Using MATLAB with CANoe

Version 2.0
2017-03-09
Application Note AN-IND-1-007

Author Vector Informatik GmbH

Restrictions Public Document

Abstract This application note describes the usage of MATLAB®/Simulink® combined with CANoe. It covers the basic principles of the CANoe/MATLAB Interface and gives an overview over the various use cases.

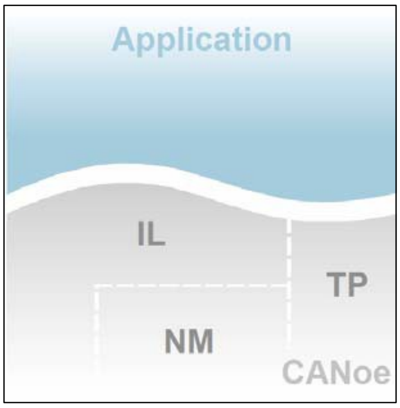
Table of Contents

1.0 Overview

This application note covers the principles and usage of the CANoe/MATLAB® Interface. The purpose of this interface is to extend CANoe's node modelling capability by adding the strength of the MATLAB®/Simulink® environment. It allows execution of Simulink® models inside the CANoe network simulation environment. Currently all MATLAB® versions starting with R2007b (MATLAB® 7.0) are supported. The CANoe/MATLAB® Interface is delivered together with CANoe. The setup program is located in the folder **<...>\Installer Additional Components\Matlab** of any CANoe installation.

2.0 Connection of CANoe and MATLAB®/Simulink®

2.1 Modeling Concept



It is important to understand how this interface communicates with CANoe in order to easily design a model. An ECU is modeled in CANoe using several software layers. Typically the application behavior is described with CAPL (Communication Access Programming Language) which is a C style programming language. This application layer uses interfaces of middleware layer that provides common services for network nodes.

It is easiest to think of the CANoe/MATLAB® interface as an extension or a complete replacement of existing CAPL code. The application of a CAPL node can be completely implemented, i.e. modelled with Simulink®.

Application Layer

The basic application behavior of an ECU should be defined with special emphasis on its functional bus behavior. The application of an ECU has two major interfaces:

- The interface to bus signals
- The interface to peripheral I/Os like sensor and actuators. In CANoe the interface to I/Os is modeled using environment variables or system variables.

With the CANoe/MATLAB® interface only the application behavior must be modeled with Simulink®. No hardware specific or bus system specific function blocks are necessary. Thus the application behavior is modeled completely separated from common network services.

Middleware

Apart from the application behavior of an ECU, there are parts of the software which are identical for each network node, such as interaction layers (IL), network management services (NM), diagnostic services and transport protocols (TP).

These layers are available as CANoe modeling libraries for all major OEMs.

Interaction Layer

An important part of the software located between an ECU's application layer and lower-level functions is the interaction layer (IL) with its signal interface. It is important because applications deal with named signals (bus signals) instead of their networked representation in bits and bytes of the data stream. This interaction layer performs mapping between the signals and their network representations.

The Vector CANoe Interaction Layer (in short CANoeIL) provides a signal-oriented means of accessing the bus. The CANoeIL also performs mapping of signals to their send messages and controls the sending of these send messages as a function of the send model. Different send models are defined depending on the network type (OEM specific), and a special CANoeIL is provided to each of these send models.

It is possible to map model outputs and inputs directly to signals. This mapping causes the changed value of the model to be routed directly to the specific CANoeIL responsible for sending the value out. The CANoeIL also stimulates the model’s input with the changed value.

Network Management Layer

The availability of the CAN bus is handled by network management. Typical features of an OSEK NM include:

- Identifies the network configuration at start-up
- Monitors the network configuration while the bus system is running
- Synchronizes transition of all network nodes to bus sleep mode (power saving mode if network is not needed)
- Controls peripheral hardware (CAN Controller and Bus Transceiver)
- Provides network-relevant status information
- Error recovery after bus-off

Each ECU is identified by a unique station number and has a special message identifier for exchanging network-relevant information. This message contains identification of the transmitting node (encoded in the CAN identifier), the address of the receiving node as well as the message type and additional sleep flags.

Other Network Management types such as AUTOSAR NM may work differently but are also available as a component for CANoe.

Transport Layer

Some information to be transmitted over the CAN bus does not fit into a single data frame because the data length exceeds 8 bytes. In such cases the sender has to split up the data into several CAN messages with the same identifier. Additional information is necessary to re-assemble the data in the receiver.

This is performed by the transport layer:

- Segmentation and reassembly of data that is larger than the underlying data link layer
- Flow control for single messages
- Error recognition

A transport layer is not only needed for diagnostics purposes but also for any large data which must be exchanged between different nodes, e.g. text information to be displayed on a dashboard.

2.2 Interface Concept

The interface consists of a block set for MATLAB®/Simulink® and the Simulink® Coder. It provides data exchange with CANoe for simulations running inside Simulink® and assures time synchronization between both tools. There are two different approaches of time synchronization with co-simulation:

1. The Simulink® time base is used for the CANoe simulation. This will be referred to as the **Offline Mode**.

2. The (real-time) time base of CANoe is used for the Simulink® simulation, which is called the **Synchronized Mode**.

The interface also provides a target file for the Simulink® Coder. Using this target file a Simulink® model will be compiled as a DLL which runs in the CANoe environment (called **HIL Mode**). Except setting the CANoe target file and compiling the model no other action is needed to create a DLL which provides best runtime behaviour and exactness of timings. Although the model is compiled, parameters of Simulink® blocks can be changed from within CANoe. For debugging purposes Simulink® signals can be easily analysed in CANoe. The Simulink® model can be viewed using CANoe’s built in model explorer. No MATLAB®/Simulink® license is needed for running or viewing a compiled model.

Besides sample applications the interface comes up with integration to the MATLAB®/Simulink® Model Advisor. The Model Advisor allows checking the model for proper configuration with the current CANoe simulation. This assures that sources and sinks provided by CANoe exist and will work properly.

Contact Information: www.vector.com or +49-711-80 670-0

2.3 Data Exchange

Data exchange between CANoe and MATLAB®/Simulink® is provided by a library for Simulink®. It contains blocks for the following CANoe elements:

- Signals
- Environment variables
- System variables
- CAPL functions

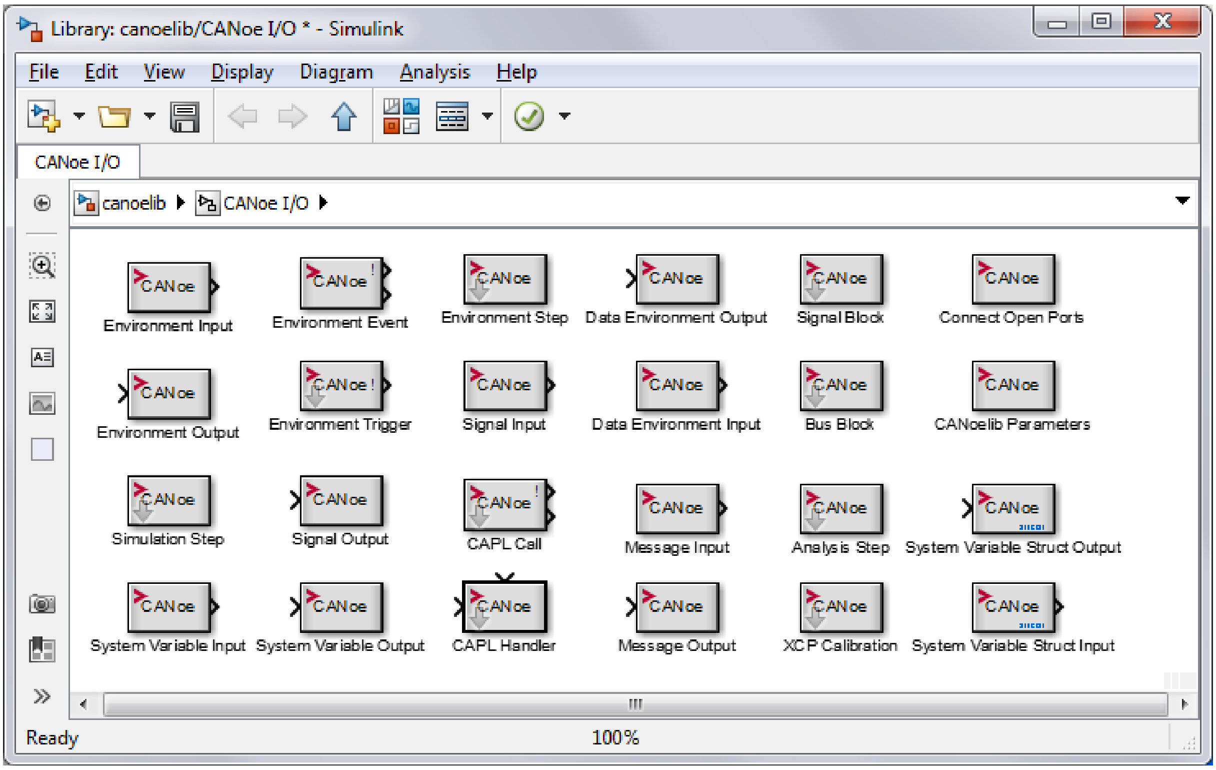


Figure 1: CANoe block set library for Simulink®

The blocks are used as sources and sinks for Simulink® models. They provide the current value of the according CANoe element. The signal input block for example returns the current value of the last CAN[1] message containing the signal. The block set is a very easy way to extend MATLAB®/Simulink® models with the ability to communicate with real bus devices. CANoe provides network management and interaction layer functionality so that the models are focused on the application logic and are separated from hardware access functions.

You may also connect the inputs and outputs of your model to CANoe System Variables. System Variables are used for many internal features as well as can be defined by the user. For example, with the option XCP, ECU internal memory addresses are mapped to System Variables. Another example is the channels of connected I/O HW such as the VT-System. Thus the Simulink® model can be directly connected to I/O lines.l

When a simulation is run in Simulink® the blocks are using the CANoe COM interface to configure which data they will provide. The data itself is transmitted via shared memory during the simulation. This provides fast transmission with very little communication overhead.

If the Simulink® Coder is used, data exchange is realized using a C++ Interface of CANoe. Therefore the generated model DLL is executed with the same performance and timing exactness as if it would be a built-in component of CANoe.

[1] The signal blocks handle all CANdb, FIBEX and ARXML signals so that the bus systems CAN, LIN and FlexRay are supported.

2.4 Execution Modes

Simulations utilizing the CANoe/MATLAB® Interface can be operated in three different modes. The following chapters introduce them exactly and explain their focus.

2.4.1 Hardware-In-The-Loop (HIL) Mode

In this mode the simulation is run in the CANoe execution environment. With the Simulink® Coder you can target CANoe and create a Windows® DLL which can be loaded in CANoe's simulation environment. With this approach it is possible to test and verify a design with real (CAN, LIN, FlexRay, I/O) hardware in a networked environment. The simulation can be used as a remaining bus simulation in a real-time environment. Several ECUs can be simulated simultaneously within a single CANoe environment. This mode is recommended if it is necessary to run huge models and the remaining bus in real time or the CANoe simulation including the model should be executing in standalone mode without connection to MATLAB® Simulink®.

Requirements: Simulink® Coder and Microsoft® Visual Studio compiler.

2.4.2 Offline Mode

In this mode the simulation is run in the MATLAB® Simulink® environment. CANoe is operated in slave mode whereas MATLAB® Simulink® is the simulation master. This is the mode for non-real-time simulation. The simulation is controlled from the MATLAB® Simulink® environment.

CANoe's slave mode is a special simulation mode in which CANoe is simulating the (CAN,LIN, FlexRay) bus and takes its measurement time base from outside (here: from Simulink®). This is especially useful in early design stages. Debugging features of the MATLAB® environment can be used. Communication between MATLAB® Simulink® and CANoe takes place over Microsoft® COM for configuration and shared memory for data exchange.

No real-time simulation is provided with this kind of simulation. The simulation is run as fast as possible. Therefore no real hardware is needed and development cycles are short.

Figure 2: Synchronization in offline mode

2.4.3 Synchronized Mode

In this mode the simulation is run in the MATLAB® Simulink® environment. In contrast to the Offline Mode the CANoe time base is used in MATLAB® Simulink®. Therefore the simulation is executed in almost real-time (typical resolution is about 1ms depending on the model). It is necessary that MATLAB® Simulink® can compute the model faster than real-time in order to use this mode.

CANoe can either run in simulated mode or in real-time mode providing real hardware access. Debugging features of the MATLAB® environment might be used but synchronization will be lost as long as the model is paused.

This mode can be used for interaction with real (CAN) hardware devices. It is highly recommend using a multi core processor for best simulation results. Communication between MATLAB® Simulink® and CANoe takes place over Microsoft® COM for configuration and shared memory for data exchange.

Figure 3: Synchronization in synchronized mode

3.0 Model Development

This chapter describes the steps which are necessary to run a MATLAB® model together with a CANoe configuration. It is recommended to start CANoe and open the desired configuration at first. The reason is that signals, messages, environment variables and system variables can be selected easily in the Simulink® model if every symbol is already present.

3.1 Setup

The first important setting of a model is the solver setting. If the model should be compiled, it is recommended to choose a fixed step solver. It depends on the size of the model how fast it can be computed. Typically a sample time of 1 ms to 100 ms is used. Currently only the tasking mode **SingleTasking** is supported. If the HIL mode is used, the CANoe target file must be selected in the Configuration Parameter settings.

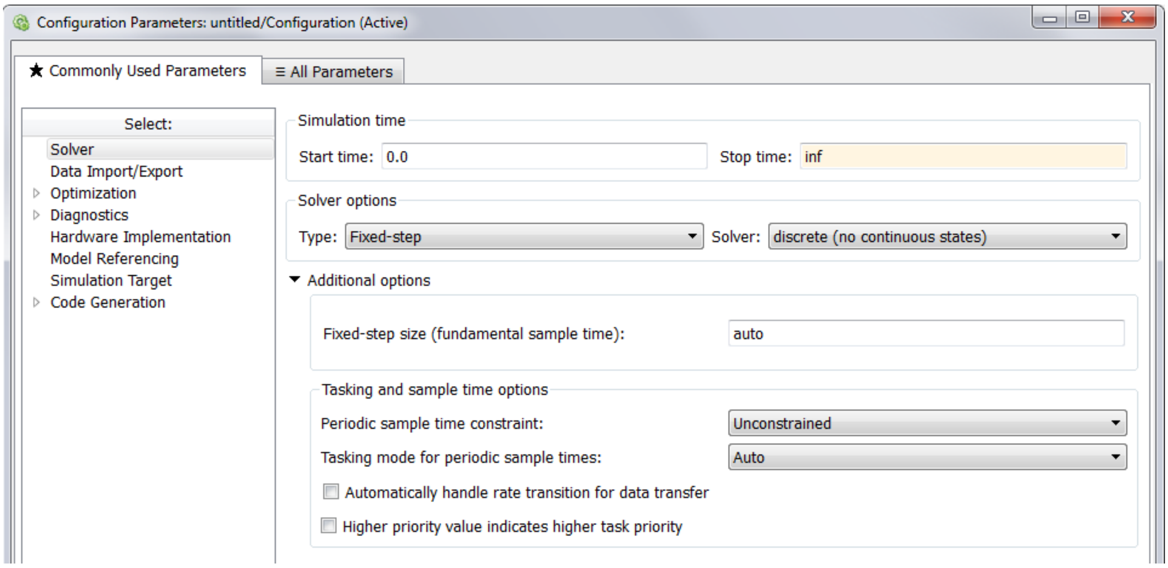


Figure 4: Solver settings

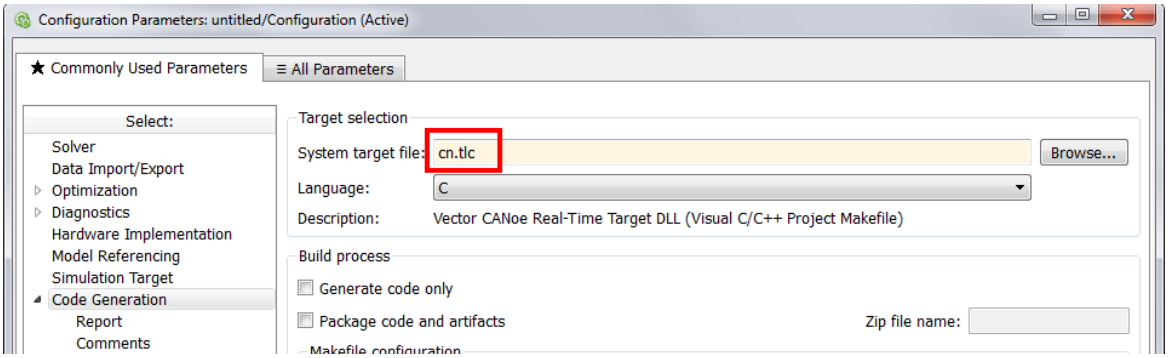


Figure 5: CANoe target file setting

A model that contains blocks of the CANoe block set library and should be executed in offline or synchronized mode must use exactly one Simulation step block. Just like other Simulink® blocks it can be dragged from the Simulink® library browser into the model. By default this block sets the execution mode to offline mode or synchronized mode. This can be changed by double clicking the block and selecting the desired mode. The simulation step block also manages the data exchange when the simulation is run in Simulink®.

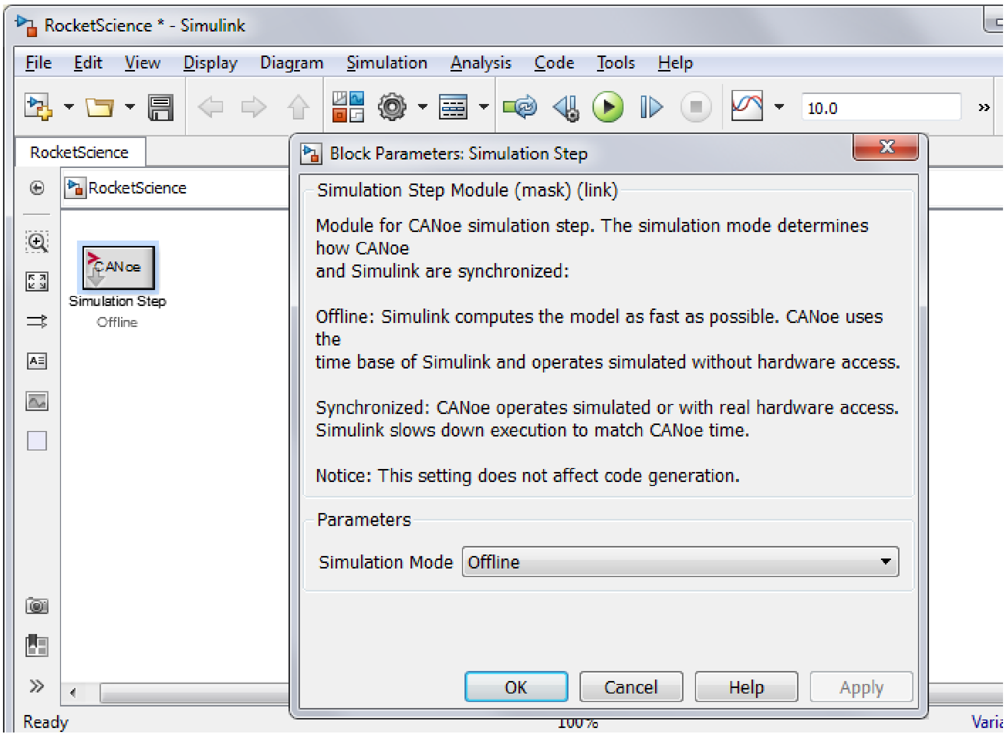


Figure 6: Simulation step block

With these preparations the model can be developed. Whenever an input of a CANoe value is needed, one of the input blocks for signals, messages, environment variables or system variables must be used. They can be configured easily by a double click on the block. A dialog appears allowing selection of an appropriate CANoe element. For each type of input block exists an output block accordingly. In the current example the signal “SigModelInput” of the CAN message “MsgModel” is multiplied by the result of a sine wave block. The result is set to the signal “SigResult”. This signal is contained by a CAN message named “MsgResult”. Both CAN messages are send in 100 ms intervals on the CAN bus.

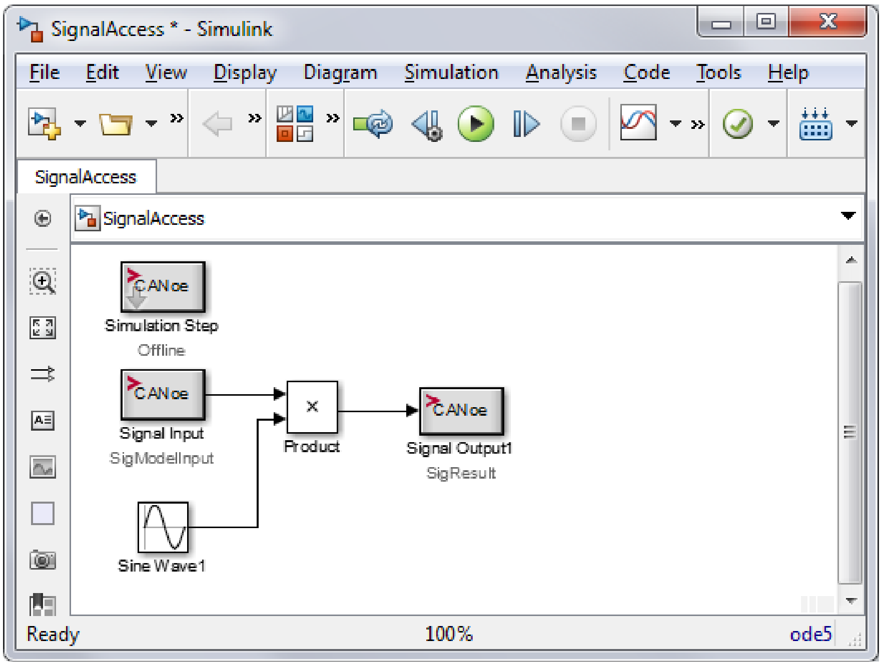


Figure 7: Simple model for interaction with CANoe

3.2 Execution

The model can now be executed in different ways. By default the simulation step block sets the execution mode to “Offline Mode”. The simulation must be started inside Simulink® which will automatically start the measurement of CANoe. If a simulation stop time is defined by the Simulink® model, the simulation and the measurement of CANoe is stopped after the time is reached.

In offline mode the execution can be paused without losing time synchronization with CANoe because the measurement will be paused, too. In synchronized mode real hardware is used and therefore pausing the simulation will break the time synchronization between Simulink® and CANoe. If the model is computed faster than real time, Simulink® will catch up and will restore synchronization. The model outputs computed during this time might be incorrect because they are based on newer input values. In general it is recommendable not to pause the simulation in synchronized mode.

During simulation the communication is monitored by CANoe and can be analyzed during the measurement or afterwards. Figure 8 shows the content of graphic and trace window during execution of the current model example. As expected it outputs a sine wave as soon as the input value changes from zero to one. The trace window shows that the output signal “SigResult” is sent correctly via the “MsgResult” CAN message.

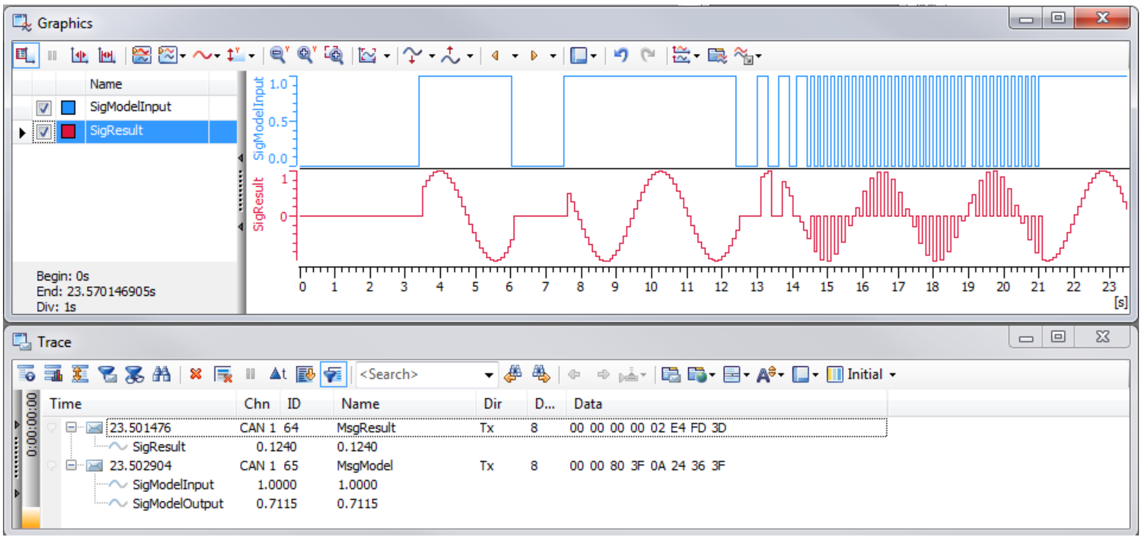


Figure 8: Analysis in CANoe

3.3 Checking Configuration

During the development of Simulink® models the CANoe configuration might change. Databases might be updated to a newer version for example. In this case all CANoe blocks of the model must be checked. The CANoe/MATLAB® interface provides custom checks for the Simulink® Model Advisor.

The Model Advisor tries to find all CANoe objects of the current model inside the currently loaded CANoe configuration. If any object cannot be found, it will add a link in the report. By following the links of erroneous blocks Simulink® opens the correct sub system and highlights the block. Using the Model Advisor it is very easy to find all CANoe blocks that won’t work properly with the current CANoe configuration.

The Model Advisor checks can also be run automatically during compilation. In the model configuration parameters dialog below “CN code generation” the option “Check CANoe block configuration at compilation” must be enabled.

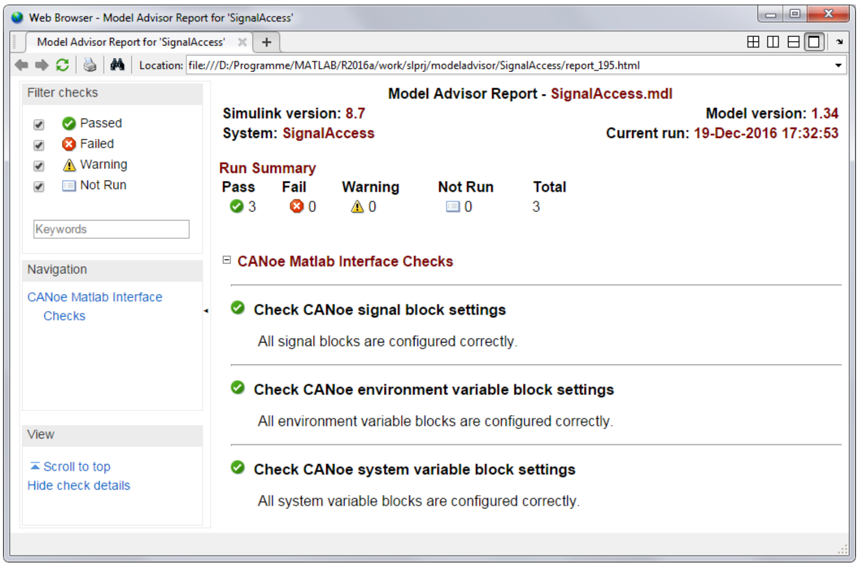


Figure 9: Model Advisor report

4.0 Support for Models in HIL Mode

When the HIL mode is used, the Simulink® model is compiled as a DLL that runs inside the CANoe environment. CANoe offers several features that allow manipulation and analysis of such model DLLs without recompiling. In addition, no MATLAB®/Simulink® license is necessary for analysis tasks.

4.1 Parameterization

Per default all parameters of every Simulink® block can be made accessible through CANoe. In the CN code generation page the option “Enable Parameterization from CANoe” must be activated:

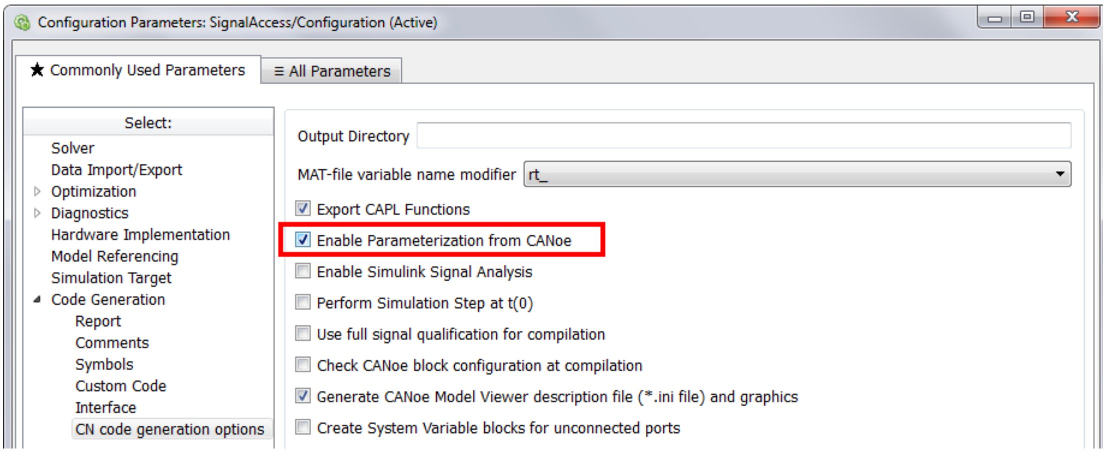


Figure 10: CN code generation options

Afterwards the model must be compiled and added to CANoe. It will automatically create a system variable for each model parameter. The number of parameters and their names might slightly differ from the ones configurable in Simulink®.

It is important to know that it depends on the implementation of a Simulink® block if its parameters are changeable during runtime. It is recommended to change the system variables in the On PreStart section of a CAPL program or to set the values before the measurement is started in order to make sure that all parameters are taken over. Another possibility is to provide a set of start values in the respective dialog. Names of namespaces and variables are derived from the structure of the model. The namespaces start with the model name and contain all sub system’s names followed by the block’s name. The variables are named according to the Simulink® internal parameter naming. The selected variable below represents the value of a constant block in the subsystem “Enabled Subsystem1 No ABS” of the model “Breaks”. It can be referenced in CAPL by its full name:

```
SysSetVariableFloat("Breaks::Parameters::Enabled_Subsystem::Constant2",
        "Value", 42.01);
```

Or just simply:

```
@sysvar::Brakes::Parameters::Enabled_Subsystem::Constant2::Value = 42.02;
```

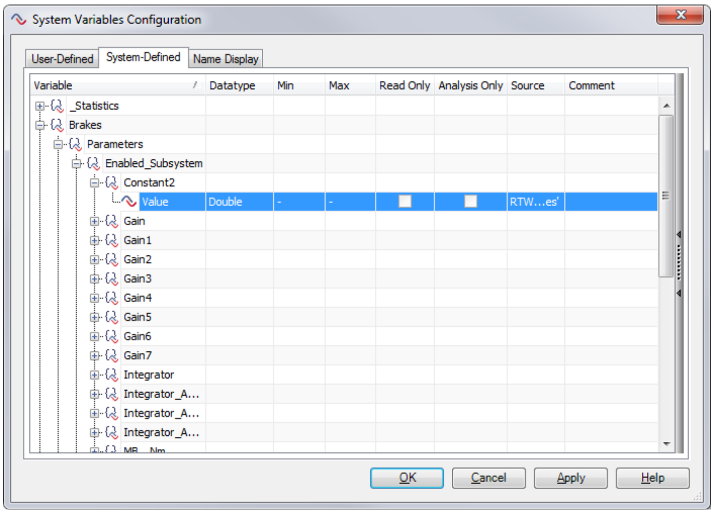


Figure 11: CANoe system variables dialog

It is possible to create a panel containing the system variables that must be changed for parameterization of a model. After installation of the CANoe/MATLAB® Interface there is a generic parameterization panel (.NET panel) located in `$(MATLABROOT)\rtw\bin\canoe\parameterization\Parameterization.dll`. It allows modification of all variables used for parameterization as well as loading and saving all variables from/to a parameterization file. With these files it is possible to parameterize the complete model at once.

Another possibility is to provide a set of start values within CANoe's start value dialog. It will execute the assignment of start values to the appropriate parameters on each measurement start.

4.2 Analysis of Simulink Signals

When developing models in Simulink® it is easy to analyze intermediate results. Typically the output signal of a questionable block would be displayed in a scope. In figure 12 the signal "lambda" is analyzed. It is the output of a gain block which was fed by the result of a product block.

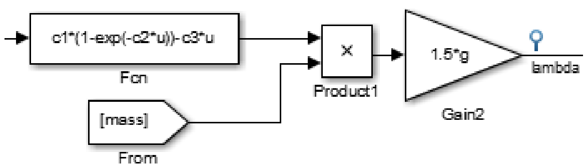


Figure 12: Simulink® signal

CANoe supports analysis of Simulink® signals even if the model is compiled. In general all named signals which contain scalar or vector values can be analyzed. But it depends on the Simulink® Coder settings if a signal exists in the compiled model at all. Further information can be found in the MATLAB® documentation concerning the "Optimization" settings of the model configuration parameters dialog.

To enable signal analysis only few configuration steps are necessary. The most important one is that the signals are named. The signal configuration dialog of Simulink® also provides an option called "Test point" which prevents the signal from being removed due to optimization:

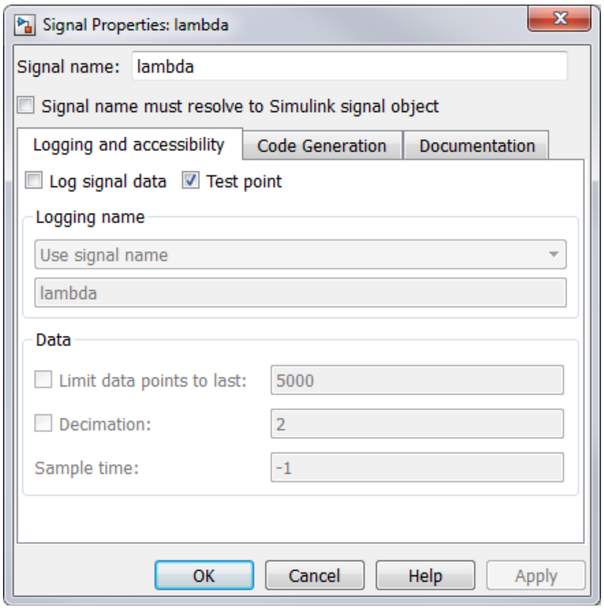


Figure 13: Signal properties dialog

Similarly to the parameterization settings, the signal analysis must be activated in the model configuration:

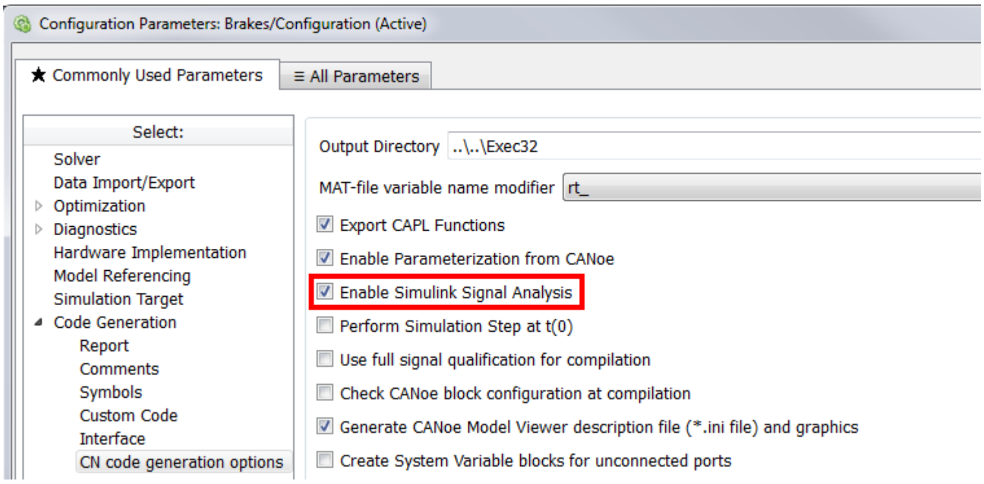


Figure 14: Interface options

Afterwards the model must be compiled and added to CANoe. It will automatically create a system variable for each Simulink® signal. The signal values can be analysed by adding the corresponding system variable to the graphics or data window. Names of namespaces and variables are derived from the structure of the model. The namespaces start with the model name and contain all sub system's names followed by the block's name that provides the signal. The desired signal can be added to panels, graphic or data window in CANoe. In all cases the following selection dialog will appear in CANoe which simplifies the configuration:

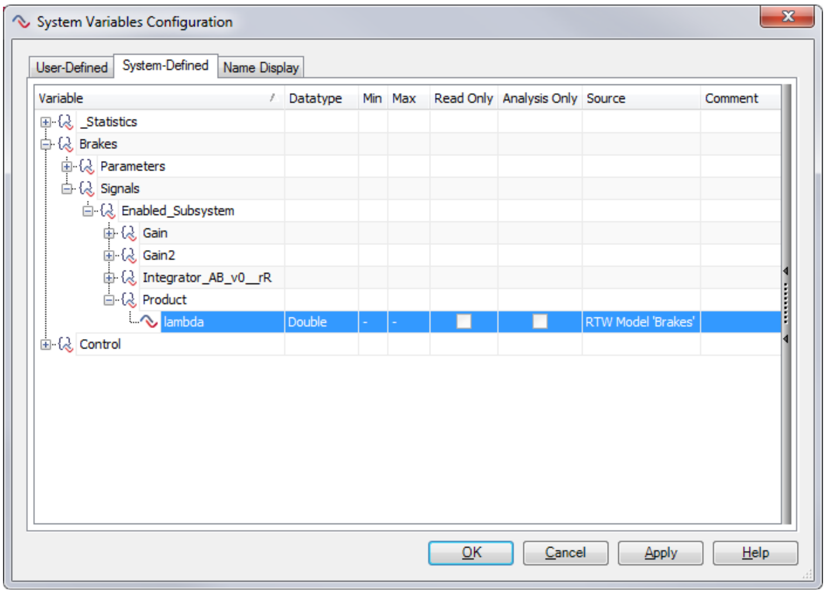


Figure 15: Selection of system variables

4.3 Calibration

The above mentioned possibilities for changing model parameters during simulation are all based on system variables. Each system variable represents one Simulink® block parameter or signal. However, CANoe can also be run in standalone mode on a dedicated hardware interface such as the VN89 series. In this case system variables cannot be accessed from outside. For such scenarios the block set of the CANoe/MATLAB® interface contains a calibration block. This block includes an XCP server into the model DLL which allows to access model's block parameters through XCP (on CAN or on Ethernet). In order to achieve this, an a2l file is generated during the DLL build process. The a2l file can be used to configure any calibration tool such as CANoe.XCP or CANape.

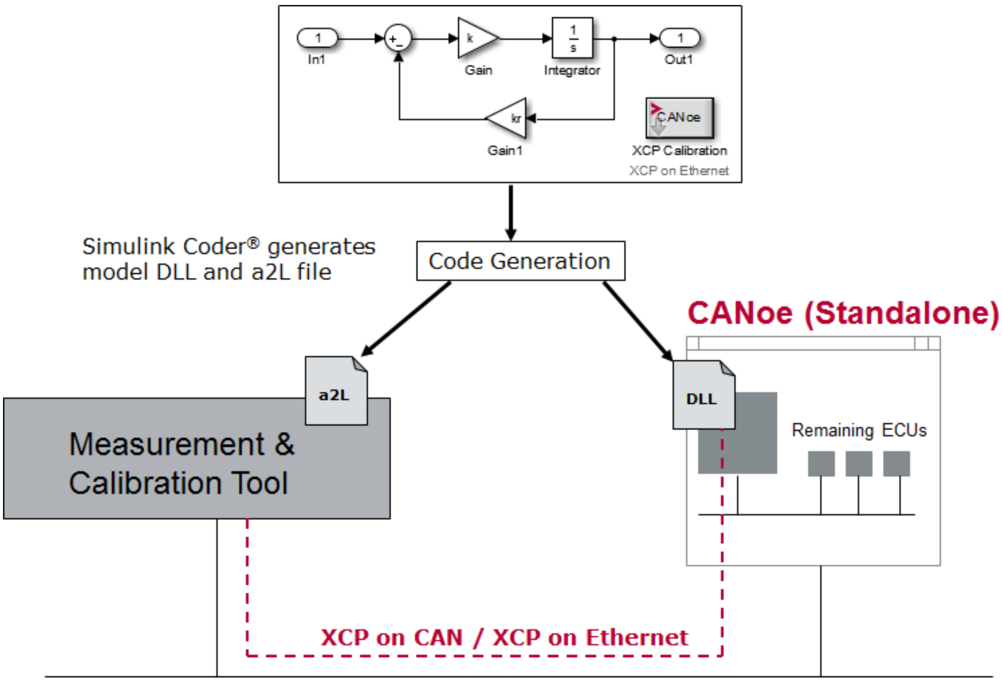


Figure 16: Calibration workflow

4.4 Model Viewer

With the CANoe Model Viewer a read-only representation of a Simulink® model can be opened inside CANoe. It gives an overview of the model structure and its subsystems as well as the usage of signals and variables. Information files used by the model viewer must be generated once by Simulink® Coder. If a model is compiled, these files are generated by default. They can be built manually from the MATLAB® prompt with the following command:

```
VGenerateModelViewerInformation('model name')
```

CANoe must be configured to use this model viewer information. The configuration dialog of a node in the simulation setup contains a Simulink® page if the CANoe/MATLAB® interface is installed. The path to the Simulink® model and the model viewer information files must be entered there:

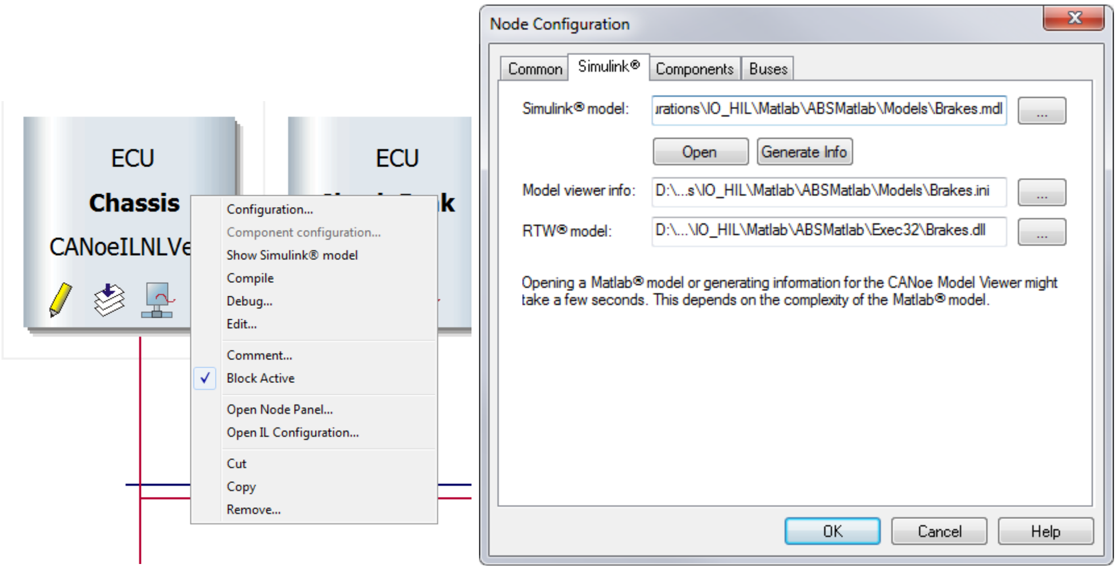


Figure 17: Model Viewer configuration

As soon as a model viewer information file is entered, the model viewer can be opened by a double click on the node in the simulation setup or via the context menu "Show Simulink® model". It will show the main model level by default. Subsystems can be opened using the tree on the left side or by a mouse click on the system on the right side. The model viewer also shows the Simulink® parameters of a block as comment if the mouse points to it. The tooltips can be enabled via the model viewer's context menu.

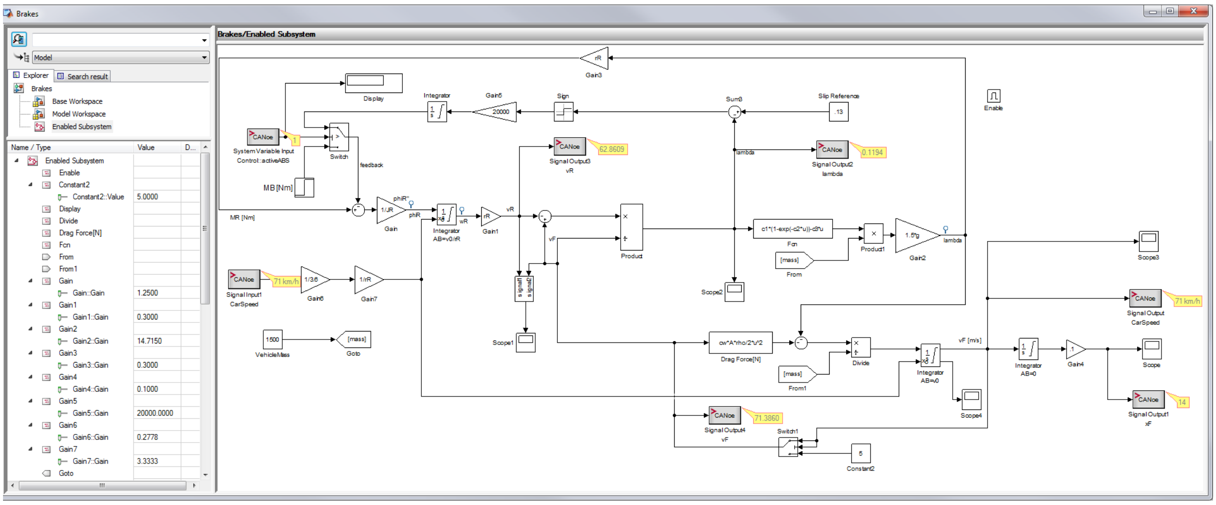


Figure 18: Model Viewer

The model viewer can not only be used for viewing but also for configuring the data or graphics window. Signal, environment and system variable blocks as well as parameter and Simulink® signal values can be dragged from the tree on the left side into graphics or data window of CANoe.

It is also possible to search for elements in the model. As soon as text is entered in the search field, the page “Search result” is opened and all elements containing the text are displayed. The selected item in the search list will automatically be highlighted in the model display area on the right hand side.

5.0 Contacts

For a full list with all Vector locations and addresses worldwide, please visit http://vector.com/contact/.