Genesis HIGH SPEED

User Manual

English

HPHV Automated Analysis Option

Perception

HBM

I2695-1.2 en

Document version 1.2 - June 2012

For Perception 6.0 or higher

For HBM's Terms and Conditions visit www.hbm.com/terms

HBM GmbH
Im Tiefen See 45
64293 Darmstadt
Germany
Tel: +49 6151 80 30
Fax: +49 6151 8039100
Email: info@hbm.com
www.hbm.com/highspeed

LICENSE AGREEMENT AND WARRANTY
For information about LICENSE AGREEMENT AND WARRANTY refer to www.hbm.com/terms.

Table of Contents

1 Getting Started

1.1 Introduction

The High Power High Voltage market contains manufacturers and testing facilities. Both are involved in the development and testing of apparatus that are used in the electrical power distribution grid. Because of its importance in the economy of a country, a set of strict regulations of how these apparatus should behave, and how they need to be tested, has been defined.

For circuit breakers several international standards have been developed. Also many local standards have been defined to describe the requirements for circuit breakers.

The “Short-Circuit Testing Liaison (STL)” has created an internationally accepted report called “Harmonization of Data Processing Methods for High Power Laboratories”. This standard describes a set of rules on where and how to measure values in the digitized information.

1.1.1 Perception and HPHV

Perception defines two layers to work with the circuit breaker signal information:

- The first layer consists of the STL library.
- The second layer consists of the HPHV Automated Analysis within Perception.

1.2 Requirements

The following section lists the software requirements.

- Perception 5.0 or higher
- Perception HPHV Automated Analysis option
- Perception STL Analysis option. Refer to "STL Library" on page 10.

Note *For more information about Requirements, see chapter "Requirements" in the Perception manual.*

1.3 Installing the software

Install the Perception program files from the CD. Note that you cannot run Perception from the CD; you must install the components onto your hard drive and run the software from this drive.

Note *For more information about **Installing the software**, see chapter "Installing the software" in the Perception manual.*

1.3.1 How to install the HPHV option

The Perception software requires a HASP key. HASP (Hardware Against Software Piracy) is a hardware-based (hardware key) software copy protection system that prevents unauthorized use of software applications.
Each HASP key contains a unique ID number used for personalization of the application according to the features and options purchased. The key is also used for storing licensing parameters, applications and customer-specific data. If you have purchased the HPHV option as a separate item, you will receive a personalized "key file". Use this file to unlock the additional features.

You can find the serial number of your key in **Help** ▸ **About Perception**

To update the key information:

1 Choose **Help** ▸ **Update Key...**
2 In the Open dialog locate the Key File (*.pKey) and click **Open**.
3 If everything is OK you will see the following message:

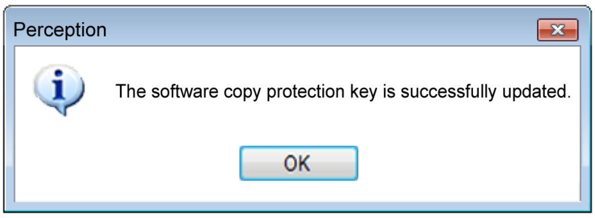

Figure 1.1: Software copy protection dialog

4 Click **OK**.

After the installation you can go to **Help** ▸ **About Perception** ▸ **More...** to see all installed options.

You will need to restart the program before the changes take effect. The HPHV option is now available.

1.4 STL Library

The methods described in the STL report are available within Perception as a library of formulas. Each formula will execute a specific STL function on a signal. The formula database of Perception must be used together with the STL library formulas to do calculations on signals. Most of these library methods require a signal, a starting point and an end point to perform a calculation. The result of this method will be available as a data source within Perception.

If the signals are very predictable and straight forward it is a very simple, customer-definable tool to perform calculations. More information about the STL formulas can be found in the corresponding manual.

Note *For more information on STL procedures please refer to:*

STL
Technical report
"Harmonisation of data processing methods for high power laboratories"

September 2004

A copy of this document can be obtained through ASTA BEAB Certification Services that holds the Secretariat of the Short Circuit Testing Liaison Agreement (STL).

ASTA BEAB is currently part of Intertek.

For circuit breakers there are a number of testing methods that are commonly used. The **HPHV Automated Analysis** sheet within Perception enables the user to do these measurements automatically. The user will have to define the relevant signals for each type of test. The software behind the sheet will take care of the calculation. It will use the functionality of the STL library modules to do the analysis of the signals and the actual calculation of the test parameters. The software is “intelligent” when working with the test signals. It can recognize events in the signals and perform the relevant calculations. The next chapters describe the user interface of the HPHV Automated Analysis sheet and the way it performs the calculations.

2 HPHV Automated Analysis Sheet

2.1 Introduction

The HPHV Automated Analysis sheet within Perception is an integral part of Perception. All Perception basic functionality also applies for this sheet. Any settings in the sheet are automatically stored in a recording when this is created. Also, when a workbench or experiment is saved or loaded, the HPHV settings are part of the save or load operation. When the sheet becomes active it has a dedicated menu and toolbar.

2.1.1 User interaction

The user interface of the sheet allows the user to control.

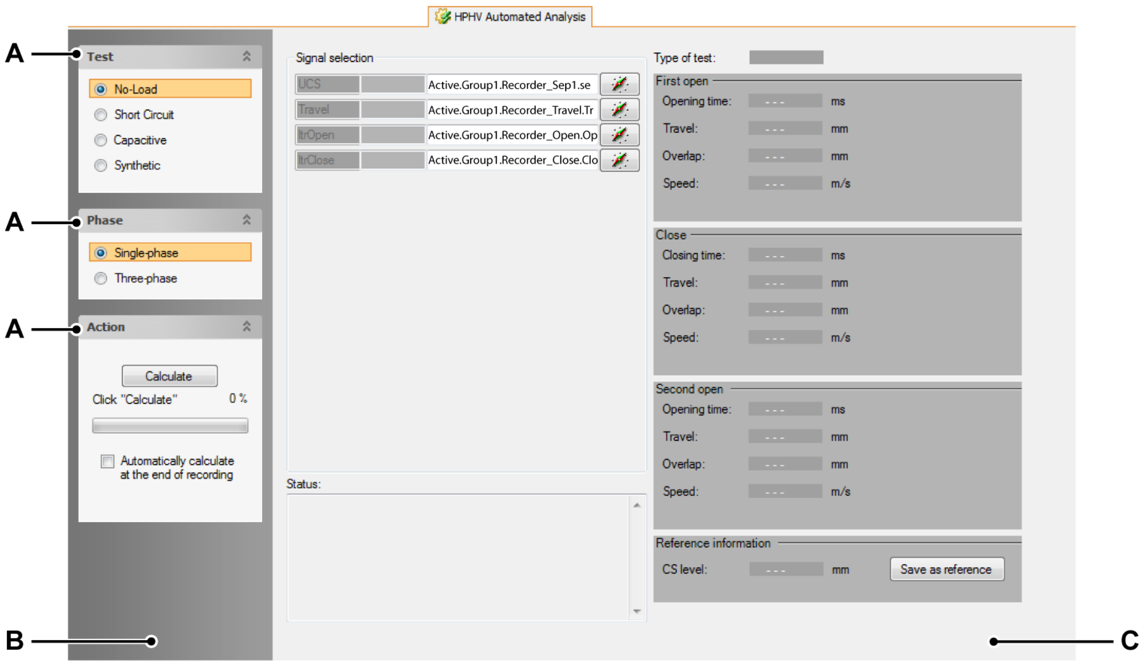

Figure 2.1: HPHV Automated Analysis Sheet

A Groups with actions and selections

B Task pane

C HPHV information

The HPHV Automated Analysis sheet contains two basic parts.
The left part is a task pane that provides a set of contextual commands and options that apply to the HPHV analysis.
When a selection is made, the content of the right part (the “HPHV information”) is changed to reflect with the selected options.

Task pane menu

The **Task pane menu** provides the user with the functionality to select a specific type of test or to perform a calculation. The menu is divided in three menu sections.

The **Test** selection allows the user to select a specific test. Currently four tests are defined: **No-Load**, **Short Circuit**, **Capacitive** and **Synthetic**.

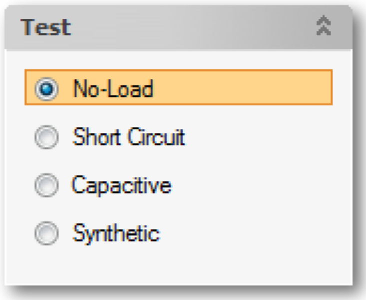

Figure 2.2: Test selection

To select a test:

1 Click the test you want to perform.

2 The right-hand side of the sheet will be updated with the relevant information.

The **Phase** section allows the user to select a **Single-phase** or a **Three-phase** test.

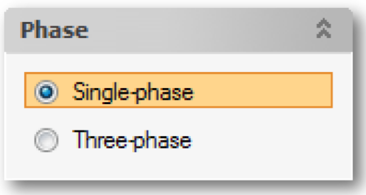

Figure 2.3: Phase selection

To select a phase:

1 Click the phase you want to use.

2 The right-hand side of the sheet will be updated with the relevant information.

The **Action** section of the menu allows the user to do the actual calculation.

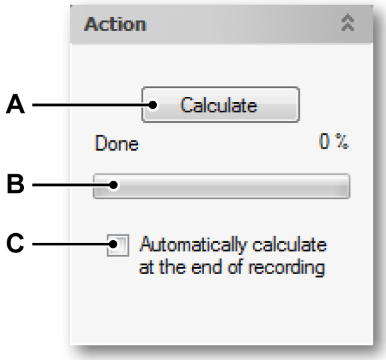

Figure 2.4: Action selection

A Calculate button

B Progress bar

C Check box for automatic calculation

To start the calculation:

1 Click the **Calculate** button **(A)**.

2 The actual calculation will take place as a “background” process. This means the computer will not be blocked while the calculation is performed. The progress of the calculation will be indicated in the progress bar **(B)**.

3 Select the check box **(C)** if you want to start the calculation automatically at the end of the recording. If checked, the calculation will start automatically after a new recording has been made. If cleared, you must click **Calculate** to start the calculation. This only works if a new recording is made, not if a recording is loaded.

During the calculation the results of the calculation will be reported in the results fields of the **HPHV information**.

HPHV information

The HPHV information part of the sheet contains the following sections: The Signal selection area, the Status area and the HPHV Results area.

Signal selection

Here you select the signals. Each type of test selection requires a number of signals to be present that contain valid information.

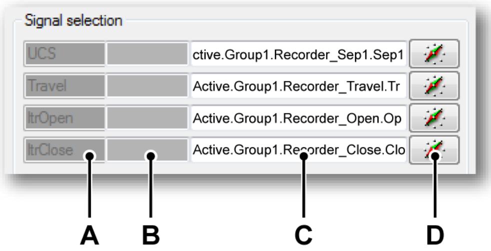

Figure 2.5: Signal selection area

A HPHV signal name

B User-definable name

C Name of the data source

D Button for selection (browse for data source)

- The left-hand field contains the **HPHV signal name (A)**. This name is used throughout this manual, and is used in the status output area in case of problems. This name is fixed and cannot be modified.
- The second field contains a **user-definable name (B)** for this signal. By default it is empty. By clicking the text box, the user can enter a meaningful name that describes the signal.
- The third part is the **data source name of the signal (C)**
- The combination of **HPHV signal name**, **user-definable name** and **selected signal** is automatically stored in Perception.
- Click the data source browse button **(D)** to select a data source.

Note *Only waveform (analog and digital) and formulas will be shown.*

Status output

This area shows the **status** of the calculation. Warning and problem reports are shown here. A warning is a non-critical message. The calculation will continue, but a message will be shown describing the warning. This can have several reasons. The most common is that a calculation is performed in a different way or a non-default behaviour is reported. A problem report is a critical message. The calculation will be aborted. A message will be shown describing the problem.

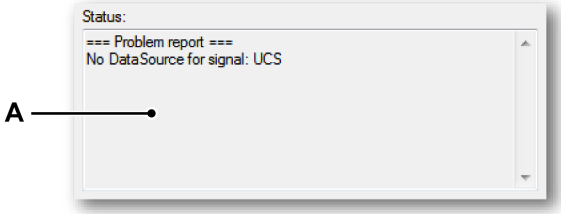

Figure 2.6: Status area

A Status area: General status and error or warning messages are shown here.

Results

This part displays the main results of the calculation. It is divided into five sections (No-Load test). The first is the type of test. The next three show the results of the switch operation for *First open*, *Close* and Second *open*.

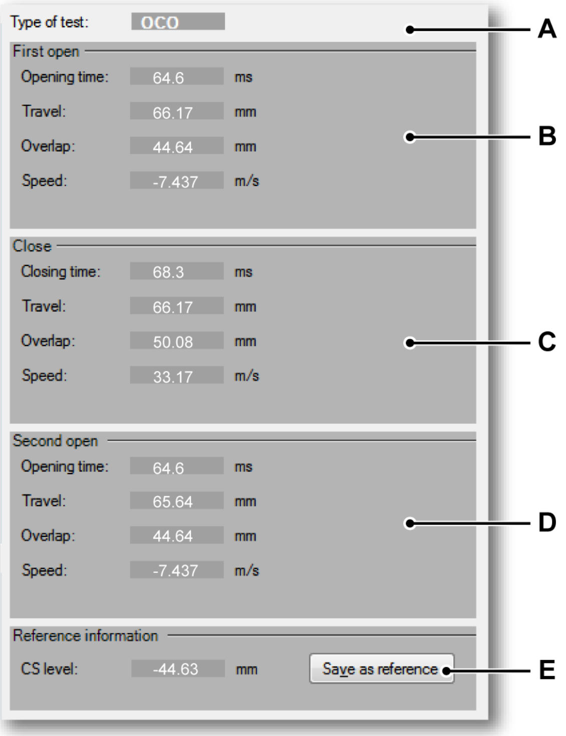

Figure 2.7: The HPHV Results menu

A Type of test (e.g. Open/Close/Open)

B First open

C Close

D Second open

E Command button **Save as reference** (Only No-Load)

- The **type of test** describes the switching operations of the circuit breaker.
- The operations are basically only an *Open* or a *Close* action.
- In some cases the testing regulations describe a fixed sequence of *Open* and *Close* operations with a predefined time between them. Because the time between the individual operations can be very short, these actions cannot be measured individually as separate recordings. In this case they can only be captured in a single recording which extends over the whole sequence.

The possible combinations of operations for type of test are: *O, C, CO, OC* or *OCO*. The analysis part of the software will determine the switching operations. The result will be shown in the *Type of test* field. For each found switching operation a set of parameters is shown. The information of the first found *Open* will be reported in the *First open* field. If there is a *Second open* switching event, it will be reported in the *Second open* field. If a type of test is a *CO*, the *First open* field shows the information of the *Open* part of the sequence.

2.1.2 Operation

General

In order to work with the HPHV Automated Analysis you must connect to a GEN DAQ product and hook up the required signals.

Make a recording

1. The amplitude scaling of the signals and the recording time information needs to be entered in Perception.
2. In the HPHV Automated Analysis sheet the user must select a specific type of the test to be performed. Usually a new series of tests will start with a *No-Load* test.

Note *For more information about **Make a recording**, see the Perception manual.*

Load a recording

If you want to continue a test after a break or repeat a previously executed tests, you he can use the *load as an experiment* function to retrieve the complete setup.

- After this initial setup the system is ready to perform the first test. Click **Start** to start an acquisition. For information about recordings, see chapter 5 "Acquisition Control and Status" in the Perception manual.

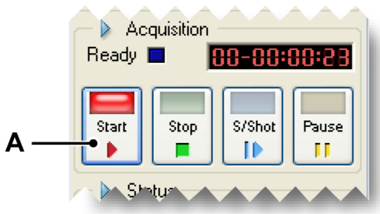

Figure 2.8: Acquisition control

A Start command

- When the actual test has been executed, the signals are shown in Perceptions Active sheet. In the HPHV Automated Analysis sheet you can now click **Calculate** to start the analysis. The results are shown as described in "Results" on page 17.

Data Sources variables

When the user has selected a specific test and phase, the HPHV sheet will show the result of this selection after the calculation. For each switching event a selected group of results will be shown. These are not all the results that are calculated. The complete set of results is available through the data sources navigator. The basic outputs of the HPHV Automated Analysis software are the variables in the data source. The HPHV Automated Analysis creates a new node entry called *HPHV* in the data sources navigator.

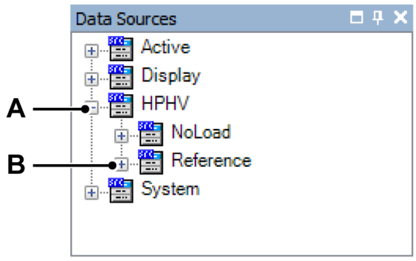

Figure 2.9: Data Sources navigator

A HPHV node

B Reference node

Under this node two new entries are present. The first one is the node called **Reference**. It contains the reference information that was calculated during the *No-Load* operation. For more information, see the paragraph "No-Load reference determination" on page 23 in this chapter.

Depending on the selection made in the HPHV Automated Analysis sheet, a second entry has the name of the type of test that was selected.

This can be either *No-Load*, *Short Circuit*, *Capacitive* or *Synthetic*. Under this node there are several entries that contain further information.

This information can be used in the other parts of Perception, just like any other data source.

Storage
When a recording is made, Perception stores all raw date and settings information in the recording file. This includes the complete setup, configuration, layout and results from Perception. It also includes all settings of the HPHV Automated Analysis sheet. If modifications or calculations are made after the test, they must be saved manually in the recording.

A specific setup can also be saved in a **virtual workbench** file. A workbench file only contains settings, no raw data. This will also include all settings of the HPHV Automated Analysis sheet.

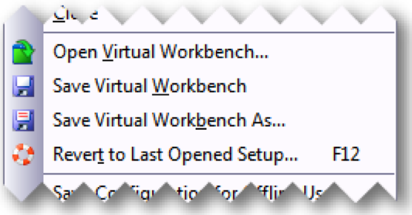

Figure 2.10: File menu

For information how to save a virtual workbench, see the Perception manual.

A setup of the HPHV Automated Analysis sheet can also be saved individually. It contains all information currently available. This will include all HPHV values that are currently available as a source.

The HPHV Automated Analysis toolbar and menu allow the user to load or save this information.

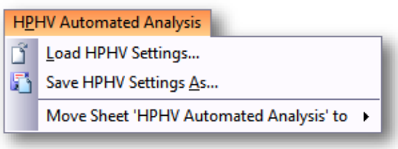

Figure 2.11: HPHV Automated Analysis menu

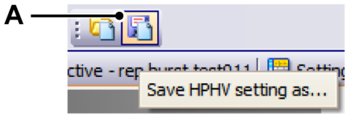

Figure 2.12: HPHV Automated Analysis toolbar

A The Icon **Save HPHV setting as...**

To save a HPHV setting as:

1. Click on the icon or select the menu command **Save HPHV setting as...**
2. Select the file type in the drop-down list box.
3. Select the file you want to save replace or type a name for a new file in the **Save as** dialog box.
4. Click **Save**.

Loading

When a recording is loaded as an experiment, all information will be restored and Perception will return to the situation at the moment of saving. Also the information of the HPHV Automated Analysis sheet will be restored exactly as it was at the moment of saving.

If the calculation was already done, the results will be shown in the sheet. The values of the HPHV variables are loaded from the information stored in the recording. This means that the values shown in the HPHV Automated Analysis sheet are the values that were stored in the recording.

When a workbench is loaded, all HPHV Automated Analysis sheet information will also be restored. This will include the reference values of the HPHV variables . The HPHV variables of the selected test are initialized to their default values. These defaults are empty text for all text-related variables, and a “Not a Number” value for numerical values.

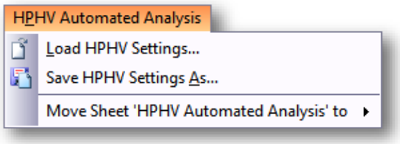

Figure 2.13: HPHV Automated Analysis menu

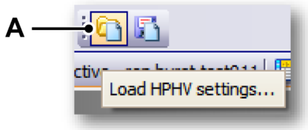

Figure 2.14: HPHV Automated Analysis toolbar

A Icon **Load HPHV settings ...**

To load HPHV settings:

1. Click on the icon or select the menu command **Load HPHV settings...**.
2. Select the file type in the drop-down list box.
3. Select the file.
4. Click **Open** to load the file.

A setup of the HPHV Automated Analysis sheet can also be loaded individually. The load option in the menu or the toolbar allows the selection of the load option.

When a load is executed, all information of the HPHV Automated Analysis sheet will be restored. The selected test, phase and reference values will be restored.

No-Load reference determination

A high power circuit breaker has some default characteristics. These depend on the construction of the circuit breaker and the switching mechanism. For the HPHV Automated Analysis, a signal is expected that measures the travel of the contacts.

- During the **No-Load** of a circuit breaker, a relatively small voltage, and thus a small current, is applied across the contacts. During the switching event this voltage across the contacts will show exactly when the contacts separate.

- The level of the **travel recorder** at the time of contact separation will be used as a reference for further tests. Also the time between the switching command and the actual contact separation or touch will be remembered. The result sheet provides an extra option to save this as reference.

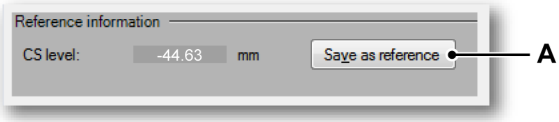

Figure 2.15: Reference information

A Click this button to save the contact separation level as reference

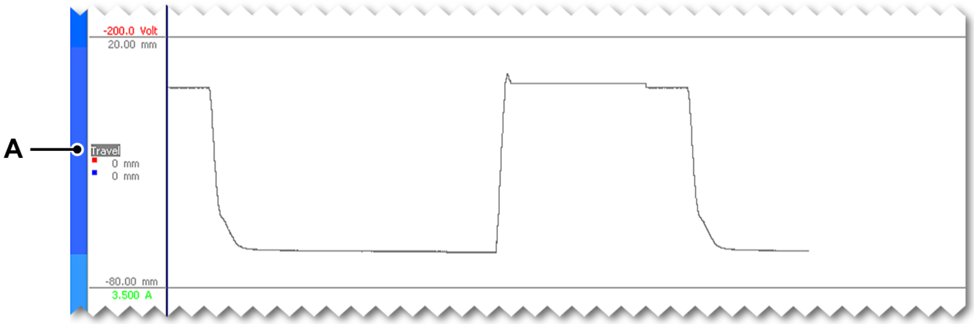

Figure 2.16: Display - Travel recorder

A Signal which represents the travel of the contacts

To save as reference:

1. Click the **Save as reference** button.
2. The *Contact separation* level is saved as a data source variables under the *HPHV.Reference* node in the Data Sources navigator. These values are used for all other tests as a reference to get the contact separation moment in the tests.

3 Analysis and Calculation

3.1 Introduction

You can select four different kinds of calculations. Each calculation requires that specific signals are present. Each calculation has its own set of signals which are used in a specific way. This document describes all the individual calculations. It describes how the calculations are done, what the results are and the names of the corresponding variables, selectable in the data sources navigator. This chapter describes the calculation process in general.

3.1.1 General calculation steps

When a **calculation** is started, some actions are common to all tests. Instead of describing them individually for each test, they will be described here. The general calculation process contains the following steps:

- Verify selected signals
- Get switching events
- Evaluate switching events

Each will be described in the next paragraphs.

Verify selected signals

- The first step in the calculation cycle is to **verify** if the selected signals match the expected information. They must be waveform signals, and they must be static, and contain a valid data source. The *ItrOpen* and *ItrClose* signals can also be digital event signals. If this check fails, an appropriate **error message** will be shown. The calculation cannot continue until this problem is resolved.

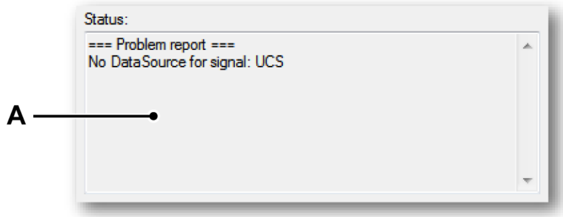

Figure 3.1: Status area

A Error message (Verify selected signals)

Get switching events

- The second step in the calculation cycle is to evaluate the switching events of the circuit breaker. See paragraph "Switching event determination" on page 30 for more information about this process.
- If no switching events can be found, an appropriate message will be shown and the calculation process will be terminated.
- If an invalid switching event is found, the system will issue an appropriate warning message and will continue with the valid events that were found.

Evaluate switching events

Once the **switching events** have been established the calculations are done for each individual switching event, one at a time. The system will use the boundaries that are defined by either the physical signal boundaries, or the end of the previously processed switching event, or the start of the next switching event.

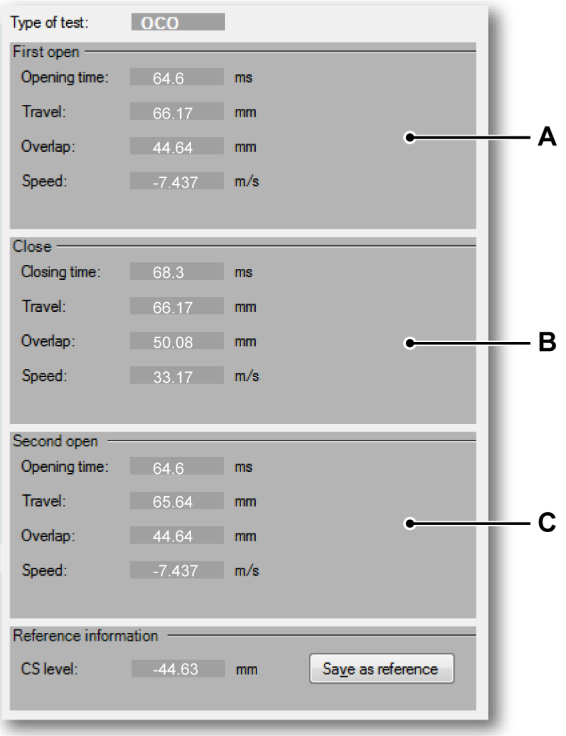

Figure 3.2: HPHV Results menu (No-Load test)

A **First open** information

B **Close** information

C **Second open** information

- The switching event *Open* or *Close* defines the calculations that need to be done. The system will use the start location as a reference point.
- For each possible switching event a node in the data sources navigator is defined.
- The first *O* event will report its information under the *Open1* node in the data sources navigator. The information of the second *O* will be saved under the *Open 2* node. The *C* information will be saved under the *Close* node in the data sources navigator.

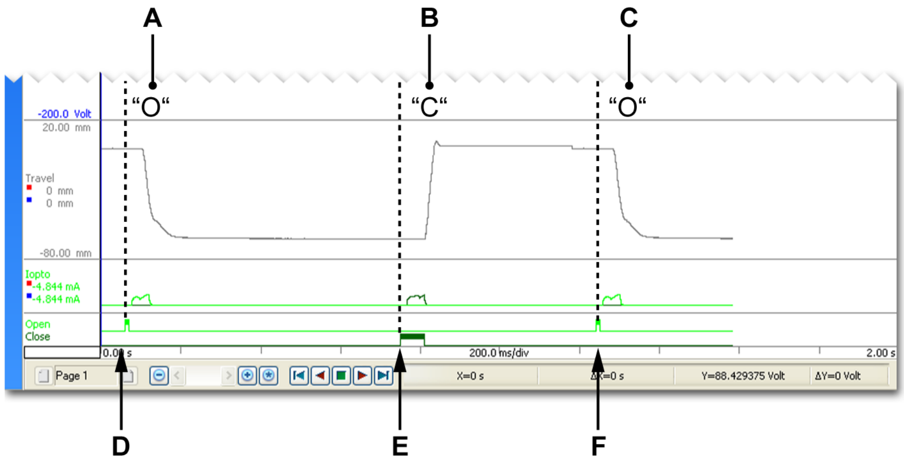

Figure 3.3: Display - Switching events

A **First open** signal

B **Close** information

C **Second open** information

D SwitchEvent1

E SwitchEvent2

F SwitchEvent3

The **switching events** are used for every individual calculation.

- When for example an *OCO* test is being examined, the first switching event is an *O*. The calculations are performed between two boundaries. The left boundary is the start of the recording, as there is no previous calculation. The right boundary is the next switching operation. This is the switching event *C* **(SwitchEvent2)**.
- The calculation on the *O* redefines a new *EndOfOperation* value somewhere between **SwitchEvent1** and **SwitchEvent2**. The exact location depends on the results of the measurement of the *O*.
- For the calculation on the *C*, again two boundaries are used. The left is the EndOfOperation of the previous calculation. The right boundary is the next switching operation **(SwitchEvent3)**. Again, the calculation of the *C* redefines the *EndOfOperation*. For the last *O* the left boundary is the previously defined *EndOfOperation*, and the right boundary side is the end of the recording, as there is no next switching event.

3.1.2 Naming conventions for data source variables

Most calculations can be done on single-phase or on three-phase circuit breakers. The calculation principle is exactly the same for both. When using multiple phases, specific information for each individual phase needs to be reported as a data source. In the **naming convention** of the variables the individual phase information is denoted by a (#) sign in the name. The (#) in the name will be replaced by the phase number 1, 2 or 3. If it is a one-phase test, the (#) will be replaced by nothing.

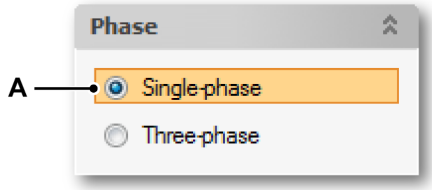

Figure 3.4: Phase - Group box

A Option button (e.g. Single-phase)

To select a phase:

1. Click the phase you want to perform.
2. The right-hand side of the HPHV Automated Analysis sheet will be updated with the relevant information.

The data source variables are located under the currently selected test entry and corresponding switching event. For example, the *ActionTime* of a No-Load test for the second *Open* event is located under the data source entry *HPHV.NoLoad.Open2.ActionTime*. This applies to all tests.

3.1.3 Frequently used calculations

Almost all kinds of tests perform some **calculations** that are common to all. This chapter describes the most commonly used calculations. The individual test descriptions will refer to these calculations without further explanation.

Switching event determination

For almost all calculations the HPHV Automated Analysis will need to establish the switching operation on the circuit breaker. This chapter explains how this is done.

Signals used:	
Travel	This is the representation of the contact movement in mm.
ItrOpen	This is the current through the tripping coil for the test objects *Open* command. Also a digital event signal can be used.
ItrClose	This is the current through the tripping coil for the test objects *Close* command. Also a digital event signal can be used.

Output information:	
Switching event	A text string representing the switching operation
Switching times	An array of maximum three times representing the start of the switching actions

The Travel Recorder signal is used to determine the start and end time of the recording. This will be used as the signal boundaries for all other signals and calculations.

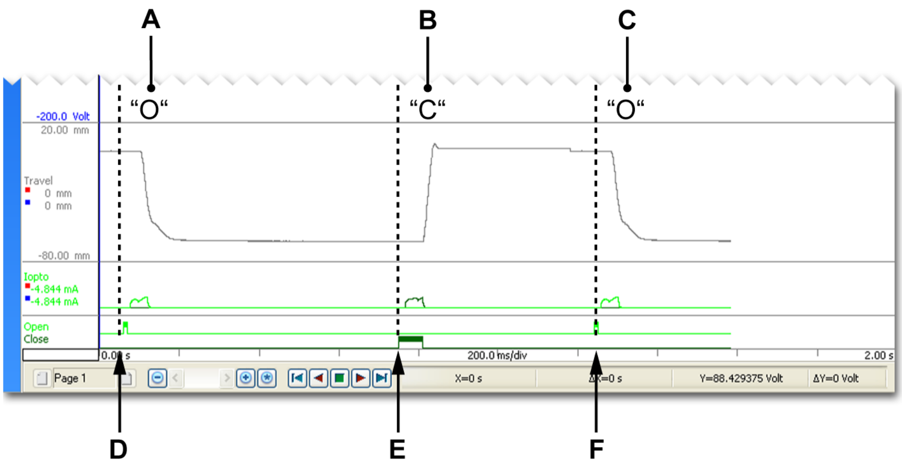

Figure 3.5: Display of the Travel Recorder Signal

A **First open** signal

B **Close** information

C **Second open** information

D SwitchTime1

E SwitchTime2

F SwitchTime3

- The system will search in the *ItrOpen* signal for the start of a signal. It will use the **STLSignalStart** method, searching in forward direction, to find the location of the start of the signal. The polarity of the signal is not important.

Figure 3.6: Display - STLSignalStart method

- When the start location is found, the **STLPrevZeroCrossing** method will be used to find the actual start of the tripping coil current. The start location (time) and switching event type *(O)* will be remembered.
- Then the system will try the same for the *ItrClose* (close command) signal. The start location(s) and switching event type *(C)* will be remembered. When one location is found, the system will skip 100 ms in forward direction. From that point on it will try to find another start and also store that information.
- If the *ItrOpen* or *ItrClose* signals are digital event signals, it will use the formula *NextLvlCross* to determine the start of the signal.
- Now the time and switching events of both *ItrOpen* and *ItrClose* will be examined to determine the switching protocol for the circuit breaker. The timing of the found events defines the switching protocol. Only the following switching events are recognized: *O*, *C*, *CO*, *OC* and *OCO*.
- If the system encounters other switching methods it will try to bring it back to one of the above mentioned switching events. It will inform the user of the choice that the computer has made.

Calculations on the travel signal

For switching events of the circuit breaker there are some **calculations** that are standardized for the operating timing of the contacts. These apply to almost all operations and are therefore explained separately in this chapter.

Signals used:	
Travel	This is the representation of the contact movement in mm.

Input information:	
CSTime	The *CSTime* is the location in the travel signal where that the contact separation/toucht was determined.
ActionTime	The *ActionTime* represents the moment in time where the circuit breaker received the command to do a switching action.
EndOfOperation	The *EndOfOperation* represents the end of the switching operation.

Output information:	
OperationTime	This is the time difference between the ActionTime and the CSTime. For an *Open* action it will be reported as *OpeningTime* in the data sources navigator. For a *Close* action it will be reported as the *ClosingTime*.
Speed	Speed of the contacts at the CSTime location
Travel	This is the total traveling distance of the contacts from the *ActionTime* to the *EndOfOperation*.
Overlap	This is the distance between the contacts in the *ActionTime* position and the CSTime position.
CSLevel	This is the value in the travel signal at CSTime.

The *CSTime* is used to do some calculations on the travel signal. It will determine the amplitude in the travel signal at *CSTime*.

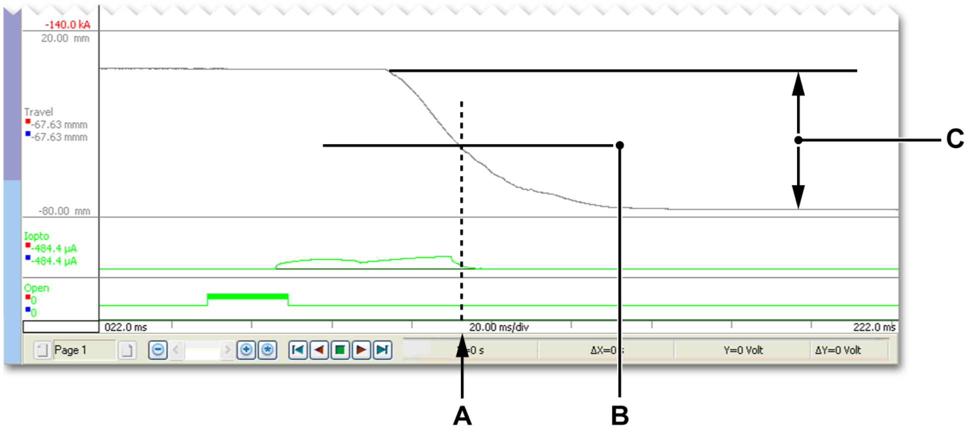

Figure 3.7: Display - CSTime

A CSTime

B CSLevel

C Travel

- A 1 ms average is used to determine the amplitude at this location. The averaging is done to eliminate spikes. The calculated level resembles the contact separation/touch position. This value is stored as *CSLevel* data source.
- The **CSTime** is also used to calculate the speed of the moving contacts. The STLContactSpeed method will be used to perform the calculation.
- The value is stored as *Speed* data source.

- It uses a 1 ms average around this location to determine the amplitude. This level is internally stored as the level before the start of the switching operation.

At the *ActionTime* the system will again calculate the level of the travel signal.

At the *EndOfOperation* it will also calculate the level. This level is stored internally as the level after the operation.

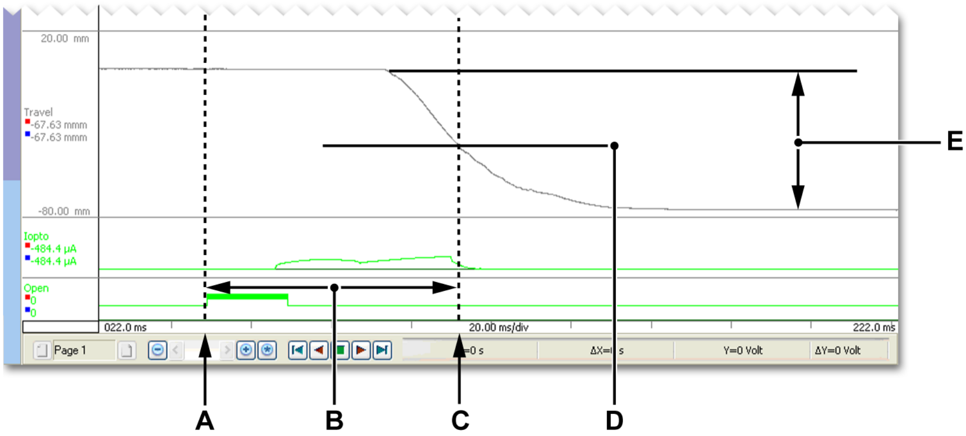

Figure 3.8: Display - Action Time

A ActionTime

B OpeningTime

C CSTime

D CSLevel

E Travel

- With the internally stored information a number of calculations will be done. The time between the *CSTime* and **ActionTime** will be calculated. Depending on the switching event it will be reported in the data sources navigator as the *Openingtime* or the *Closingtime*.

The calculation of the **Overlap** depends on the type of switching event.

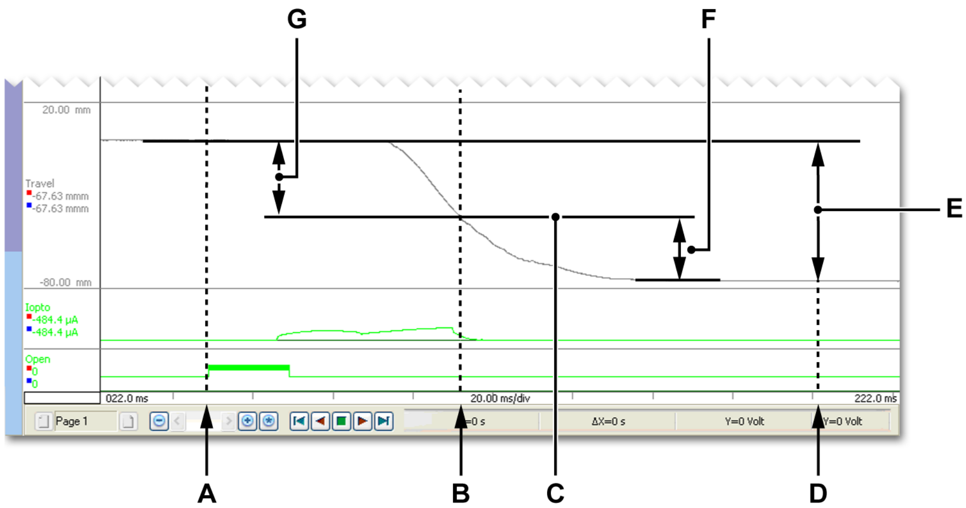

Figure 3.9: Display - Overlap

A ActionTime

B CSTime

C CSLevel

D EndOfOperation

E Travel

F Overlap close

G Overlap open

- If it is an *O*, the system will calculate the difference in the travel levels at *ActionTime* and at CSTime.
- If it is a *C*, it will take the difference between the levels at *CSTime* and *EndOfOperation*. This will be stored as the *Overlap* data source.
- HPHV will calculate the difference between the level of the travel signal at the *ActionTime* and at the end of its transition at the *EndOfOperation*. This value is stored as the *Travel* data source.

Location of contact separation/touch

This describes the actions used to find the **contact separation** or **contact touch** moment in the travel signal.

Signals used:	
Travel	This is the representation of the contact movement in mm.

Input information:	
CSLevel	This is the value of the travel signal.
ActionTime	The *ActionTime* represents the moment in time where the circuit breaker received the command to do a switching action.
EndOfOperation	The *EndOfOperation* represents the end of the switching operation.

Output information:	
CSTime	The *CSTime* is the location in the travel signal where the contact separation/touch was determined.

During the *No-Load* operations the level in the travel signal is calculated where the **contact separation** takes place. This value can be stored as a reference. During switching operations this value can be used to get the actual position in the travel where the contact separation takes place.

From the *ActionTime* location the system will search for the *CSLevel* in the travel signal. It uses the 1 ms average to locate the position. It stops looking at the *EndOfOperation*. When found, this location is stored as *CSTime* in the data source.

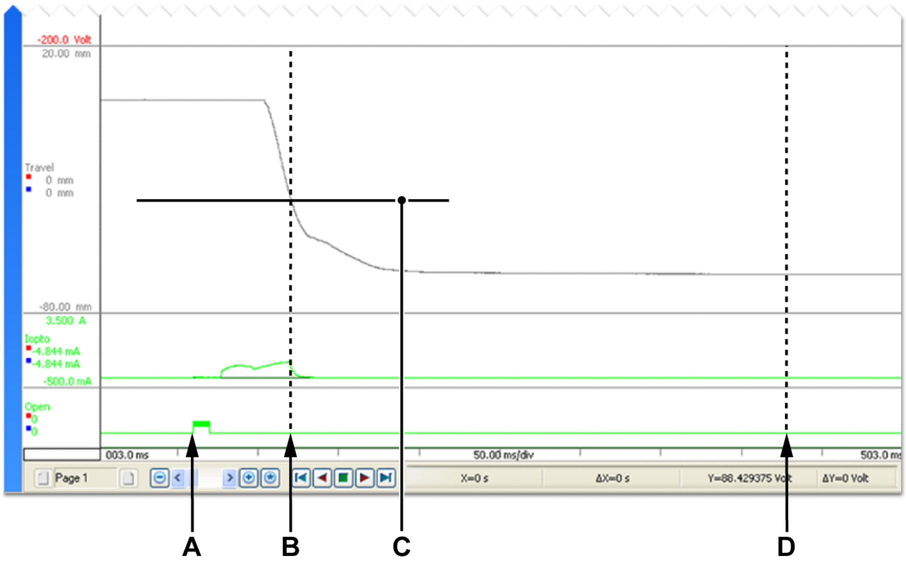

Figure 3.10: Display - Location of contact separation/touch

A ActionTime

B CSTime

C CSLevel

D EndOfOperation

Signal normalization

Sometimes the signals are not nicely scaled to the available signal input range. The standard STL routines depend on the fact that the signal occupies at least 50-100 % of the static signal range. For maximum accuracy this is what should be done. In some cases this is not a problem. The **signal normalization** will artificially increase the static signal range so that the STL routines work properly. If the signal is increased by more than 10 times, a warning will be issued.

Signals used:	**none**
Any signal	This can be any kind of signal.

Input information:	
Start time	Start location in the signal
End time	End location in the signal
Filter frequency	Frequency of the filter

Output information:	
Derived signal	The normalized signal

The input signal will be filtered between the supplied start and end location with the supplied frequency. The filtering is done to remove spikes or other disturbances. From the resulting waveform, the maximum and minimum value is determined. The absolute largest value of the maximum and minimum is used as the new static range. A new signal will be returned that will have the new static range.

Note *The new derived signal is only scaled, not filtered.*

Frequency determination
The STL routines require the **frequency** of the signal to do accurate calculations. Therefore the frequency of the signals will have to be determined. This is needed for almost all HPHV tests and will be explained here.

Signals used:	**none**
Any signal	This can be any kind of signal.

Input information:	
Start time	Start location in the signal
End time	End location in the signal

Output information:	
Frequency	The frequency found

The frequency will be determined from the supplied signal.

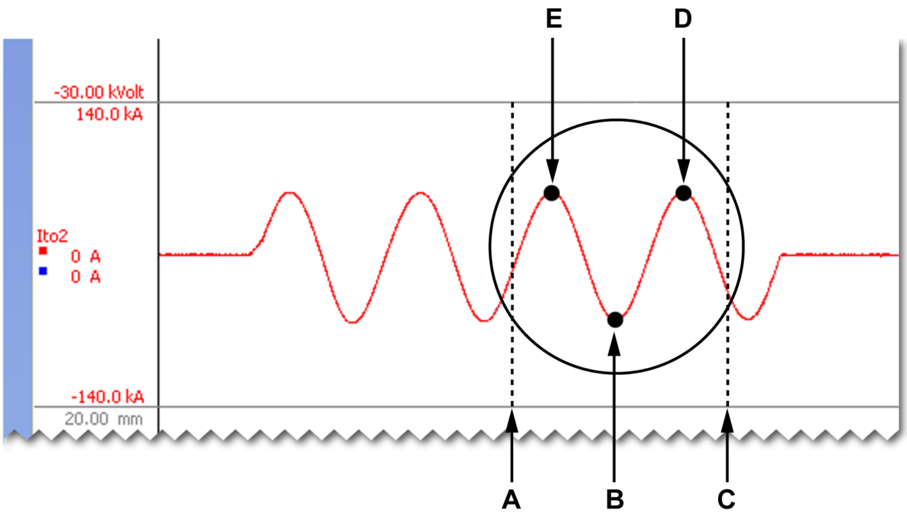

Figure 3.11: Display - STL3CrestCalculation

A StartTime

B Crest 2

C EndTime

D Crest 3

E Crest 1

- The signal will use the **STL3CrestCalculation** method to get three crests between the supplied start and end location.

- These three crests are used to determine the frequency and thus the period time.
- In case of multiple phases the system will also verify if all three phases are present. If one phase is missing, it assumes a two phase system measurement.

4 Calculations on No-Load

4.1 Introduction

The purpose of the **No-Load calculations** is to investigate the timing issues and corresponding signal levels during the switching operations on circuit breakers. A high-power circuit breaker will have some default characteristics. These depend on the construction of the circuit breaker and the switching mechanism.

HPHV Automated Analysis expects a signal that measures the traveling of the contacts. This travel must be entered in mm. The action that trips the circuit breaker to perform a switch operation must also be available as an electrical impulse. This can be either an analog waveform of the current through the tripping coil, or a digital event that signals the tripping moment. Both the *Open* command and the *Close* command must be available. During a **No-Load** these characteristics are calculated and remembered. A simplified diagram of the test circuit used during the No-Load test is shown below.

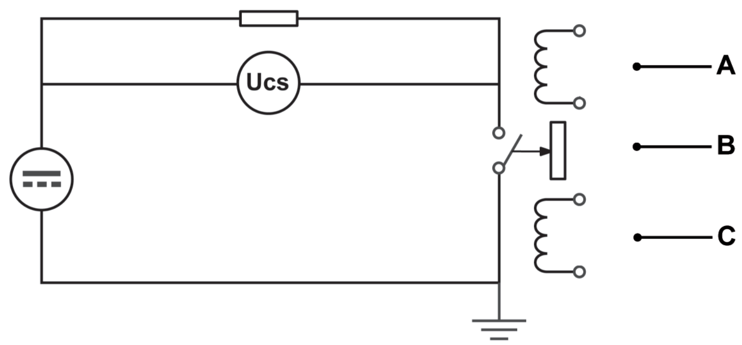

Figure 4.1: Diagram - Circuit breaker during a No-Load calculation

A ItrOpen

B Travel

C ItrClose

4.1.1 Input and output (Calculations on No-load)

When the No-load option is selected, HPHV will enter a *NoLoad* node entry under the HPHV node entry in the data source. There the *Open1*, *Open2* and *Close* nodes will be created. See paragraph "Naming conventions for data source variables" on page 29 for more information.

Signals used:	
Travel	This is the representation of the contact movement in mm.
ItrOpen	This is the current through the tripping coil for the test objects *Open* command.
ItrClose	This is the current through the tripping coil for the test objects *Close* command.
Ucs(#)	This is the representation of the contact open/close transition. It should be a “high” value when closed, and a “low” value when opened.

Input information:	None

Output information:	Nodes in the data source
Type	String value expressing the type of operation. It can only be one of the following options: *O*, *C*, *CO*, *OC* or *OCO*.

Output Open/Close:	
ActionTime	Start time of the switching action
CSTime	Time of contact separation/touch, or the shortest of the three-phases
CSTime(#)	Time of contact separation/touch for each phase
Speed	Speed of contacts at CSTime
Overlap	Movement in travel between contact separation/touch and the closed location.
Travel	Movement in travel for the whole switching level
CSLevel	Level in travel signal where contact separation/touch takes place

Output Open:	
MinimumOpeningTime	Minimum opening time of all phases
OpeningTime	Time between ActionTime and CSTime, or the shortest of the three-phases
OpeningTime(#)	Time between ActionTime and CSTime for each phase for an opening event

Output Close:	
MinimumClosingTime	Minimum closing time of all phases
ClosingTime	Time between ActionTime and CSTime, or the shortest time of the three-phases
ClosingTime(#)	Time between ActionTime and CSTime for each phase for a closing event

4.1.2 Initial calculation (Calculations on No-load)

As described in paragraph "Switching event determination" on page 30, the switching events will be established. Then one by one, each event is processed.

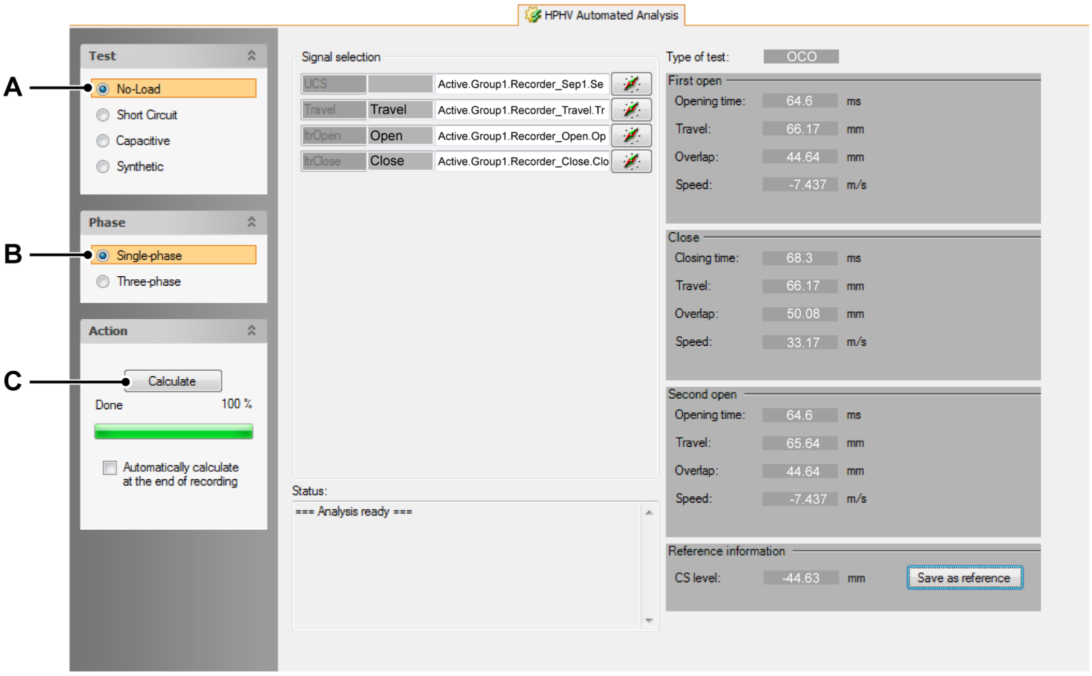

Figure 4.2: HPHV Automated Analysis Sheet - Initial calculation process - No-Load

A **No-Load** option button

B **Single-phase** option button

C **Calculate** command button

To start a calculation in No-Load:

1. Click on the **No-Load** option button.
2. Click on the (e.g.) **Single-phase** option button.
3. Click on the **Calculate** command button to start the calculation.

Depending on the switching event, the data source values are entered under the *Open1*, *Open2* or *Close* node. The first found *O* switching event will be stored under *Open1*. The second *O* will be reported in the *Open2* node. The start location of the switching event will be saved in the data source as *ActionTime*.

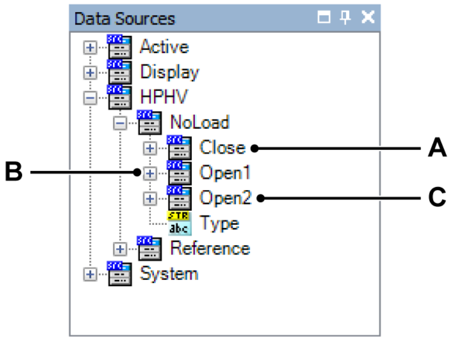

Figure 4.3: Data Sources Navigator - Calculation on No-Load

A Close node

B Open1 node

C Open2 node

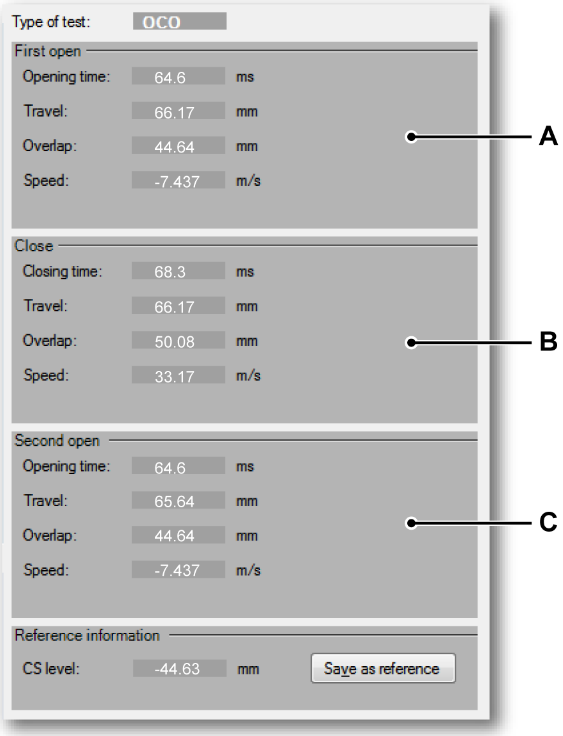

Figure 4.4: Switching events

A **First open** information

B **Close** information

C **Second open** information

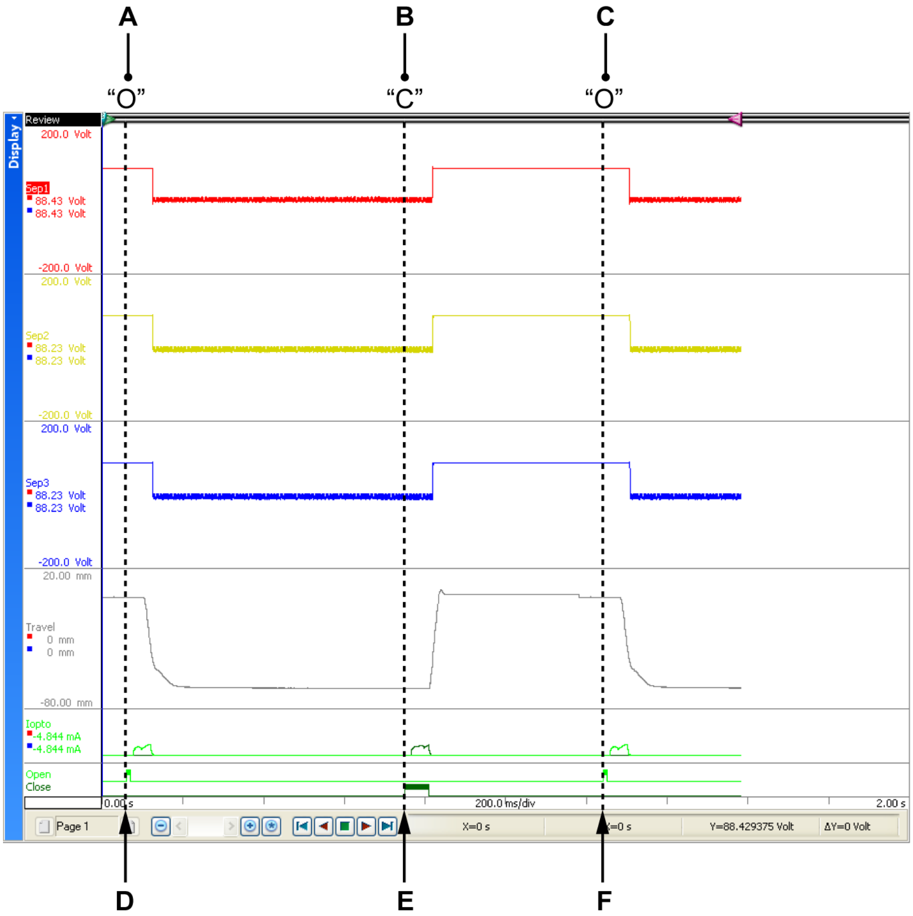

Figure 4.5: Display - Three-phase No-Load test

A **First open** signal

B **Close** signal

C **Second open** signal

D **ActionTime** for "O"

E **ActionTime** for "C"

F **ActionTime** for "O"

Overview of a typical three-phase No-Load test. The next paragraphs explain the calculation based on a single phase.

4.1.3 Open event (Calculations on No-load)

For an *O* switching event the system will look in the UCS signal for the **open** location. It will use the *STLNoLoadOpen* method to find the location.

Figure 4.6: Display - Open event - Calculations on No-Load

A **ActionTime**

B **CSTime**

C **CSLevel**

D **CSTime**

E Location found through STLNoLoadOpen

- The *STLNoLoadOpen* method starts searching at the *ActionTime* and ends at the next switching event. The signal transition should go from High to Low.
- If the location is found, it will be saved as *CSTime* in the data source.
- If the location is not found within the specified region, an error message will be issued and the operation will be terminated.

- At the *CSTime* the system will determine the level of the travel signal, using a 1 ms average around the CSTime. This value is saved as the CSValue in the data source. This is also the value that will be saved as reference for other tests. See paragraph "Saving as reference" on page 54.
- After this it will do the calculation as described in paragraph "Calculations on the travel signal" on page 33. The opening time, travel, overlap and speed will be calculated using the start of the next switching event as the EndOfOperation. If there is no more event, the physical end boundary of the signal will be used.
- The *CSTime* will be remembered internally as the previous contact separation time. It will add 10 ms to the *CSTime* position and remember this time internally as the end of the previous switching event. This way it can be used as the start of the next switching event.

4.1.4 Close event (Calculations on No-load)

For a *C* switching event the system will look in the UCS signal for the **close** location.

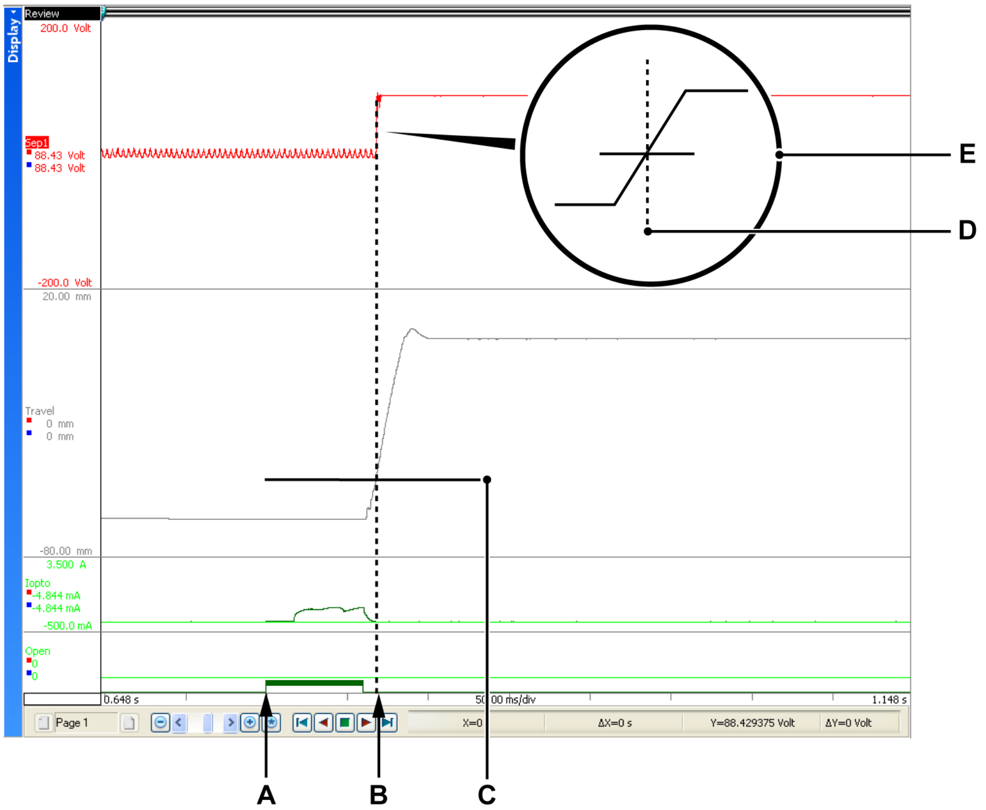

Figure 4.7: Display - Close event - Calculations on No-Load

A **ActionTime**

B **CSTime**

C **CSLevel**

D **CSTime**

E Location found through *STLNoLoadClose*

- The calculation will use the *STLNoLoadClose* method to find the location. The signal transition should go from Low to High.
- If the location is found it will be stored as *CSTime*.
- If the location is not found within the specified region, it will issue an appropriate error message and terminate.
- At the *CSTime* the system will determine the level of the travel signal, using a 1 ms average around the CSTime. This is also the value that will be saved as a reference for other tests. See paragraph "Saving as reference" on page 54.

- After this it will do the calculation as described in paragraph "Calculations on the travel signal" on page 33. The opening time, travel, overlap and speed will be calculated using the start of the next switching event as the EndOfOperation. If there is no more event, the physical end boundary of the signal will be used.
- The *CSTime* will be remembered internally as the previous contact separation time. The system will add 10 ms to the *CSTime* position and remember this time internally as the end of the previous switching event. This way it can be used as the start of the next switching event.

4.1.5 Multiple phase

If the calculation is done for a **three-phase system**, the *O* or *C* calculation will be performed on each individual phase. The information of each phase is stored as described in paragraph "Naming conventions for data source variables" on page 29. The extension 1, 2 or 3 will be used for the individual phases. Example: *CSTime1* for phase 1 and so on. The time of the three-phases is determined and stored as *CSTime*. The calculated information is shown in the HPHV Automated Analysis Sheet for each individual phase.

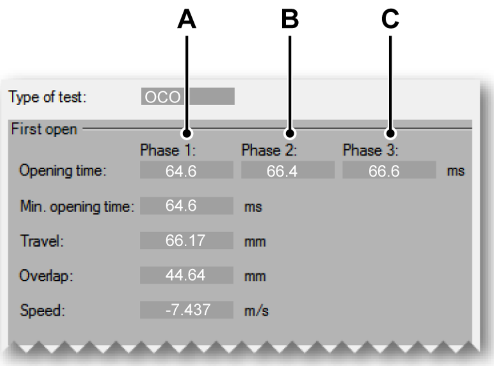

Figure 4.8: Type of test area

A Phase 1

B Phase 2

C Phase 3

4.1.6 Saving as reference

The No-Load calculation menu allows you to store the *CSLevel*, *OpeningTime* and *ClosingTime* in the **Reference** node in the data source. The CSLevel of the first calculated Open will be used to save. In absence of a CSValue for an Open, the one of the Close event will be used.This will be remembered throughout the Perception session.

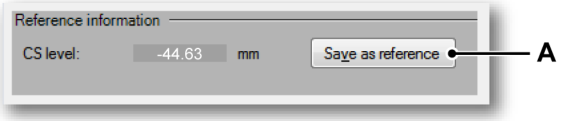

Figure 4.9: Reference information area

A Value as reference for other calculation methods

Note *This is a global reference.*

- Click **Save as reference** .
- If Perception is terminated and started again, the user will have to reload the reference information. This can be done by loading the latest recording as an experiment, loading the latest saved workbench, or loading a previously saved HPHV settings file.
- These values are used as references for the other calculation methods.

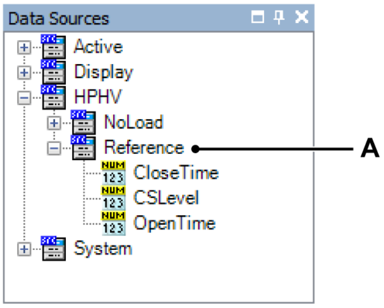

Figure 4.10: Data Sources list view - Save as reference

A Reference

The CSLevel is the most important data to save. The information saved is visible in the data source under HPHV.Reference. This includes: *CSLevel, OpenTime, CloseTime*.

5 Calculations for Short Circuit

5.1 Introduction

A **short circuit** test is basically used to test the ability to switch a current on and off. The timing of the opening and closing actions with respect to voltages and current define the severity of the test. During a **short circuit** test the circuit breaker will perform either an *Open* and/or a *Close* action. A simplified diagram of the test circuit used during the short circuit test is shown below.

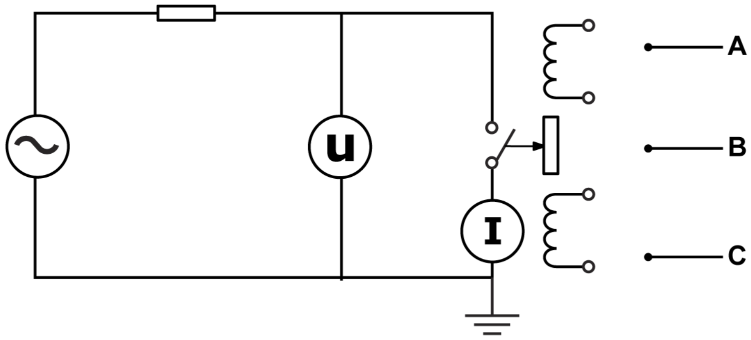

Figure 5.1: Diagram - Circuit breaker during a short circuit test

A ItrOpen

B Travel

C ItrClose

- During an *Open* action the circuit breaker will try to interrupt or break the current flow. Because of the very high amplitude the current will not be interrupted when the contacts are separated. An arc will be created through which the current continues to flow. Only during the zero crossings of the current the arc can be extinguished, thus interrupting the current.
- During a *Close* action the contacts will close while a voltage is applied over the contacts. When the contacts close, a current will start to flow. Depending on the type of circuit breaker and the applied voltage the current can already begin to flow prior to contact touch. This is called *pre-arcing*. The voltage at the moment of contact touch defines the asymmetry of the current.
- When a **short circuit** test is selected, an entry in the data source called *ShortCircuit* is created. Each stored value is added below this new entry as defined in paragraph "Naming conventions for data source variables" on page 29.

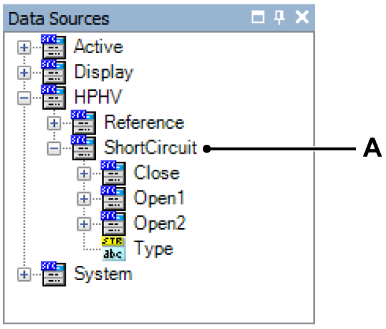

Figure 5.2: Data Sources Navigator - Calculations for Short Circuit

A **Short circuit** node

5.1.1 Input and output (Calculations for Short Circuit)

Signals used:	
Travel	This is the representation of the contact movement in mm.
ItrOpen	This is the current through the tripping coil for the test objects *Open* command.
ItrClose	This is the current through the tripping coil for the test objects *Close* command.
I(#)	This represents the current through the circuit breaker.
U(#)	This represents the voltage across the circuit breaker.

Input information:	
None	

Output information:	**Nodes in the data source**
Type	String value expressing the type of operation. This is one of the following strings: *O*, *C*, *CO*, *OC* or *OCO*.

Output Open/Close:	
ActionTime	Start time of the switching action
CSTime	Time of contact separation/touch
Speed	Speed of contacts at CSTime
Overlap	Movement in travel between ActionTime and CSTime
Travel	Movement in travel for the whole switching level

Output Open:	
Open	Node for the open information of the test
OpeningTime	Time between ActionTime and CSTime of an *Open* switching event
ArcingTime(#)	Time between CSTime and zero current for each phase
DC(#)	DC component in the current at CSTime
IBreaking(#)	RMS value of the current just before CSTime for each phase
IBreakingAverage	Average of the IBreaking currents per phase
URecovery(#)	RMS value of the voltage after the TRV for each phase
Breaking(#)	Node for current breaking information per phase
RMS	3 Crest RMS value
Crest(x)	Node for the crests used for RMS calculation, where (x) is the crest number from left to right
Value	Crests used for RMS calculation
Time	Time of the crest used for RMS calculation
StartCurrent	Start time of current
EndCurrent	End time of current
Recovery(#)	Node for recovery information per phase
RMS	RMS value after the TRV
Crest(x)	Node for the crests used for RMS calculation
Value	Crests used for RMS calculation
Time	Where (x) is the crest number from left to right
TRV	Node in data source for TRV information. See STL documentation for more information about the TRV.
Uc	Node for the Uc information of the TRV
Value	Value at the peak of the TRV
Time	Time of the peak of the TRV
Uo	Node for the Uo information of the TRV
Value	Value at the beginning of the TRV
Time	Time of the beginning of the TRV
Ua	Node for the tangent line touch in the signal
Value	Value of the tangent line touch in the signal
Time	Time of the tangent line touch in the signal
t3	t3 time of the TRV
td	td time of the TRV

Output Close:	
Close	Node for the close information of the test
ClosingTime	Time between ActionTime and CSTime of a *Close* switching event
PrearcingTime(#)	Time between current start and CSTime
IMakePeak(#)	Highest peak in the current just after contact touch
UApplied(#)	RMS value of the voltage just before CSTime
UAppliedAverage	Average of the UApplied voltage per phase
Making(#)	Node in the data source for making information per phase
DC	DC component at CSTime
Tau	Time constant in current signal
to	Parameter for time constant
Alpha	Parameter for time constant
RMS	RMS value after making
Crest(x)	Node for the crests used for RMS calculation
Value	Crests used for RMS calculation
Time	Where (x) is the crest number from left to right
Peak	Node for the peak current of the making
Value	Peak current for making
Time	Time location of peak
Applied(#)	Node in data source for the applied voltage information per phase
RMS	three-crest RMS value
Crest(x)	Node for the crests used for RMS calculation
Value	Crests used for RMS calculations
Time	Where (x) is the crest number from left to right

5.1.2 Initial calculations (Calculations for Short Circuit)

The **calculation** starts with the determination of the switching events. This is already described in paragraph "Switching event determination" on page 30. The time of the switching event is stored in the *ActionTime*. The switching event is stored in the *Type* variable.

The *CSLevel*, *OpeningTime* and *ClosingTime* are read from the *HPHV.Reference* node in the data source. These are the values that were saved as reference points during the No-Load operations. The *CSLevel* level specifies the contact separation/touch within the travel signal. The *OpeningTime* is the time needed to open, the *ClosingTime* the time needed to close the switch.

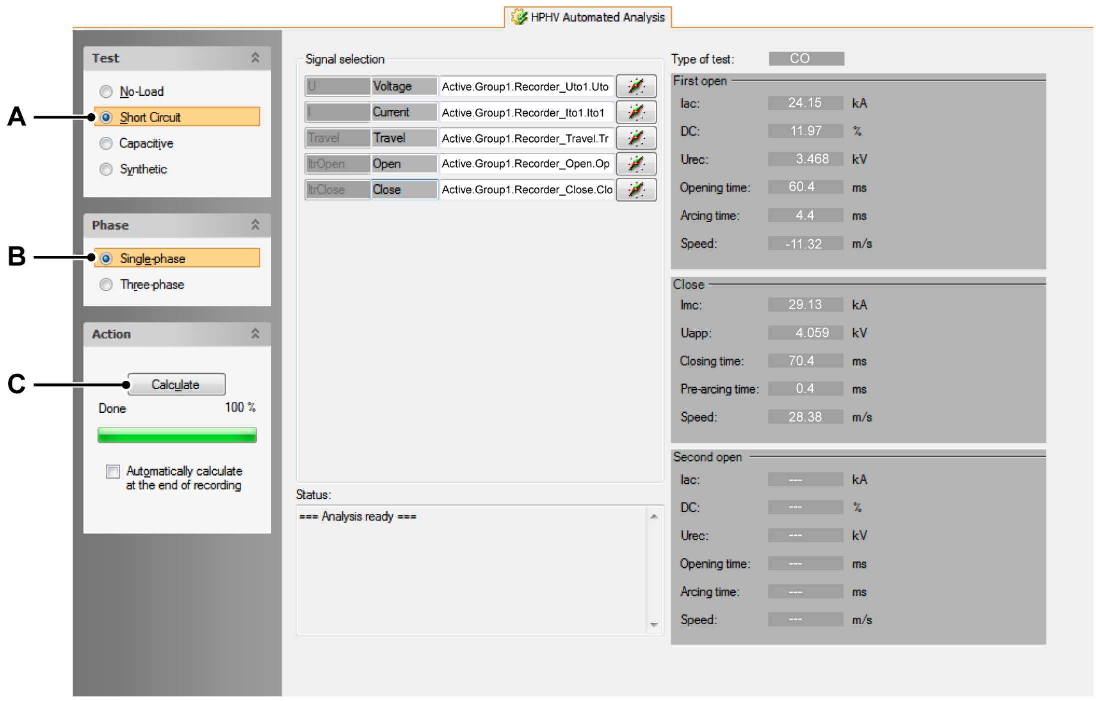

Figure 5.3: HPHV Automated Analysis sheet - Initial calculation process - Short circuit

A **Short circuit** option button

B **Single-phase** option button

C **Calculate** command button

To start a calculation for a short circuit process:

1 Click on the **Short circuit** option button.

2 Click on the (e.g.) **Single-phase** option button.
3 Click on the **Calculate** command button to start the calculation.

The switching events are calculated one by one. This is described in the following paragraphs.

5.1.3 Open event (Calculations for Short Circuit)

- All information will be saved in the data source under the **Open1** or **Open2** node, depending whether it is the first or second open in the recording. Primarily the *CSLevel* will be used to find the location of the contact separation. See paragraph "Location of contact separation/touch" on page 37. The value is stored in the data source variable *CSTime*.
- If the signals have not been **normalized** yet, the signals *U* and *I* will now be normalized. See paragraph "Signal normalization" on page 38 for more information. For the current, 100 ms before *CSTime* will be used for normalization. For the voltage, 100 ms after *CSTime* will be used. In case of multiple phases, the normalization is determined for one phase and applied to all other phases of the signal.
- If the **frequency** has not been calculated yet, it will be calculated now. The frequency will be determined from the current signal *I*. See paragraph "Frequency determination" on page 40. The time before *CSTime* will be used to determine the frequency. If the frequency can not be determined, an error message will be shown and the calculation will be terminated.

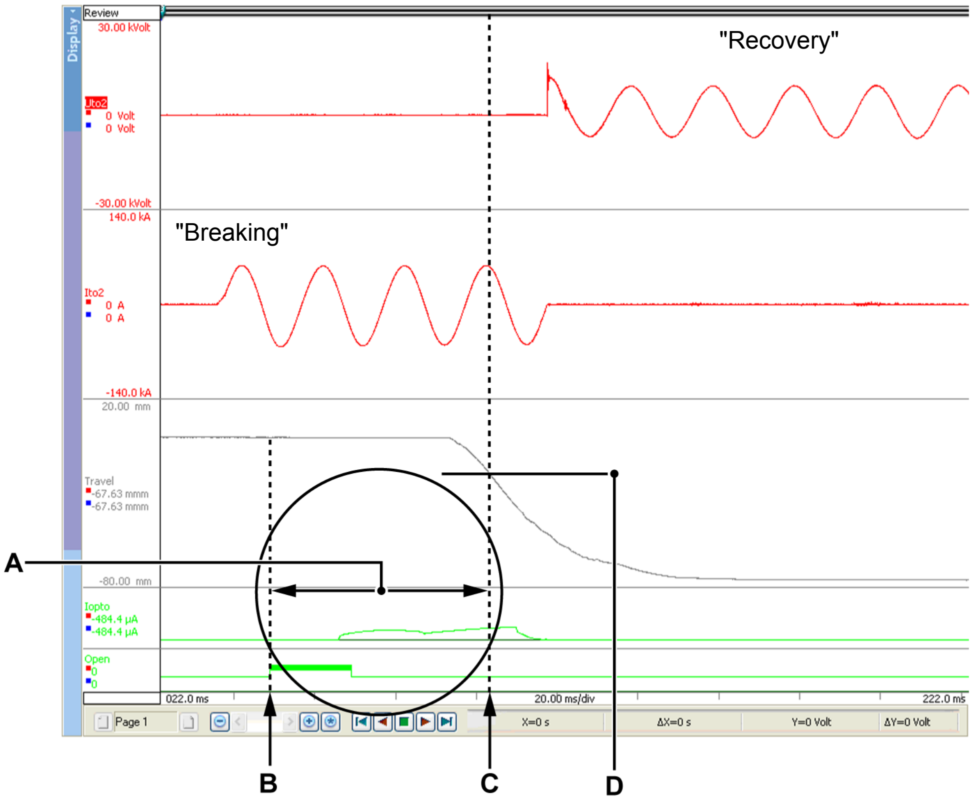

Figure 5.4: Display - Short circuit open event

A Opening Time

B ActionTime

C CSTime

D CSLevel

Breaking current calculations

The *I* signal (current) will be used to perform the calculations.

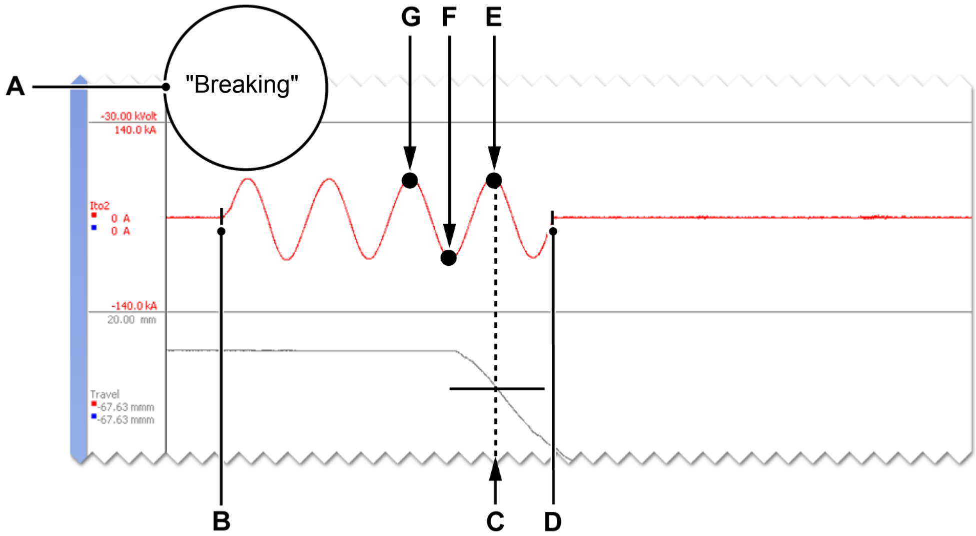

Figure 5.5: Display - Breaking current calculation process

A Breaking current calculation

B StartCurrent

C CSTime

D EndCurrent

E Crest 3

F Crest 2

G Crest 1

- From *CSTime* to the next switching event, the system will search backwards for the end of the current. It will use the STLSignalEnd method to find it.
- When found, it will use this point to search forward to find the zero crossing. It will use the STLNextZeroCrossing method to find this position. The location is saved in the data source as *EndCurrent*.
- The system will use the STLSignalStart method to find the start of the current. It will use the end of the previous switching event to start the search and the *CSTime* as the end. The determined value will be saved in the data source as *StartCurrent*.
- From the *EndCurrent* position it will search backwards in the current signal for the previous crests up to the *StartCurrent*. It will use the first three-crests prior to *CSTime* to make a three-crest RMS measurement. This value is saved as **IBreaking** in the data source.

- HPHV will take the three-crests before the contact separation point to avoid possible disturbance or deformation of the current due to the arcing. The used three-crest values and their times are saved under the *Breaking* node. The values are stored as *Crest1.Value*, *Crest1.Time* to *Crest3.Value* and *Crest3.Time*. They can be used to show the locations of the calculation.

Note *When there are not enough crests before the CSTime, the system will use the crest just after CSTime if it is close enough (less than 10 % of the period time) to the CSTime position. The user will be notified when this choice is made.*

HPHV will use the same three-crests to determine the DC component of the current. This value will be saved in the *DC* entry in the data source.

- Now the travel will be calculated as described in paragraph "Calculations on the travel signal" on page 33.
- Finally the *EndOfOperation* value will be set to the calculated *EndCurrent* time plus one loop. This value will be used internally as the start of the next switching event evaluation.
- The functionality described for one phase will also be used for three-phase calculations. Phase numbers 1 to 3 will be appended to the corresponding data source variables.
- The RMS value of the current per phase will be averaged and placed in the data source variable *IBreakingAverage*.

Recovery voltage calculations
From the *CSTime* location until the EndCurrent + 50 % of the period time the system will look for the start of the **recovery voltage** in the *U* signal.

- The system will use the STLTRV2Param(xxxx) method to find this location.
- The information of the TRV measurement is saved under the *Recovery* node in the data source. It has its own data source node called *TRV*. The *Uo.Value* and *Uo.Time* are the start value and time of the TRV. The *Ua.Value* and *Ua.Time* are the tangent of the ramp of the TRV and the *t3* and *td* information. The *Uc.Value* and *Uc.Time* are the peak value and time.
- If this method fails, the system will use the *STLSignalStart* followed by *STLZeroCrossing* to determine the start of the **recovery voltage**.

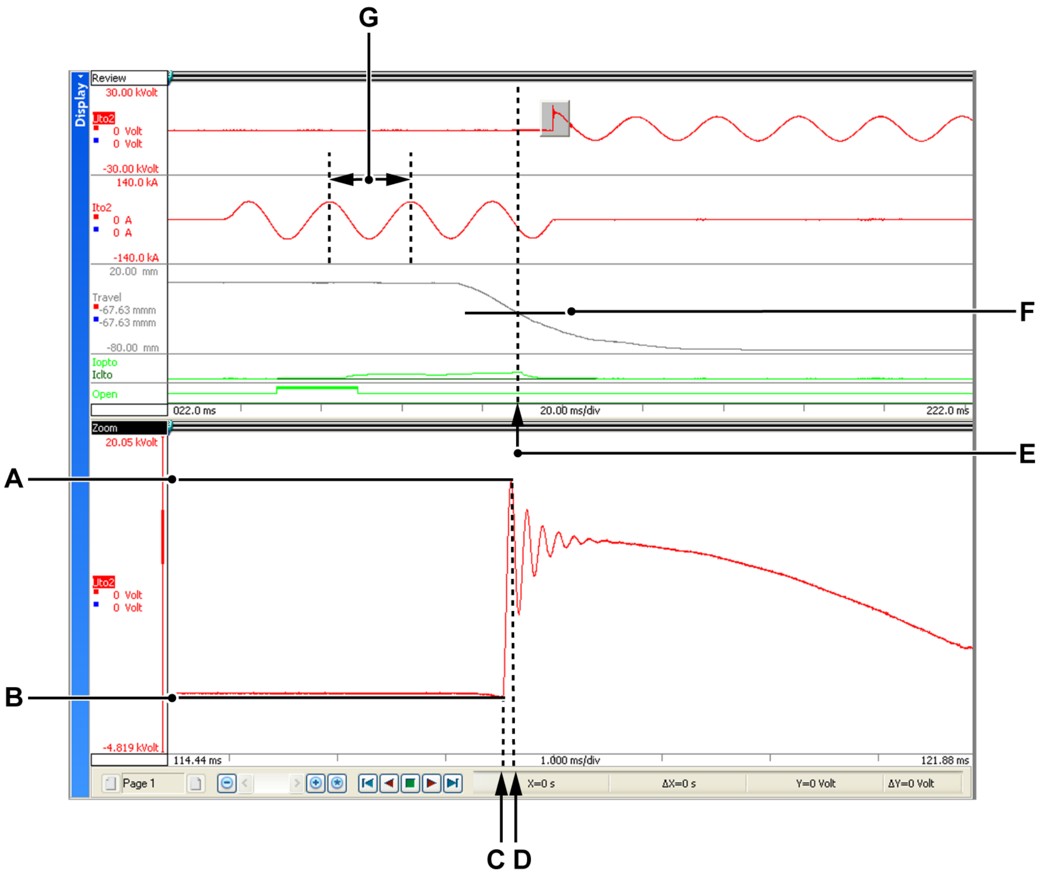

Figure 5.6: Display - UcValue/UoValue

A UCValue

B UoValue

C UoTime

D UCTime

E CSTime

F CSLevel

G Period Time

- The start of the TRV is internally stored for further processing (StartOfTRV). From the *StartOfTRV* the system will look for the next zero crossing in the recovery voltage signal. It will use the STLNextZeroCrossing method to find it.
- From this previously calculated *STLNextZeroCrossing* point it will use the next three-crests to determine the RMS value of the signal. It will use the STLNext3CrestRMS method. The value is saved as **URecovery** in the data source. The RMS value and the value and time of the three-crests are saved under the *Recovery* node.

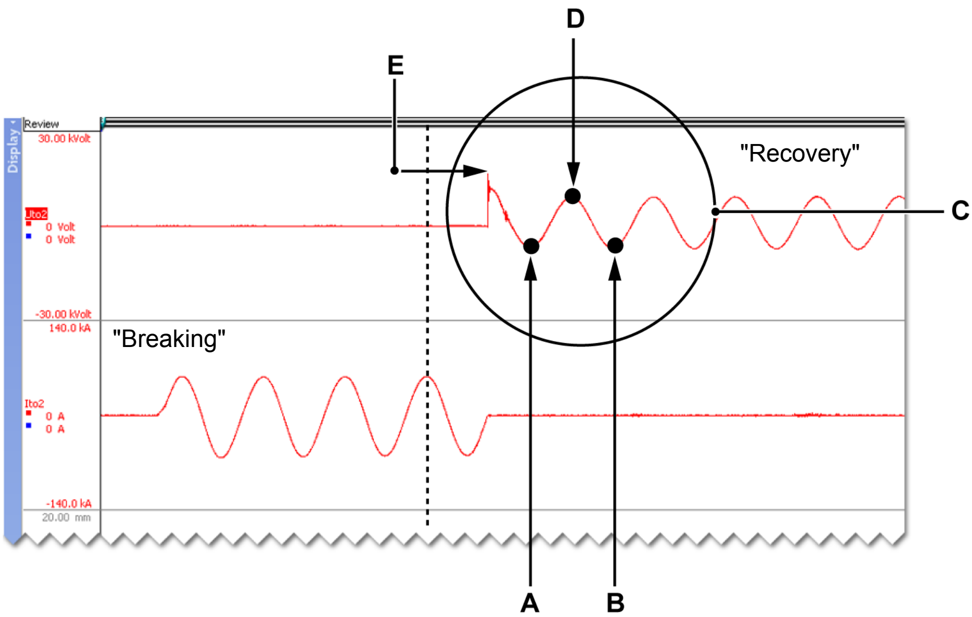

Figure 5.7: Display - Recovery voltage calculation

A Crest 1

B Crest 2

C Three-crest recovery voltage calculation

D Crest 3

E TRV

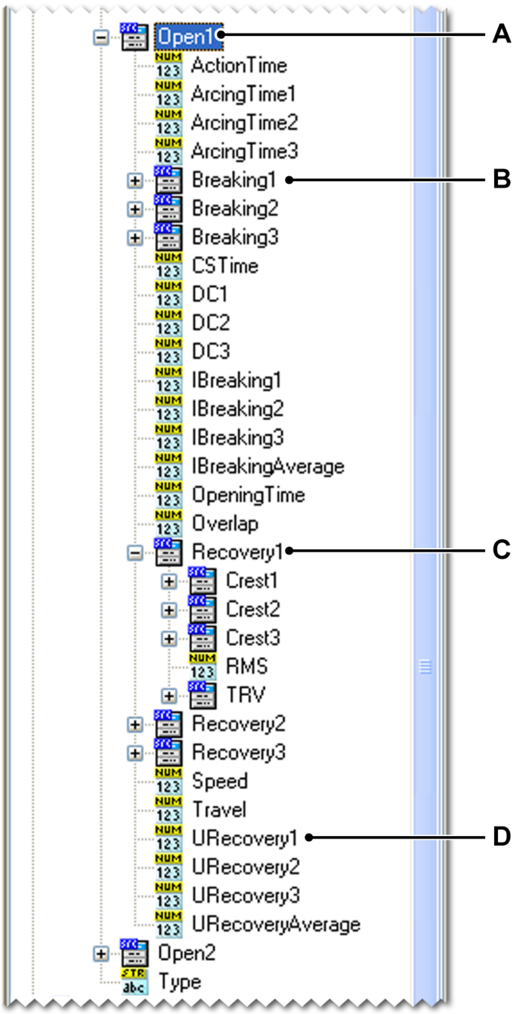

Figure 5.8: Data Sources Navigator (detail) - URecovery node

A Open information

B Breaking information

C Recovery information

D URecovery information

- For a three-phase system, calculations will be performed on each individual phase. The information is stored in the corresponding data source names. These names are extended with the phase number. The RMS value of the three-phases is averaged and stored in the **URecoveryAverage** variable in the data source.

5.1.4 Close event (Calculations for Short Circuit)

All calculated information will be saved in the data source under the **Close** node.

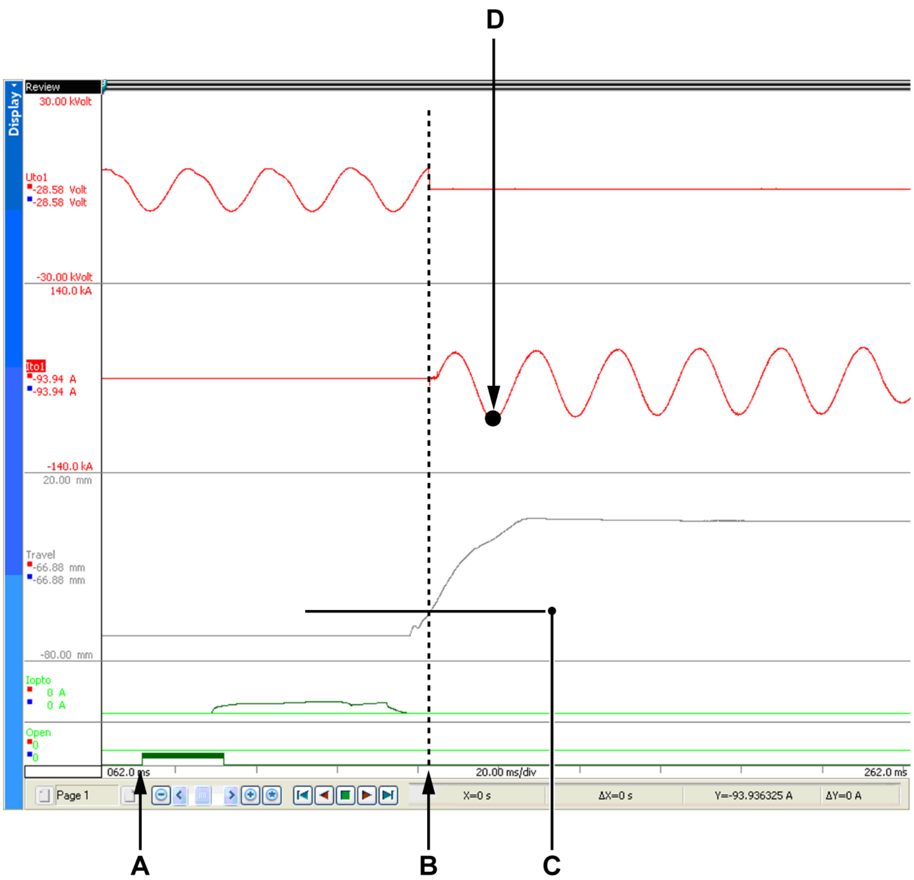

Figure 5.9: Display - Short circuit close event

A ActionTime

B CSTime

C CSLevel

D IMakePeak

- First the location of the contact touch is determined, using the method described in paragraph "Location of contact separation/touch" on page 37.

- If the signals have not been normalized yet, the signals *U* and *I* will now be normalized. See paragraph "Signal normalization" on page 38 for more information. For the voltage, 100 ms before *CSTime* will be used for normalization. For the current, the 100 ms after *CSTime* will be used. In case of multiple phases, the normalization is determined for one phase and applied to all other phases of the signal.
- If the frequency has not been defined yet, the system will search for the three-crests in the *I* signal (current) using the STLnext3Crest method. It will search forward from the *CSTime* to the next switching event. If the system cannot able to determine the frequency in this way, it will try to use the three-crests of the voltage before the *CSTime*. If the frequency cannot be determined, an error message will be shown and the calculation will be terminated.

Making current calculation
The system will use the *I* signal (current) to calculate the **making current**.

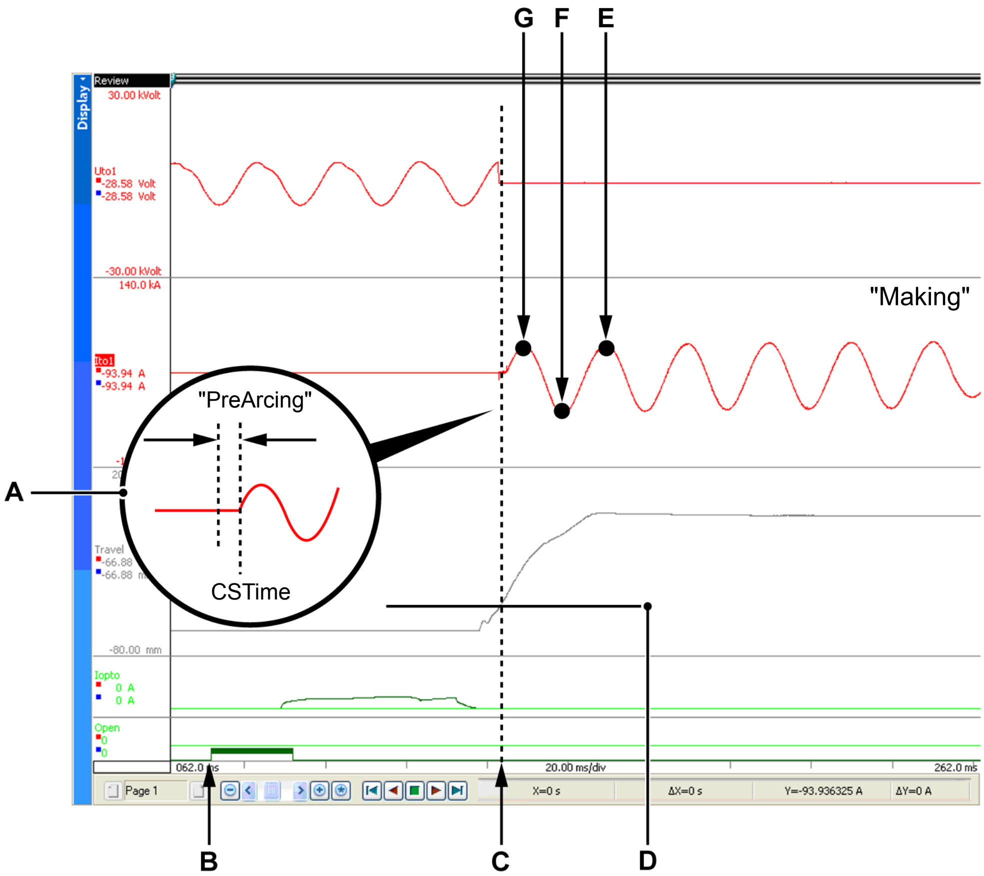

Figure 5.10: Display - Making current calculation

A Pre-arcing - Close event

B ActionTime

C CSTime

D CSLevel

E Crest 3

F Crest 2

G Crest 1

- From the *ActionTime* to the start of the next operation, the system will search for the start of the current using the STLSignalStart method. This position is then used with the STLZeroCrossing method to find the real zero crossing. The location is stored internally as *StartCurrent*.

- Now it will look backwards to find the end of the signal, using the STLSignalEnd method. When found, it will use the STLZeroCrossing method to find the actual end of the current. The value is internally stored as *EndCurrent*.
- Now the crests between *StartCurrent* and the next switching event will be calculated. The system will investigate the first two crests. The real value of the absolute maximum of the two is saved in the data source under the *Making* node as *IMakePeak.Value* and *IMakePeak.Time*.
- The first three peaks are used to calculate the making current. The system will use the STLNext3CrestRMS method. This will be stored in the data source as *IMaking*.
- The values of the three-crests that were used are saved in the data source as *Crest1.Value*, *Crest1.Time* up to *Crest3.Value*, *Crest3.Time* under the *Making* node.

Applied voltage calculation

The *U* signal (voltage) will be used for the calculations.

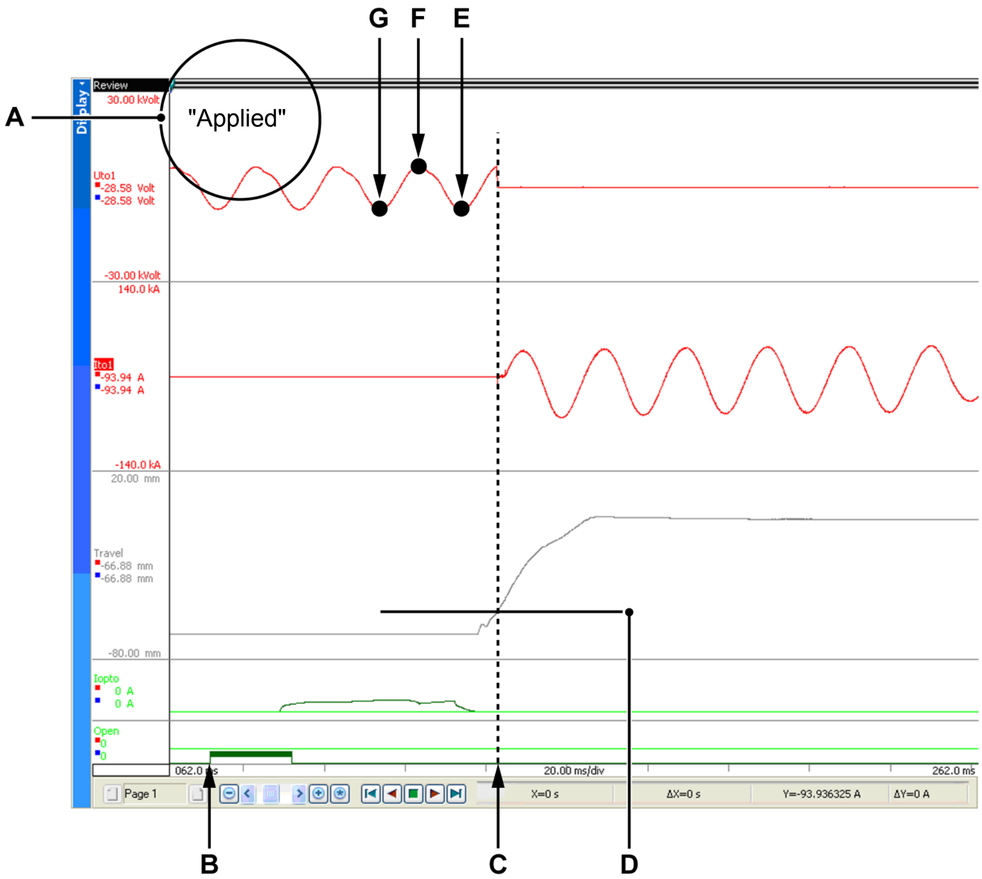

Figure 5.11: Display - Applied voltage calculation process

A Applied voltage calculation

B ActionTime

C CSTime

D CSLevel

E Crest 3

F Crest 2

G Crest 1

- From the *CSTime* location the system will search backwards for the previous four crests. This leaves three-crests out of the four to do the calculation. It will investigate the last two crests if the expected period time matches. The last crest should be at least 10 % of the period time before the contact touch location to avoid voltage distortion due to pre-arcing.

- The previously calculated three-crests are then used to calculate the applied voltage. The value is saved in the data source as *UApplied*. The three-crests that were used for the calculation are saved in the data source under the *Applied* node as *Crest1.Value*, *Crest1.Time* up to *Crest3.Value*, *Crest3.Time*.
- In case of a calculation for a three-phase system, the three voltages are averaged and stored in the *UAppliedAverage* variable in the data source.

5.1.5 Miscellaneous

The method described in paragraph "Calculations on the travel signal" on page 33 will be used to determine the travel information of this switching event.

6 Calculations for Capacitive Test

6.1 Introduction

The purpose of the capacitive test is to test the ability to “make” or “break” a current at capacitive loads at high voltages. During a capacitive test, the circuit breaker will switch a capacitive load. The voltages involved can be very high. The corresponding currents are relatively small. A simplified diagram of the test circuit used during the capacitive test is shown below.

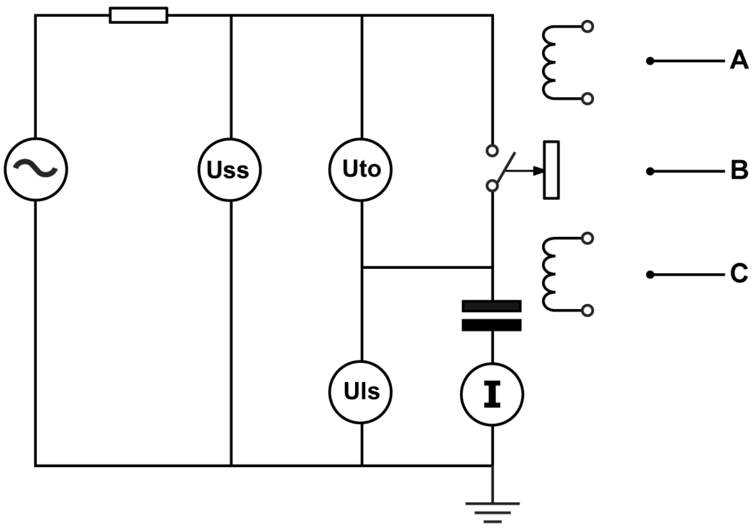

Figure 6.1: Diagram - Circuit breaker during a capacitive test

A ItrOpen

B Travel

C ItrClose

- Due to the nature of this test circuit and the extra elements (capacitor bank), the voltage and current are not always sine wave shaped signals. The STL committee has not made specific exceptions for distorted signals; they recommend to use true RMS calculations in case of such signals. The reason is the parabolic fit that will be used over the expected top of the sine wave. This can produce unexpected results when used for non sine wave shaped signals.
- The HPHV Automated Analysis uses true RMS calculation on all current and voltage signals, because they are not perceptive to signal distortion. For each true RMS calculation the two points between which the RMS calculation is executed will also be saved as source variables.
- However, the RMS calculations are also available as three-crest RMS calculations. The crests that were used to get the three-crest RMS are also saved as a data source.

- In case of errors in the three-crest RMS calculations, a warning message will be issued, but the calculations will continue, because they are primarily based on the true RMS calculations.

When a capacitive test is selected, an entry in the data sources navigator named *Capacitive* is created.

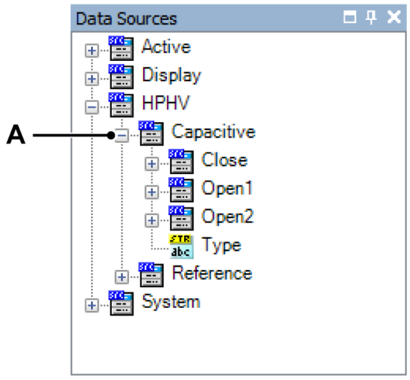

Figure 6.2: Data Sources Navigator - Capacitive test

A **Capacitive** node

Each stored value is added below this new entry as defined in paragraph "Naming conventions for data source variables" on page 29.

6.1.1 Input and output (Calculations for Capacitive Test)

Signals used:	
Travel	This is the representation of the contact movement in mm.
ItrOpen	This is the current through the tripping coil for the test objects *Open* command.
ItrClose	This is the current through the tripping coil for the test objects *Close* command.
I(#)	This represents the current through the circuit breaker.
Uto(#)	This represents the voltage across the circuit breaker.
Uls(#)	This represents the voltage at the load side.
Uss(#)	This represents the voltage at the supply side.

Input information:	**None**

Output information:	**Nodes in the data source**
Type	String value expressing the type of operation. This is one of the following strings: *O*, *C*, *CO*, *OC* or *OCO*.

Output Open/Close:	
ActionTime	Start time of the switching action
CSTime	Time of contact separation/touch
Speed	Speed of contacts at CSTime
Overlap	Movement in travel between ActionTime and CSTime
Travel	Movement in travel for the whole switching level

Output Open:	
Open	Node for the Open information of the test
OpeningTime	Time between ActionTime and CSTime of an *Open* switching event
ArcingTime(#)	Time between CSTime and current zero for each phase
IBreaking(#)	True RMS value of the current just before CSTime for each phase
IBreakingAverage	Average of the IBreaking currents per phase
UBefore(#)	True RMS value of the USS just before CSTime
UBeforeAverage	Average of the UBefore voltage per phase
UAfter(#)	True RMS value of the USS after CSTime
UAfterAverage	Average of the UAfter voltage per phase
Breaking(#)	Node for current breaking information per phase
RMS	True RMS value after making

Output Open:	
RMSTime1	Start time of true RMS calculation
RMSTime2	End time of true RMS calculation
Crest(x)	Node for each individual crest used for the RMS calculation, where (x) is the crest number from left to right
Value	Crests used for the three-crest RMS calculation
Time	Time of the crest used for the three-crest RMS calculation.
StartCurrent	Start time of current
EndCurrent	End time of current
After(#)	Node for recovery information per phase
RMS	True RMS value after making
RMSTime1	Start time of true RMS calculation
RMSTime2	End time of true RMS calculation
Crest(x)	Node for each individual crest used for the RMS calculation, where (x) is the crest number from left to right.
Value	Crests used for the three-crest RMS calculation
Time	Time of the crest used for the three-crest RMS calculation
Peak	Node for the peak current of the making
Value	Peak voltage
Time	Time location of peak
TimeToPeak	The time from CSTime to the peak voltage
Before(#)	Node for recovery information per phase
RMS	True RMS value after making
RMSTime1	Start time of true RMS calculation
RMSTime2	End time of true RMS calculation
Crest(x)	Node for each individual crest used for the RMS calculation, where (x) is the crest number from left to right
Value	Crests used for the three-crest RMS calculation
Time	Time of the crest used for the three-crest RMS calculation

Output Close:	
Close	Node for the Close information of the test
ClosingTime	Time between ActionTime and CSTime of a *Close* switching event
PreArcingTime(#)	Time between current start and CSTime

Output Close:	
IMake(#)	Highest peak in the current just after contact touch
UApplied(#)	True RMS value of the voltage just before CSTime
UAppliedAverage	Average of the UApplied voltage per phase
Making(#)	Node in data source for making information per phase
RMS	True RMS value after making
RMSTime1	Start time of true RMS calculation
RMSTime2	End time of true RMS calculation
Crest(x)	Node for each individual crest used for the RMS calculation, where (x) is the crest number from left to right
Value	Crests used for the three-crest RMS calculation
Time	Time of the crest used for the three-crest RMS calculation
Peak	Node for the peak current of the making
Value	Peak current for making
Time	Time location of peak
Applied(#)	Node in data source for the applied voltage information per phase
RMS	True RMS value after making
RMSTime1	Start time of true RMS calculation
RMSTime2	End time of true RMS calculation
Crest(x)	Node for each individual crest used for the RMS calculation, where (x) is the crest number from left to right
Value	Crests used for the three-crest RMS calculation
Time	Time of the crest used for the three-crest RMS calculation

6.1.2 Initial calculation (Calculations for Capacitive Test)

The calculation starts with the determination of the switching events, as described in paragraph "Switching event determination" on page 30. The time of the switching event is stored in the *ActionTime*. The switching event is stored in the *Type* variable.

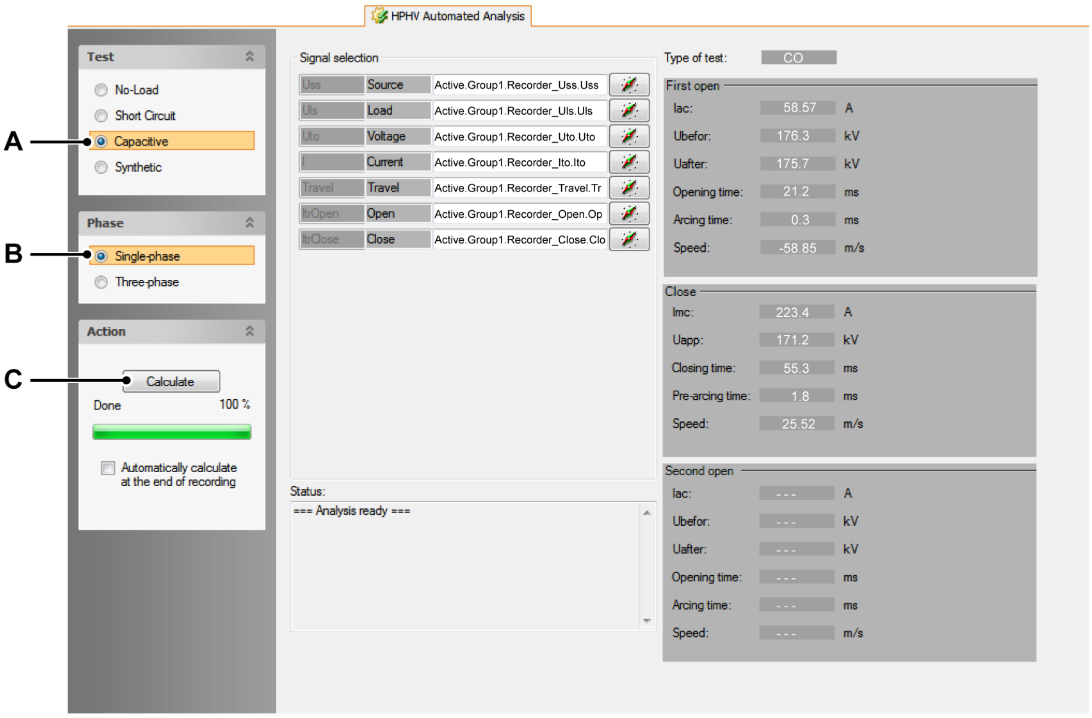

Figure 6.3: Perception HPHV window - Initial calculation process - Capacitive test

A **Capacitive** test option

B **Single-phase** option button

C **Calculate** command

To start a calculation for a capacitive process:

1. Click **Capacitive**
2. Select the required phase.
3. Click **Calculate** to start the calculation.

- The *CSLevel*, *OpeningTime* and *ClosingTime* are read from the *Reference* node in the data source. These are the values that were saved as reference points during the No-Load operations.
- The *CSLevel* specifies the contact separation/touch within the travel signal.

- The OpeningTime is the time needed to open, the ClosingTime the time needed to close the switch.

The switching events are now calculated one by one, as is described in the following paragraphs.

6.1.3 Open event (Calculations for Capacitive Test)

All information will be saved in the data source under the *Open1* or *Open2* node, depending on whether it is the **first open** or **second open** in the recording.

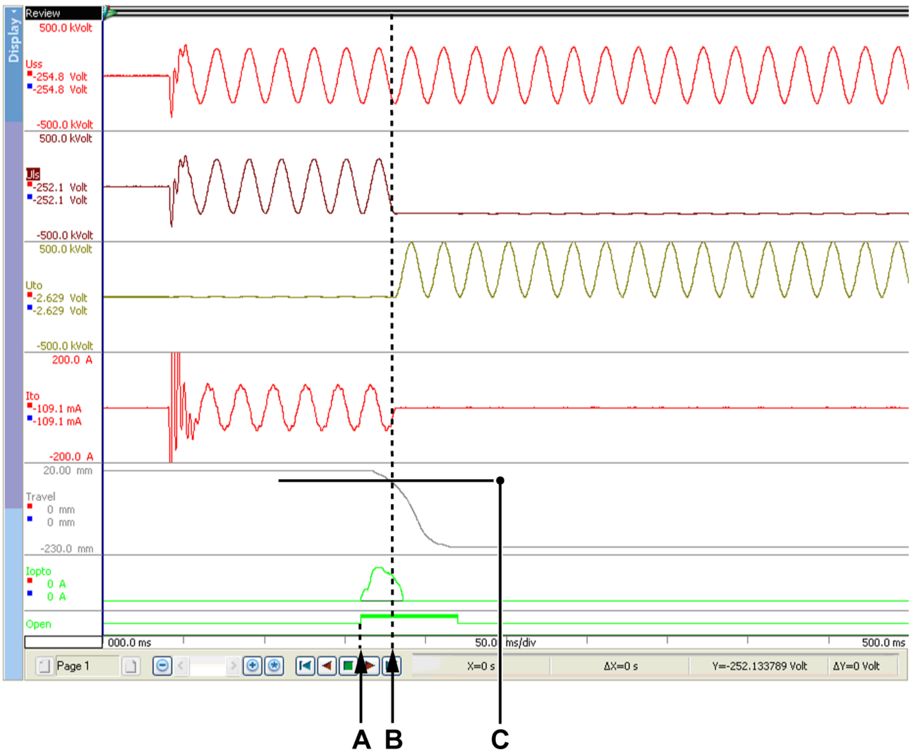

Figure 6.4: Display - Open event - Capacitive test

A ActionTime

B CSTime

C CSLevel

- First the *CSLevel* will be used to find the location of the contact separation. See paragraph "Location of contact separation/touch" on page 37. Now the *CSTime* is known.
- If the signals have not been normalized yet, the signals *U* and *I* will now be normalized. See paragraph "Signal normalization" on page 38 for more information. For the voltage, 100 ms before *CSTime* will be used for normalization. For the current, the 100 ms after the *CSTime* will be used. In case of multiple phases, the normalization is determined for one phase and applied to all other phases of the signal.

- If the frequency has not been calculated yet, it will be calculated now. The frequency will be determined from the current signal *I*. See paragraph "Frequency determination" on page 40. The time before *CSTime* will be used to determine the frequency. If the frequency cannot be determined, an error message will be shown and the calculation will be terminated.

Breaking current calculations
The current signal *I* will be used for this calculation.

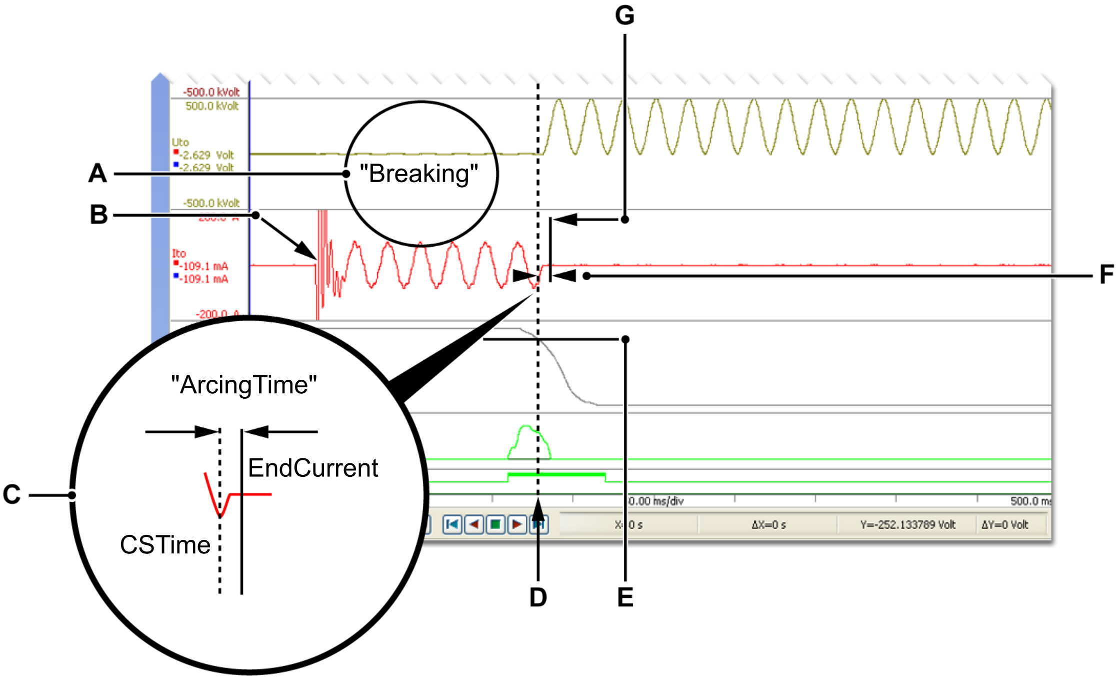

Figure 6.5: Display - Breaking current calculation process

A Breaking current calculation

B StartCurrent

C Arcing Time

D CSTime

E CSLevel

F ArcingTime

G EndCurrent

- First the start and end of the current will be determined. The system will search backwards between the *CSTime* position and the end to find the end of the current. It uses the STLSignalEnd method.
- When this position is found, it will use the STLNextZeroCrossing method to determine the real zero crossing.
- It will search forward between the *StartOfOperation* and *CSTime* position to find the start of the signal, using the STLSignalStart method. Because of the high spike density at the start of the current due to the pre-arcing this method is not very accurate. Therefore the sytem will search before the found start position to find the first spike with an amplitude larger than 1 % of the amplitude range and with a duration longer than 0.1 % of the period.

- The start and end value will be saved in the data source as *Breaking(#).StartCurrent* and *Breaking(#).EndCurrent*.
- Now the **breaking current** will be calculated searching backwards in the signal between the *CSTime* and *StartOfOperation* to find three zero crossings. The first and last will be used to perform a true RMS calculation, using the STLPrevTrueRMS method. The values are stored in the data source variables *IBreaking*, *Breaking(#).RMS* for the RMS value, and *Breaking(#). RMSTime1* and *.RMSTime2* for the RMS time boundaries.

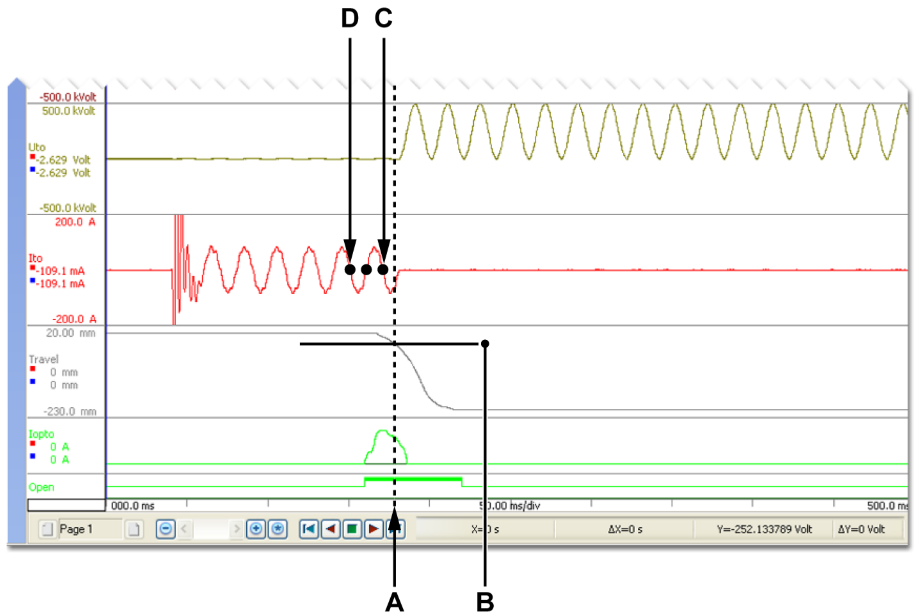

Figure 6.6: Display - Breaking current calculation process - RMS time

A CSTime

B CSLevel

C RMSTime 2

D RMSTime 1

- As from now the three-crest RMS values will be calculated. The system will search backwards from the CSTime to get the three-crest RMS and crest position, using the STLPrev3Crest method. A warning message will be issued in case of calculation problems. The values are saved in the data source under the *Breaking(#)* node as *.3CrestRMS* for the three-crest RMS and *.Crest(x).Value* and *.Crest(x).Time* for each crest.

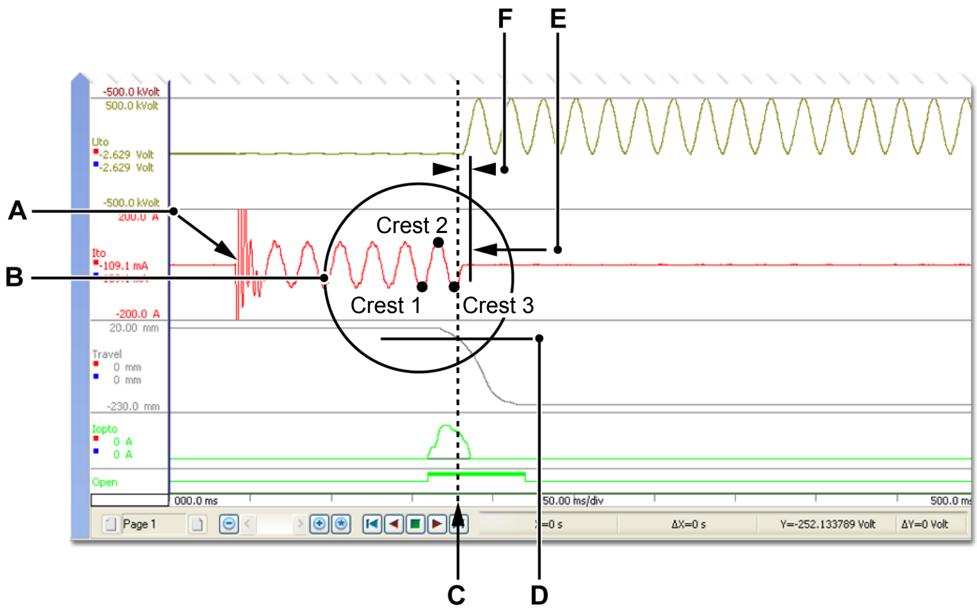

Figure 6.7: Display - Breaking current calculation process - Three-crest

A StartCurrent

B Three-crest breaking current calculation

C CSTime

D CSLevel

E EndCurrent

F Arcing Time

- If this is a calculation for three phases, the three true RMS values will be averaged and stored in the *IBreakingAverage* variable in the data source.

Voltage before calculations
The **voltage** signal *Uls* will be used for the calculations.

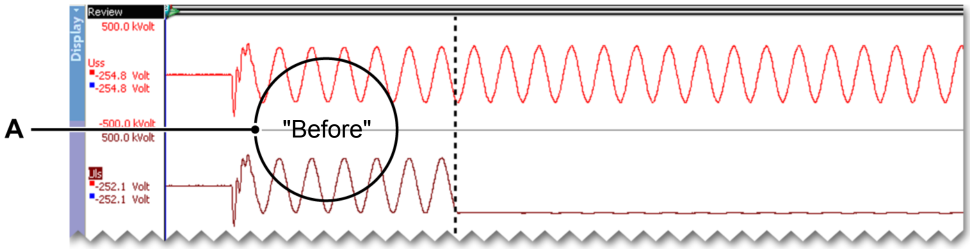

Figure 6.8: Display - Calculation on the voltage before contact separation

A Voltage before calculations

- Between the *StartOfOperation* and *CSTime* the system will search backwards for three zero crossings, using the STLPrevZeroCrossing method.
- The first and last are used to do a true RMS calculation, using the STLPrevTrueRMS method. The values are stored in the data source as **UBefore** and *Before(#).RMS* for the RMS value and *Before(#).RMSTime1* and *.RMSTime2* for the true RMS boundaries.

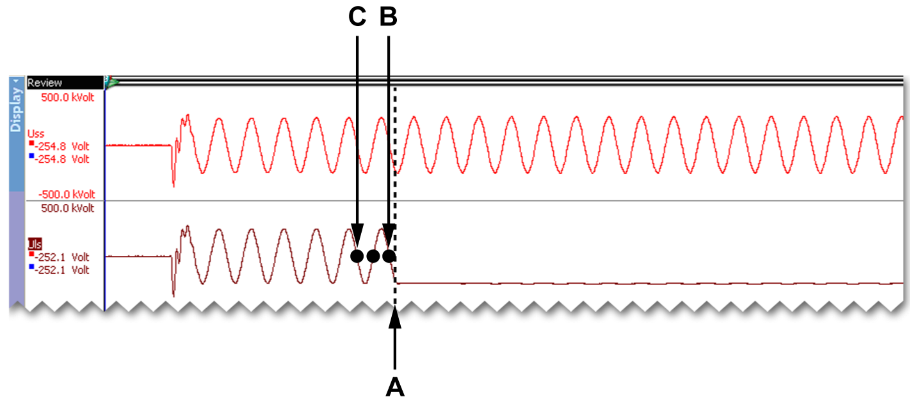

Figure 6.9: Display - Calculation on the voltage before contact separation (RMSTime)

A CSTime

B RMSTime 2

C RMSTime 1

- Now the three-crest RMS values will be calculated. The system will search backwards from *CSTime* to determine the three-crest RMS and crest position, using the STLPrev3Crest method. A warning message will be issued in case of calculation problems. The values are saved in the data source under the *Before(#)* node as *.3CrestRMS* for the three-crest RMS and *.Crest(x).Value* and *.Crest(x).Time* for each crest.

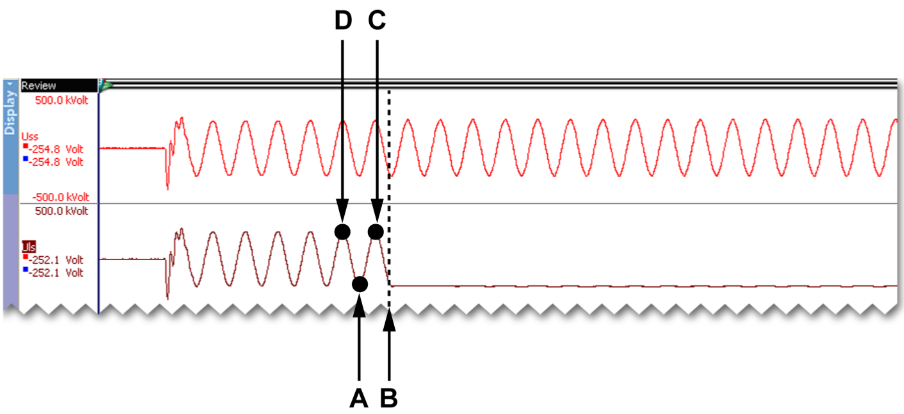

Figure 6.10: Display - Calculation on the voltage before contact separation (RMS Crests)

A Crest 2

B CSTime

C Crest 3

D Crest 1

- If this is a calculation for three phases, the three true RMS values will be averaged and stored in the *UBeforeAverage* variable in the data source.

Voltage after calculations
The **voltage** signal *Uto* will be used for calculation.

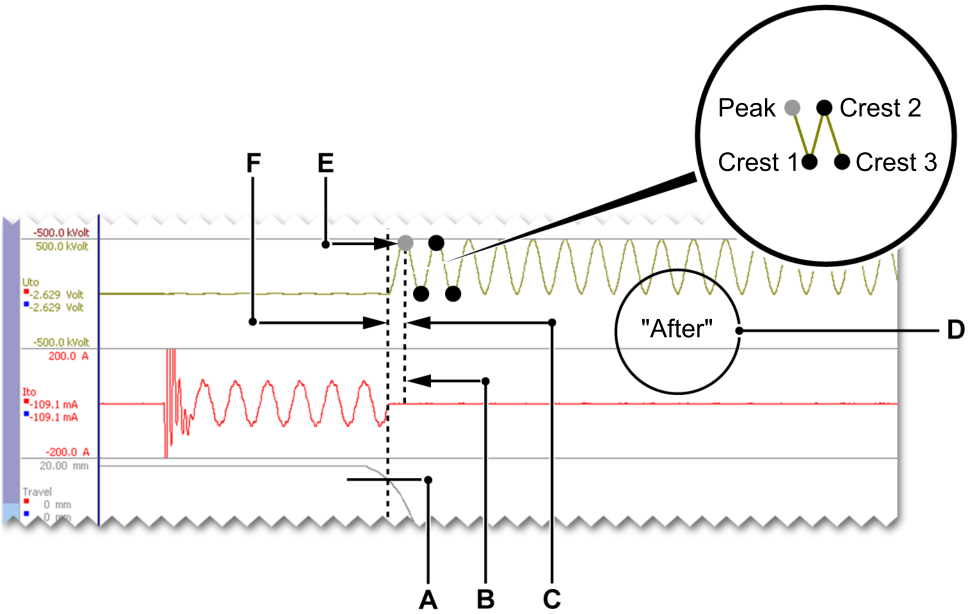

Figure 6.11: Display - Calculation on the voltage after contact separation

A CSLevel

B EndCurrent

C TimeToPeak

D Voltage after calculations

E Peak

F TimeToPeak

- To begin, the first peak and location will be calculated using the STLNextCrest method, starting at the *EndCurrent*. The values are stored in the data source under the **After(#)** node as *.Peak.Value* and *.Peak.Time*. The time between the peak and the *EndOfCurrent* is stored in the variable *After(#).TimeToPeak* value.
- For the next calculation of the true RMS and three-crest RMS values the *Uto* signal needs to be centered around zero. The new voltage signal to work on is the formula:
  Uto – Peak.Value / 2.

- Between *Peak.Time* and *EndOfOperation* the system will search forward for three zero crossings, using the STLNextZeroCrossing. The first and last are used to do a true RMS calculation, using the STLNextTrueRMS method. The values are stored in the data source as *UAfter* and *After(#).RMS* for the RMS value and *After(#).RMSTime1* and *.RMSTime2* for the true RMS boundaries.

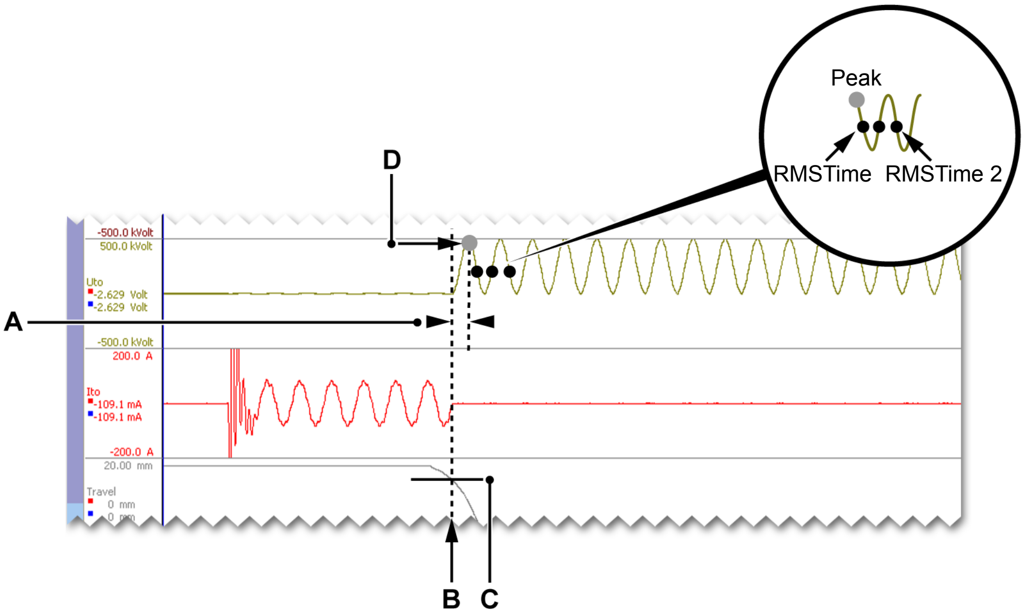

Figure 6.12: Display - Calculation on the voltage after contact separation (RMS Time)

A TimeToPeak

B CSTime

C CSLevel

D Peak

- Now the **three-crest RMS** values will be calculated. The system will search forward from *RMSTime1* to determine the three-crest RMS and crest position, using the STLNext3Crest method. A warning message will be issued in case of calculation problems. The values are saved in the data source under the **After(#)** node as *.3CrestRMS* for the three-crest RMS and *.Crest(x).Value* and *.Crest(x).Time* for each crest.

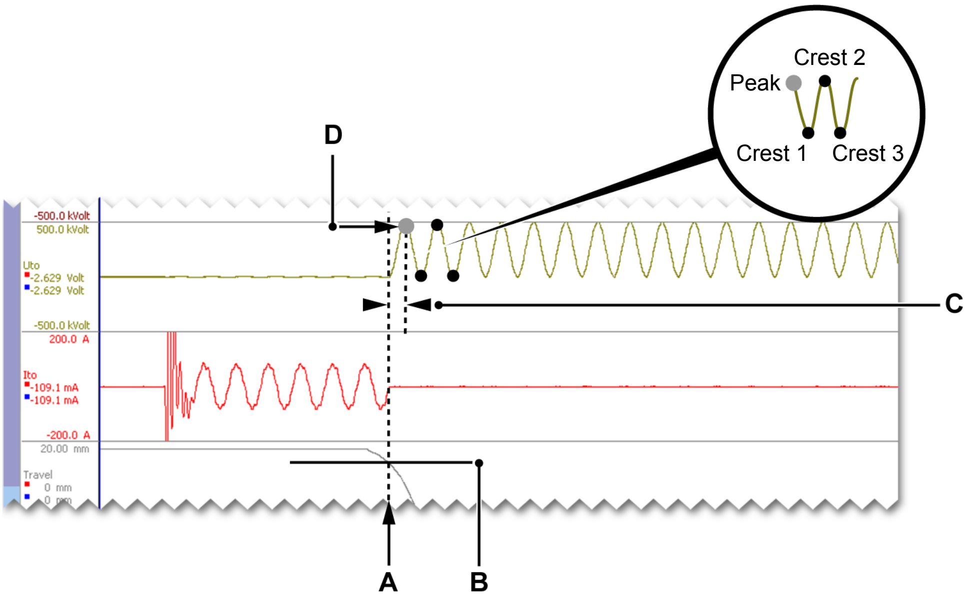

Figure 6.13: Display - Calculation on the voltage after contact separation (RMS Crests)

A CSTime

B CSLevel

C TimeToPeak

D Peak

- If this is a calculation for three phases, the three true RMS values will be averaged and stored in the *UAfterAverage* variable in the data source.

6.1.4 Close event (Calculations for Capacitive Test)

All information will be saved in the data source under the **Close** node.

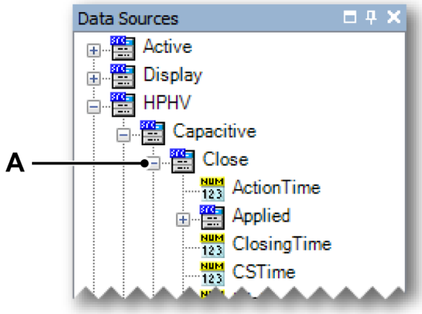

Figure 6.14: Data Sources Navigator - Close node - Capacitve test

A Close node

- First the *CSLevel* will be used to find the location of the contact separation. See paragraph "Location of contact separation/touch" on page 37. Now the *CSTime* is known.
- If the signals have not been normalized yet, the signals *Uss, Uls, Uto* and *I* will now be normalized. See paragraph "Signal normalization" on page 38 for more information. For the current, 100 ms after *CSTime* + 5 ms will be used for normalization. For the voltage, the 100 ms before *CSTime* will be used. In case of multiple phases, the normalization is determined for one phase, and applied to all other phases of the signal.
- If the frequency has not been calculated yet, it will be calculated now. The frequency will be determined from the supplied signal *Uss*. See paragraph "Frequency determination" on page 40. The time before *CSTime* will be used to determine the frequency. If the frequency cannot been determined, an error message will be shown and the calculation will be terminated.

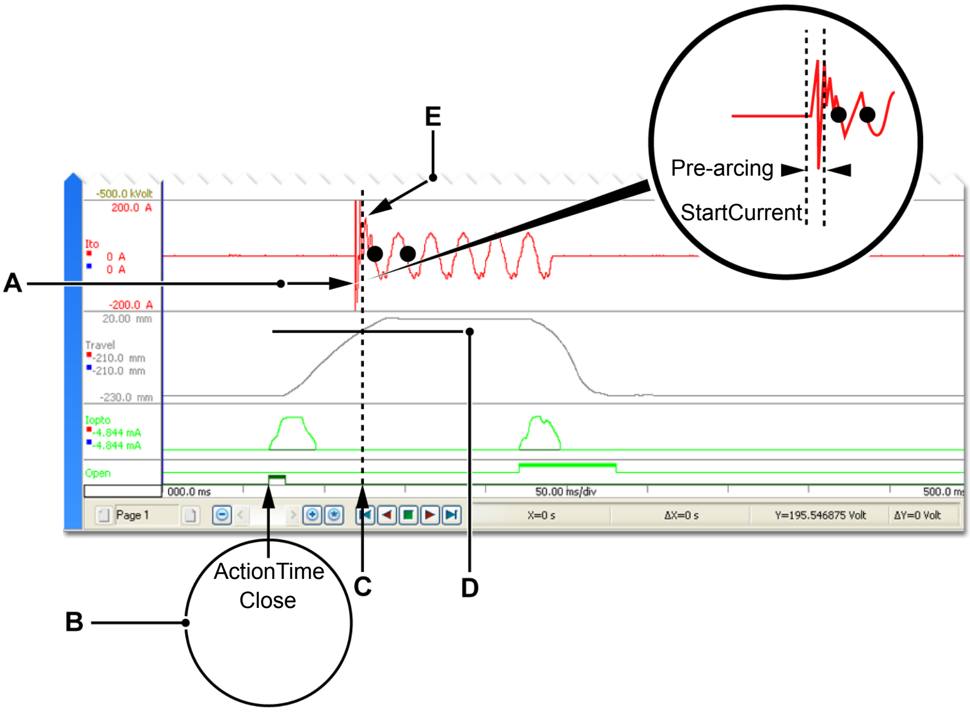

Figure 6.15: Display of a Close event - Capacitive test

A Pre-arcing

B ActionTime - Close event

C CSTime

D CSLevel

E Peak

Closing current calculations
The *I* signal will be used for calculation.

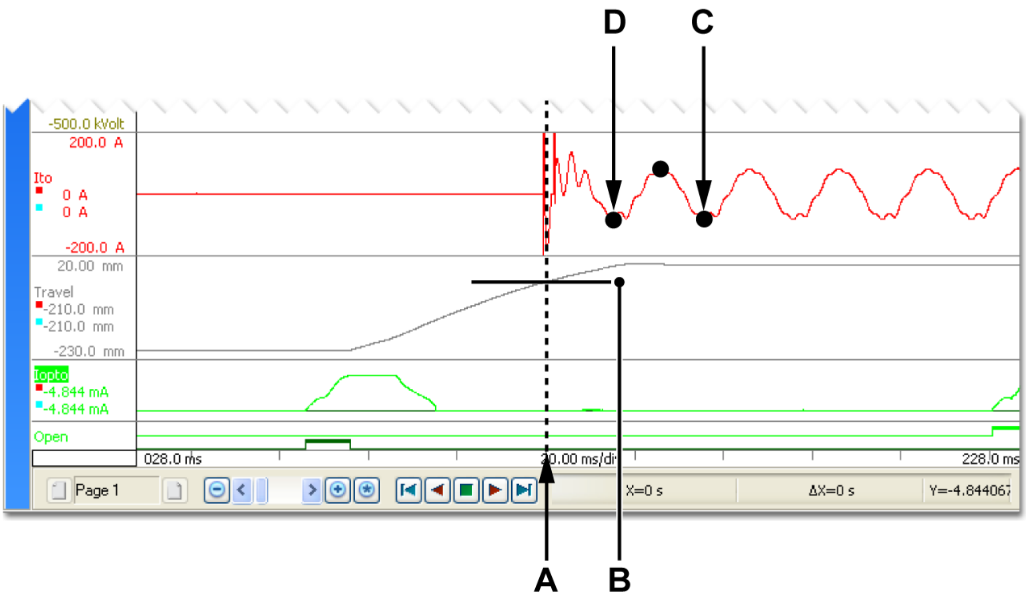

Figure 6.16: Display - Closing current calculations - RMS Time

A CSTime

B CSLevel

C RMSTime 2

D RMSTime 1

- First the system will find the next three zero crossings after *CSTime* using the STLNextZeroCrossing method. From the first and last the true RMS value will be calculated using the STLNextTrueRMS method. The values are reported in the data source variables. The information is stored under the *Making(#)* node. The RMS value is saved as *.RMS*, the times used to do the RMS calculation as *.RMSTime1* and *.RMSTime2*.

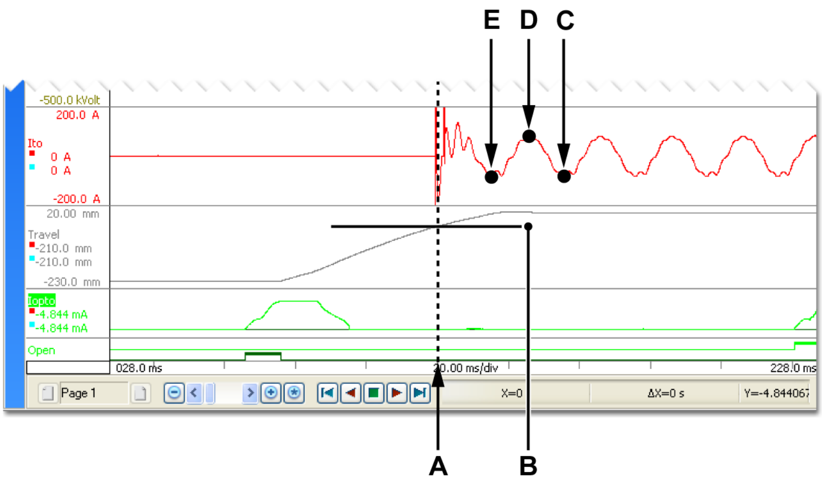

Figure 6.17: Display - STLNext3CrestRMS method

A CSTime

B CSLevel

C Crest 3

D Crest 2

E Crest 1

- From the first determined zero crossing, the three-crest RMS calculations will be performed using the STLNext3CrestRMS method. The values are stored under the *Making(#)* node as *.3CrestRMS* for the three-crest RMS and *.Crest(x).Value* and *.Crest(x).Time* for each crest. If this is a calculation for three phases, the three true RMS values will be averaged and stored in the *IMakeAverage* variable in the data source.
- Between the *CSTime* and the first determined zero crossing the absolute maximum value will be found. The formula functions *NextHillPos* and *NextValleyPos* are used to find this value. It will be reported as *IMake* in the data source.

Applied voltage calculations
The *Uto* signal will be used for calculation.

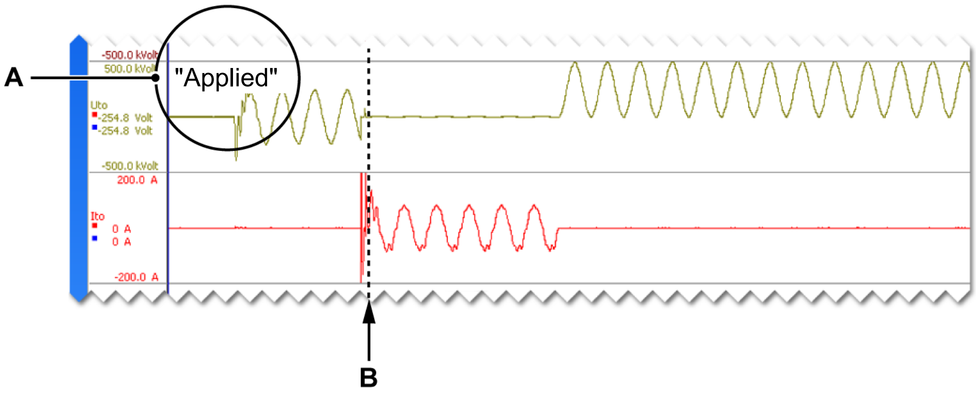

Figure 6.18: Display of an applied voltage calculation process

A Applied voltage calculation

B CSTime

- Between the *StartOfOperation* and *CSTime* the system will search backwards for three zero crossings using the STLPrevZeroCrossing method. The first and last are used to perform a true RMS calculation using the STLPrevTrueRMS method. The values are stored in the data source as **UApplied** and *Applied(#).RMS* for the RMS value and *Applied(#).RMSTime1* and *.RMSTime2* for the true RMS boundaries.

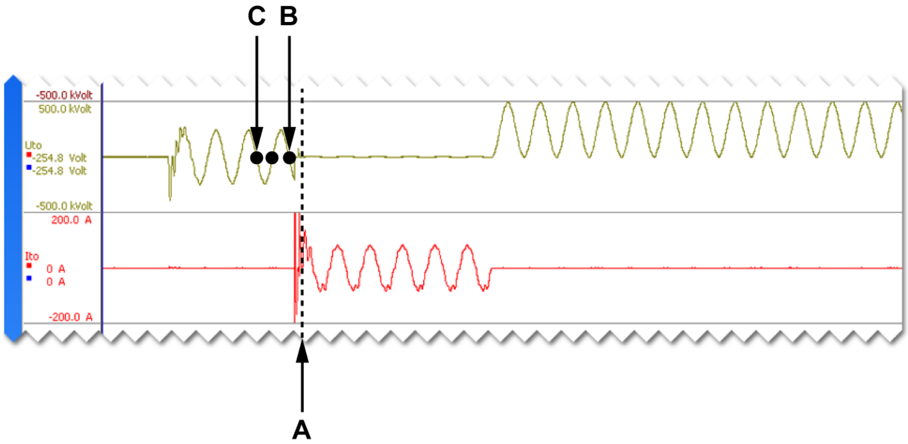

Figure 6.19: Display - RMS Time

A CSTime

B RMSTime 2

C RMSTime 1

- Now the three-crest RMS values will be calculated. The system will search backwards from *CSTime* to determine the three-crest RMS and crest position, using the STLPrev3Crest method. A warning message will be issued in case of calculation problems. The values are saved in the data source under the *Applied (#)* node as *.3CrestRMS* for the three-crest RMS and *.Crest(x).Value* and *.Crest(x).Time* for each crest.

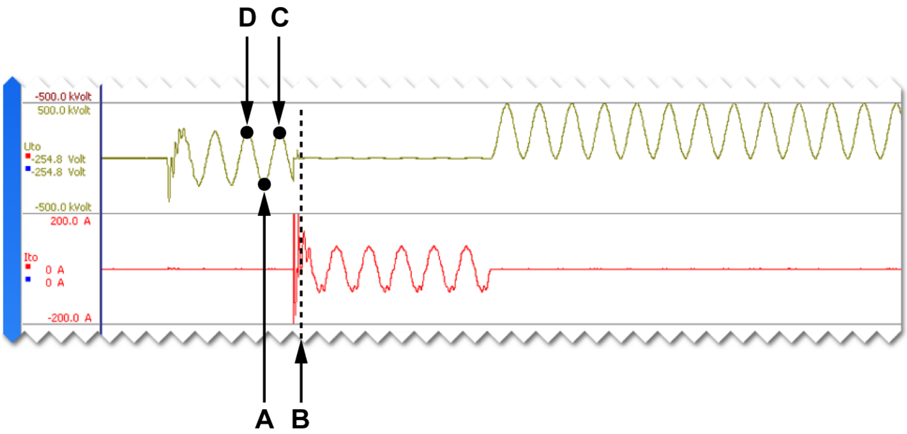

Figure 6.20: Display - STLPrev3Crest method

A Crest 2

B CSTime

C Crest 3

D Crest 1

- If this is a calculation for three phases, the three true RMS values will be averaged and stored in the *UAppliedAverage* variable in the data source.

6.1.5 Miscellaneous

The method described in paragraph "Calculations on the travel signal" on page 33 will be used to determine the travel information of this switching event.

7 Calculations for Synthetic Test

7.1 Introduction

The purpose of the synthetic test is to test the ability to “make” or “break” a current. A **synthetic test** is basically a short circuit test. A simplified diagram of the test circuit used during the **synthetic test** is shown below.

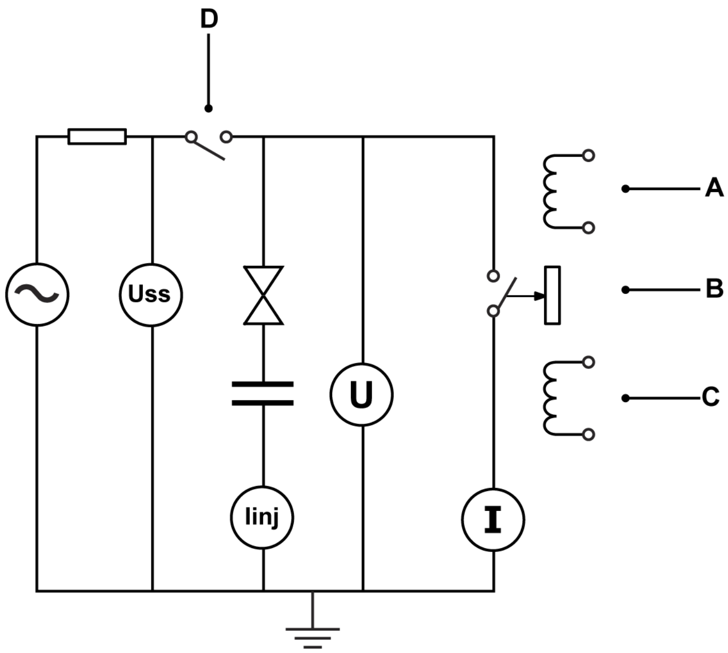

Figure 7.1: Diagram - Circuit breaker during a synthetic test

A ItrOpen

B Travel

C ItrClose

D Auxillary breaker

- The extra circuit with an auxiliary breaker and a current injection circuit makes it possible to inject current at the last loop of the current of an *Open* switching event. This forces a much larger di/dt value at the current zero crossing. For the circuit breaker this behaviour is as if a much larger voltage were being applied, hence the name **synthetic**.

- When a *synthetic* test is selected, an entry in the data source called *Synthetic* is created. Each stored value is added below this new entry as defined in paragraph "Naming conventions for data source variables" on page 29.

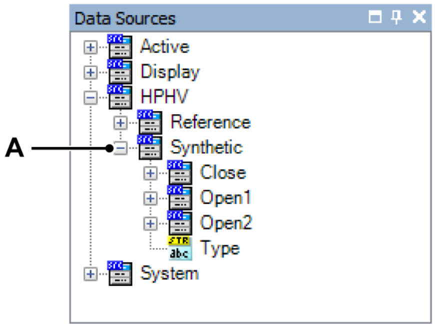

Figure 7.2: Data Sources Navigator - Synthetic test

A **Synthetic** node

7.1.1 Input and output (Calculations for Synthetic Test)

Signals used:	
Travel	This is the representation of the contact movement in mm.
ItrOpen	This is the current through the tripping coil for the test objects *Open* command.
ItrClose	This is the current through the tripping coil for the test objects *Close* command.
I(#)	This represents the current through the circuit breaker.
U(#)	This represents the voltage across the circuit breaker.
Iinj(#)	This represents the current injected through the circuit breaker.
Uss(#)	This represents the voltage at the supply side.

Input information:	
None	

Output information:	**Nodes in the data source**
Type	String value expressing the type of operation. This is one of the following strings: *O*, *C*, *CO*, *OC* or *OCO*.

Output Open/Close:	
ActionTime	Start time of the switching action
CSTime	Time of contact separation/touch
Speed	Speed of contacts at CSTime
Overlap	Movement in travel between ActionTime and CSTime
Travel	Movement in travel for the whole switching level

Output Open:	
Open	Node for the open information of the test
OpeningTime	Time between ActionTime and CSTime of an *Open* switching event
ArcingTime(#)	Time between CSTime and current zero for each phase
DC(#)	DC component in the current at CSTime
IBreaking(#)	RMS value of the current just before CSTime for each phase
IBreakingAverage	Average of the IBreaking currents per phase
URecovery(#)	RMS value of the voltage after the TRV for each phase
URecoveryAverage	Average of the URecovery voltage per phase
Injection(#)	Injection current
InjectionTime(#)	Injection time

Output Open:	
USource	RMS value at the supply side
Breaking(#)	Node for current breaking information per phase
RMS	Three-crest calculation of the breaking current
Crest(x)	Node for each individual crest used for RMS calculation, where (x) is the crest number from left to right
Value	Crests used for RMS calculation
Time	Time of crest used for the RMS calculation
StartCurrent	Start time of current
EndCurrent	End time of current
Recovery(#)	Node for recovery information per phase
RMS	RMS value after the TRV
Crest(x)	Node for each individual crests used for the RMS calculation, where (x) is the crest number from left to right
Value	Crests used for RMS calculation
Time	Time of crest used for the RMS calculation
TRV	Node in data source for TRV information. See STL documentation for more information about the TRV.
Uc	Node for the Uc information of the TRV
Value	Value at the peak of the TRV
Time	Time of the peak of the TRV
Uo	Node for the Uo information of the TRV
Value	Value of the start of the TRV
Time	Time of the start of the TRV
Ua	Node for the tangent line touch in the signal
Value	Value of the tangent line touch in the signal
Time	Time of the tangent line touch in the signal
t3	t3 time of the TRV
td	td time of the TRV
Injection(#)	Node for injection information
Peak	Node for the peak current of injection
Value	Peak value of the injection current
Time	Time of the peak value
StartTime	StartTime of the injection current
EndTime	EndTime of the injection current
CouplingMoment	Time where there is only the injection current through the circuit breaker
IncouplingCurrent	Injection current at the CouplingMoment

Output Open:	
IncouplingTime	Time between CouplingMoment and end of injection
Source(#)	Node for the source voltage information
RMS	Three-crest RMS value
Crest(x)	Node for each individual crest used for RMS calculation, where (x) is the crest number from left to right
Value	Crests used for RMS calculation
Time	Time of crest used for the RMS calculation

Output Close:	
Close	Node of the close information of the test
ClosingTime	Time between ActionTime and CSTime of a *Close* switching event
PreArcingTime(#)	Time between current start and CSTime
IMakePeak(#)	Highest peak in the current just after contact touch
UApplied(#)	RMS value of the voltage just before CSTime
UAppliedAverage	Average of the UApplied voltage per phase
Making(#)	Node in data source for making information per phase
DC	DC component at CSTime
Tau	Time constant in current signal
to	Parameter for time constant
Alpha	Parameter for time constant
RMS	RMS value after making
Crest(x)	Node for each individual crest used for RMS calculation, where (x) is the crest number from left to right
Value	Crests used for RMS calculation
Time	Time of crest used for the RMS calculation
Peak	Node for the peak current of making
Value	Peak current for making
Time	Time location of peak
Applied(#)	Node in data source for the applied voltage information per phase
RMS	Three-crest RMS value
Crest(x)	Node for each individual crest used for RMS calculation, where (x) is the crest number from left to right
Value	Crests used for RMS calculation
Time	Time of crest used for the RMS calculation

The basic synthetic calculation is exactly the same as for the short circuit test. For a description refer to the corresponding chapter .

An extra is the calculation at the *Open* event. This will be described in more detail in the "Open event (Calculations for Synthetic Test)" on page 105.

7.1.2 Initial calculation (Calculations for Synthetic Test)

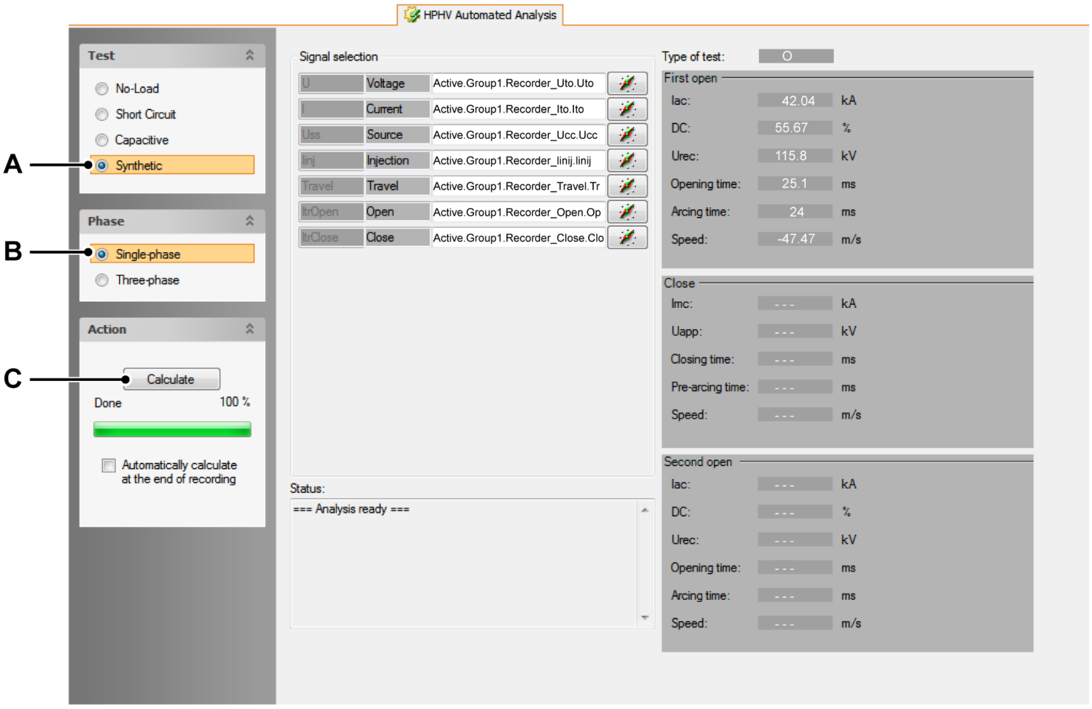

Figure 7.3: HPHV Automated Analysis Sheet - Initial calculation process - Synthetic test

A **Synthetic** test option

B **Single-phase** option

C **Calculate** command

To start a calculation for a synthetic test:

1 Click **Synthetic**
2 Select the required phase.
3 Click **Calculate** to start the calculation.

As the synthetic test is basically a short circuit test, please refer to chapter 5 "Introduction" on page 55 (Calculations for Short Circuit) for this information. This chapter describes the specific synthetic test calculations.

7.1.3 Open event (Calculations for Synthetic Test)

If a normalization has not yet been performed on the injection current and source voltage, it will be performed now. See paragraph "Signal normalization" on page 38 for more information. The system will use the *Iinj* and the *Uss* signal.

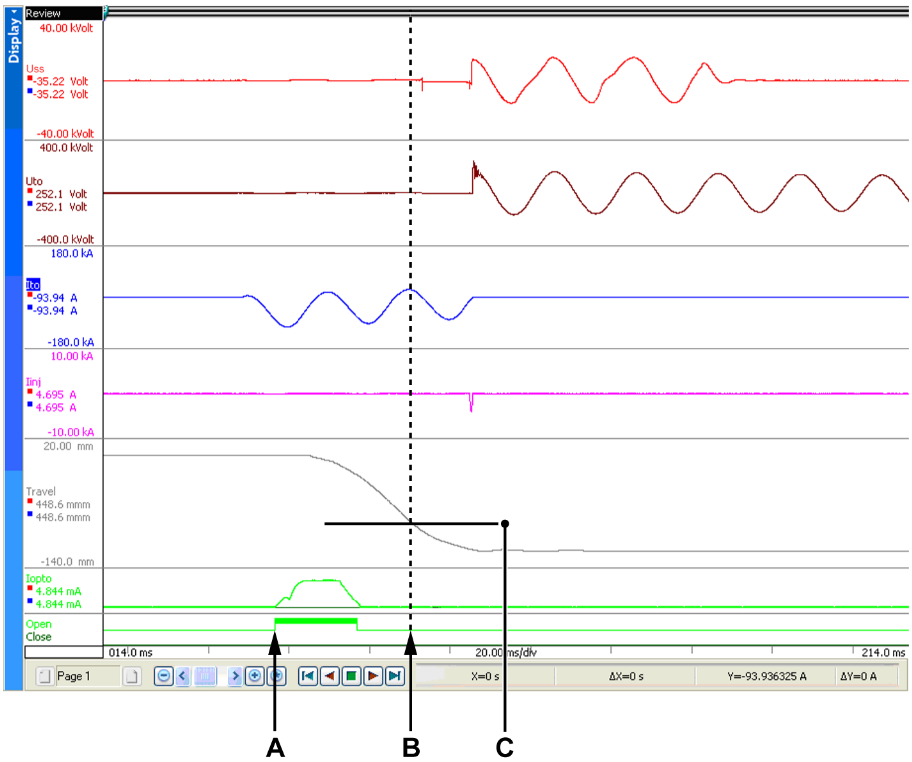

Figure 7.4: Display - Open event - Calculations for Synthetic test

A ActionTime

B CSTime

C CSLevel

The *Iinj* signal **(injection current)** is used to find the start and end time of the **injection** current.

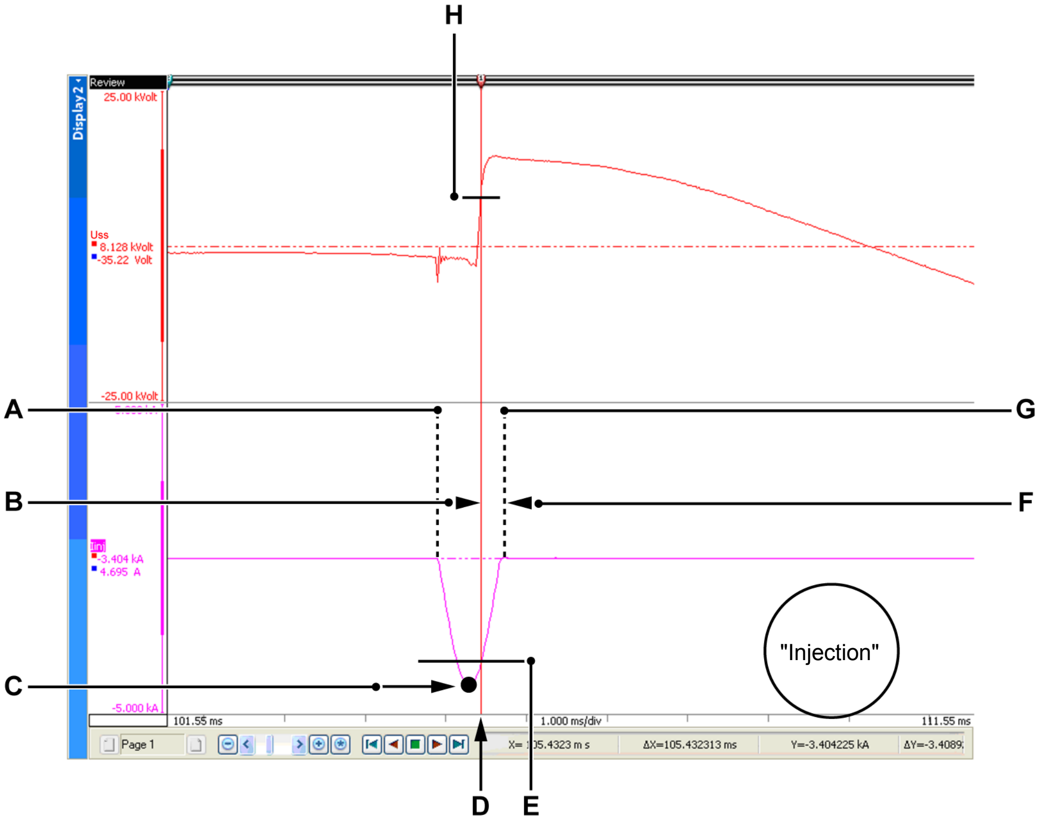

Figure 7.5: Display of an injection current - Synthetic test

A Injection(#).StartTime

B IncouplingTime

C Peak

D CouplingMoment

E IncouplingCurrent

F IncouplingTime

G Injection(#).EndTime

H 50% Level

- The calculations use the STLSignalStart and STLSignalEnd followed by STLNextZeroCrossing to determine the positions. These positions are stored in the data source variables *Injection(#).StartTime* and *.EndTime*.

- Between the found values the peak and its time position are determined using the STLNextCrest method. The information is stored in the data source variables *Injection.Peak.Value* and *.Time*. The peak value is also saved in the variable *Injection*.
- The *Uss* signal is used to find the 50% level. The system uses the static amplifier range to determine the 50% level. The calculation uses the NextLvlCross method to find the location. The location is saved in the data source as *Injection.CouplingMoment*.
- In the *Iinj* signal the signal value is calculated at the *Injection.CouplingMoment*. This value is saved in the data source variable *Injection.IncouplingCurrent*. The value is negative if the *Injection.CouplingMoment* is before the *Injection.Peak.Time*.
- The time difference between the *Injection.CouplingMoment* and the *Injection.EndTime* is saved in the data source variable *Injection.IncouplingTime*. The time difference is also saved in the *InjectionTime* variable.

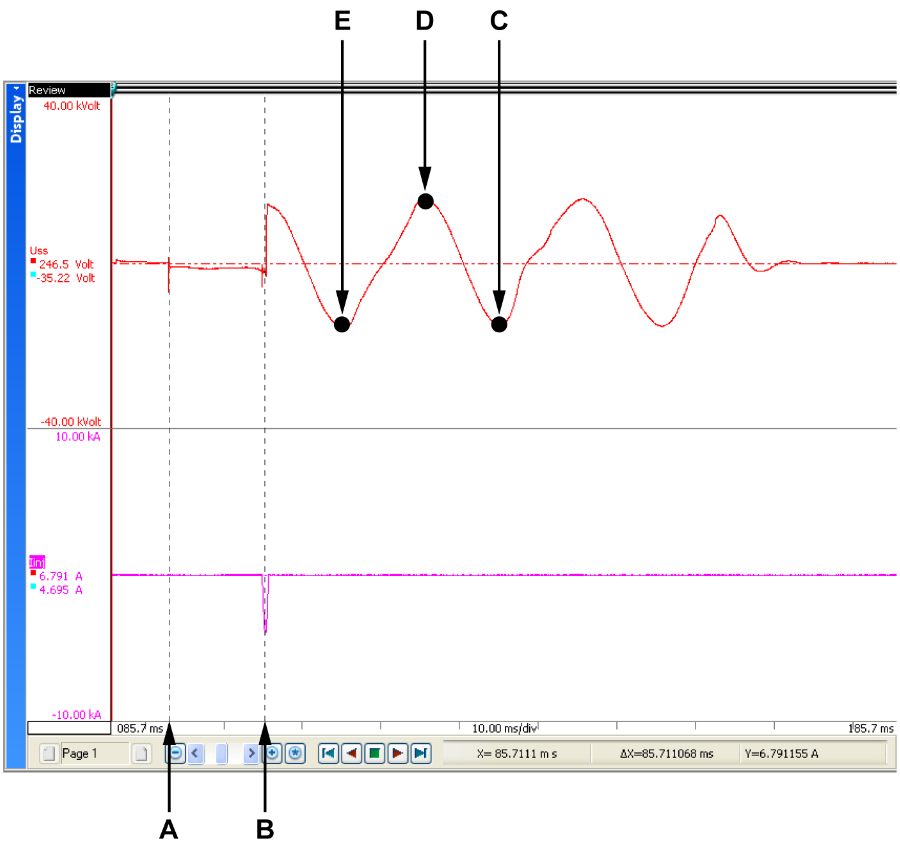

Figure 7.6: Display - Three-crest calculation on supply voltage

A CSTime

B CouplingMoment

C Crest 3

D Crest 2

E Crest 1

- On the *Uss* signal a three-crest RMS calculation will be executed using the STL3CrestRMS method. The loop just after the *Injection.CouplingMoment* will be skipped by finding the next zero crossing using the STLNextZeroCrossing method.
- The RMS value and the corresponding crests are saved in the data source as *USource.RMS*, *.Crest(x).Value* and *.Time*. The RMS value is also saved in the *USource* variable.

Index

Head Office
HBM
Im Tiefen See 45
64293 Darmstadt
Germany
Tel: +49 6151 8030
Email: info@hbm.com

France
HBM France SAS
46 rue du Champoreux
BP76
91542 Mennecy Cedex
Tél:+33 (0)1 69 90 63 70
Fax: +33 (0) 1 69 90 63 80
Email: info@fr.hbm.com

UK
HBM United Kingdom
1 Churchill Court, 58 Station Road
North Harrow, Middlesex, HA2 7SA
Tel: +44 (0) 208 515 6100
Email: info@uk.hbm.com

USA
HBM, Inc.
19 Bartlett Street
Marlborough, MA 01752, USA
Tel : +1 (800) 578-4260
Email: info@usa.hbm.com

PR China
HBM Sales Office
Room 2912, Jing Guang Centre
Beijing, China 100020
Tel: +86 10 6597 4006
Email: hbmchina@hbm.com.cn

measure and predict with confidence

I2695-1.2 en