CarSIM: Math Models

CarSim solves equations that represent the dynamic behavior of four-wheeled vehicles, possibly towing a trailer. The VehicleSim® (VS) architecture includes the built-in VS Command scripting language to add new capabilities at runtime to automate tests or add features to the math models. Further, the models can work with other software (Simulink, LabVIEW, ETAS ASCET, Custom C/C++ programs, Visual Basic, etc.) for automation or extensions to the math models.

Vehicle Math Models

Configurable Table Functions

- Potentially nonlinear relationships between independent and dependent variables are defined with VS Configurable Functions that are set at runtime to use:
  - Constants
  - Linear coefficients
  - Nonlinear tables with several interpolation methods involving one or two independent variables
  - Algebraic formulas involving other variables
- When linear coefficients or linear interpolation methods are selected, the simulations can run even faster.
- No built-in limit to the length of tables.
- Independent and dependent variables can be transformed in support of normalized functions.

VS Reference Paths

- Up to 200 VS Reference Paths, each with a sequence of segments (straight, arc, clothoid, X-Y table).
- Reference paths are used for driver controls, traffic vehicles, and road definitions.

Driver Controls

- All driver controls can be specified using built-in model options, defined by equations added at runtime with VS Commands, or imported from other software.
- The built-in driver model can steer to follow a target path, which can be changed during the run.
- Closed-Loop Speed Controller based on target speed and acceleration limits, curvature of the target path, and 3D road geometry (banking, grade, curvature).
- Gear shifting and clutch controls can be handled with shift schedules and automatic throttle-clutch interactions.
- Closed-loop and open-loop controls can be combined to simulate intervention systems.
- Open-loop steering can be angle or torque.
- Open-loop braking can be pedal force or fluid pressure.

3D Road Geometry and Friction

- Up to 100 road surfaces are supported, each with its own reference path.
- Vertical elevation is added with several components, each specified as a function of path S (station) and L (lateral offset) coordinates.
- Roads can have variable width, allowing highly efficient descriptions of complex geometries.
- Road profiles are included that "wander" to follow the vehicle wherever it goes. This provides efficient use of high-frequency measured road roughness data.
- Friction is specified relative to the reference path with a variable-width function of station and lateral distance.
- Road geometry can be imported from other software or defined by equations that are added at runtime.
- The VS API includes functions to provide access to the 3D road geometry for user-supplied equations for model extensions or additional outputs.

Vehicle Configurations

- CarSim trailer includes a trailer with one or two suspensions in a tandem with load-sharing effects. An optional license is needed to include a trailer.
- The sprung masses may be simple rigid bodies, or include frame rails account for torsional compliance. An optional license is needed for torsional compliance.
- CarSim includes an option to include extra multibody degrees of freedom (DOF) for engine movement due to mount compliances. An optional license is needed.

Wind and Aerodynamic Effects

- Six aerodynamic forces and moments are applied to the sprung mass (both sprung masses if there is a trailer).
- These forces and moments are Configurable Functions of aerodynamic slip, pitch, and ride height.
- Ambient wind speed and heading can be set with tables, runtime equations, or imported from other software.

Suspensions

- Suspensions can be generic or a solid axle. Generic suspensions specify wheel movements with jounce on both sides, possibly with coupling. They can represent independent and twist-beam suspensions.
- Suspensions are either un-steered, or they include full 3D kinematics for inclined steering axes of the wheels.
- The suspension models have full nonlinear kinematical behavior and are asymmetric.

Mechanical Simulation
755 Phoenix Drive, Ann Arbor MI, 48108, USA • 734 668-2930 • Fax: 734 668-2877 • Email: info@carsim.com
carsim.com

- Suspension springs and dampers are nonlinear and include hysteresis due to friction.
- Every suspension has compliance in the lateral and longitudinal directions, and every wheel has toe and camber compliance effects.
- All compliances can be represented with linear coefficients or nonlinear configurable functions.
- Separate forces are included for bump stops.
- Suspension roll moments include a nonlinear auxiliary roll moment and linear coefficient roll damper.

Steering System

- The interactions between the suspension, steering, tire, and ground are handled with a detailed multibody model that uses an inclined kingpin axis.
- The steering model includes specific details for rack-and-pinion and recirculating ball-type systems.
- Steer angle of each road wheel is available as rotation as measured in a K&C rig (in the sprung mass coordinate system), or as rotation about the kingpin axis.
- The steering system includes detailed options for manual or dynamic power boost, including column assist.
- The steering system includes hysteresis, compliance, inertia, and damping.
- Special equations are used for low-speed conditions to simulate ground friction steer torque.

Brake System

- Brake control can be set with pedal force (with or without boost) or master cylinder pressure.
- The control input pressure from the master cylinder is proportioned for each wheel-end brake actuator.
- Brake torque is modeled as a nonlinear function of actuator pressure and thermal effects.
- The brake system has several options for providing ABS control from external programs such as Simulink, or using a simple built-in controller.
- Special equations handle wheel lockup to obtain the correct reaction torque and avoid numerical instability.

Tires

- CarSim includes several tire models. It is ready to run with a fully asymmetric table-based model, an extended model (more tables for camber effects), the Pacejka 5.2 version of the magic formula, and MF-Tyre from TASS/TNO.
- CarSim runs with MF-Swift from TASS/TNO and FTire from COSIN (extra licenses are required from TASS and COSIN, respectively, to use their models).
- Different models can be applied to different wheels of the same vehicle.
- The built-in models use nonlinear tables for lateral force, longitudinal force, aligning moment, and overturning moment as functions of slip, load, and camber.
- Variable friction conditions are handled using similarity, to maintain both linear and limit properties of the tire. Separate coefficients are used for X and Y components.
- Transient effects of rolling are included using relaxation length. Relaxation lengths can be constant or defined as nonlinear functions of vertical force and slip.
- Special equations are used to keep realistic tire transient behavior at very low speeds and very high speeds.
- External tire models can apply forces at either the ground contact point or the wheel center.
- Tire contact can be handled with one to four points to account for the contact patch dimensions, which can be defined as nonlinear functions of tire compression.
- Dual tires are available for all wheels.

Powertrain

- CarSim has detailed powertrain models for front-wheel drive, rear-wheel drive, and four-wheel drive (all-wheel drive). There is also a minimal model used for speed control in which torque is applied directly to the wheels.
- Engine torque is defined with a Configurable Function of throttle input and crankshaft angular velocity.
- Fuel consumption is defined with a 2D table.
- The engine feeds torque to the transmission either through a hydraulic torque converter or through a mechanical clutch.
- The transmission converts torque and speed based on the current gear selection, with spin inertias and efficiencies that depend on the gear selection.
- Continuously variable transmissions (CVT) are supported.
- The torque from the transmission goes to a differential for front or rear-wheel drive systems. With four-wheel drive systems, the transmission applies torque to a center unit (e.g. transfer case) with a torque bias.
- The center unit and differential models are similar. All have four model options:
    1. Always locked.
    2. Viscous coupling.
    3. Coupling applied using a clutch. The clutch can be controlled externally or with built-in logic.
    4. Yaw control differential system, which involves two clutches with reduction gears in parallel over a differential. The system allows control of the torque distribution between left and right, or front and rear.
- The center unit has a torque bias for non-locked options.

- Twin-clutch is an alternative to an axle differential. The system involves a gear box in the middle of the axle and two clutches between each wheel and the gearbox.
- Torsional compliance of the driveline can be characterized by a natural frequency and damping ratio.
- CarSim provides tight, low-level integration with the commercial tool AVL Cruise. The GUI in CarSim and AVL-Cruise support automatic connections. An optional license is needed to use the Cruise interface.

Sensors and Traffic

- The models include several kinds of virtual sensors that detect various types of vehicle motion, including acceleration, speed, and previews of the road ahead.
- Up to 200 moving objects can be added that are updated automatically to convert simple road-based commands into full 3D geometry. The objects can be recycled for extensive runs, to reappear after they go out of view.
- Motion of an object can be constant, set by specifying speed, set with algebraic equations, or imported from third-party software.
- Objects that move based on speed support offtracking, to produce realistic low-speed traffic turns at intersections.
- Up to 99 range and detection sensors can be included that detect the moving objects. An optional license is needed to use the sensor feature.
- Each detection includes 22 variables that can be exported to external controllers (e.g., ADAS).
- Objects can block each other (occlusion). The sensor detection variables respond only to the portion of the object that is within the field of view.

Solver Program Input and Outputs

CarSim uses standard VS library routines for processing input files, performing standard calculations, and generating output files.

Input Data Files

- CarSim reads all input from text files that are normally generated automatically. These files can also be made externally for advanced applications.
- Input files for CarSim follow a simple keyword-based format called the Parsfile. CarSim can recognize thousands of keywords when processing input files.
- Parsfiles are efficient for software to read and write, while also being easy for people to read and edit.
- Each input line can optionally specify alternate units for a parameter.
- Values can be assigned directly to model parameters with numbers, numerical expressions (e.g., 1/16), or symbolic algebraic expressions involving other model variables.
- CarSim processes VS Commands at runtime that define new variables, add equations to the model, change units for variables, and otherwise extend the original CarSim model to meet custom requirements.
- Parsfiles support the INCLUDE capability, allowing advanced applications such as design of experiments (DOE), sensitivity, and customized automation methods.

Output Variables

- CarSim generates from 600 to several thousand built-in output variables, depending on whether there is a trailer, sensors, traffic vehicles, etc.
- Commands used at runtime add sets of motion outputs, driver preview points, and other outputs as needed.
- A subset of the available outputs can be specified at run-time, to control the size and organization of output files.
- Writing to file can be enabled and disabled during the run, to save only interesting results from long simulations.
- New output variables can be defined at runtime.
- CarSim provides a GUI for browsing the lists of available variables, sorting by several categories.
- All variables are described in documentation files in both text and spreadsheet format.
- Output files may be written in several binary forms (32-bit and 64-bit) or CSV (text) spreadsheet format.
- Output variables are used for several purposes:
  - Make plots that show vehicle behavior.
  - Input to post-processing software.
  - Motion information for video visualization.
  - Define conditions for VS Events when new vehicle or control properties take effect.
  - Used in formulas added at runtime.
  - Export to other software during the simulation.

Working with Simulink® and External Models

- On Windows machines, the CarSim math model is a DLL file that runs in many environments:
  - CarSim runs the models with no additional software.
  - Models run as blocks in MATLAB/Simulink, LabVIEW, and other simulation environments.
  - Models can run as functional mockup units (FMU) using the functional mockup interface (FMI). FMUs may access database files or be defined as completely self-contained files.
  - Models work with Visual Basic, MATLAB, and other programming languages that can load DLL files and access their functions with the VS API.
  - CarSim includes both 32-bit and 64-bit DLLs.

- C/C++ can be used to extend the math models, accessing thousands of parameters and variables using the VS API.
- MATLAB, Visual Basic (VB), and other languages can run the models with Windows COM for automation.
- Math model solver programs are compiled to native code for real-time systems to interface with the RT test control software.
- CarSim has a LINEARIZE command to generate linearized A, B, C, and D matrices for use in MATLAB.
- CarSim has built-in commands for adding forces and moments to extend the model.

Input Variables

- Calculations from external models and measurements from hardware-in-the-loop (HIL) can be imported into CarSim. These include most forces and moments, fluid pressures, controls, ground geometry under each tire, etc.
- The vehicle models can import values for hundreds of built-in variables.
- Most of the import variables can be combined with native internal variables with one of three modes:
  1. replace the native variable,
  2. add to the native variable, or
  3. multiply with the native variable.
- CarSim provides a browser for activating import variables from the lists of all those that are available.
- New import variables can be defined at runtime to pass through data from other software. E.g., variables from Simulink can be passed through to the animator.

Export Variables

- All variables available for writing to output files are also available for export to Simulink or external code.
- Variables are exported only if activated at runtime, as needed to be compatible with the external model.
- New export variables can be defined at runtime.

Multibody Model Specifications

State Variables and Degrees of Freedom

CarSim has ordinary differential equations (ODEs) for the dynamics of multibody systems, including rigid bodies, fluids, tires, controllers, and other dynamic parts.

- The multibody mechanical system for a four-wheeled vehicle has 15 mechanical degrees of freedom (DOF):
  - The sprung mass is a rigid body with six DOF.
  - Each suspension has two DOF. (The 3D motions follow constraint data from K&C tests.)
  - Each wheel has one spin DOF.
  - The steering system has one DOF.
- The math model for a four-wheeled vehicle has over 110 ordinary differential equations (ODEs). Each multibody DOF has two equations; other equations represent the dynamics of components:
  - Each tire has two DOF for lagged response.
  - MF-Tyre / MF-Swift and FTire add more DOF.
  - The brake fluid in each actuator has one DOF.
  - The temperature for each brake has one DOF.
  - The engine crankshaft has one DOF.
  - Throttle has a lag with one DOF.
  - Fuel consumption has one DOF.
  - Power steering and power brake each add one DOF.
- The engine mount models add six DOF.
- The math model has over 250 state variables, needed (along with parameters and configurable function definitions) to fully define the state of the system. These include the ODE variables plus others:
  - Each friction element has a state variable for hysteresis (suspension springs, steering system, low-speed tire steer).
  - Clutches and built-in controllers (e.g., ABS) have locked states.
  - Other dynamic mode conditions have state variables.

Equation Form

- The equations of motion are derived from first principles for 3D motions of multiple connected rigid bodies.
- The equations of motion are ordinary differential equations (ODE's) that are well behaved (not stiff).
- The built-in VS library provides six methods for solving the ODE's (Adams-Bashforth, Adams-Moulton, Runge-Kutta, and Euler methods).
- All methods run at a fixed time step and work well for real-time applications.
- The algorithms work well with measured and sampled data sources, even when there are discontinuities.
- The equations are compiled with extensive optimizations for efficient use either alone or with other software (e.g., Simulink, LabVIEW).

Initialization and Restarts

- CarSim supports many initialization options, from automatic to detailed specification of any state variable.
- The complete state of the vehicle model is saved at the end of each run, to support continuation of advanced automation and optimization methods.
- The state of the model can be saved during a run and fully restored during the run, in support of advanced optimization methods and repetitive test sequences.