components, brake reaction distance (distance traveled from the instant of object detection to the instant the brakes are applied) and braking distance (distance traveled from the instant the brakes are applied to when the vehicle is decelerated to a stop). The objective of this research is to find such segments on the highways in the state. Calculation of the stopping sight distance is based on the parameters like speed limit and profile of the road.

2. Parameters for Measuring Sight Distance

Sight distance is the distance along a roadway throughout which an object of specified height is continuously visible to the driver. This distance is dependent on the height of the driver's eye above the road surface, the specified object height above the road surface, and the height and lateral position of sight obstructions within the driver's line of sight.

2.1 Height of Driver's Eye

Driver eye height values are a combination of the height of driver stature and driver seat height. The design value for driver eye height is selected so that the majority of driver eye heights in current vehicles will be greater than design values. For sight distance calculations for passenger vehicles, the height of the driver's eye is considered to be 3.5 feet above the road surface. Fambro, Fitzpatrick and Koppa found that average vehicle heights have decreased to 4.3

feet with a comparable decrease in average eye heights to 3.5 feet. Because of various factors that appear to place practical limits on further decreases in passenger car heights and the relatively small increases in the lengths of vertical curves that would result from further changes that do occur, 3.5 feet is considered to be the appropriate height of driver's eye for measuring stopping sight distances (AASHTO, 2001).

2.2 Height of Object

For stopping sight distance calculations, the height of object is considered to be 2 feet above the road surface. The basis for selection of a 2 feet object height was largely an arbitrary rationalization of the size of object that might potentially be encountered in the road and of a driver's ability to perceive and react to such situations. It is considered that an object 2 feet high is representative of an object that involves risk to drivers and can be recognized by a driver in time to stop before reaching it (AASHTO, 2001). Using object heights of less than 2 feet for stopping sight distance calculations would result in longer crest vertical curves without documented safety benefits (Fambro, Fitzpatrick and Koppa, 1997). Object height of less than 2 feet could substantially increase construction costs because additional excavation would be needed to provide the longer crest vertical curves. It is also doubtful that the driver's ability to perceive situations involving risk of collisions would be increased because recommended stopping sight distances for high-speed design are beyond most driver's capabilities to detect small objects (Fambro, Fitzpatrick and Koppa, 1997).

2.3 Lane Width

No feature of a highway has a greater influence on the safety and comfort of driving than the width and condition of the surface. There is an obvious need for a smooth, skid-resistant, all-weather surface on highways. 10 feet to 13 feet lane widths are generally used, with a 12 feet lane predominant on most typical highways. Studies on two-lane two-way rural highways by the Highway Research Board show that undesirable conditions (inadequate vehicle clearances and edge-of-pavement clearances) exist on surfaces less than 22 feet wide carrying even moderate volumes of mixed traffic. A 24 feet surface is required to permit desired clearance between commercial vehicles. It is generally accepted that lane widths of 12 feet should be provided on main highways (AASHTO, 2001).

2.4 Clear Zone

The term "clear zone" is used to designate the unobstructed, relatively flat area provided beyond the edge of the traveled way for the recovery of errant vehicles. The traveled way does not include shoulders or auxiliary lanes (AASHTO, 2001).

The growth of an urban area typically extends outward along major arterial highways. The nature of the land use along the highways gradually changes from rural and agricultural to suburban with strip commercial developments, such as service stations, fast food restaurants and shopping centers. The resulting growth in traffic volume and frequent turning movements can cause congestion and increase accident experience, which may necessitate widening the existing two-lane highways to four or more lanes. Also, in anticipation of future growth, these suburban arterial highway sections are designed with curb-and-gutter cross sections and often with two-way left-turn center lanes, typical of urban type roadways. However, these highway sections will remain suburban in nature for a period of time, i.e. with moderate traffic volume and high speed, and speed limits ranging from 50 to 55 mph). The land use and resulting traffic volume will continue to grow until these highway sections become urban roadways with high traffic volume and lower traffic speed limits [i.e. 45 mph (72.4 km/hr) or less] (Mak, Bligh and Ross, 1995).

These suburban arterial highway sections pose some interesting problems because they serve as a transition from rural to urban type highways at the fringes of urban areas. Under current AASHTO design guidelines, low speed [i.e. 45 mph (72.4 km/hr) or less] urban roadways with curb-and-gutter cross sections and no shoulders must have a minimum clear zone width of 18 inches beyond the face of the curb. On the other hand, high-speed rural arterial highways with shoulders and parallel drainage ditches are typically required to have a clear zone width of 30 feet or more beyond the edge of the travel way (i.e. edge line or edge of pavement) (Mak, Bligh and Ross, 1995). The AASHTO Roadside Design Guide (1989) discusses clear zone widths as related to speed, volume and embankment slope.

2.5 Headlight Height

In general, headlight height of 2 feet and a 1° upward divergence of the light beam from the longitudinal axis of the vehicle are used (AASHTO, 2001).

2.6 Stopper

The SSD model has two components, available sight distance (SSD_{avail}) and required stopping sight distance (SSD_{req}). SSD_{avail} is equal to the distance along the highway within which the driver is able to perceive the existence of an unexpected object, of a specific height, stationary or slowly moving ahead. SSD_{req} is equal to the distance traversed by the vehicle, from the instance the driver sights an object necessitating a stop, to the instant the vehicle is fully immobilized. Both the components of SSD are calculated based on the GPS data. Although we calculate SSD_{req}, in this study, any region with SSD_{req} below 1,000 feet is considered as a stopper (KDOT).

3. Stopping Sight Distance Algorithm

3.1 Methodology

Sight distance may be obstructed by vertical alignments or horizontal alignments or a combination of both. On vertical alignments, if the sight line from the driver's eye to the object should pass over a crest curve, it may intersect the road itself. The point of intersection may be a tangent segment, a crest curve, or a sag curve. Subsequently, the sight distance is limited by having the sight line tangent to the crest curve. In addition, sight distance on sag curves at night time is limited by the distance ahead covered by the vehicle headlight.

GPS data collected by Kansas Department of Transportation (KDOT) has been used for the analysis. GPS raw data for the roads is collected in the form of latitude, longitude and elevation and stored in databases. This data is organized by the years of data collection and driving direction, from the raw GPS data databases. The terrestrial coordinates (GPS data) are transformed into Cartesian coordinates using coordinate transformation. Lambert conformal projection with two standard parallels transformation method is used for coordinate transformation. The elevation is checked for outliers and those found are eliminated. Control points thus obtained from the data are used for SSD analysis.

Parametric equations in the form of cubic B-splines are used to represent the highway surface and sight obstructions, including tangents (grades), horizontal curves, and vertical curves. The available sight distance is found by examining the intersection between the sight line and the elements representing the highway surface and sight obstructions. A profile of available sight distance is established and used to evaluate sight distance deficiency.

The road obtained using GPS data is represented here as a curve rather than a surface. The advantage of representing it, as a curve is the ease of analysis and lesser time required for calculations. Not many powerful and fast calculation techniques are available for analyzing the road surface planes. Therefore, the road for the purpose of calculating SSD has been represented as a curve using parametric equations.

Every point in space can be represented by 3-dimensions viz. (x,y,z) to determine its position with reference to the origin. Figure 3.1 shows a point P in space with coordinates (x, y, z). OP makes an angle α with the XOZ plane and an angle β with the XOY plane.

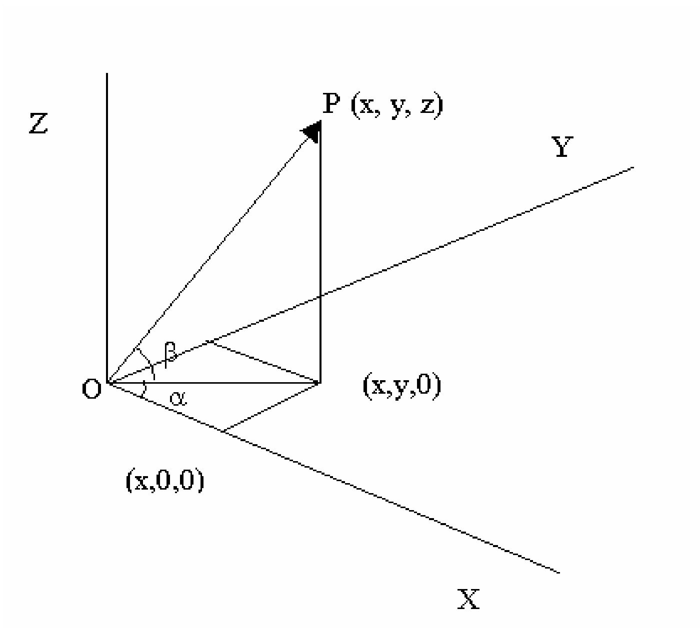

Figure 3.1: 3D-Coordinate Geometry

In Figure 3.2, curve AB represents the road with the start point A and end point B and the travel direction A to B. At the point S, the tangent is represented by dS. dS has components dx, dy and dz in directions X, Y, and Z respectively. dS makes an angle θ in the horizontal plane with the X-Z plane and an angle ϕ in the vertical plane with the X-Y plane. X-Z and X-Y planes are the planes of references for the calculation of angles θ and ϕ.

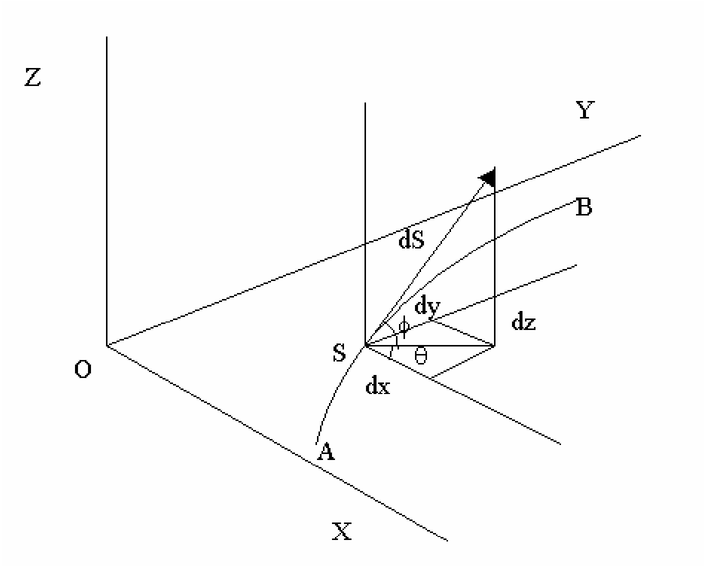

Figure 3.2: 3D Representation of Curve AB

3.2 Construction of B-Spline

Choose four consecutive points to generate a B-spline curve that fit these five points. Suppose, the points form the control sequence $P_0P_1P_2P_3$ the curve between $P_j(x_j,y_j,z_j)$ and $P_{j+1}(x_{j+1},y_{j+1},z_{j+1})$ is given by:

$$(x(t),y(t),z(t)) = \frac{1}{6}\begin{bmatrix} t^3 & t^2 & t & 1\end{bmatrix}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}\begin{bmatrix} x_{j-1} & y_{j-1} & z_{j-1} \\ x_j & y_j & z_j \\ x_{j+1} & y_{j+1} & z_{j+1} \\ x_{j+2} & y_{j+2} & z_{j+2} \end{bmatrix} \qquad (3.1)$$

where $0<t<1$.

A control point sequence $P_0P_1P_2P_3$ will generate a curve that passes through the points P_0 and P_3. By fitting consecutive points in this way, the profile of the road is obtained. By varying the value of t between 0 and 1, the intermediate points are obtained. In the algorithm, t is varied by 0.05 i.e. 20 points are generated between two control points. These points are used for calculation purposes.

By taking the differential of the equation 3.1, we get the components in the three directions x, y and z at each point as shown in equation 3.2.

$$(dx(t), dy(t), dz(t)) = \frac{1}{6}\begin{bmatrix} 3t^2 & 2t & 1 & 0 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_{j-1} & y_{j-1} & z_{j-1} \\ x_j & y_j & z_j \\ x_{j+1} & y_{j+1} & z_{j+1} \\ x_{j+2} & y_{j+2} & z_{j+2} \end{bmatrix} \qquad (3.2)$$

Converting the components of derivatives from Cartesian to spherical coordinates, we get the values of ϕ and θ. Note that the values of ϕ and θ are in radians. Using the above logic for calculations of intermediate points and angles, data for the entire road is calculated and stored in arrays.

The length of the road from the first point is calculated by an incremental method. The increment in length between two consecutive points is calculated and added to the length, giving the length of the road from the first point. This length is also stored in an array. Thus, the distance between any two points can be easily found by just subtracting the values of the length at each of these points.

The increment in length (l) between two consecutive points A(x_a, y_a, z_a) and B(x_b, y_b, z_b) is calculated as a straight line approximation given by:

$$l = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2 + (z_b - z_a)^2} \qquad (3.3)$$

the required SSD is calculated using the formula as:

$$SSD_{req} = 0.278*V*T_{pr} + \frac{V^2}{254(\frac{a}{9.81} + G)} \qquad (3.4)$$

where: SSD_{req} = required stopping sight distance

V = initial speed of the vehicle (km/h)

T_{pr} = driver perception-brake reaction time (sec)

a = deceleration rate, m/s^2

G = grade of the braking section

+ for upgrade

- for downgrade

The value ϕ gives the vertical elevation of the road with the horizontal at the point of consideration. Thus,

$$G = \tan(\phi) \quad (3.5)$$

On upgrade, G is positive, while on downgrades G is negative. The above method is used to calculate the required SSD value at each point along the road for which coordinates are calculated.

3.3 Object Visibility Studies

The Swedish Design Standards (National Swedish Road Administration, 1986) state that under bright light conditions, 1 min of arc is the minimum angle that part of the obstruction must cover to allow a driver with 20/20 static visual acuity to perceive it as an object if he or she is looking for it. McLean stated that an observer could resolve detail under lighting and contrast conditions when an object spanned 1 min of arc (McLean, 1988). Considering the atmospheric and environmental conditions that a driver encounters on the roadway, 5 min of arc would be necessary to perceive an object on the roadway surface (McLean, 1988). Under these conditions, 0.039 inches of an object must be above the line of sight to detect it at a distance of 213 feet. At a distance of 1280 feet, the object must have 2 feet above the line of sight to be detected (McLean, 1988).

Therefore, a moving window of 1312 feet is used as a threshold value. For calculations, sight distance is checked for a distance up to 1312 feet. If the sight distance is more than 1312 feet, it is not calculated as the above object visibility studies show that a normal person cannot identify an obstruction beyond this distance.

3.4 Calculation of Available SSD

The parameter ‘t’ used in the calculations represents the index for the arrays used to store the values. The first point t=1. e.g. x(t=1) is represented as x(1) and is the abscissa of the first point. The parameter ‘t’ thus gives the location of the point along the road.

Sight distance is calculated for each point along the road in one direction, i.e. the data direction. A complete sight distance profile for the road can be constructed by repeating the process for a single point.

The available sight distance on vertical alignments may be governed at daytime by the road surface and at night-time by the vehicle headlight beam. Sight distance calculations are done for all the points that are generated between the control points. Calculations are done for vertical as well as horizontal alignments and the lower bound is taken as the available sight distance at that point.

As stated previously, the road itself may obstruct the sight line in the case of a crest vertical curve. Although the road cannot obstruct the sightline unless there is a crest curve, a sag curve or a tangent segment. The available sight distance using vertical geometry is determined assuming an initial value of 0 for the sight distance. The following steps explain this procedure with reference to Figure 3.3.

1. Determine the coordinates of the point A at which sight distance is to be calculated. The coordinates of P are then calculated by adding the height of driver’s eye.
2. Determine the point B along the path such that the line PB and the tangent at B coincide. An iterative procedure is followed to get the location of B. The angle made by the line PB with the horizontal plane i.e. ϕ_2 is found

and compared with the angle ϕ made by the tangent at B with the horizontal plane. Here, some approximate means of determining whether the condition is met is necessary. For this purpose, the following approximation is used:

$$(\phi_2 - \phi) < \delta \qquad (3.6)$$

for some small value of tolerance δ.

1% tolerance means that any answer, which is within 1% deviation in either direction from the correct answer, will be accepted as correct. In the application developed, tolerance value of 0.001 is accepted to give reasonable results.

A sight line is thus established which is obstructed by the road surface itself.

If no point of intersection is found till a distance of 1312 feet, then the sight distance is assumed as 1312 feet, which is the distance that can be seen by a driver with normal vision.

3. Point C is determined when the following condition is met. The condition being the tangent at B should coincide with line BQ. An iterative procedure is followed to get the location of C. The coordinates of point Q are found by adding the object height to a point C. Again, the angle made by line BQ with the horizontal plane i.e. ϕ_3 is found and compared with the angle ϕ made by the tangent at B with the horizontal plane. For this purpose, the following approximation is used:

$$(\phi_3 - \phi) < \delta \qquad (3.7)$$

where δ is the tolerance.

4. The sight distance is the length AC along the road.

Since the above method only makes use of ϕ, it is similar to taking vertical projections of the road for the calculations and makes no difference even if the road has horizontal curvature at the same time.

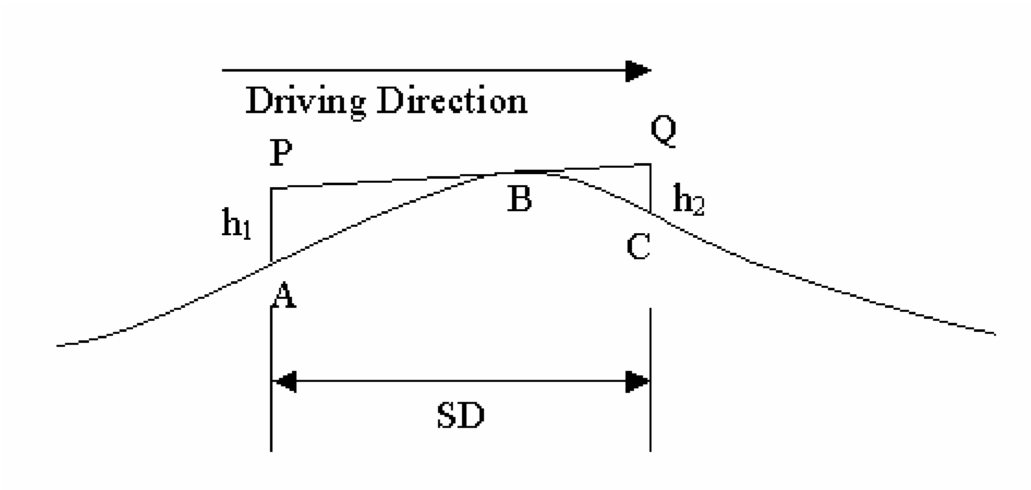

Figure 3.3: Sight Distance Limited by Crest Vertical Curve (Profile view)
(h_1, height of driver's eye; h_2, height of object; SD, sight distance)

The above procedure also accounts for sight hidden dips as shown in Figure 3.4. The reason being, sight distance is calculated in the direction of travel starting from point A. Once point B is found, as explained in the above procedure, the next task is to find the point closest to point B that coincides with line BQ. Point C is the nearest point to point B than point D. Moreover, point C comes prior to point D in the driving direction.

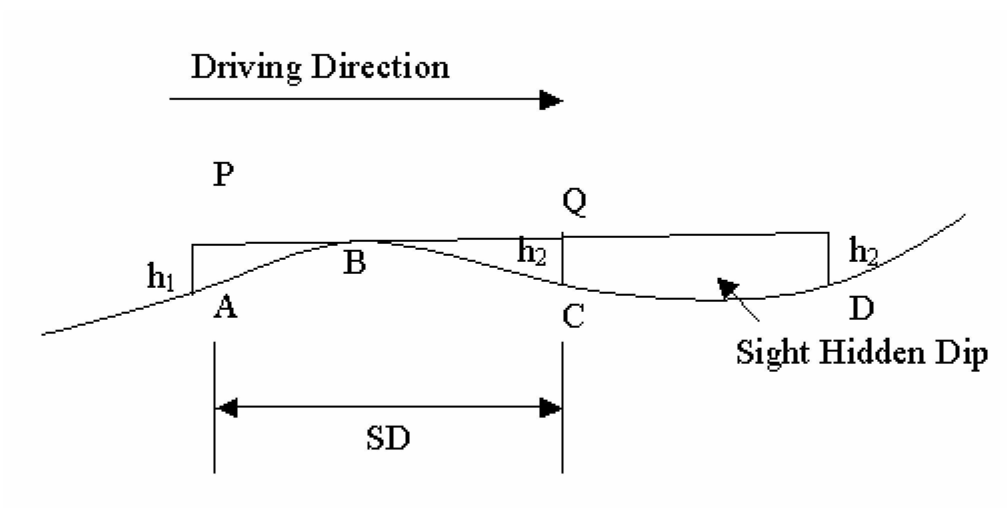

Figure 3.4: Sight Hidden Dip (Profile view)
(h_1, height of driver's eye; h_2, height of object; SD, sight distance)

At night-time, the available SSD on sag vertical curves is limited to the farthest point lighted by the vehicle's headlight. Fig 3.5 shows the graphical methodology followed for finding the sight distance limited by headlight on sag vertical curves. A simple procedure as given below is followed to find the sight distance due to headlight control.

1. Determine the coordinates of the point A at which sight distance is to be calculated. The coordinates of P are then calculated by adding the height of driver's eye.
2. Determine the point B along the path such that the line PB makes an angle equal to sum of the measures of the angle made by the tangent at A with the horizontal plane i.e. ϕ and the measure of the angle β made by the headlight in the upward direction. An iterative procedure is followed to get the location of B. The angle made by the line PB with the horizontal plane i.e. ϕ_2 is found and compared with the angle $(\phi+\beta)$ made at A with the horizontal plane. Here, some approximate means of determining whether the condition is met is necessary. For this purpose, the following approximation is used:

   $$(\phi_2 - (\phi+\beta)) < \delta \qquad (3.8)$$

   where δ is the tolerance.

   A sight line is thus established which is obstructed by the road surface itself.

   If no point of intersection is found till a distance of 1312 feet, then the sight distance is assumed as 1312 feet.
3. The sight distance is length AB along the road.

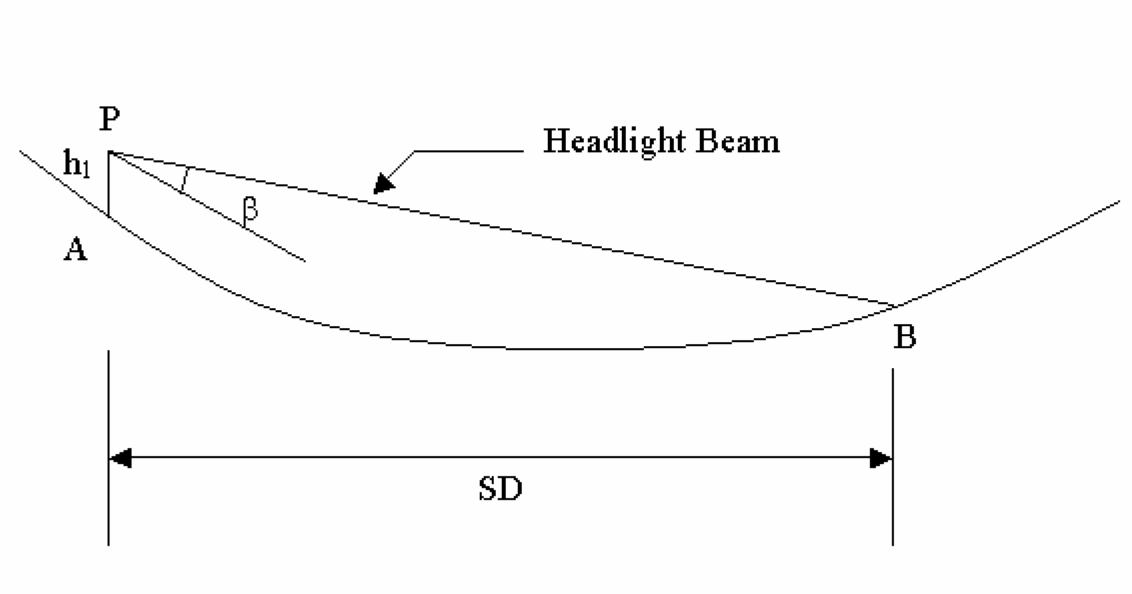

Figure 3.5: Sight Distance Limited by Headlight on Sag Vertical Curves (Profile view)
(h_1, height of vehicle's headlight; β, angle of headlight in upward direction; and SD, sight distance)

3.5 Discussion

Sight distance is calculated from vertical alignments at each point along the road. The distance between the points can be increased or decreased by varying the value of "t" in the equation 3.1 while determining the intermediate points.

Sight distance for vertical geometry is calculated using the above procedures for each point. The available sight distance at a point on the road is the lower bound of the sight distances obtained from vertical crest curve analysis and headlight sight distance

4. Application of Algorithm

4.1 Introduction

Computer software is developed for the stopping sight distance algorithm using the theoretical procedures stated in Chapter 3 in Matlab v6.1 and it can be seen in Appendix A. The algorithm was used to find stopping sight distances on several two-lane rural highways in

Kansas, USA. GPS data was made available by KDOT. KDOT takes GPS data runs for each highway in Kansas every year. Every highway had at least 2 data sets of GPS data.

4.2 Stopper

An instance when the required stopping sight distance is less than 1000 feet is called a stopper (KDOT). In order to calculate the number of stoppers along a highway, KDOT has taken data of the state grading plans. Stopping sight distance is calculated using equation 3.4 and 3.5 from the algebraic grade differences of the road curves. The stoppers data was originally collected state wide in the mid 1980's and then updated periodically. Stopper information is updated as new plans and major modifications are accomplished. The current method is manual calculations from design plans.

4.3 Testing SSD Algorithm

KDOT has used the AASHTO 1994 standards for calculations of stoppers. The only difference between the latest AASHTO 2001 standards and the AASHTO 1994 standards is the height of object. Height of object as per the AASHTO 2001 standards is 2 feet and that as per the AASHTO 1994 standards is 6 inches. Obviously, the AASHTO 2001 is more conservative in its calculations as the object height increases; the stopping sight distance becomes larger. The algorithm was tested for both, AASHTO 2001 standards as well as the AASHTO 1994 standards for several highways

4.3.1 Illustration for K-99 in Pottawatomie County

The algorithm is run as per the AASHTO 2001 standards on the highway K-99 in Pottawatomie County, Kansas. Table 4.1 gives the parameters used for the application example.

Table 4.1: Parameters Used for the Application Example - AASHTO 2001 Standards

Parameter	Value
Height of driver's eye	3.5 feet
Height of object	2 feet
Height of vehicle's headlight	2 feet
Angle of headlight beam	1°
Lane width	12 feet
Clear zone	30 feet
Stopper (SSD_{req})	1000 feet

Figure 4.1 shows the results of the run in metric units, while Figure 4.2 shows the results in US customary units. The speed limit was assumed as 60 mph, which is the speed limit on most of the two-lane rural highways in Kansas. The available SSD and the required SSD for a driver traveling in the direction of x-axis are shown. The x-axis is the length of the road in the direction of increasing mileage. The red points on the y-axis shows the available stopping sight distance while the blue points show the required stopping sight distance. The highway has all kinds of curves, namely crest vertical curves, sag vertical curves, horizontal curves and spans over a length of about 27 miles. The software was able to determine the profile of the available SSD due to all kinds of road curves.

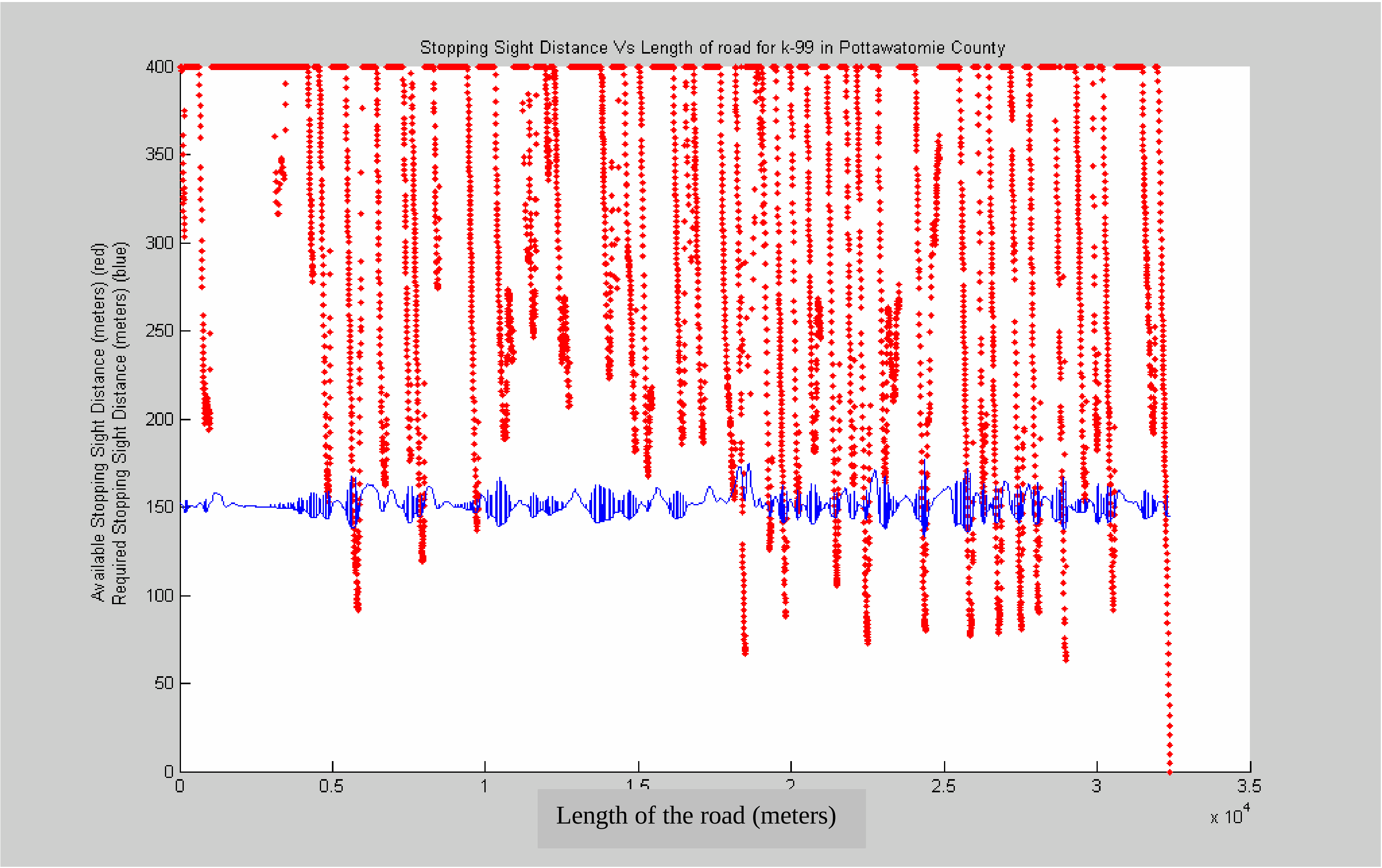

Figure 4.1: Stopping Sight Distance Vs Length of Road for K-99 in Pottawatomie County, Kansas, USA (Metric units).

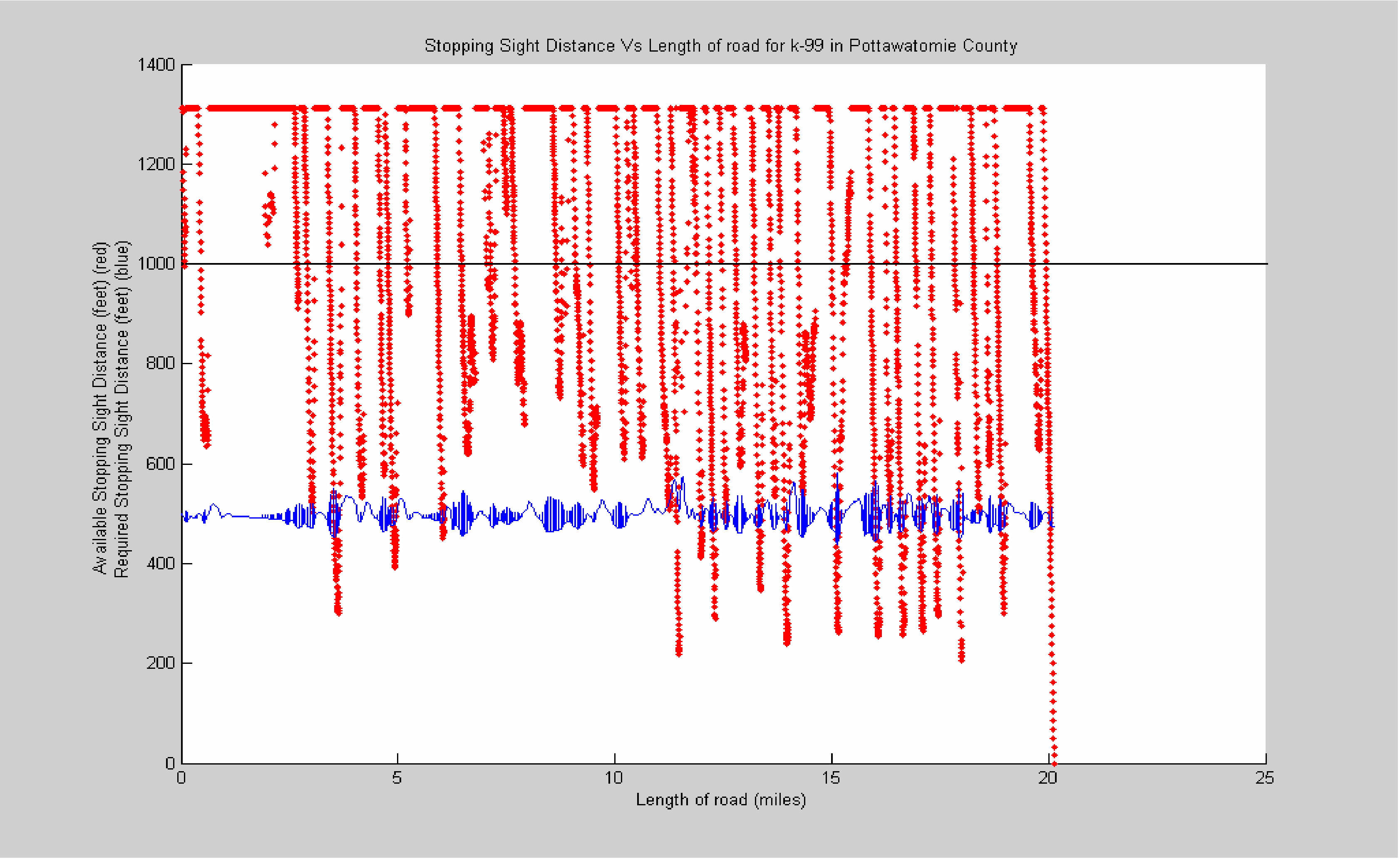

Figure 4.2: Stopping Sight Distance Vs Length of Road for K-99 in Pottawatomie County, Kansas, USA (US customary units).

Figure 4.2 shows the section (0 to 20 miles) of highway K-99 in Pottawatomie County, Kansas. A black line is drawn at 1000 ft on this figure for reference of stoppers. The red points, as previously depicted, are the available stopping sight distances, while the blue ones give the required stopping sight distance. When the red points go below the black line, there is a stopper. It can be seen from the figure that there are 38 instances of stoppers for this portion of the highway.

On every run of the algorithm, a database of the results is created. Table 4.3 shows a portion of the data created for highway K-99 in Pottawatomie county. Such a database shows the exact location of a point and the corresponding available and required stopping sight distance. The exact location of deficient sight distance is available from the database. Depending on the severity of difference between the required SSD and available SSD, corrective measures may be taken. Table 4.2 gives the details of each stopper.

Table 4.2: Snapshot of Final Database Generated for Highway K-99 in Pottawatomie County

Starting_position (feet) (Position of the stopper from the start)	**Location_ssd_min (feet) (Position of the lowest SSD_{avail} on that stopper from the start)**	**Intensity_ssd_min(feet) (Lowest SSD_{avail} on that stopper)**
1802.19	1802.19	825.10
5209.67	5527.52	816.52
7329.68	7386.38	124.13
12002.15	12002.15	929.89
12248.81	12248.81	579.09
14969.61	15476.18	622.72
17258.23	17702.76	705.19
18137.14	18233.08	619.23
18443.61	18443.61	987.67
18582.82	18615.89	221.79

Table 4.3: Snapshot of Database Generated for Highway K-99 in Pottawatomie County

X (feet)	Y (feet)	Z (feet)	Length from start (miles)	Theta (°)	Phi (°)	SSD_{reqd} (feet)	SSD_{avail} (feet)	Aindex
26306.38	-18262.13	959.32	134.52	-1.5525	-0.0020	495.52	1011.57	427
26306.38	-18262.13	959.32	134.52	-1.5525	-0.0020	495.52	1011.57	427
26306.68	-18278.17	959.28	134.53	-1.5518	-0.0025	495.89	995.53	427
26317.21	-18803.80	954.37	134.63	-1.5555	-0.0072	499.98	469.77	427
26317.39	-18815.69	954.29	134.63	-1.5556	-0.0064	499.26	457.88	427
26317.58	-18828.00	954.22	134.63	-1.5557	-0.0055	498.54	445.56	427
26317.77	-18840.78	954.15	134.63	-1.5559	-0.0047	497.82	432.79	427
26317.97	-18854.05	954.10	134.64	-1.5560	-0.0039	497.10	419.51	427
26318.17	-18867.84	954.05	134.64	-1.5561	-0.0030	496.40	405.72	427
26318.38	-18882.20	954.01	134.64	-1.5562	-0.0022	495.71	391.36	427
26318.60	-18897.14	953.98	134.64	-1.5563	-0.0014	495.03	376.41	427
26318.82	-18912.71	953.97	134.65	-1.5564	-0.0007	494.38	360.84	427
26290.47	-16465.19	957.10	132.15	-1.5691	0.0034	490.94	1312.32	427
26290.71	-16603.43	957.57	132.17	-1.5691	0.0034	490.96	1312.32	427
26290.92	-16721.25	957.97	132.19	-1.5689	0.0034	490.98	1312.32	427
26291.11	-16818.53	958.30	132.21	-1.5688	0.0034	491.00	1312.32	427
26291.27	-16895.12	958.56	132.23	-1.5685	0.0033	491.03	1312.32	427
26291.40	-16950.91	958.74	132.24	-1.5681	0.0033	491.07	1312.32	427
26291.52	-16985.76	958.86	132.24	-1.5667	0.0032	491.15	1312.32	427
26291.60	-16999.52	958.90	132.25	-1.5476	0.0019	492.22	1312.32	427
26291.66	-16992.08	958.87	132.25	1.5681	-0.0036	496.87	1312.32	427
26291.70	-16963.30	958.77	132.25	1.5702	-0.0034	496.74	1312.32	427

4.3.2 Testing the Algorithm on Several Highways

KDOT made available stoppers data on several highways in Kansas along with GPS data for those highways. The algorithm was run on those highways with the parameters specified in Table 4.1 and Table 4.4.

Table 4.4: Parameters Used for the Testing – AASHTO 1994 Standards

Parameter	Value
Height of driver's eye	3.5 feet
Height of object	2 feet
Height of vehicle's headlight	2 feet
Angle of headlight beam	1°
Lane width	12 feet
Clear zone	30 feet
Stopper (SSD_{req})	1000 feet

Table 4.1 shows the AASHTO 2001 standards, while Table 4.4 shows AASHTO 1994 standards. The results of running the algorithm based on both AASHTO standards on several two-lane rural highways are shown in Table 4.5. The algorithm is run for each of the highways and the number of stoppers is automatically counted.

Table 4.5: Results from SSD algorithm for Highways in Kansas, USA

County	Highway	Length (miles)	No. of stoppers (KDOT)	No. of stoppers (AASHTO 2001)	No. of stoppers (AASHTO 1994)	Difference (KDOT & AASHTO 1994)	Difference (KDOT & AASHTO 2001)	Difference (AASHTO 2001 & 1994)
75	99	27.1	57	56	60	3	-1	4
81	113	5.63	2	2	3	1	0	1
81	177	7.91	1	1	3	2	0	2
61	68	24.55	15	17	23	8	2	6
3	59	16.36	9	10	12	3	1	2
53	284	5.61	7	7	7	0	0	0
56	170	8.03	27	25	25	-2	-2	0
64	77	12.06	8	5	8	0	-3	3
65	56	21.87	6	4	5	-1	-2	1
70	75	31.11	15	11	12	-3	-4	1
80	14	25.1	10	5	9	-1	-5	4
52	32	17	31	29	30	-1	-2	1
104	54	25.82	7	6	7	0	-1	1
35	50	26	2	3	4	2	1	1
38	50	28.49	1	0	1	0	-1	1
43	75	30.74	13	8	15	2	-5	7
50	101	10	18	14	16	-2	-4	2
53	232	4.99	2	2	2	0	0	0

4.4 Modules of the Algorithm

The entire procedure is automated, right from the importing of the data from a database to transferring the results into two databases. The model consists of six modules and uses three databases. The first module imports the data from the 'control_points' database using the database toolbox. The 'control_points' database has a table called points1 that has the following information:

Table 4.6: Information on the Control_Points Database

Field Name	Data Type	Description
aindex	number	aindex number
fragment	number	fragment number
direction	number	1 - forward direction 2 - reverse direction
control_lon	number	longitude of the control point
control_lat	number	latitude of the control point
control_z	number	altitude of the control point
err_xy	number	horizontal error (in meters)
err_z	number	vertical error (in meters)
quality_xy	number	1 - horizontal possible bad control point 0 - good point
quality_z	number	1 - vertical possible bad control point 0 - good point
distance	number	distance from the first point (miles)

The first module establishes connection to the database and retrieves all the records of a highway segment from the database based on the a-index number. The connection is then closed and the records are then converted into matrix form for analysis.

The second module transforms the terrestrial coordinates into Cartesian coordinates using coordinate transformation. Lambert conformal projection with two standard parallels transformation method is used for coordinate transformation. The elevation is checked for outliers and those found are eliminated. Control points that are now in Cartesian coordinate system are used for analysis.

The third module performs the analysis. For the details on analysis refer to Appendix A. The road obtained through the GPS data is represented as a curve. Since cubic B-spline curves give a good fit, we use this curve to fit the road data into equations. We then use four consecutive points to generate a B-spline curve to fit these five points. We then calculate the sight distance at each point for vertical alignments. At each point we find the sight distance limited by crest vertical curve.

The fourth module performs the analysis for headlight sight distance. It is because in the night, the available SSD on sag vertical curves is limited to the farthest point lighted by the vehicle's headlight. In this module we are finding the sag vertical curves because at night the headlights limit the sight distance due to the sag.

In the fifth module the results of the analysis are written into a database. We establish a connection to the 'data' database and then transfer all the intermediate results into it. This database consists of the analysis at every control point. It consists of the following fields.

Table 4.7: Information on the Data Table of the Results Database

Field Name	Data Type	Description
X	number	distance of the point in the direction of x-axis (feet)
y	number	distance of the point in the direction of y-axis (feet)
z	number	distance of the point in the direction of z-axis (feet)
len	number	distance travelled from the starting point (miles)
theta	number	angle made by the tangent in the horizontal plane (radians)
phi	number	vertical elevation of the road with the horizontal (radians)
ssd	number	required stopping sight distance (feet)
sd	number	available stopping sight distance (feet)
aindex	number	aindex number

In the penultimate module we find the location of the points where the sight distance is below 1000 feet. From the previous analysis, we find the location of the stoppers, its starting point and the point where the sight distance is minimum. We also find the location of the minimum sight distance for each of these stoppers.

In the final module, the results are transferred into a database called ‘final’. The results are stored in the following format.

Table 4.8: Information of the Final Table of the Results Database

Field Name	Data Type	Description
Starting_position	number	Position of the stopper in from the start (feet)
Location_ssd_min	number	Position of the point having minimum sight distance from the start (feet)
Intensity_ssd_min(meters	number	The minimum sight distance at a stopper (feet)
Aindex	number	aindex number

4.5 Discussion of Results

Table 4.5 gives the results of running the algorithm based on both AASHTO standards on several two-lane rural highways. It can be seen from Table 4.5 for most of the highways, the stopper data from KDOT matches that to the algorithm when run with AASHTO 1994 standards. In most of the cases, the algorithm is seen to overestimate the number of stoppers, with the exception of a few. A closer look at the table shows suspect cases for Highway 68 in County 61 (Miami County) and Highway 75 in County 70 (Osage County).

4.6 Accidents Data

Accidents data was made available by KDOT for the roads on which stopping sight distance algorithm was run. There were 18 such roads that varied in the number of stoppers and length. Data was available for years 1996 through 2000, i.e. for 5 years for each of the 18 roads. It was intended to find if there is any correlation between the number of accidents on the roads and the number of stoppers.

The format for the accident's data is as shown in Table 4.9.

Table 4.9: Accident Data Format

Column	Short Description
Year	Accident year
Case #	Accident case number
Date	Accident date
Co	County
RCRP	Route County Reference Post
Dist	Distance
UOM	Units of Measurement
Dir	Direction of motion
Atrd	At road
Class	Accident Class
Cwov	Collision with other vehicle
Loc	Location
Vehs	Vehicles involved in accident
Fatl Accs	Fatal Accidents
Inj Accs	Injury Accidents
PDO	Property Damage Only
Deaths	Deaths in accident
Injd	Injured in accident
Time	Time of accident
Light	Light conditions
Weather	Weather Conditions

Table 4.10 shows the accident class codes used in the data set and the description of the accident class.

Table 4.10: Accident Class Format

Accident class code	Description
00	Other non-collision
01	Overturn
02	Collision With Pedestrian
03	Collision With Other Motor Vehicle
04	Collision With Parked Motor Vehicle
05	Collision With Railway Train
06	Collision With Pedalcycle
07	Collision With Animal
08	Collision With Fixed Object

Table 4.11 shows the weather conditions codes used in the data set and the description of the weather code.

Table 4.11: Accident Class Format

Weather Code	Description
01	Rain
02	Sleet
04	Fog
05	Smoke
06	Strong winds
07	Blowing dust
08	Freezing rain
14	Rain & Fog
16	Rain & Wind
24	Sleet & Fog
36	Snow & Wind
88	Other
99	Unknown

Table 4.12 shows the light conditions codes used in the data set and the short description of the light code.

Table 4.12: Light Conditions Codes

Light code	Description
01	Daylight
02	Dawn
03	Dusk
04	Dark; street lights on
05	Dark; no street lights
99	Unknown

KDOT data was compiled and was assimilated as shown in Table 4.13.

Table 4.13: Accident Data Format Used for Analysis

Column	Short Description
County	County
Route	Highway route
Year	Data year
Stoppers	Number of stoppers calculated by SSD algorithm using AASHTO 2001 standards
Accidents	Number of accidents
Collisions	Number of collision related accidents
Inj accd	Number of accidents causing injuries
Injrd	Number of people injured
Fatal accd	Number of fatal accidents
Deaths	Number of deaths
Daytime	Number of accidents caused during daytime
Normal weather	Number of accidents caused during normal weather conditions

Statistical correlation analysis was performed on the data compiled. Table 4.14 shows the Pearson’s correlation coefficients obtained in the SAS analysis. Appendix B shows the output file generated by SAS.

It can be seen that there was no significant correlation (0.09) between the number of stoppers and the number of accidents occurred on a highway. Also, since the correlation between collisions and accidents is 0.99, it is inferred that most of the accidents are collision related accidents involving another vehicle. Most of the accidents are injury related accidents and are happening during daytime and under normal weather conditions.

Table 4.14: Pearson's Correlation Coefficients Obtained for Accident's Data

Variable	**Stoppers**	**Accidents**	**Collisions**	**Injury related accidents**	**Number of Injured**	**Fatal accidents**	**Deaths**	**Daytime**	**Normal weather**
Stoppers	1.00	0.09	0.08	0.03	0.02	0.00	-0.02	0.03	0.10
Accidents		1.00	1.00	0.93	0.92	0.52	0.50	0.92	0.99
Collisions			1.00	0.93	0.92	0.51	0.50	0.92	0.99
Injury related accidents				1.00	0.97	0.53	0.51	0.89	0.93
Number of Injured					1.00	0.60	0.59	0.84	0.91
Fatal accidents						1.00	0.98	0.37	0.52
Deaths							1.00	0.36	0.51
Daytime								1.00	0.93
Normal weather									1.00

The results obtained from the above study are in agreement with the studies done in the past. Kahl and Fambro, 1995, found that more than 90 percent of the accidents occurred on flat roadways in conditions on which sight distance is not limited by the roadway's geometry. Some of the possible reasons for such a kind of result is that the driver finds a route more challenging for driving when it has sight distance deficiency and drives with more care and alertness, than on a straight flat roadway where he/she tends to dose off. On hilly terrains or on curvy paths, the driver tends to enjoy the ups and downs and the turns.

5. Conclusions

5.1 Conclusions

Sight distance is the most basic and critical element of highway design. It controls all aspects of design from the establishment of alignments to the development of the cross-section elements of the roadway.

Finding the stopping sight distance on actual roadways after years of maintenance and reconstruction has always been a challenge. Going back to design plans does not help as the roadway surface may not always comply to the designs. Surveying the actual road surface is a tremendous task and is time consuming. Gathering GPS data for such roads and then using this data for analysis for stopping sight distance is the solution. A methodology to find the available stopping sight distance and required stopping sight distance has been presented in this report. The algorithm developed takes into account the vertical highway alignments. The algorithm automatically calculates the sight distance profile along any given highway for which GPS data is available. The results of actual data show that the algorithm is able to identify sight distance restrictions.

The sight distance profile i.e. the plot of the required SSD and available SSD vs the length of the road, may be used to find places of deficient stopping sight distance along the highway. The profile gives the location of the deficiency and the available sight distance at that point. The computer software that has been developed can help highway designers and professionals locate the actual substandard sections. At times when financial resources become scarcer, the funds available for improvements can be effectively allocated to sections that need the improvement the most. Corrective measures may be then taken depending upon the severity. The remedial measures may include cutting vertical crest curves, filling up sag curves or building bridges on them, clearing up clear zones, or changing speed limits.

The analysis performed in this report is based on the assumptions that the GPS data collection vehicle moves along the center of the right lane at a steady pace; a constant bias may be introduced by the vehicle during GPS data collection, but is neglected as only the profile of the road is important for stopping sight distance calculations; and no banking effects are considered i.e. the road is assumed flat.

The objective of this project was to find the location of the stoppers and to calculate the minimum sight distance (intensity) at each of these stoppers based on the vertical geometry for all the highways in Kansas. The analysis has been done on 613 highways.

The existing routes have been refined so that the highway system can be processed automatically. The current GPS database is based on GPS attributes while the GPS highway model uses a B-Spline fitting approach. Appropriate data structures were developed to preserve the highway model in an efficient manner with high accessibility to the users. The final results shows each highway segment, sorted by aindex number, giving the location of the stoppers or the segments that have SSD_{avail} less than 1,000 feet, the lowest SSD_{avail} and its location.

The entire procedure was automated, right from the importing of the data from the GPS database to transferring the results into the two tables of the results database.

5.2 Scope for Future Research

The software developed calculates stopping sight distances along the roadway for vertical alignments. The algorithm can be extended to determine the available sight distance on 3-D combined horizontal and vertical alignments. A need for developing software for determining passing and no-passing zones has been long in waiting. It is reasonable to assume that accidents due to stopping sight distance restrictions occur within the SSD-deficient road segments. Consequently, it might be worthwhile to conduct further research to record accidents occurring within SSD-deficient segments on existing crest/sag/horizontal curves. As the sight distance has been thoroughly researched, more research should be directed at 3D analysis of other design bases – namely, vehicle stability, driver comfort, drainage and aesthetics.

APPENDIX A

PROGRAMS LISTING FOR SSD ALGORITHM

Program for SSD Calculations using AASHTO 2001 standards

for ai=1:613

% module 1

```
logintimeout(5)
conn = database('stopper', '','')
ping(conn)
curs=exec(conn,['select * from control_points where aindex=',num2str(ai), 'and direction =', num2str(1)])
curs=fetch(curs)
data3 = curs.Data
save data3.mat data3
close(curs)
close(conn)
```

% module 2

```
data(:,1) = [data3{:,1}]'
data(:,2) = [data3{:,2}]'
data(:,3) = [data3{:,3}]'
data(:,4) = [data3{:,4}]'
data(:,5) = [data3{:,5}]'
data(:,6) = [data3{:,6}]'
data(:,7) = [data3{:,7}]'
data(:,8) = [data3{:,8}]'
data(:,9) = [data3{:,9}]'
data(:,10) = [data3{:,10}]'
data(:,11) = [data3{:,11}]'

p=[];
p=sortrows(data,4)

save_lon = p(:,4)
save_lat = p(:,5)
save_z   = p(:,6)

nu=size(save_lat(:,1))
lon0 = save_lon(1,1);
lat1 = save_lat(1,1)-0.1;
lat2 = save_lat(nu(1,1),1)+0.1;
if lat1 == lat2
   lat2=lat2-0.1;
end
size_con=size(save_lon,1);
x=zeros(size_con,1);
y=zeros(size_con,1);
z=save_z;
control_data = [];
```

```
for i=1:size_con
    [x(i),y(i)]=lambertterr2cart1(lon0, lat1,lat2,save_lon(i),save_lat(i));
    control_data = [control_data;x(i),y(i),save_z(i)];
end

save control_data.mat control_data;


%function ssd
%----------------------INPUT PARAMETERS------------------------
fr = 0.28
a = 3.4
V = 55*1.609344
T = 2.5

driver_ht = 1.080
object_ht = 0.60
headlight_ht = 0.60
tol = 0.001

lane_width = 3.6
clear_zone = 9.1

load control_data.mat;
xxx=control_data(:,1);
yyy=control_data(:,2);
zzz=control_data(:,3);

%----------------------INPUT PARAMETERS------------------------

% ssd = 0.278*V*T + V^2 / (254*(a/9.81 + G ))
% Here ssd = stopping sight distance
% V = velocity in km/hr
% a = deceleration rate m/s2
% fr = friction between the tire and the road surface
% G = Grade of the barking section + for upgrade and - for downgrade
% T = perception reaction time (seconds)

% Units Convertion
% 1 mile = 1609.344 m

% open the data file
% load data
% generate points
% find theta and phi for each point
% find the length of the curve from the initial point
% calculate ssd also at this point

d_left = (lane_width*1.5) + clear_zone
d_right = (lane_width*0.5) + clear_zone

A=[-1 3 -3 1;3 -6 3 0;-3 0 3 0;1 4 1 0;]
size_data = size(xxx,1);
count=1;
for i=2:size_data-2
    for u=0:0.05:1
```

```
        X(count) = 1/6*[u^3 u^2 u 1]*A*[xxx(i-1);xxx(i);xxx(i+1);xxx(i+2)];
        Y(count) = 1/6*[u^3 u^2 u 1]*A*[yyy(i-1);yyy(i);yyy(i+1);yyy(i+2)];
        Z(count) = 1/6*[u^3 u^2 u 1]*A*[zzz(i-1);zzz(i);zzz(i+1);zzz(i+2)];
        if (count-1) == 0
          len(count) = 0;
        else
          len(count) = len(count-1) + sqrt((X(count)-X(count-1))^2 + (Y(count)-Y(count-1))^2 + (Z(count)-
Z(count-1))^2 );
        end
        % write the equation of the curve
        % write the equation of the derivatives
        dx = 1/6*[3*u^2 2*u 1 0]*A*[xxx(i-1);xxx(i);xxx(i+1);xxx(i+2)];
        dy = 1/6*[3*u^2 2*u 1 0]*A*[yyy(i-1);yyy(i);yyy(i+1);yyy(i+2)];
        dz = 1/6*[3*u^2 2*u 1 0]*A*[zzz(i-1);zzz(i);zzz(i+1);zzz(i+2)];
        [theta(count),phi(count),r]=cart2sph(dx,dy,dz);

        % find the required stopping sight distance
        if (count-1) == 0
          var_sign = 1;
        elseif Z(count) > Z(count-1)
          var_sign = 1;
        else
          var_sign = -1;
        end
        G = var_sign*abs(tan(phi(count)));
        ssd(count) = 0.278*V*T + V^2 / (254*((3.4/9.81) + G ));
        count = count + 1;
      end
  end
  count = count - 1
  t_max = count;
```

% module 3

```
  %script file to calculate the SSD for a vertical curve

  % Calculate the sight distance for points at an increment of t_incr
  % This loop is for calculating SSD using the elevation
  current_t = 0;
  for current_t = 1:t_max
    % (xp,yp,zp) is the point under consideration for which sd is to be calculated
     xp = X(current_t);
     yp = Y(current_t);
     zp = Z(current_t);

     zpd = zp + driver_ht;

    inner_current_t = current_t;

    continue1 = 1;

    while continue1    % the condition is true
        inner_current_t = inner_current_t + 1;

        if  inner_current_t > t_max | (len(inner_current_t)-len(current_t))>400    % greater than the length of the
curve
```

```
            continue1 = 0;
            sdtype(current_t) = 1;
            if inner_current_t > t_max
               sd(current_t) = len(t_max) - len(current_t);
            else
               sd(current_t) = 400;
            end
         else
            % (xp,yp,zp) is the point under consideration for which sd is to be calculated
            % (x_current,y_current,z_current) is the current point
            x_current = X(inner_current_t);
            y_current = Y(inner_current_t);
            z_current = Z(inner_current_t);

            % find the slope of the line joining (xp,yp,zpd) and (x_current,y_current,z_current)
            [theta2,phi2,r2]=cart2sph((x_current-xp),(y_current-yp),(z_current-zpd));
            if abs(phi2-phi(inner_current_t)) < tol
               continue1 = 0;
               outer_current_t = inner_current_t;
               continue2 = 1;
               while continue2
                  outer_current_t = outer_current_t + 1;
                  if  outer_current_t > t_max | (len(outer_current_t)-len(current_t))>400    % greater than the length of
the curve
                     continue2 = 0;
                     sdtype(current_t) = 1;
                     if outer_current_t > t_max
                        sd(current_t) = len(t_max) - len(current_t);
                     else
                        sd(current_t) = 400;
                     end
                  else
                     xpo = X(outer_current_t);
                     ypo = Y(outer_current_t);
                     zpo = Z(outer_current_t) + object_ht;
                     % find the slope of the line joining (xpr,ypr,zpr) and (xpo,ypo,zpo)
                     [theta3,phi3,r3]=cart2sph((xpo-x_current),(ypo-y_current),(zpo-z_current));
                     if abs(phi3-phi(inner_current_t)) < tol
                        continue2 = 0;
                        sdtype(current_t) = 0;
                        sd(current_t) = len(outer_current_t) - len(current_t);
                     end
                  end
               end
            end
         end
      end
   end
```

% module 4

```
   % second loop for headlight sight distance

   current_t = 0;
   for current_t = 1:t_max
      % (xp,yp,zp) is the point under consideration for which sd is to be calculated
```

```
      xp = X(current_t);
      yp = Y(current_t);
      zp = Z(current_t);

      zpd = zp + headlight_ht;

      inner_current_t = current_t;

      continue1 = 1;

      while continue1    % the condition is true
          inner_current_t = inner_current_t + 1;

          if  inner_current_t > t_max | (len(inner_current_t)-len(current_t))>400    % greater than the length of the
curve
              continue1 = 0;
              sdtype(current_t) = 1;
              if inner_current_t > t_max
                  sd(current_t) = min(sd(current_t),len(t_max) - len(current_t));
              else
                  sd(current_t) = min(sd(current_t),400);
              end
          else
              % (xp,yp,zp) is the point under consideration for which sd is to be calculated
              % (x_current,y_current,z_current) is the current point
              x_current = X(inner_current_t);
              y_current = Y(inner_current_t);
              z_current = Z(inner_current_t);

              % find the slope of the line joining (xp,yp,zpd) and (x_current,y_current,z_current)
              [theta2,phi2,r2]=cart2sph((x_current-xp),(y_current-yp),(z_current-zpd));
              if abs(phi2-(phi(inner_current_t)+pi/180)) < tol
                  continue1 = 0;
                  sdtype(current_t) = 1;
                  sd(current_t) = min(sd(current_t),len(inner_current_t) - len(current_t));
              end
          end
      end
  end


  %
  %
  % % Calculate the sight distance for points at an increment of t_incr
  % % This loop is for calculating SSD using the horizontal geometry
  % % Loop to calculate the SSD using the right hand side
  %
  for current_t = 1:t_max
      % (xp,yp,zp) is the point under consideration for which sd is to be calculated
      xp = X(current_t);
      yp = Y(current_t);
      zp = Z(current_t);

      inner_current_t = current_t;

      continue1 = 1;
```

```
while continue1    % the condition is true
    inner_current_t = inner_current_t + 1;

    if  inner_current_t > t_max | (len(inner_current_t)-len(current_t))>400    % greater than the length of the
curve
        continue1 = 0;
        sdtype(current_t) = 1;
        if inner_current_t > t_max
            sd(current_t) = min(sd(current_t),len(t_max) - len(current_t));
        else
            sd(current_t) = min(sd(current_t),400);
        end
    else
        % (xp,yp,zp) is the point under consideration for which sd is to be calculated
        % (x_current,y_current,z_current) is the current point
        x_current = X(inner_current_t);
        y_current =  Y(inner_current_t);
        z_current = Z(inner_current_t);

        eata = theta(inner_current_t);
        if (eata >= 0) & (eata <= pi/2)
            xpr = x_current + d_right*cos((pi/2) - eata);
            ypr = y_current - d_right*sin((pi/2) - eata);
            zpr = z_current;
            xpl = x_current - d_left*cos((pi/2) - eata);
            ypl = y_current + d_left*sin((pi/2) - eata);
            zpl = z_current;
        elseif (eata > pi/2) & (eata <= pi)
            xpr = x_current + d_right*cos(eata - (pi/2));
            ypr = y_current + d_right*sin(eata - (pi/2));
            zpr = z_current;
            xpl = x_current - d_left*cos(eata - (pi/2));
            ypl = y_current - d_left*sin(eata - (pi/2));
            zpl = z_current;
        elseif (eata > pi) & (eata <= 3*pi/2)
            xpr = x_current - d_right*cos((3*pi/2) - eata);
            ypr = y_current + d_right*sin((3*pi/2) - eata);
            zpr = z_current;
            xpl = x_current + d_left*cos((3*pi/2) - eata);
            ypl = y_current - d_left*sin((3*pi/2) - eata);
            zpl = z_current;
        else
            xpr = x_current - d_right*cos(eata - (3*pi/2));
            ypr = y_current - d_right*sin(eata - (3*pi/2));
            zpr = z_current;
            xpl = x_current + d_left*cos(eata - (3*pi/2));
            ypl = y_current + d_left*sin(eata - (3*pi/2));
            zpl = z_current;
        end

        % find the slope of the line joining (xp,yp,zp) and (xpr,ypr,zpr)
        [theta2,phi2,r2]=cart2sph((xpr-xp),(ypr-yp),(zpr-zp));
        if abs(theta2-theta(inner_current_t)) < tol
            continue1 = 0;
            outer_current_t = inner_current_t;
```

```
                continue2 = 1;
                while continue2
                   outer_current_t = outer_current_t + 1;
                   if  outer_current_t > t_max | (len(outer_current_t)-len(current_t))>400    % greater than the length of
the curve
                      continue2 = 0;
                      sdtype(current_t) = 1;
                      if outer_current_t > t_max
                         sd(current_t) = min(sd(current_t),len(t_max) - len(current_t));
                      else
                         sd(current_t) = min(sd(current_t),400);
                      end
                   else
                      xpo = X(outer_current_t);
                      ypo = Y(outer_current_t);
                      zpo = Z(outer_current_t);
                      % find the slope of the line joining (xpr,ypr,zpr) and (xpo,ypo,zpo)
                      [theta3,phi3,r3]=cart2sph((xpo-xpr),(ypo-ypr),(zpo-zpr));
                      if abs(theta3-theta(inner_current_t)) < tol
                         continue2 = 0;
                         sdtype(current_t) = 0;
                         sd(current_t) = min(sd(current_t),len(outer_current_t) - len(current_t));
                      end
                   end
                end
             end
          end
       end
    end

    % Loop to calculate the SSD using the left hand side
    for current_t = 1:t_max
       % (xp,yp,zp) is the point under consideration for which sd is to be calculated
       xp = X(current_t);
       yp = Y(current_t);
       zp = Z(current_t);

       inner_current_t = current_t;

       continue1 = 1;

       while continue1    % the condition is true
          inner_current_t = inner_current_t + 1;

          if  inner_current_t > t_max | (len(inner_current_t)-len(current_t))>400    % greater than the length of the
curve
             continue1 = 0;
             sdtype(current_t) = 1;
             if inner_current_t > t_max
                sd(current_t) = min(sd(current_t),len(t_max) - len(current_t));
             else
                sd(current_t) = min(sd(current_t),400);
             end
          else
             % (xp,yp,zp) is the point under consideration for which sd is to be calculated
             % (x_current,y_current,z_current) is the current point
```

```
x_current = X(inner_current_t);
y_current = Y(inner_current_t);
z_current = Z(inner_current_t);

eata = theta(inner_current_t);
if (eata >= 0) & (eata <= pi/2)
    xpr = x_current + d_right*cos((pi/2) - eata);
    ypr = y_current - d_right*sin((pi/2) - eata);
    zpr = z_current;
    xpl = x_current - d_left*cos((pi/2) - eata);
    ypl = y_current + d_left*sin((pi/2) - eata);
    zpl = z_current;
elseif (eata > pi/2) & (eata <= pi)
    xpr = x_current + d_right*cos(eata - (pi/2));
    ypr = y_current + d_right*sin(eata - (pi/2));
    zpr = z_current;
    xpl = x_current - d_left*cos(eata - (pi/2));
    ypl = y_current - d_left*sin(eata - (pi/2));
    zpl = z_current;
elseif (eata > pi) & (eata <= 3*pi/2)
    xpr = x_current - d_right*cos((3*pi/2) - eata);
    ypr = y_current + d_right*sin((3*pi/2) - eata);
    zpr = z_current;
    xpl = x_current + d_left*cos((3*pi/2) - eata);
    ypl = y_current - d_left*sin((3*pi/2) - eata);
    zpl = z_current;
else
    xpr = x_current - d_right*cos(eata - (3*pi/2));
    ypr = y_current - d_right*sin(eata - (3*pi/2));
    zpr = z_current;
    xpl = x_current + d_left*cos(eata - (3*pi/2));
    ypl = y_current + d_left*sin(eata - (3*pi/2));
    zpl = z_current;
end

% find the slope of the line joining (xp,yp,zp) and (xpl,ypl,zpl)
[theta2,phi2,r2]=cart2sph((xpl-xp),(ypl-yp),(zpl-zp));
if abs(theta2-theta(inner_current_t)) < tol
    continue1 = 0;
    outer_current_t = inner_current_t;
    continue2 = 1;
    while continue2
        outer_current_t = outer_current_t + 1;
        if  outer_current_t > t_max | (len(outer_current_t)-len(current_t))>400    % greater than the length of 
the curve
            continue2 = 0;
            sdtype(current_t) = 1;
            if outer_current_t > t_max
                sd(current_t) = min(sd(current_t),len(t_max) - len(current_t));
            else
                sd(current_t) = min(sd(current_t),400);
            end
        else
            xpo = X(outer_current_t);
            ypo = Y(outer_current_t);
            zpo = Z(outer_current_t);
```

```
                % find the slope of the line joining (xpl,ypl,zpl) and (xpo,ypo,zpo)
                [theta3,phi3,r3]=cart2sph((xpo-xpl),(ypo-ypl),(zpo-zpl));
                if abs(theta3-theta(inner_current_t)) < tol
                   continue2 = 0;
                   sdtype(current_t) = 0;
                   sd(current_t) = min(sd(current_t),len(outer_current_t) - len(current_t));
                end
             end
          end
       end
    end
  end
end
```

% module 5

```
data1 = [];
data2 = [];
for t = 1:t_max
   data1 = [data1;X(t),Y(t),Z(t),len(t),theta(t),phi(t),ssd(t),sd(t),ai];
end

for t = 1:t_max
   xa(t)=X(t)*3.2808;
   ya(t)=Y(t)*3.2808;
   za(t)=Z(t)*3.2808;
   lena(t)=len(t)*0.00062;
   ssda(t)=ssd(t)*3.2808;
   sda(t)=sd(t)*3.2808;
end
for t = 1:t_max
   data2 = [data2;xa(t),ya(t),za(t),lena(t),theta(t),phi(t),ssda(t),sda(t),ai];
end
timeoutA=logintimeout(5)
connA=database('results','','')
ping(connA)
get(connA,'AutoCommit')
colnames={'x','y','z','len','theta','phi','ssd','sd','aindex'};
insert(connA,'data',colnames, data2)
close(connA)
```

% module 6

```
dist = data1(:,4)
sd = data1(:,8)
size_data = size(sd,1)
fordiscard = sd(:,1)<400;
i=1;
while i<size_data
   while i<=size_data & fordiscard(i)==0
      i=i+1;
   end
   discard = 0;
   while i<=size_data & fordiscard(i)==1
      i=i+1;
      discard = discard+1;
```

```
    end
end
forstoppers = sd(:,1)<304.8;
i=1;
num_stoppers = -1;
while i<(size_data-discard)
    while i<=size_data & forstoppers(i)==0
        i=i+1;
    end
    while i<=size_data & forstoppers(i)==1
        i=i+1;
    end
    num_stoppers = num_stoppers+1;
end
display(discard);
display(num_stoppers);

i=1;
t=0;
if num_stoppers~=0
    length=zeros([num_stoppers,2]);
    while i<(size_data-discard)
        while sd(i)>=304.8 & i<(size_data-discard)
            flag=1;
            i=i+1;
        end
        if flag~=0
            if sd(i)<304.8
                t=t+1;
                flag=0;
                length(t,1) = data1(i,4);
                length(t,2) = data1(i,4);
                sd_lowest(t)=sd(i);
                loc(t)=i;
            end
        end
        if sd_lowest(t)>sd(i)
            length(t,2) = data1(i,4);
            sd_lowest(t)=sd(i);
            loc(t) = i;
        end
        i=i+1;
    end
    st=length(:,1);
    de=length(:,2);
    data3 = [];
    data4 = [];
    for t = 1:t
        data3 = [data3;st(t,1),de(t,1),sd_lowest(1,t),ai];
    end
    for t=1:t
        sta(t)= st(t,1)*3.2808;
        dea(t)= de(t,1)*3.2808;
        sd_lowesta(t) = sd_lowest(1,t)*3.2808;
    end
    for t = 1:t
```

```
        data4 = [data4;sta(t),dea(t),sd_lowesta(1),ai];
    end
```

% module 7

```
    timeoutA=logintimeout(5)
    connA=database('results','','')
    ping(connA)
    get(connA,'AutoCommit')
    colnames={'start','deepest ','depth','aindex'};
    insert(connA,'final',colnames, data4)
    close(connA)
  end

end
```

APPENDIX B

PROGRAM OUTPUT LISTING FROM SAS

```
                              The SAS System
                                                                          1
                                   12:10 Thursday, April 11, 2002

                           Correlation Analysis

9 'VAR' Variables:  STOPS    ACCDS    COLLS    I_ACCD   INJRD    F_ACCD
                       DEATHS   DAYTIME  N_WEAT


                            Simple Statistics

Variable            N        Mean      Std Dev          Sum      Minimum
Maximum

STOPS              89    11.49438     13.45964         1023            0
56.00000
ACCDS              89    38.50562     27.90533         3427      1.00000
109.00000
COLLS              89    36.17978     26.88467         3220      1.00000
104.00000
I_ACCD             89     8.97753      7.90710    799.00000            0
37.00000
INJRD              89    14.92135     14.39557         1328            0
71.00000
F_ACCD             89     0.53933      0.89260     48.00000            0
5.00000
DEATHS             89     0.58427      0.96306     52.00000            0
5.00000
DAYTIME            89    18.26966     13.84719         1626            0
58.00000
N_WEAT             89    32.13483     23.10362         2860      1.00000
91.00000
```

Correlation Analysis

Pearson Correlation Coefficients / Prob > |R| under Ho: Rho=0 / N = 89

	STOPS	ACCDS	COLLS	I_ACCD	INJRD
STOPS	1.00000 0.0	0.09200 0.3912	0.07873 0.4633	0.02541 0.8131	0.02267 0.8330
ACCDS	0.09200 0.3912	1.00000 0.0	0.99726 0.0001	0.93093 0.0001	0.91753 0.0001
COLLS	0.07873 0.4633	0.99726 0.0001	1.00000 0.0	0.92935 0.0001	0.91666 0.0001
I_ACCD	0.02541 0.8131	0.93093 0.0001	0.92935 0.0001	1.00000 0.0	0.97175 0.0001
INJRD	0.02267 0.8330	0.91753 0.0001	0.91666 0.0001	0.97175 0.0001	1.00000 0.0
F_ACCD	-0.00164 0.9879	0.51769 0.0001	0.51018 0.0001	0.53306 0.0001	0.60471 0.0001
DEATHS	-0.02341 0.8276	0.50348 0.0001	0.49712 0.0001	0.51061 0.0001	0.58941 0.0001
DAYTIME	0.02690 0.8024	0.91897 0.0001	0.92382 0.0001	0.88971 0.0001	0.83913 0.0001
N_WEAT	0.10324 0.3357	0.99445 0.0001	0.99274 0.0001	0.92854 0.0001	0.91376 0.0001

Correlation Analysis

Pearson Correlation Coefficients / Prob > |R| under Ho: Rho=0 / N = 89

	F_ACCD	DEATHS	DAYTIME	N_WEAT
STOPS	-0.00164	-0.02341	0.02690	0.10324
	0.9879	0.8276	0.8024	0.3357
ACCDS	0.51769	0.50348	0.91897	0.99445
	0.0001	0.0001	0.0001	0.0001
COLLS	0.51018	0.49712	0.92382	0.99274
	0.0001	0.0001	0.0001	0.0001
I_ACCD	0.53306	0.51061	0.88971	0.92854
	0.0001	0.0001	0.0001	0.0001
INJRD	0.60471	0.58941	0.83913	0.91376
	0.0001	0.0001	0.0001	0.0001
F_ACCD	1.00000	0.97764	0.37057	0.51882
	0.0	0.0001	0.0004	0.0001
DEATHS	0.97764	1.00000	0.35532	0.50663
	0.0001	0.0	0.0006	0.0001
DAYTIME	0.37057	0.35532	1.00000	0.92735
	0.0004	0.0006	0.0	0.0001
N_WEAT	0.51882	0.50663	0.92735	1.00000
	0.0001	0.0001	0.0001	0.0

REFERENCES

1. AASHTO. 2001; “A Policy on Geometric Design of highways and streets”, American Association of State highway and transportation officials.
2. AASHTO. 1994; “A Policy on Geometric Design of highways and streets”, American Association of State highway and transportation officials.
3. AASHTO. 1990; “A Policy on Geometric Design of highways and streets”, American Association of State highway and transportation officials.
4. AASHTO. 1989; “Roadside Design Guide”, American Association of State highway and transportation officials.
5. Easa, S.M. (1991). “Lateral clearance to vision obstacles on horizontal curves.” Transportation Research Record 1303, National Research Council, Washington, D.C. pp. 22-32.
6. Easa, S.M. (1991). “Sight distance model for unsymmetrical crest curves.” Transportation Research Record 1303, National Research Council, Washington, D.C. pp. 39-50.
7. Easa, S.M. (1994). “Design considerations for highway sight-hidden dips.” Journal of Transportation Research 28A(1), pp. 17-29.
8. Easa S. M. (1998). ‘‘Three-arc vertical curve for constrained highway alignments.’’ Journal of Transportation Engineering, ASCE, 124(2), 163–171.
9. Easa, S. M., Abd El Halim, A. O., and Hassan, Y. (1996). ‘‘Sight distance evaluation on complex highway vertical alignments.’’ Canadian Journal of Civil Engineering, Ottawa, 23, 577–586.
10. Easa, S. M., and Hassan, Y. (1997). ‘‘Headlight sight distance on separate highway alignments: a new approach’’ Canadian Journal of Civil Engineering, 24(6), 1007–1018.
11. Fambro, D. B., Fitzpatrick, K., and Koppa, R. (1997). “Determination of stopping sight distance” National Cooperative Highway Research Program Report 400, National Research Coouncil, Washington, D.C.
12. Johansson, G., and K. Rumar. (1971) “Drivers’ brake reaction times” Human Factors, Vol. 13, No. 1, Feb 1971: 23-27.

13. Hall, J.W., and Turner, D.S. (1989). “Stopping sight distance: can we see where we now stand?” Transportation Research Record 1208, National Research Council, Washington, D.C., pp 4-13.
14. Harwood, D. W., Fambro, D. B., Fishburn, B., Joubert, H., Lamm, R., and Psarianos, B. (1995). ‘‘International sight distance design practices.’’ Procedures, International Symposium on Highway Geometric Design Practices.
15. Hassan, Y., Easa, S. M., and Abd El Halim, A. O. (1996). ‘‘Analytical model for sight distance analysis on three-dimensional highway alignments. ’’ Transportation Research Record 1523, Transportation Research Board, Washington, D.C., 1–10.
16. Hassan, Y., Easa, S. M., and Abd El Halim, A. O. (1997). ‘‘Automation of determining passing and no-passing zones on two-lane highways.’’ Canadian Journal of Civil Engineering, Ottawa, 24, 263–275.
17. Hassan, Y., Easa, S. M., and Abd El Halim, A. O. (1997). ‘‘Design considerations for combined highway alignments.’’ Journal of Transportation Engineering, ASCE, 123(1), 60–68.
18. Hassan, Y., Easa, S. M., and Abd El Halim, A. O. (1995). ‘‘Sight distance on horizontal alignments with continuous lateral obstructions.’’ Transportation Research Record 1500, Transportation Research Board, Washington, D.C., 31–42.
19. Hassan, Y., Easa, S. M., and Abd El Halim, A. O. (1998). ‘‘State of the art three dimensional highway geometric design’’ Canadian Journal of Civil Engineering, Ottawa, 24, 500-511.
20. Hooper, K.G. and H. W. McGee. “Driver Perception Reaction Time: Are revisions to current specification values in order?” Transportation Research Record 904, Transportation Research Board, Washington, D.C., 21–30.
21. Kahl, K. B., and Fambro, D. B. (1995). ‘‘Investigation of object-related accidents affecting stopping sight distances.’’ Transportation Research Record 1500, Transportation Research Board, Washington, D.C., 25–30.
22. Lovell, D. J (1999). “Automated calculation of sight distance from horizontal geometry” Journal of Transportation Engineering 125(4), 297-304.

23. Mak, K. K., Bligh, R. P., and Ross, H. E., Jr. (1995). ‘‘Clear zone requirements for suburban highways.’’ Transportation Research Record 1500, Transportation Research Board, Washington, D.C., 119–126.
24. McLean, J.R. (1988). “Speeds, Friction Factors and Alignment Design Standards” Research Report ARR No. 154, Australian Road Research Board, Victoria, Australia.
25. Neuman, T.R. (1989). “New approach to design for stopping sight distance” Transportation Research Record 1208, National Research Council, Washington, D.C., 14-22.
26. Neuman, T.R., and Glennon, J.C. (1984). “Cost effectiveness of improvements to stopping sight distance.” Transportation Research Record 923, National Research Council, Washington, D.C., 26-34.
27. Normann, O. K. (1953) “Braking distances of vehicles from high speeds” Proceddings HRB, Vol. 22, Highway Research Board, 1953: 421-436.
28. Olson, P.L., Cleveland, D.E., Fancher, P.S., Koystyniuk, L.P. and Schneider, L.W. (1984) “Parameters affecting stopping sight distance”. National Cooperative Highway Research Program Report 270, National Research Council, Washington, D.C.
29. Sanchez, E. (1994). ‘‘Three-dimensional analysis of sight distance on interchange connectors.’’ Transportation Research Record 1445, Transportation Research Board, Washington, D.C., 101–108.
30. Stroud, K.A. (2001) “Engineering Mathematics”.
31. Taiganidis, I. (1998). ‘‘Aspects of stopping sight distance on crest vertical curves.’’ Journal of Transportation Engineering, ASCE, 124(4), 335–342.
32. Taiganidis, I., and Kanellaidis G. (2001). ‘‘Required stopping sight distance on crest vertical curves.’’ Journal of Transportation Engineering, ASCE, 127(4), 275–282.
33. Trafikleder Pa Landsbygd (1986). National Swedish Road Administration, Borlange, Sweden.
34. Turner, D. S. (1987). ‘‘A primer on the clear zone.’’ Transportation Research Record 1122, Transportation Research Board, Washington, D.C., 86–95.
35. http://www.garmin.com
36. http://www.cee.hw.ac.uk/~ian/hyper00/curvesurf/bspline.html)