# bikeSIM®: Math Models

BikeSim provides custom computer programs optimized for solving equations in math models that represent the dynamic behavior of motorcycles. 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 with built-in gains and offsets 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.

### Rider Controls

- All rider controls can be specified using built-in model options, or defined by equations added at runtime with VS Commands, or imported from other software.
- The built-in rider model can steer to follow a target path, which can be changed during the run.
- A Closed-Loop Lean Target (for the bike) can be combined with a Rider Body lean angle.
- The rider model can control speed 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 braking is represented as lever force (front and rear).
- Open-loop rider body control.

### Wind and Aerodynamic Effects

- Six aerodynamic forces and moments are applied to the sprung mass.
- These forces and moments are configurable functions of aerodynamic slip and pitch.
- Ambient wind speed and heading can be set with tables, runtime equations, or imported from other software.

### 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.

### Suspensions

- The suspension models have full nonlinear kinematical behavior.
- Front suspension has compliance in longitudinal (bend) and roll (twist) relative to the main frame.
- The front suspension can be many types: *telescopic fork*, *McPherson strut*, *double wishbone*, *springer*, *bottom link*, etc. The detailed kinematic motion is able to represent *anti-dive* geometry.
- Rear suspensions can be a swing arm with or without a parallel link. The model specifies detailed kinematic


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

motion of the wheel hub to represent *anti-jacking* geometry.

- Each wheel moves vertically. Longitudinal movement and dive angle of wheel hub are related to vertical position by nonlinear tables.
- Suspension springs are nonlinear and include hysteresis due to friction.
- Damper forces are nonlinear functions of stroke rate.
- Both the spring and damper use nonlinear lever ratios.
- Front *lean-linkage* suspension for the three-wheeled motorcycles is included.

## Steering System

- The interactions between the suspension, steering, tire, and ground are handled with a detailed multibody model with steering axis.
- The steering system geometry is parameterized by caster angle, wheel axle height, fork length, and fork offset.
- The steering system includes mechanical limits and damping.
- Caster angle can be fixed relative to the main frame or variable with suspension stroke.

## Brake System

- Master cylinder pressure is calculated by the lever/pedal force input through the booster mechanism.
- Brake torque is calculated by actuator pressure and disc mechanism.
- The brake system can involve external programs (e.g. Simulink) to provide advance control such as ABS, TCS, and stability control.
- Special equations handle wheel lockup to obtain the correct reaction torque and avoid numerical instability.

## Tires

- BikeSim includes several tire models, along with a program interface that supports external tire models, such as MF-MC from TASS/TNO.
- BikeSim 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 the front and rear wheels.
- The original model uses the *Magic Formula* to represent longitudinal force, lateral force, and aligning moment as functions of slip, load, and camber. The shear forces are applied at a single contact point which moves around the tire circumference and laterally around the side wall: this automatically defines overturning moment.
- External tire models can apply forces at either the ground contact point or the wheel center.
- An alternate model uses fully nonlinear asymmetric tables to represent lateral force, longitudinal force, and aligning moment as functions of slip, load, and camber.
- Overturning moment due to the tire contact kinematical effect can be replaced with non-linear tables as functions of slip, load, and camber.
- Variable friction conditions are handled using similarity, allowing BikeSim to maintain both linear and limit properties of the tire (for table look up model).
- 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 maintain realistic tire behavior at low speeds, when the assumptions of a rolling tire are not valid.

## Powertrain

- BikeSim has detailed powertrain models for chain drive and shaft drive. There is also a minimal model used for speed control in which torque is applied directly to the wheel(s).
- Engine torque is defined with a 2D table that relates torque to throttle input and crankshaft angular velocity.
- Fuel consumption is defined with a 2D table.
- The engine feeds torque to the transmission either through a mechanical clutch or through a hydraulic torque converter with a primary gear.
- 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 either a sprocket and chain mechanism or a driveshaft.
- The chain has tensional stiffness and damping. The force vectors for driving and engine braking affect the swing arm motion.
- The driveshaft has torsional stiffness and damping.

## 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

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

### Input Data Files

- BikeSim reads all input from text files that are normally generated automatically. These files can also be made externally for advanced applications.
- Input files for BikeSim follow a simple keyword-based format called the Parsfile. BikeSim 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.
- BikeSim processes VS Commands at runtime that define new variables, add equations to the model, change units for variables, and otherwise extend the original BikeSim 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

- The solver programs generate over 400 output variables.
- 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.
- BikeSim provides a GUI for browsing the lists of available variables, sorting by several categories.
- New output variables can be defined at runtime.
- 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 the animator.
  - Possible inputs for external model extensions.
  - Define conditions for VS Events when new vehicle or control properties take effect.

### Working with Simulink® and External Models

- On Windows machines, the BikeSim math models are DLL files that run in many environments:
  - BikeSim runs the models with no additional software.
  - Models run as blocks in MATLAB/Simulink, LabVIEW, and other simulation environments.
  - Models 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.
  - BikeSim S-Function supports multiple-port connections, with Import and Export variables activated with point and click browsing in the GUI.
  - Models work with Visual Basic, MATLAB, and other programming languages that can load DLL files and access their functions with the VS API.
  - BikeSim includes both 32-bit and 64-bit DLLs.
- Multiple instances of a math model can run simultaneously to simulate multiple vehicles in Simulink, LabVIEW, and other environments.
- 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 for automation and extend the models using import and export variables.
- Math model solver programs are compiled to native code for real-time systems to interface with the RT test control software.
- BikeSim has a LINEARIZE command to generate linearized A, B, C, and D matrices at any time during the run for MATLAB analyses.
- BikeSim solvers have 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

BikeSim. These include most forces and moments, fluid pressures, controls, ground geometry of each tire, etc.

- The vehicle models can import nearly 120 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.
- BikeSim 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 runtime. Most Simulink models receive only a small number of the potential export variables from BikeSim, simplifying the integration with other software.
- BikeSim 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

BikeSim Math Models have ordinary differential equations (ODEs) for the multibody system dynamics, including rigid bodies, fluids, tires, controllers, and other dynamic parts.

- The multibody mechanical model for a two-wheeled vehicle has 16 mechanical degrees of freedom (DOF); the model for a three-wheeled vehicle has 22 DOF:
  - The sprung mass is a rigid body with six DOF.
  - Each suspension has one DOF for stroke. Other suspension motions, such as spindle pitch and longitudinal position are constrained by configurable functions.
  - Each wheel has one spin DOF.
  - The steering system has three DOF (twist, bend and steer).
  - Rider body has two DOF (upper body lean and lateral translation relative to main frame).
  - Drivetrain has one gearbox sprocket spin DOF.
- The math model has 58+ ordinary differential equations (ODEs) for a two-wheeled vehicle, and 87+ ordinary differential equations (ODEs) for a three-wheeled vehicle. Each multibody DOF has two equations (position and velocity); other equations represent the dynamics of components:
  - Each tire has two DOF for lagged response.
  - MF-MC/MF-Swift and FTire add more DOF.
  - The brake fluid in each actuator has one DOF.
  - The engine crankshaft has one DOF.
  - Throttle has a lag with one DOF.
  - Clutch torque has a lag with one DOF.
  - Fuel consumption has one DOF.
- BikeSim Math Models have about 100 state variables for a two-wheeled vehicle and about 150 state variables for a three-wheeled vehicle, 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 and low-speed tire steer).
  - Clutches and wheels have locked states.
  - Other dynamic mode conditions have state variables.

### Equation Form

- The equations of motion are derived from first principals 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).

### Initialization and Restarts

- BikeSim 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.