LIST OF EXECUTABLE STAND-ALONE EXAMPLES

This section describes context models use with stand-alone executable vehicle simulations. The simulations were developed in Matlab, C++, Excel, either directly linked to context models, or run-time linked via FMU interfaces.

COLD START DEMO

This simple code demonstrates the usage of the environmental context models for C2M2L-1. In particular, the AR-70-32 environmental specification is used to evaluate an abstract vehicle design.

`Models/Composite/FluidThermal/doc/Cold Start Demo Notes.docx`

RUN SIX WATT STAND ALONE EXAMPLE

The run six watt stand alone example is a demo example that connects to the terrain server and pulls a synthesized terrain profile for simulation analysis. The terrain examples all employ a terrain generator shown in Figure 1.

`Models/Land/terrains/grades_slopes/doc/Demo_RunSixWattStandAlone.docx`

DISCRETE OBSTACLES EXAMPLE

The Discrete Obstacles Pitch Plane demo example demonstrates the use of discrete obstacle context with a simulation model to determine the go/no-go performance of the design.

`Models/Land/obstacles/doc/Demo_DiscreteObstaclesPitchPlane.docx`

DRIVER SPECTRA AND OBSTACLES EXAMPLE

The driver spectra and obstacles example demonstrates the use of a spatial design model with suitable spatial context.

`Models/Land/terrains/grades_slopes/doc/Demo_DriverSpectraNObstacles.docx`

DRIVER LIMITED SPEED SPECTRA EXAMPLE

The Driver Limited Speed Spectra demo example demonstrates the generation of a synthetic terrain for simulated model crossing at speed.

`Models/Land/terrains/grades_slopes/doc/Demo_DrvLmtdSpdSpctr.docx`

FUEL EFFICIENCY EXAMPLE

The Fuel Efficiency demo example demonstrates the generation of a synthetic terrain for use in a fuel efficiency calculation for design evaluation.

`Models/Land/terrains/grades_slopes/doc/Demo_Fuel_Efficiency.docx`

AMPHIBIOUS CONTEXT MODEL

To determine the reserve buoyancy, range and stability of an amphibious vehicle within a water condition an example vehicle was chosen.

```
Models/Aquatic/properties/buoyancy/Summary of Amphibious Context Model.docx
```

SOIL TRACTION EXAMPLE

Demonstrate how a soil cone model can be used to determine the soil cone index with a given humidity value, along with a simple application to demonstrate how a mobility model can be used to determine the rolling resistance and the theoretical max speed of a vehicle.

```
Models/Land/doc/SoilConeIndexDemo.docx
Models/Land/doc/MobilityDemo.docx
```

GENERATING OBSTACLES AND TERRAINS EXAMPLES

The generation of discrete obstacles parameterized with static friction and other properties is demonstrated for two specific geometries, along with a random terrain generator.

```
Models/Land/obstacles/doc/DiscreteObstacleDemo.docx
Models/Land/obstacles/doc/AngleDitchCrossingDemo.docx
Models/Land/terrains/grades_slopes/doc/TerrainGeneratorDemo.docx
```

COMMON FEATURES

Several of the examples use PSD representations to synthesize terrain (see Figure 1). For discrete obstacles, deterministic profiles were used.

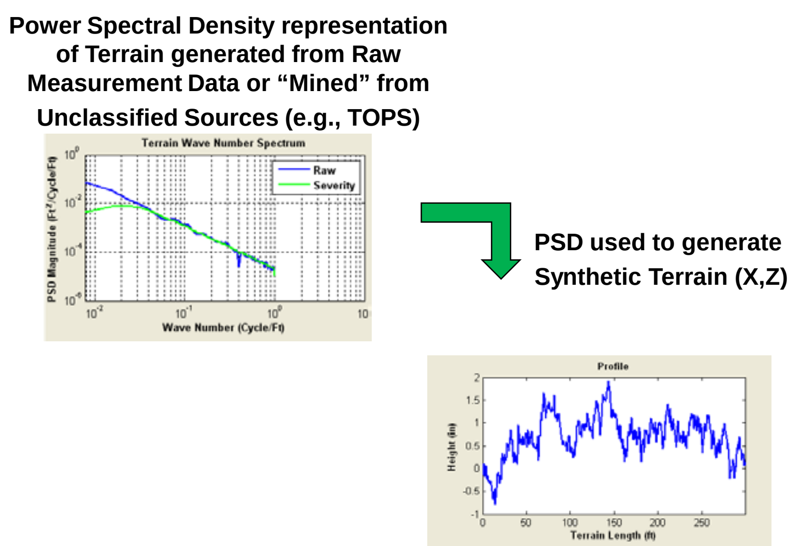


Figure 1: PSD to terrain generation

COLD START DEMO

This simple code demonstrates the usage of the environmental context models for C2M2L-1. In particular, the AR-70-32 environmental specification is used to evaluate an abstract vehicle design. The situation is similar to a typical US military vehicle requirement verification test. The vehicle is conditioned to temperatures corresponding to a climate type indicated in AR-70-32 and starting attempted. In the verification test, the vehicle must be "combat ready" within a certain time limit. After the vehicle starts, the engine warms up to an operational threshold, the electronics and computer systems are booted and ready. In this example, the "vehicle" is deemed able to start and combat ready if the available battery power is equal to or greater than the power needed to turn over the engine.

LIMITATIONS OF THE COLD START DEMO CODE:

In general, the code is in the alpha stage. Limited testing has occurred but is has not been finished on all aspects of the models. Documentation is incomplete. Some inputs are parameterized in an input file, but the vehicle architecture is hard-coded and unalterable except through recompilation of the code. The code was developed using the Microsoft Visual C++ 2010 compiler. Some of this code uses features of the C++ 11 language standard not available in earlier compiler versions.

The vehicle is abstract. Nothing is assumed or modeled about the internal heat generation of components due to running the engine, engine heaters, or other sources. The vehicle model is a simple thermal network solver that uses lumped-mass approximations for the individual components. The vehicle data is local to the simulation. The vehicle behavior is local to the simulation. Vehicle data and behavior will ultimately be defined within META.

The thermal network solver used to define the vehicle's response to the environment is limited in functionality. A 4th order Runge-Kutta solver is used, but with a fixed time step. Full vehicle models would be sufficiently complex that an automatically adjustable time step solver would be preferred to keep the simulation time to a reasonable limit. The thermal network solver does not yet poses a feature to allow component heat generation, a critical element in the thermal architecture of a vehicle. The thermal network solver does not possess interfaces for more complex components. For example, a 1D or heat conduction model of an insulated vehicle wall would be useful in some situations.

Code for the thermal and fluid context models included, but the feature set is incomplete. Humidity information, which is specified in AR-70-38 is not yet available.

The Alglib math library used in this example has been replaced in more recent versions of the fluid and thermal context models.

Figure 2: Cold Start GUI

RUN SIX WATT STAND ALONE EXAMPLE

The run six watt stand alone example is a demo example that connects to the terrain server and pulls a synthesized terrain profile for simulation analysis. The demo example accepts an input file describing the terrain spectra to be analyzed and some user configurable design model parameters. The simulation model is repeatedly exercised across the terrain until a limit speed is reached. The limit speed is either a six watt average absorbed power limit or an upper bound speed set in the code. The simulation outputs two files, one providing chart data showing driver absorbed power as a function of terrain crossing speed, the other listing the driver absorbed power and terrain rms value. The terrain server must be started prior to calling the six watt stand alone model. The terrain server window is shown below.

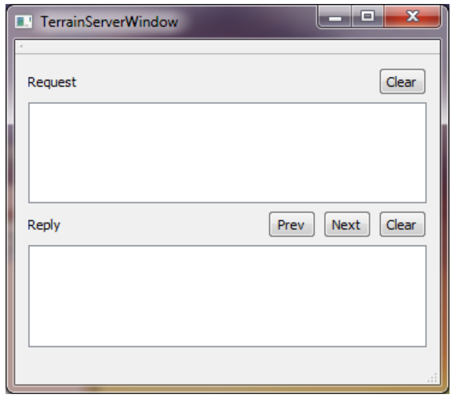


Figure 3

The call to the six watt stand alone demo example accepts input arguments. The first input argument is taken to be the model input file. A sample input file is shown below. The model input is kept relatively simple for this demo. The parameters listed below from ServerHost to seed are specifications for the call to the terrain server to retrieve a terrain profile. The terrain profile requested in this case is spectral representation number 20 "Perryman 3", with a synthesized length of 30 meters at 0.025 meter resolution. Additionally, the context model is seeded with a seed value of 1. The sprungmass and wheelposition values are design model parameters. The design model in this case consists of a 1000 kilogram sprung mass supported by three wheels on each side. The simulation model is only in the pitch plane for this example as the point is to illustrate the effect of design and context model changes on performance.

```
<?xml version="1.0" encoding="UTF-8"?>

<ModelInput>
  <ServerHost string_val="127.0.0.1"/>
  <ServerPort int_val="8080"/>
  <spectral int_val="20"/>
  <length double_val="30"/>
  <int double_val="0.025"/>
  <seed int_val="1"/>
  <SprungMass double_val="1000"/>
  <WheelXPosition.0 double_val="1.67258"/>
  <WheelXPosition.1 double_val="2.67258"/>
  <WheelXPosition.2 double_val="3.67258"/>
 </ModelInput>
```

The simulation model exercises the design model repeatedly across the terrain profile at increasing speeds. At the end of each crossing, the driver average absorbed power is examined and recorded. The speed is increased until the limit speed is achieved. The results are placed in two output comma separated value files. The names of the files can be passed as the optional second and third arguments to the model call.

Influences of either design changes or context model changes can be studied and quantified by changes to the input file. Suggested changes for the context include the selection of different terrain spectra profiles, different terrain lengths, and different terrain seeds. Potential changes for the design model include changes to the sprung mass, the number of supporting wheels, and the spacing between the supporting wheels.

DISCRETE OBSTACLES EXAMPLE

The Discrete Obstacles Pitch Plane demo example demonstrates the use of discrete obstacle context with a simulation model to determine the go/no-go performance of the design. Course profiles and discrete obstacle profiles are loaded individually and the simulation model is exercised at a user-controlled speed across the terrain. Success or failure to cross the terrain profile is observed from the simulation graphic.

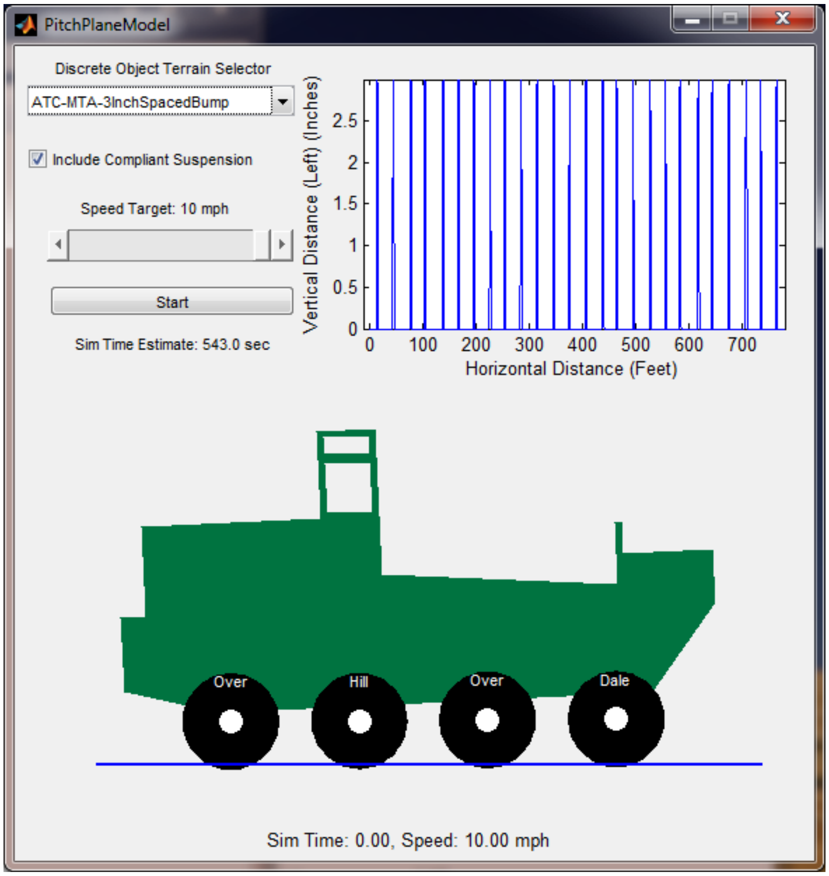


Figure 4

The demo GUI, shown above, consists of one plot, the simulation graphic, and four user interface controls. The demo combines the terrain context modeling with a simple design simulation model. The simulation model is a multiple degree-of-freedom pitch plane model that queries the terrain at multiple points beneath the wheels. This design model terrain contact interface is not constrained to the terrain and can separate.

The discrete object terrain selector in the upper left corner select terrain context from the available courses and profiles and loads them individually into the simulation model. The include compliant checkbox changes the simulation model from having a compliant suspension when checked, to a non-compliant suspension when unchecked. Performance differences can be observed in the simulations to the selection of compliant and non-compliant. This option is not selectable during run-time. The simulation speed target is user-adjustable before and during run-time. The simulation start pushbutton starts the model execution. The model execution will stop after an estimated terrain crossing time is reached or when the user presses the simulation start pushbutton again.

The simulation graphic and terrain profile plot are updated continuously during the simulation. The terrain profile displays a windowbox to indicate the current portion of the terrain profile that the simulation model is in. This is the terrain portion shown in the simulation graphic window on the bottom. The design response to the terrain context can be viewed here as the simulation progresses. Failure to cross/traverse an obstacle/terrain profile is observed as a stagnant design model, or a design model in an undesirable configuration (such as inverted).

DRIVER SPECTRA AND OBSTACLES EXAMPLE

The driver spectra and obstacles example demonstrates the use of a spatial design model with suitable spatial context. The demo example can use either discrete obstacles, profiles, or generate synthetic terrain from the spectral representations. The demo uses context in a manner similar to the Driver Limited Speed demo example when selecting terrain spectra. The demo example generates synthetic terrain with wavenumber content from the reference spectra profile. The demo just loads up the obstacle or course profile content when selecting a discrete obstacle or profile, similar to the Discrete Obstacles Pitch Plane demo model. The key difference is the use of a spatial design model to assess the performance across the context.

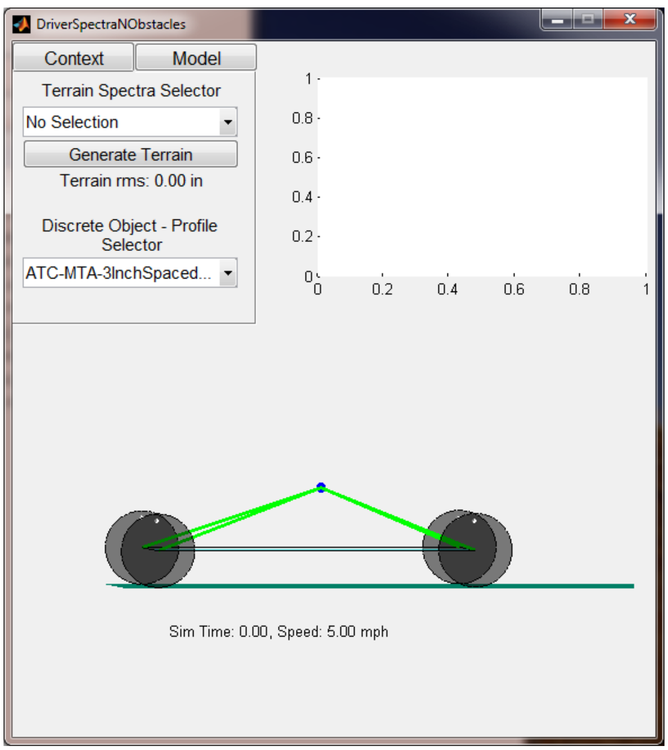


Figure 5

The driver spectra and obstacles gui, with initial appearance shown above, consists of context and model control panels, a context profile window, and a simulation graphic. The context panel, displayed in above graphic, permits the selection of either terrain spectra or discrete object profiles. If a spectral terrain representation is selected, terrain must be generated before continuing to the model panel for simulation. The model panel contains the controls for interaction with the design model, displayed in the figure below. The include compliant checkbox changes the simulation model from having a compliant suspension when checked, to a non-compliant suspension when unchecked. Performance differences can be observed in the simulations to the selection of compliant and non-compliant. This option is not selectable during run-time. The simulation speed target is user-adjustable before and during run-time. The simulation start pushbutton starts the model execution. The model execution will stop after an estimated terrain crossing time is reached or when the user presses the simulation start/stop pushbutton again.

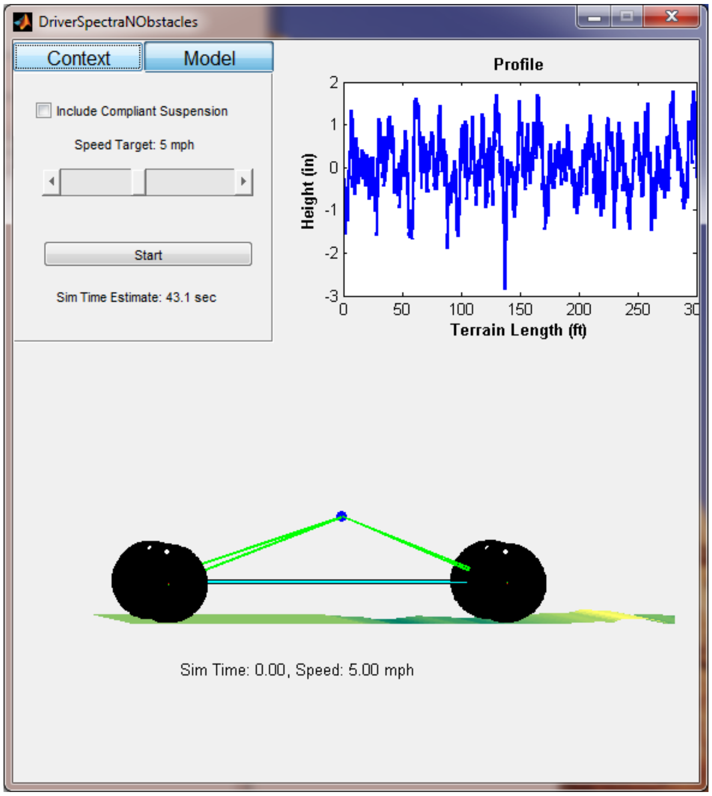


Figure 6

The simulation graphic and terrain profile plot are updated continuously during the simulation. The terrain profile displays a windowbox to indicate the current portion of the terrain profile that the simulation model is in. This is the terrain portion shown in the simulation graphic window on the bottom. The design response to the terrain context can be viewed here as the simulation progresses. Failure to cross/traverse an obstacle/terrain profile is observed as a stagnant design model, or a design model in an undesirable configuration (such as inverted).

DRIVER LIMITED SPEED SPECTRA EXAMPLE

The Driver Limited Speed Spectra demo example demonstrates the generation of a synthetic terrain for simulated model crossing at speed. This demo illustrates the coordination of context modeling and system modeling for simulated evaluation of design performance against the requirement of driver limited speed. The driver limited speed requirement is the evaluation of the average absorbed power to the driver for crossing severe terrains at speed. There exists a power limit at which the driver will slow down for a smoother ride versus subjecting him/herself to excessive vibration. The limiting speed is typically termed 6 watts and the design will likely be required to cross a specific severity terrain at a speed to keep combat formation.

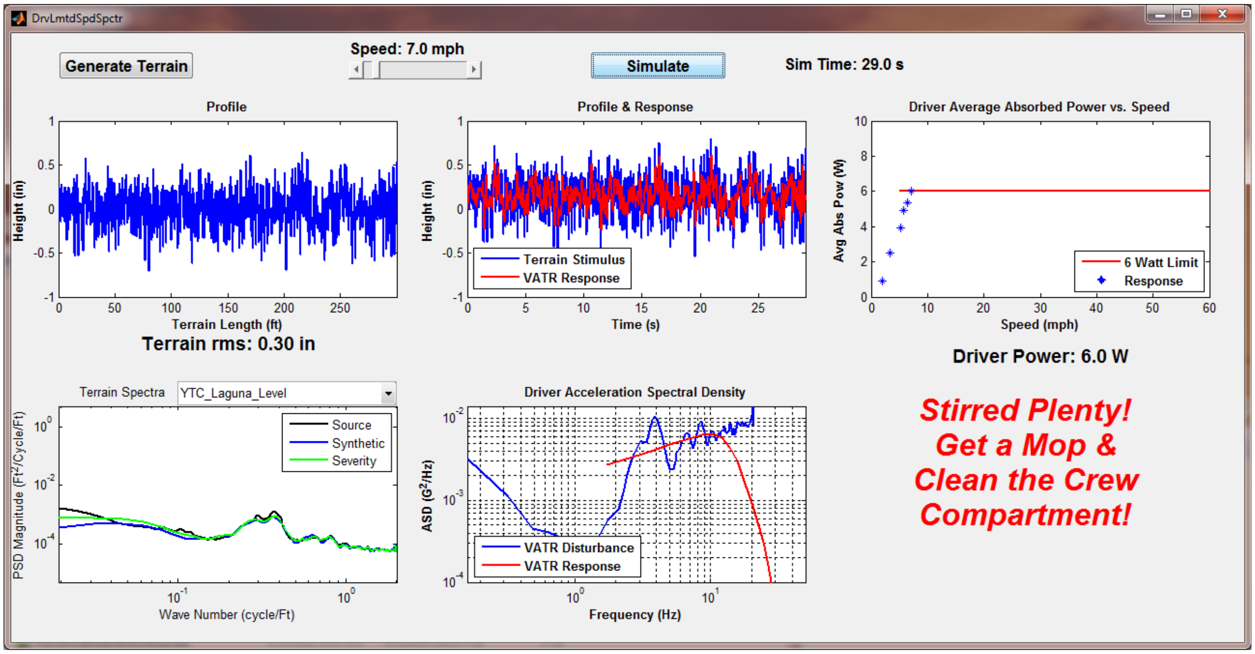


Figure 7

The driver limited speed demo GUI, shown above, consists of five plots and four user interface controls. The demo combines the terrain context modeling with a simple design simulation model. The simulation model is a single degree-of-freedom sprung mass model that queries the terrain at a singular point. This design model terrain contact interface is constrained to the terrain and does not separate.

Terrain spectra is selected in the lower left pull-down menu from the terrain spectra JSON files in the data directory. The "Generate Terrain" pushbutton synthesizes a 300 foot terrain length containing spectral wavelength content shown by the blue curve in the lower left plot, with similar content to the reference black curve. This range of wavelength content represents the wavelength relevant to terrain severity calculation. The terrain severity is shown by the "Terrain rms" value between the upper and lower left plots. This particular example synthesizes terrains within that rms range and contains spectral content along the logarithmic line shown in the lower plot. Note that the terrain profile consistently contains similar spectral content, yet is not unique in the terrain profile plot with each press of the "Generate Terrain" pushbutton. The green plot in the wave number spectrum represents the detrended terrain content that is used in the severity calculation (the Terrain rms number).

The synthetic terrain shown in the upper left is used as context input to the simulation model. The "Speed" slider control adjusts the model's speed of crossing the terrain profile. The terrain profile is turned into model stimulus by crossing the terrain at the specified speed.

The "Simulate" pushbutton simulates the system response to the context model with the stimulus and response shown in the central top plot in the GUI. The driver acceleration spectral density and corresponding stimulus input is shown below that. The average absorbed power to the driver is then calculated and plotted as a data point in the right plot for comparison to the six watt limit. The intent is to adjust the driven speed to find the limit speed at which the six watt line is crossed. Suggestions to speed adjustment are provided in the lower right, below the most recent simulated driver absorbed power value.

FUEL EFFICIENCY EXAMPLE

The Fuel Efficiency demo example demonstrates the generation of a synthetic terrain for utilization in a fuel efficiency calculation for design evaluation. This demo shows the coordination of context modeling and system modeling for simulated evaluation of design performance against the requirement of vehicle range with on-board consumables. The vehicle range is specified in program requirements and is limited to on-board fuel as the consumable source. The evaluation of compliance to this requirement is a necessary design effort. This demo illustrates the coupling of context and system modeling with a simple arithmetic design model.

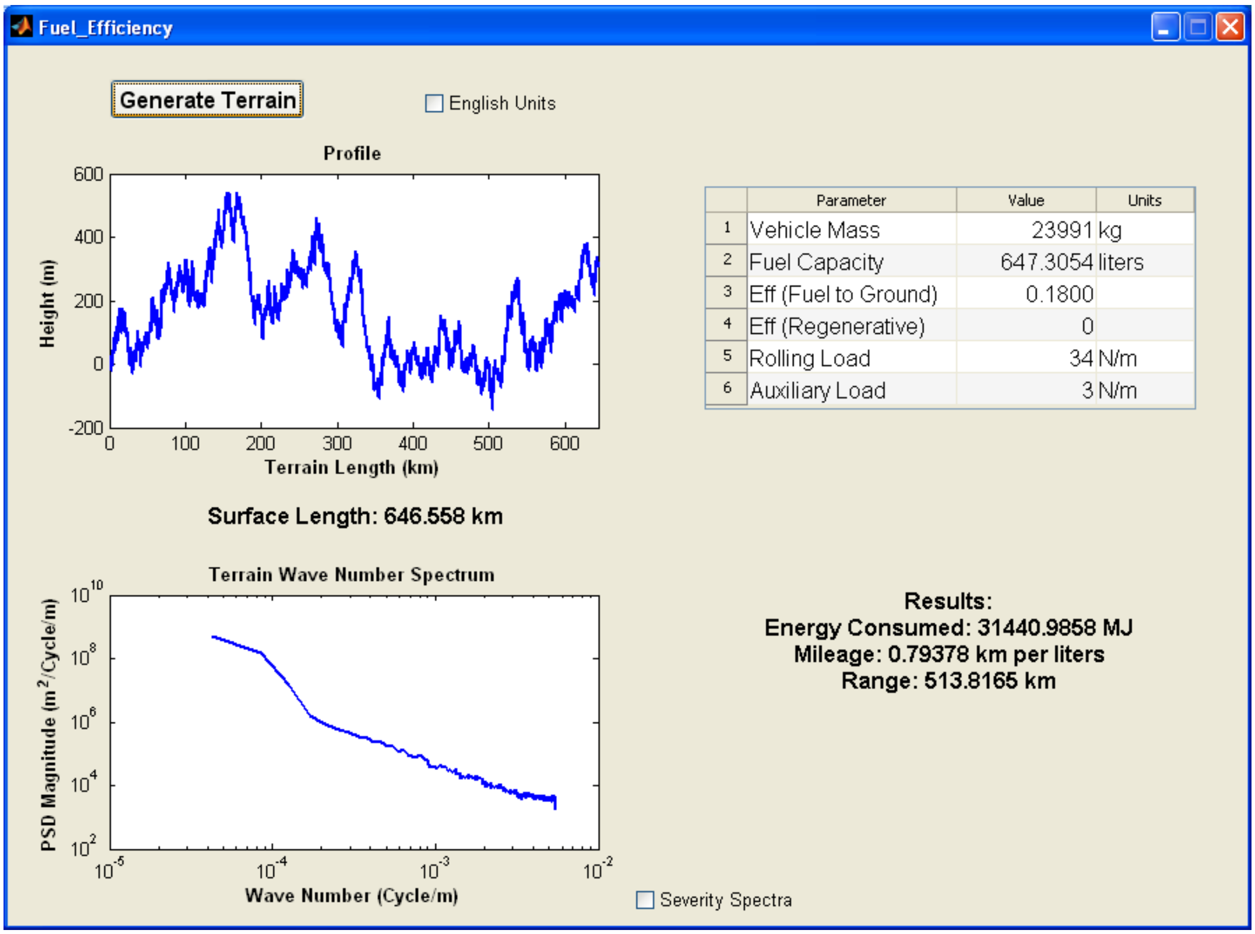


Figure 8

The fuel efficiency demo GUI, shown above, consists of two plots, a results statement, and four user interface controls. The demo combines the terrain context modeling with a simple design simulation model. The simulation model is an arithmetic calculation based on the synthetic terrain and design parameters.

The “Generate Terrain” pushbutton user interface control synthesizes a 650 km terrain from a probabilistic density function fitted to CONUS DEM data. This is the context model. It is displayed in the upper left plot as a terrain profile. It changes with every press of the “Generate Terrain” button from a series of randomly generated seed values.   The terrain is synthesized from a random walk on the probability density function.

Design model influential characteristics of the terrain profile are extracted and design efficiency results are displayed in the lower right. The energy consumed, mileage, and design range are shown for the design parameters in the upper right. The energy content of the fuel is coded into the example for the results calculations.

The wave number spectrum is shown in the lower plot for continuity with the driver limited speed example. The wave numbers are limited to long wavelengths that represent large changes in slope and elevation. This plot introduces users to the concept of wave spectrum for terrains.

A units checkbox is provided to express the model in English or Metric units. When the model is in the English units configuration, the “Severity” checkbox will permit the severity spectra of the generated terrain to be displayed (green plot). This capability illustrates separation of the wave number ranges for severity and fuel efficiency. The relative intensity of the severity spectra is substantially lower than the synthetic terrain generated.

Model parameters can be set in the upper right dialog to determine the influence and sensitivity of the model parameters.

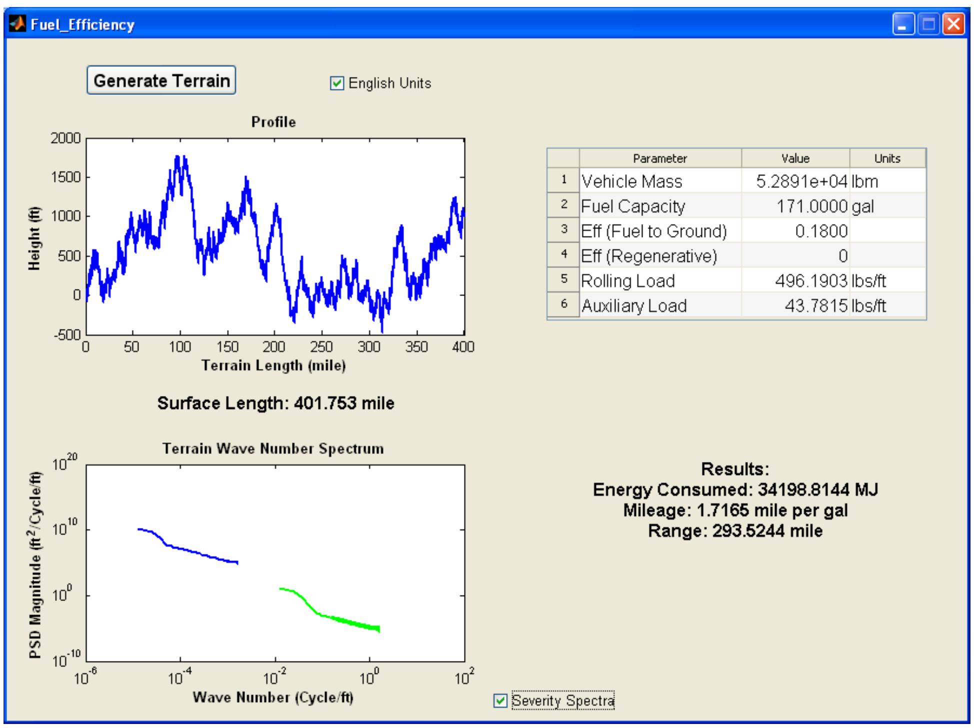


Figure 9

AMPHIBIOUS CONTEXT MODEL

In order to determine the reserve buoyancy, range and stability of an amphibious vehicle within a water condition an example vehicle was chosen. The example vehicle is a commercial amphibian (hydrotrek) whose weight and dimensional values are obtainable to the public online. This vehicle was set up within Pro/Engineer and created in such a way that the major dimensions were parameterized. Any change to one of the parameters will adjust the rest of the model through relation calculations. This allows for several size conditions to give the outputs needed for the buoyancy and stability calculations. The inputs needed are length, width, height, wheel diameter and either 2 or 4 occupants. The outputs produced are total volume, center of gravity, and vehicle mass. Again these outputs are based on relations to the initial model.

Once the outputs are obtained, either through the Pro/e model or direct inputs, visual basic code is utilized to determine the reserve buoyancy, range and stability of the model.

- For the buoyancy calculations fluid type, temperature and elevation can be taken into account or can be left as the baseline values. The temperature and elevation adjust the result slightly while the fluid type can have a larger effect if it is a mixture. Options are in place in order to allow for 5 different particulate materials mixed with fresh and/or salt water to create the fluid mixture. The correct fluid density is determined by the percentage of each.
- Along with fluid density several other outputs are calculated including object density, buoyant mass and buoyant force, but reserve buoyancy is most important. The reserve buoyancy is the percentage of the vehicle which remains above water therefore as long as the value is positive the vehicle will float. If the value is negative, than the vehicle will be completely submerged under the water.
- In order to calculate the range and maximum speed of the amphibious vehicle within the water several inputs need to be known. These include, fuel efficiency, fuel tank capacity, run time, power, drag coefficient and reference area. A table is provided for several different frontal geometry configurations to estimate the drag coefficient. Range is given in both miles and nautical miles and speed is given in miles per hour as well as knots.
- The stability of the model is determined by using the weight, center of gravity and the reserve buoyancy of the vehicle in self righting calculations. The calculations use the maximum width and height of the vehicle as a bounding box in order to determine the stability curve. The stability curve shows at any particular heel angle, or angle at which the vehicle is tilted in the water, the likelihood that the vehicle will right itself or will tip over. If the slope of the curve is positive at the angle of interest than the vehicle will right itself back to equilibrium. If the slope is negative the angle will continue to increase and the vehicle will tip over. The static attitude, or forward tilt angle, of the vehicle is also calculated. This is done using a bounding box for the length and height and depends on the height of the water and the location of the center of gravity.

Appendix P contains more details on buoyancy properties. The files describing the Pro-E model and calculations for buoyancy, reserve buoyancy, and stability can be found in following location in SVN: \Models\Aquatic\properties\buoyancy.

SEA-STATE PROTOTYPE

Due to the effect of waves, an additional sea-state-influenced buoyancy example was prototyped

```
./Aquatic/features/sea_state/test/PotentialFlowSeaApp
```

<u>Inputs</u>:

- Sea State
  - Implies values for significant wave height, wave period and wave length
  - Limited to 0-3. Non-linear effects, which are not modeled, participate in higher sea states.
- Vessel heading, radians – can change during simulation, affects wave encounter frequency
  - Vessel speed, m/s – can change during simulation, affects wave encounter frequency
  - Position:

- x (m), y(m), z(m)
- Time: from simulation, seconds

Outputs:

- Surface height, η(x, y, t), m
- Pressure, P(x, y, z, t), Pascals
- Water velocity components u(x , y, z, t) , v(x , y, z, t) , w(x , y, z, t), m/s

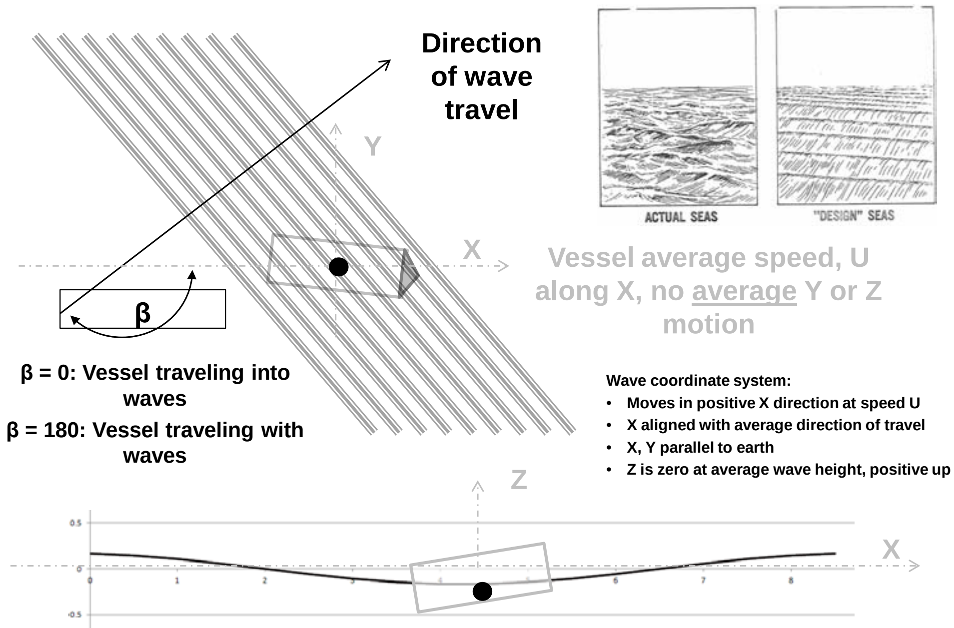


Figure 10: Wave coordinate system geometry

The introduction of waves to buoyancy requires dynamic modeling as shown below:

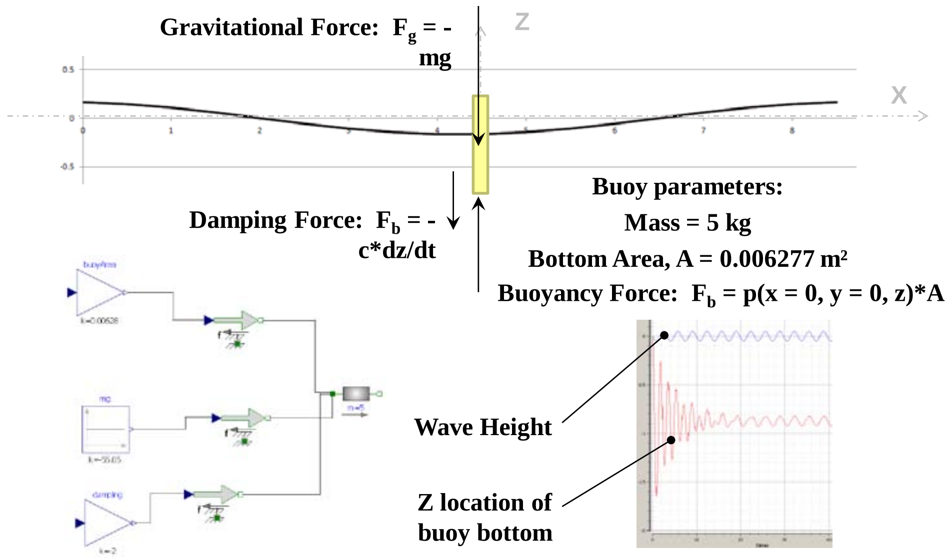
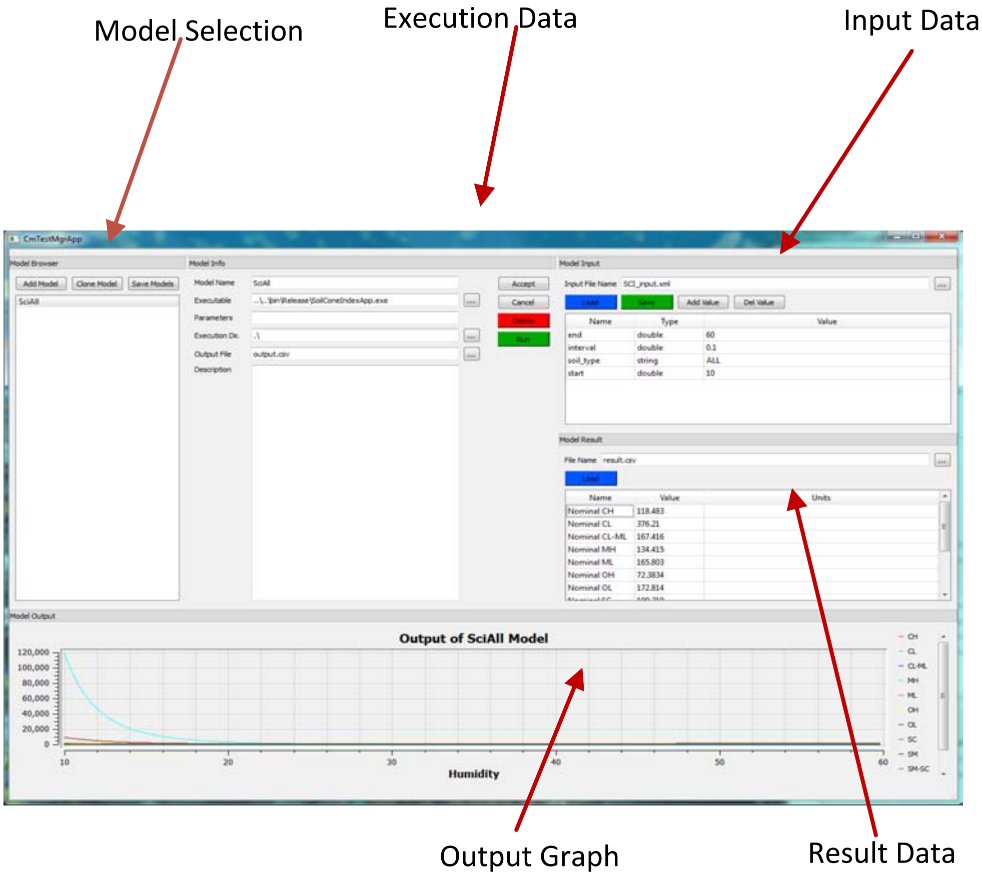


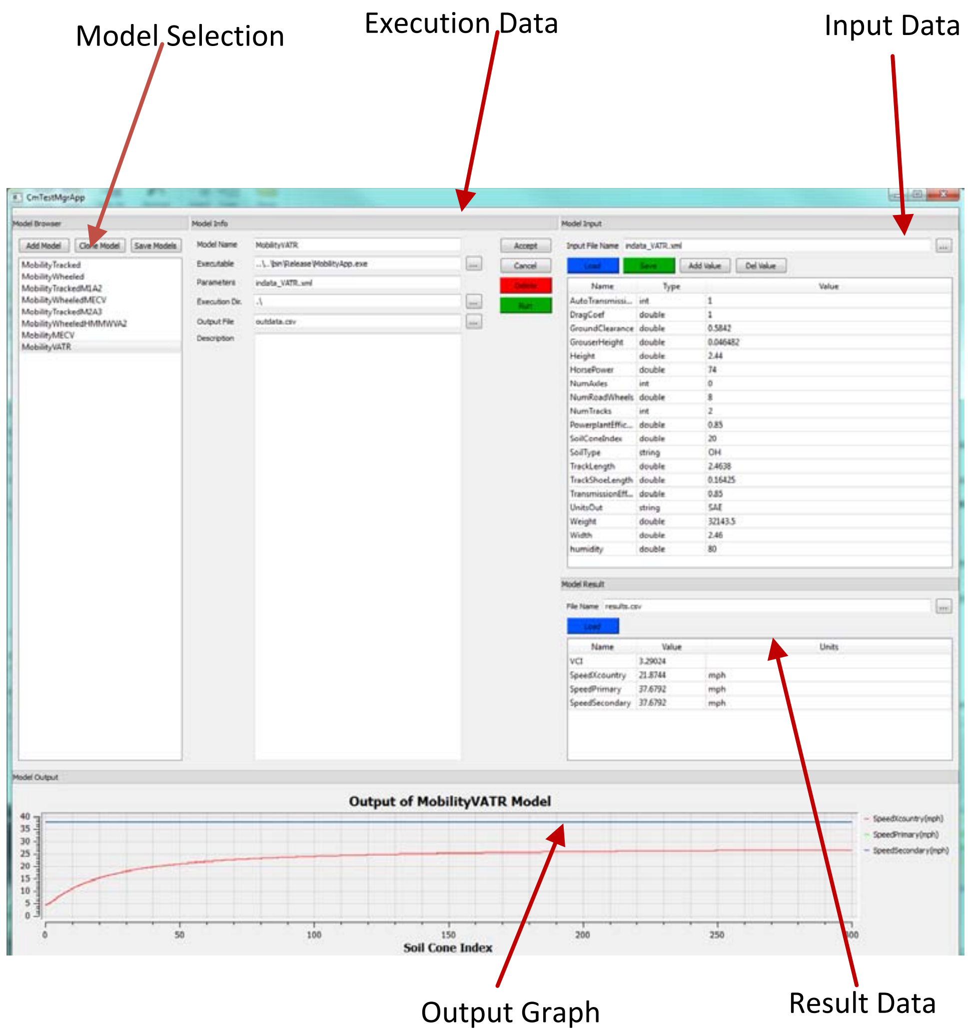
Figure 11: Example of dynamic output

SOIL TRACTION EXAMPLE

The Soil Cone Index Demo is a simple application to demonstrate how soil cone model can be used to determine the soil cone index with a given humidity value. It can be run as a stand-alone application or within the demo GUI as seen below. The GUI allows the user to create multiple data sets to be able to show multiple runs of the same demo executable or runs of different demo executables.

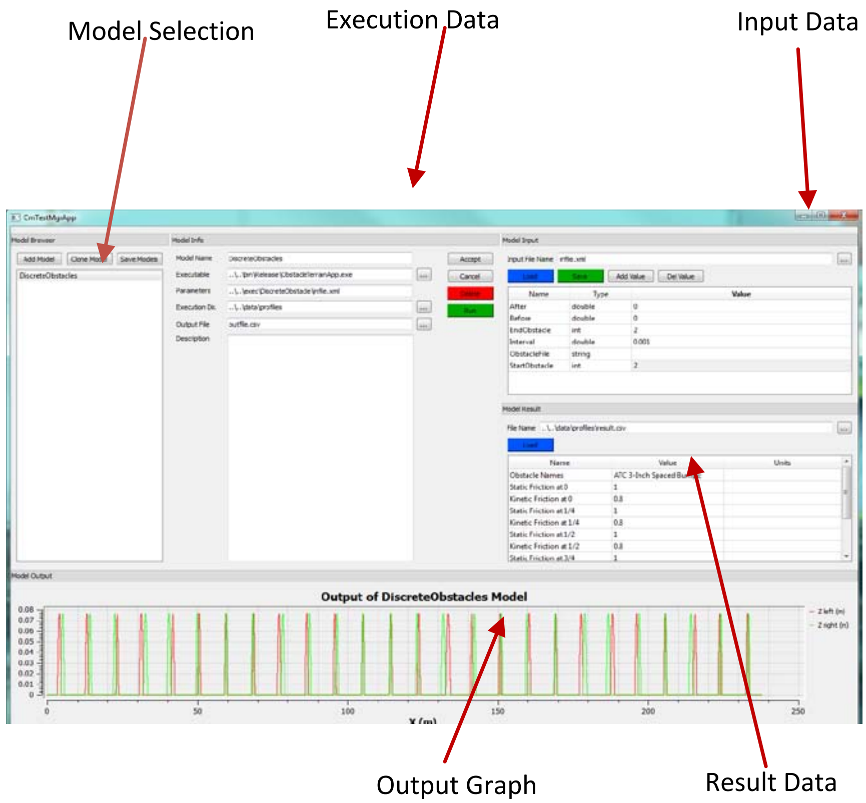


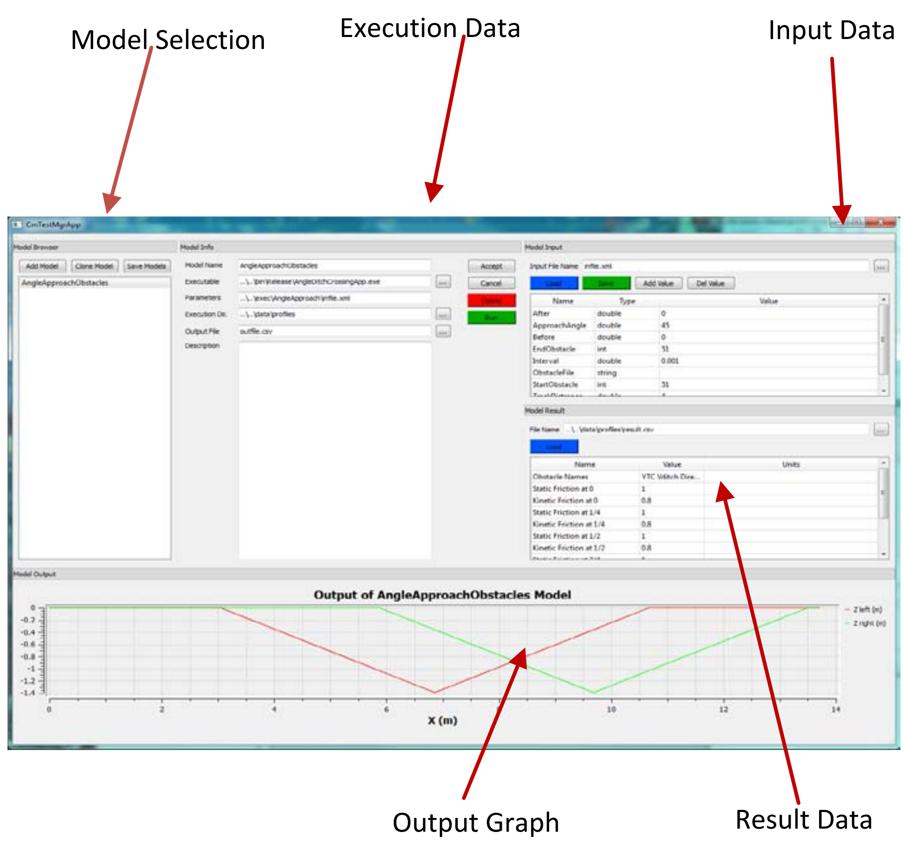
The Mobility Demo is a simple application to demonstrate how mobility model can be used to determine the rolling resistance and the theoretical max speed of a vehicle. It can be run as a stand-alone application or within the demo GUI as seen below. The GUI allows the user to create multiple data sets to be able to show multiple runs of the same demo executable or runs of different demo executables.



GENERATING OBSTACLES AND TERRAINS EXAMPLES

The Discrete Obstacle Demo is a simple application to demonstrate how the discrete obstacle model can be used to create terrains. It can be run as a stand-alone application or within the demo GUI as seen below. The GUI allows the user to create multiple data sets to be able to show multiple runs of the same demo executable or runs of different demo executables.


Model Selection
Execution Data
Input Data
Output of DiscreteObstacles Model
X (m)
Output Graph
Result Data


Model Selection
Execution Data
Input Data
Output of AngleApproachObstacles Model
X (m)
Output Graph
Result Data

The Terrain Generator Demo is a simple application to demonstrate how the spectral terrain model can be used to generate terrains.