CarSim® 7.1: Math Models

CarSim 7.1 provides highly efficient custom computer programs optimized for solving equations in math models that represent the dynamic behavior of vehicles. The built-in VehicleSim® command language can be used to add new capabilities at run time to customize and extend the math models. Further, the models can communicate with other software (Simulink, LabView, ETAS ASCET, Custom C/C++ programs, Visual Basic, etc.).

Vehicle Math Models

Nonlinear Properties and Tables

- Nonlinear properties are typically specified with VehicleSim tables. These can be set to use constants, linear coefficients, nonlinear tables with linear or spline interpolation, and in some cases, 2D interpolation. (Up to 11 calculation options are available.)
- When simpler methods are selected (coefficients or linear interpolation), the simulations can run even faster.
- There are no built-in limits to the lengths of tables.

Control Inputs

- All control inputs can be specified using built-in model options. They can also be defined by equations at runtime using VehicleSim commands. Or, they can be imported from other software such as Simulink, LabView, or custom programs written in C, VB, etc.
- Built-in steering control can be specified with open-loop steering wheel angle or open-loop steering wheel torque. Alternatively, a built-in driver model can be selected; it follows a path relative to the road centerline.
- Brake control is provided by several options, including pedal force and master cylinder pressure. Or, a speed-controller can control throttle and brakes using a target speed relative to either time or position along the road.
- Throttle control is specified as an open-loop function of time, or, throttle is handled by the speed-controller.
- Gear shifting and clutch controls are specified as open-loop functions of time, or, automatic shifting and clutch control are used.

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 nonlinear tabular functions of aerodynamic slip, pitch, and ride height.
- Wind amplitude and heading can be specified as tabular functions of time, or with equations specified at run time, or by importing from other software.

3D Road Geometry and Friction

- Horizontal geometry of a road design path (typically the centerline) is defined by a sequence of X-Y coordinates. They are fitted by a spline to define a continuously curved horizontal path. They are also used to define station (distance along the path).
- Vertical geometry of the design path is defined by a sequence of elevation values as a function of station. The elevation values are interpolated by a method specified at run time.
- Off-center elevation relative to the design path is specified with a 2D table, based on station and lateral distance from the design path. Two tables are used to help separate design geometry (superelevation, ditches, road camber, etc.) from special events (bumps, potholes, etc.) or roughness data.
- Road profiles are also included that follow the vehicle, to enable the use of high-frequency measured roughness data without requiring a fully detailed 3D map.
- Friction is specified relative to the design path with a 2D table based on station and lateral distance from the path.
- As an alternative to the road concept, a 3D ground surface can be specified as a table of Z values for a grid of global X and Y values.
- Road geometry can be imported from other software or defined by equations that are added at runtime.
- The VehicleSim API includes functions to provide access to the 3D road geometry for user-supplied equations for model extensions or additional outputs.

Suspensions

- The suspension models have full nonlinear kinematical behavior and can be asymmetric.
- Every suspension has compliance in the lateral and longitudinal directions.
- Every wheel has compliance that affects toe and camber in response to tire shear forces and moments.
- The front suspension can be independent or solid axle.

Mechanical Simulation Corporation
912 North Main, Suite 210, Ann Arbor, MI 48104 • 734.668.2930 • fax 734.668.2877

- Rear suspensions can be independent, twist axle, or solid axle.
- In an independent suspension, each wheel moves vertically. Longitudinal and lateral movements, camber, toe, and dive angles are all related to vertical position by nonlinear tables.
- In a solid axle suspension, the main variables are vertical movement and roll of an axle rigid body. Forward movement, lateral movement, steer, and dive are defined by nonlinear tables.
- CarSim obtains roll and jacking forces as the natural results of full 3D kinematical curves and compliance effects for independent and solid-axle suspension.
- In a twist axle suspension, the wheels move together when there are no external forces, similar to a solid-axle suspension. Compliance effects are included to account for the twisting of a connecting beam.
- Suspension springs are nonlinear and include hysteresis due to friction.
- Separate forces are included for bump stops.
- Damper forces are nonlinear functions of stroke rate.
- Suspension roll moments include a nonlinear auxiliary roll moment.

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 system includes specific details for rack-and-pinion and recirculating ball.
- The steering system includes detailed options for manual or power boost (with a time constant).
- 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

- The control input pressure from the master cylinder is proportioned for each actuator.
- Brake torque is modeled as a nonlinear function of actuator pressure.
- The brake system has several options for providing ABS control from external programs such as Simulink, or using a simple built-in controller.
- Thermal effects are included.
- Hydraulic dynamics are modeled with a first-order transient lag, plus a pure time delay for each actuator.
- The condition of wheel lockup is handled with a torsional spring-damper system to obtain the correct reaction torque when the wheel is locked.

Tires

- CarSim includes several tire models, along with a program interface that supports external tire models.
- The main model uses nonlinear tables to represent lateral force, longitudinal force, aligning moment, and overturning moment as functions of slip, load, and camber.
- A simpler model is included that neglects overturning moment and nonlinear camber effects.
- Lateral and longitudinal forces and moments are combined using combined slip theory as published by Pacejka and Sharp.
- 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.
- Dynamics due to rolling are included using relaxation length for lateral and longitudinal slip.
- Special equations are used to maintain realistic tire behavior at low speeds, when the assumptions of a rolling tire are not valid.
- The TNO Delft Tyre model is included.
- The Pacejka 5.2 version of the Magic Formula is included as an alternative model option.
- External tire models can apply forces at either the ground contact point or the wheel center.
- Different models can be applied to different wheels of the same vehicle.

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 power is applied directly to the wheels.
- Engine torque is defined with a 2D table that relates torque to throttle input and crankshaft angular velocity.
- 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.

- 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. In this case the locking is modeling with a torsional spring and damper.
  2. Viscous coupling. In this case the torque differential is defined with a table based on the speed difference.
  3. Coupling applied using a clutch. The differential clutch has built-in control logic (i.e. LSD: limited slip differential) that can be used, or it can be controlled externally. This can be with or without viscous coupling.
  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. This can be with or without viscous coupling.
- The center unit has a torque bias that can be used with any of the above 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. The clutch positions are controlled externally.
- Fuel consumption is defined with a 2D table based on engine speed and throttle.

Solver Program Input and Outputs

The CarSim solver programs use VehicleSim library routines for processing input files, performing standard calculations, and generating output files.

Input Data Files

- The solver programs read all inputs from text files. (These text files are normally generated automatically by CarSim; users typically do not view them.) The files can be controlled from within CarSim or externally.
- Input files for CarSim math models follow a simple keyword-based format called the Parsfile. A typical CarSim solver program 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.
- Parsfiles support the INCLUDE capability, allowing many advanced applications such as design of experiment (DOE), sensitivity, and customized automation methods.
- Values can be assigned directly to model parameters with numbers. Users can also specify values with numerical expressions (e.g., 1/16) or symbolic algebraic expressions involving other model variables.
- The solver programs process VehicleSim commands at run time that define new variables, add equations to the model, change units for variables, and otherwise extend the original CarSim model to meet custom requirements.
- The animator, plotter, and graphical user interface also use Parsfiles to store data.

Output Variables

- The solver programs generate between 700 and 1200 output variables, depending on whether there is a trailer.
- Any subset of the list of variables can be specified at run time, to control the size and organization of output files.
- New output variables can be defined at runtime.
- CarSim provides a graphical interface 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 variables are used for several purposes:
  - Make plots that show vehicle behavior.
  - Input to post-processing software.
  - Motion information for the animator.
  - Possible inputs for external model extensions.
  - Define conditions for “events” when new vehicle or control properties take effect.

Working with Simulink® and External Models

- On Windows machines, the CarSim math models are DLL files that run in many environments:
  - The CarSim graphical database runs the models with no additional software.
  - They run as blocks in MATLAB/Simulink, LabView, and other simulation environments.
  - They work with Visual Basic, MATLAB, and other programming languages that can load DLL files and access their functions.
  - They run under the control of Windows commands.
- Multiple instances of a math model can run simultaneously in Simulink and other environments to simulate multiple vehicles that interact dynamically.
- C/C++ can be used to extend the math models, accessing thousands of parameters and variables using the VehicleSim API.

- Math model solver programs are compiled to native code for real-time systems to interface with the RT test control software.

Input Variables

- Calculations from external models and measurements from hardware in the loop can be imported into CarSim. These include most forces and moments, fluid pressures, control variables, ground geometry under each tire, etc.
- The vehicle models without trailers can import almost 300 variables. Those with trailers can import over 450.
- Most of the import variables can be combined with native internal variables. Users can specify one of three possible actions for each activated import variable:
  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 run time 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 run time. Most Simulink models receive only a small number of the potential export variables from CarSim, simplifying the integration with other software.
- CarSim provides a browser for activating export variables from the lists of all those that are available.
- New export variables can be defined at run time.

Multibody Model Specifications

State Variables and Degrees of Freedom

- The sprung mass of the motor vehicle is a rigid body with six degrees of freedom (DOF).
- Each wheel has one spin DOF.
- Each independent suspension has two independent multibody DOF—the vertical movements of the wheel centers.
- Each solid-axle suspension or twist-axle has two independent multibody DOF—the relative bounce and roll of the axle.
- Other suspension motions, such as pitch, camber, lateral position, etc., are constrained as nonlinear functions of the independent variables.
- Each suspension has six compliance DOF.
- The brake fluid in each actuator has one DOF.
- The temperature of each brake has a DOF.
- Each tire has two dynamic DOF: one for lagged lateral slip, the other for lagged longitudinal slip.
- The TNO Delft Tyre model has additional DOF.
- The engine crankshaft has one DOF, and the engine torque has a dynamic lag in response to throttle change.
- Fuel consumption has one DOF.
- Every suspension spring and tire has a friction DOF, as does the steering system.
- Each wheel has a low-speed (parking lot) steering friction DOF.
- Power steering boost and power brake boost each have a DOF.
- The ABS system has one DOF per brake actuator.

Equation Form

- The equations of motion are derived from first principals for full nonlinear 3D motions of multiple connected rigid bodies.
- The equations of motion are ordinary differential equations (ODE's) that are "non-stiff" and can be solved with most numerical integration methods.
- The built-in VehicleSim library provides five methods for solving the ODE's. These include three second-order methods (Adams-Bashforth, Adams-Moulton, and Runge-Kutta), plus third- and fourth-order Adams-Moulton methods.
- All methods run at a fixed time step and work well for real-time applications with hardware in the loop (HIL).
- The algorithms work well with measured and sampled data sources (tables), eliminating the need for spline interpolation for most tables used in the models. (However, spline interpolation is supported to provide a full range of run-time options.)
- The native equations are compiled with extensive optimizations and are embedded in the solver programs where they maintain their high efficiency regardless of what other software that might be running concurrently (e.g., Simulink).