

Road Characterization for the Simulation of Automotive Vehicle Dynamics

2011-01-0185

Published
04/12/2011Michael W. Sayers
Mechanical Simulation Corp.

Copyright © 2011 SAE International

doi:10.4271/2011-01-0185

ABSTRACT

The major actions that move a highway vehicle are the forces and moments generated between the tire and ground; hence, the validity of a simulated vehicle test depends on the quality of both the tire model and the characterization of the ground surface. Other actions come from aerodynamic forces and moments that are affected by the relation of the vehicle body to the ground surface. This paper describes how the ground can be characterized to cover features of interest for most vehicle simulation scenarios involving pavements or other rigid surfaces. The 3D surface is built from tabular data related to specified properties of a road surface such as horizontal geometry, design elevation changes related to curves and drainage (i.e., banking of turns, cross-slope, ditches, etc.), elevation changes due to hills and other major grades, and disturbances and unique features such as bumps and holes. Broadband random-type road roughness is also included. The road model is intended to work with data from many sources, including GPS measurements, design data, road roughness profile measurements, 3D laser-scanned terrain topology, and specific scenarios created by engineers. A road axis system is defined for describing vehicle motions relative to an inclined road surface for aerodynamics and applications where engineers are concerned with motions of the vehicle relative to the ground surface. The methods described in this paper are demonstrated using the commercial CarSim® vehicle dynamics simulation package.

INTRODUCTION

The major actions that move a vehicle with tires are the forces and moments generated between the tire and ground; hence, the validity of a simulated vehicle test depends on the quality of both the tire model and the characterization of the ground surface. Other actions come from aerodynamic forces

and moments that depend in part on the movement of the vehicle body relative to the ground surface. Although a 3D road model is not needed for every simulated test, there are many conditions where the road geometry is an essential part of the scenario being simulated. Testing of advanced driver assistance systems can involve major 3D terrain features such as road crown, banking, ditches, median-strip features, etc.; highly-asymmetric NASCAR racecars mostly turn left at high speeds on banked tracks with known roughness features; and advanced driving simulators provide urban and rural scenarios that must provide realistic interactions in response to a driver taking the vehicle in any direction, over any part of a road network.

This paper describes the 3D road model in vehicle simulation software from Mechanical Simulation Corporation used for simulating and analyzing the dynamic behavior of motor vehicles in response to steering, braking, and acceleration control inputs. The road model is part of a system of modeling methods and library routines called VehicleSim® that underlies the software packages CarSim® (three- and four-wheeled road vehicles) [1], TruckSim® (vehicles with dual tires per axle side, multiple trailers, motor vehicles with more than two axles), and BikeSim® (motorcycles and scooters).

There were several objectives in characterizing 3D road geometry for these simulation programs.

First and foremost, the road surface description must be sufficient to work with the tire model to produce valid tire forces and moments. The vehicle models support several tire models which all assume a rigid surface (in contrast to deformable terrain such as dirt, gravel, snow, etc.). The tire models used for most simulations have a simple interface with the ground that involves a single point of the center of

tire contact (CTC). However, other tire models are available. For example, TNO MF-Tyre/MF-Swift [2] and COSIN FTire [3] locate multiple points of contact to account for detailed structural interactions with the tire. For each point of contact, the road model must provide the global Z coordinate of the ground surface when given global coordinates X and Y . Also, the model must produce the road slope ($\partial Z/\partial X$ and $\partial Z/\partial Y$) and material properties such as friction coefficient and rolling resistance coefficient.

Road geometry is also needed to provide a reference line for closed-loop path followers sometime used in vehicle models to simulate driver steering.

Information about road geometry can be obtained from many sources. Some road information might be based on design features such as the horizontal geometry of a reference line, grades, and banking. However, the geometry of a physical road does not perfectly match the design information. In many cases, design information does not even exist. Although it is important for vehicle simulation software to work with road design data, it is even more important to be able to work with physical measurements such as GPS data, road roughness profile measurements, and 3D laser-scanned terrain topology. In addition to design data from civil engineers, other sources of road data include 3D mesh descriptions from video games and 3D modeling software, and specific scenarios created by engineers to study some behavior of interest.

One more consideration is that calculations involving the interactions between the vehicle and the road must be computationally efficient. Besides the general convenience of providing simulation results rapidly for any user, vehicle simulations that include road interactions must support operation in real-time for driving simulators and hardware-in-the-loop (HIL) testing.

THE VEHICLESIM ROAD MODEL

The Federal Highway Administration (FHWA) organized a research project in 1994 to use existing vehicle dynamics simulation models from TruckSim and link them to commercial roadway programs to simulate a range of highway vehicles navigating existing or proposed highway designs [4]. (At that time, TruckSim was provided and supported by the University of Michigan.) In this project, road design datasets from civil engineers were used that involved curvature, elevation, and banking. The software package delivered to FHWA was called VDM RoAD (Vehicle Dynamic Models Roadway Analysis and Design). Although the road model in VDM RoAD met the requirements of the project, it was not well suited for handling measurements from GPS and similar methods. Nor was it convenient for handling disturbances such as bumps, holes, curbs, etc.

The road model described in this paper was built from the coordinate system from the VDM RoAD project, replacing the design parameters (curvature, banking, grade) with tabular interpolation functions to provide a more comprehensive model that works with data from a variety of sources.

HORIZONTAL GEOMETRY - STATION AS A FUNCTION OF X AND Y

Road designs typically rely on station number (longitudinal distance along the design centerline) as the primary variable for locating features along the road. For the case of a curving road with elevation change, the curved path is projected onto a flat and level plane (that is, Z coordinates are ignored), and station is the length along that projected path.

For any given value of S (station), there is a unique set of X and Y coordinates (Figure 1). The change in *true station* between two points can be slightly larger than the straight-line distance between points, because the true station is defined along the curved path. Given a set of points on this projected path, the calculation of true station along the curve depends on the mathematical function used to define the curve. Slightly different results are obtained if it is a constant-radius curve, or a spiral, or a spline function, etc. If design data are not available or the mathematical function for the curvature is not known, then there is a small ambiguity in determining the true distance along the path.

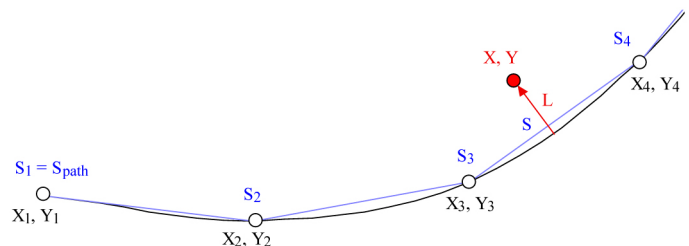


Figure 1. Horizontal geometry of a curved reference line.

The VehicleSim road model defines horizontal geometry using a set of X - Y coordinates for a reference line in the ground plane. Station S is calculated using straight line segments; the length of each line segment is added to the previous value of S when going from point to point in a direction for increasing station:

$$S_i = S_{i-1} + \sqrt{(X_i - X_{i-1})^2 + (Y_i - Y_{i-1})^2} \quad \{i > 1\} \quad (1)$$

This definition of station is deliberately simple - straight lines connect points as viewed from a satellite in space. Effects of elevation change and curvature are not included, meaning that S from this definition is slightly less than the distance

alone the curved path (the curved black line in the figure). This simplified definition of S is used to provide a connection between multiple properties of the road surface (elevation, banking, friction, etc.). Station is also used to control inputs such as target speed and lateral position.

Although the calculation of station is based on straight line segments, the reference line between the points is indeed curved, as indicated in [Figure 1](#). At station values between the original data points, there can be an offset between the straight connecting line and the curved path. Internally, the VehicleSim solver programs generate tables of X vs. S and Y vs. S and use spline interpolation to calculate X and Y as continuous functions of S for points on the curved reference line. The spline functions also provide $\partial X/\partial S$ and $\partial Y/\partial S$ as functions of S .

The second road coordinate is L , the lateral distance from a point perpendicular to the road reference line (see the red circle in [Figure 1](#)). As is the case with station, L is defined in a horizontal plane, as if viewed from space by a satellite. Lateral position is used to specify a target path for the closed-loop steering controller and for specifying elevation changes as described in the next section.

VERTICAL GEOMETRY - Z AS A FUNCTION OF S AND L

Starting with the horizontal reference, elevation at an arbitrary location on the road (Z_{road}) is summed from a contribution from the reference line, defined as a function Z_{ref} of S , and two off-center contributions defined using Z_{map} , a function of S , L , and an ID number.

$$Z_{road} = Z_{ref}(S) + Z_{map}(L, S, 1) + Z_{map}(L, S, 2) \tag{2}$$

Configurable Functions

VehicleSim solver programs represent many potentially nonlinear relationships between two or three variables with *configurable functions* of the form $f(X_{col}, X_{row}, id)$ where X_{col} and X_{row} are independent variables and id is an integer used to identify a specific dataset. Most options for a configurable function involve tabular data provided in a spreadsheet form with columns and rows of numbers. For example, for the function Z_{map} , the first row contains values of L (the X_{col} variable) associated with each column. If there are N_{col} values of L provided, then each row has $N_{col} + 1$ numbers - a value of S (the X_{row} variable) followed by corresponding values of elevation for each combination of L and S .

If there is only one independent variable (e.g, the function Z_{ref} depends only on S), then each row in the table has just

two numbers: the independent variable X_{row} and a corresponding value of the dependent variable.

In most cases, configurable functions that allow two independent variables can also be used with data for only one variable, with simpler datasets and faster calculation times. Further, most functions can also be configured to skip the nonlinear calculations completely. These options are specified at runtime by keywords used in a simulation input data file to identify parameters and tables associated with vehicle properties and test conditions. [Table 1](#) lists the options available for the internal function Z_{map} that is used to calculate the two components of road surface elevation indicated in [Eq. \(2\)](#).

The first option (keyword = ROAD_DZ_CONSTANT) specifies that calculations are skipped; the function simply provides a constant, such as zero. The second option (keyword = ROAD_DZ_COEFFICIENT) would instead specify that S be multiplied with a linear coefficient. The next seven options specify various methods for calculating the elevation component from S using interpolation methods based on a dataset with values of S and associated values of the elevation contribution. The final six options specify methods for calculating the elevation contribution using interpolation methods applied to a dataset involving a matrix of numbers with elevation values for combinations of S and L .

In addition to the options shown in [Table 1](#), most configurable functions also have two scalar parameters that apply a gain and offset to the calculated function, to allow datasets for basic shapes to be rescaled easily.

Table 1. Configurable function options and keywords.

Example Keywords	Description
ROAD_DZ_CONSTANT 0.0	Constant
ROAD_DZ_COEFFICIENT 0.02	Linear coefficient for X_{row}
ROAD_DZ_TABLE STEP	Table lookup based on X_{row} ; there are seven calculation options involving interpolation and extrapolation
ROAD_DZ_TABLE LINEAR	
ROAD_DZ_TABLE LINEAR_LOOP	
ROAD_DZ_TABLE LINEAR_FLAT	
ROAD_DZ_TABLE SPLINE	
ROAD_DZ_TABLE SPLINE_LOOP	
ROAD_DZ_TABLE SPLINE_FLAT	
ROAD_DZ_CARPET 2D_STEP	Table lookup involving both X_{col} and X_{row} ; there are six calculation options involving interpolation and extrapolation
ROAD_DZ_CARPET	
ROAD_DZ_CARPET LOOP	
ROAD_DZ_CARPET VAR_WIDTH	
ROAD_DZ_CARPET VAR_WIDTH_STEP	
ROAD_DZ_CARPET 2D_SPLINE	

In the case of the function Z_{ref} in Eq. (2), the internal function has only a single independent variable (S). Therefore, keywords related to interpolation with two independent variables (those specified with keywords containing “_CARPET”) are not installed in the math models.

Figure 2 illustrates the method implied by Eq. (2) for building a 3D road surface from tabular data. Images in the left side of the figure show screens from the VS browser used to enter and/or visualize tabular data. (The VS browser is a database manager with the main GUI seen in VehicleSim products such as CarSim.) Images on the right-hand show the road geometry as it is built from each component.

First, a set of X - Y coordinates is provided (see the upper-left part of Figure 2). In this example, the road turns to the left, then to the right, and then to the left to resume the original direction. As mentioned earlier, the road model generates values of S corresponding to each pair of X and Y coordinates, and uses internal spline interpolation methods to calculate X and Y coordinates for the road reference line for any value of S . The method was used to generate the continuously curving reference line shown in the upper-right portion of the figure. If this were the only description of the road, the reference line could be used to generate paths for driver controllers and to make shapes for animations, but the ground would be flat and level. Banking of turns involves changes in elevation away from the reference line. The middle-left part of the figure shows a 3D map of elevation in terms of S and L . The geometry here is simple; there are three banking angles for the three curves, separated with linear transitions. The road shown in the middle-right part of the figure shows the effect of taking the Z_{map} shape and distorting it to follow the curved path, with S and L defined from the curved reference line. The lower-left part of the figure shows a screen for setting Z_{ref} as a function of S , with the road dropping 10 m in elevation about halfway through the second curve. In this case, the configurable function was set to use spline interpolation within the range of data provided, with flat-line extrapolation. The view of the road in the lower-right part of the figure shows the combined result of the horizontal curve geometry, the off-center elevation due to banking, and the 10-m dip in the middle of the section.

Two of the options listed in Table 1 account for variations in road width along the road. Internally, this is accomplished by an option where the function Z_{map} is set to calculate the elevation component in terms of S and an intermediate variable associated with each column of data. (The variable is simply the column number, with values 1, 2, ...). A separate table defines L for each column as a function of S , using linear interpolation. For example, values of elevation provided in column 2 of a table of input data can correspond to values of L that in turn depend on S . This option can greatly reduce the amount of data needed to define simple

geometric features that are not parallel or perpendicular to the reference line. For example, Figure 3 shows rounded curbs at the edges of a road and in the median. (The geometry of the curbs can be seen in the figure from the shapes of the shadows cast by the vehicles.) The curbs follow the road edge as the road changes width due to lanes merging.

Coordinate Transformations

The S - L - Z coordinate system is convenient for specifying road geometry and features. However, most physics models and visualization software require locations and directions to be specified in a global X - Y - Z Cartesian system. Given that both systems use the same global Z coordinate, the challenge is to convert from X - Y coordinates to S - L (where vertical geometry is described), and from S - L to X - Y .

Figure 4 shows a curved section of road with the original data points shown as yellow circles on the reference line. The points are connected with straight lines shown in light blue, and the curved reference line is shown in black. A point of interest is shown with another yellow circle.

Consider the two types of transformations: calculate X - Y for a point given S - L ; or, calculate S - L given X - Y . Both cases require knowledge of the direction of the reference line at S and the line of constant S that is perpendicular (lines of constant S are shown with red arrows in the figure). In the road model, the directions of the lines of constant S are defined at each of the original data points (yellow circles in the figure) by bisecting the angles between the connecting line segments. For values of station between two of the original points, the direction changes linearly from one end of the segment to the other.

Transforming from S - L to X - Y is accomplished with a direct sequence of calculations (no iteration is required). As mentioned earlier, internally generated tables are used to calculate X and Y on the reference line. The X and Y coordinates of the point of interest are then calculated taking into account the known value of L and the partial derivatives $\partial X/\partial L$ and $\partial Y/\partial L$.

Going from X - Y coordinates to S - L involves similar calculations if the segment of road perpendicular to the point of interest is known. Given directions of the lines of constant S , the coordinates of an original point (yellow circle on the reference line in Figure 4), and direction of the connecting line segment, the values of S and L can be calculated. If the value calculated for S is within the segment used for the calculations, then the calculation is complete. However, if the calculated value of S is not within the assumed segment, then the segment containing the calculated value of S is found by searching the table, and the process is repeated. In vehicle simulation programs that advance through time in steps, the segment used at the previous simulation time is used for the