

bikeSIM[®]: Math Models

BikeSim 3.1 provides highly efficient custom computer programs optimized for solving equations in math models that represent the dynamic behavior of motorcycles. The VehicleSim[®] architecture includes the built-in VS command language that can add new capabilities at run time to customize and extend the math models. Further, the models can work with other software (Simulink, LabVIEW, 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 simpler methods are selected (coefficients or linear interpolation), the simulations can run even faster.
- There is no built-in limit to the length of tables.
- The independent and dependent variables can be transformed in support of normalized functions.

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.
- A built-in rider model can steer either to control specified lean angle or to follow a target path. Open-loop steering torque input is also available.
- 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).
- The target path for the rider model can be relative to the road reference line, or independent of the road line.
- 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 can be lever/pedal force or master cylinder pressure.

- Open-loop rider body control is provided including upper body lean, lateral and vertical hip movement as a function of time or position along the road.

Wind and Aerodynamic Effects

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

3D Road Geometry and Friction

- Horizontal geometry of a road reference line is defined by X-Y coordinates that are fitted by spline to define a continuously curved path.
- Vertical elevation of the reference line is defined as a configurable function of station, where station is the distance along the reference line.
- Off-center elevation is specified with a configurable function of station and lateral distance.
- The road can have variable width, allowing highly efficient descriptions of complex geometries such as lane merging, ruts and ditches that “wander,” variable-width banked turns, etc.
- 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 line 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.
- An alternative to the road concept is a 3D ground surface specified with a grid of X, Y and Z values.

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 parallel link. The model specifies detailed kinematic motion of 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 spring and damper involve nonlinear kinematic lever ratio, which can represent many kinds of cushion mechanisms.
- An alternate model uses nonlinear 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.
- Lateral and longitudinal forces and moments are combined using combined slip theory as published by Pacejka and Sharp (for table look up model).
- Variable friction conditions are handled using similarity, to maintain both linear and limit properties of the tire (for table look up model).

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.
- 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.
- Rolling resistance is included.
- External tire models can apply forces at either the ground contact point or the wheel center.
- Different models can be applied to the front and rear wheels.

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 TNO.
- BikeSim is also ready to run with FTire from COSIN (a license is required from COSIN).
- 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.

Powertrain

- BikeSim has detailed powertrain models for chain drive and shaft drive. There is also a minimal model used for speed control in which power 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.
- 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 of 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.
- Fuel consumption is defined with a 2D table based on engine speed and throttle.

Sensors and Traffic

- The models include motion sensors that can be attached to the bodies (main frame: 10, steer system: 4, front unsprung mass: 4, and rear unsprung mass: 4). Available outputs are: speed, acceleration, and angular rate on three axis directions.
- The models include reference points and associated forces and moments that are defined at runtime to extend the model.
- Up to 99 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.
- Up to 20 range and detection sensors can be included that detect the moving objects.
- Each detection includes 11 variables that can be exported to external controllers (e.g., ADAS).
- Motion of an object can be constant, set with algebraic equations, set with differential equations, or imported from third-party software.
- 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

The BikeSim solver programs use VS 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 BikeSim; users typically do not view them.) The files can be controlled from within BikeSim or externally.
- Input files for BikeSim math models follow a simple keyword-based format called the Parsfile. A typical BikeSim 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 VS commands at run time that define new variables, add equations to the model, change units for variables, and otherwise extend the original BikeSim model to meet custom requirements.
- The animator, plotter, and graphical user interface also use Parsfiles to store data.

Output Variables

- The solver programs generate over 800 output variables.
- 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.
- BikeSim 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 BikeSim math models are DLL files that run in many environments:
 - The models run 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 BikeSim. These include most forces and moments, fluid pressures, control variables, ground geometry under each tire, etc.
- The vehicle models can import nearly 200 variables.
- 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.
- 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 run time. 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

The math models have ordinary differential equations for the dynamics of multibody system, including rigid bodies, fluids, tires, controllers, and other dynamic parts.

- The multibody mechanical system for a two-wheeled vehicle has 16 mechanical degrees of freedom (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 for a two-wheeled vehicle has 58+ 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.
 - TNO Delft-Tyre and COSIN 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.
- The math model has about 100 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 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 five methods for solving the ODE's (Adams-Bashforth, Adams-Moulton, and Runge-Kutta 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 later in support 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.